

# Schnelle, konservative Greif- und Ablageplanung von unbekanntem Objekten

Von der Universität Bayreuth  
zur Erlangung des Grades eines  
Doktors der Naturwissenschaften (Dr. rer. nat.)  
genehmigte Abhandlung

von

Johannes Baumgartl  
aus Erlangen

1. Gutachter: Prof. Dr. Dominik Henrich
2. Gutachter: Prof. Dr. Michael Guthe

Tag der Einreichung: 22.12.15

Tag des Kolloquiums: 6.5.16



Im großen Garten der Geometrie kann sich jeder nach  
seinem Geschmack einen Strauß pflücken.

D. HILBERT



## DANKSAGUNG

Ich danke meinem Doktorvater Professor Dominik Henrich für die intensive Betreuung und den Freiraum meine Ideen umzusetzen. Ich danke meinen Kollegen für die fruchtbaren und kaffeehaltigen Diskussionen sowie für die tolle Arbeitsatmosphäre, insbesondere Eric Alborn, Thomas Buchmann, Michael Gradmann, Christian Groth, Maria Hänel, Maximilian Sand, Michael Spangenberg und Tobias Werner.

Weiter möchte ich mich bei meinen Studenten bedanken, die mit mir zusammen unzählige Ideen ausprobiert und spannende Projekte umgesetzt haben, insbesondere Per Kaminsky, Dominik Riedelbauch, Michael Riedl, Georg Rollinger und Tim Werner.

Ich bedanke mich bei meiner Familie für die Unterstützung und das stets offene Ohr. Abschließend danke ich allen Freunden, die mir die Zeit in Bayreuth versüßt haben und die Stadt für mich zu einer Heimat gemacht haben.



# ZUSAMMENFASSUNG

Seit einigen Jahren rücken Roboter, die im Haushalt eingesetzt werden, immer mehr in den Fokus der breiten Öffentlichkeit. Auf dem Consumer-Markt befinden sich verschiedene solcher spezialisierter Roboter, wie beispielsweise ein Staubsaugerroboter. Die Vision ist, dass diese Roboter sich soweit entwickeln, dass sie auch Aufgaben in der menschlichen Umgebung übernehmen können, bei denen Objekte manipuliert werden müssen.

Diese Umgebung ist eine besondere Herausforderung. Sie verändert sich von Tag zu Tag. Der Roboter kommt ständig mit *unbekannten Objekten* in Kontakt. Daher ist die übergeordnete Fragestellung dieser Arbeit, inwieweit es möglich ist ausschließlich auf Basis der geometrischen Rekonstruktion eines unbekanntes Objektes dieses präzise zu greifen und stabil abzulegen, obwohl auf weitere Sensorik, wie beispielsweise eine Kraftmessdose zur Messung der auftretenden Kräfte und Momente, verzichtet wird. Für die Erfassung der Informationen über die Umwelt und die zu manipulierenden Objekte wird ausschließlich eine Tiefenkamera eingesetzt. Die Berechnungen sollen so schnell ausgeführt werden, dass diese während der Bewegungen des Roboters erfolgen können, ohne dass dieser auf die Ergebnisse warten muss.

Zum Erreichen der Zielsetzung werden in dieser Arbeit drei Verfahren vorgestellt: eine konservative Oberflächenrekonstruktion, ein präziser Greifplaner und ein Ablageplaner für nicht-ebene Ablagen.

Es wird ein Verfahren zur geometrischen Rekonstruktion vorgestellt, das auf Basis unvollständiger Sensordaten aus einer Tiefenkamera ein Oberflächenmodell berechnet, welches Oberflächenbereiche unterscheidet, die sensorisch validiert sind und solche die konservativ abgeschätzt sind. Durch die konservative Abschätzung können ungewollte Kollisionen zwischen Objekt und Roboter oder Umgebung vermieden werden. Die Unterscheidung der Oberflächenbereiche wird an Greif- und Ablageplaner weitergereicht, damit sie diese Information als Unsicherheit bezüglich des Planungsergebnisses berücksichtigen können.

Weiter wird ein Greifplaner vorgestellt, der mit Hilfe einer geschlossenen geometrischen Form Teilgeometrien des rekonstruierten Objektmodells approximiert. Aus der Parametrisierung dieser geometrischen Form berechnet sich die vollständige Greifervorkonfiguration für verschieden Mehrfingergreiferkinematiken. Nachdem der Konfigurationsraum des Greifers nicht a priori eingeschränkt wird, werden keine möglichen Griffe ausgeschlossen und damit die Möglichkeiten einer stabilen Ablage diesbezüglich nicht eingeschränkt.

Der entwickelte Ablageplaner berechnet ausgehend von einer gegebenen Lagebeziehung stabile Zielposen zusammen mit einer Sequenz von Kontaktzuständen. Dazu wird eine kontinuierliche Kontaktposenberechnung eingeführt, die es ermöglicht ausschließlich auf einem polygonalen Oberflächenmodell von einer Kontaktpose zu einer neuen Kontaktpose zu gelangen. Durch dieses Verfahren ist gleichzeitig auch sichergestellt, dass zwischen den Kontaktposen eine kollisionsfreie Objektbewegung möglich ist. Um die Stabilität einer Zielpose zu evaluieren wird eine Heuristik eingeführt, die auf Basis der Kollisionspunkte und -normalen und dem geometrischen Schwerpunkt des Objektmodells eine Stabilitätsbewertung vornimmt.

Mit einem prototypischen Hardwareaufbau wird experimentell evaluiert, inwieweit die drei eingeführten Verfahren harmonieren, um ohne sensorische Rückmeldung Objekte von unterschiedlich komplexer Geometrie zu greifen und in unterschiedlichen Ablageszenarien wieder präzise zu platzieren. Dazu wird evaluiert wieviele Kameraposen für diesen experimentellen Aufbau nötig sind, um eine ausreichend genaue Rekonstruktion der Objekte zu erstellen, so dass keine nachträgliche Korrektur der relativen Pose zwischen Greifer und Objekt nötig ist, um anschließend eine präzise Ablage des Objektes zu ermöglichen.



# ABSTRACT

During the last few years robots became increasingly present in the public domain. There are different models on the consumer market for specialized helper tasks like cleaning the floor. The vision is that personal robots gain the ability to do tasks that relate to object manipulation. In order to assist a human at tasks in the household or workshop. One of the main challenges is that a robot has to deal with an highly unstructured and continuously changing environment. Thus a robot needs the ability to manipulate unknown objects.

The general research question of the thesis is, to what extent a robot is able to precisely grasp objects and place them on non-flat placement areas while there is only the geometric reconstruction of the object and environment available. Hence there will be no use of force/torque information for the planning and execution phase. Only a depth camera is used to gain information of the objects and environment. There is another demand regarding the computation time. The computation processes need to be as fast as the robot needs not to stop its motion, because it must wait for results of the processing.

To achieve this aim, this thesis introduces a conservative geometric reconstruction algorithm, a precise grasp planner for multi-finger grippers and a placement planner that is able to compute object placement poses on non-flat placement areas.

The developed surface reconstruction algorithm deals with non-dense depth images. The reconstructed surface gets divided into sensor-validated patches and conservative estimated ones. This information is provided to the grasp planner and placement planner for them to use this uncertainty during the computation.

The grasp planner designed for multi-finger grippers makes use of a quadric to approximate parts of the complete surface model. The parameters of this form are used to compute the initial configuration of the gripper finger as well as the pose of the gripper base. Due to the dynamic pregrasp computation the amount of available grasps is enlarged. Thus the placement planner is not hindered by missing grasp possibilities.

The placement planner computes stable placement poses combined with sequences of contact poses for reconstructed surface models and placement areas. The planner can deal with different requested object pose relations and does not require force values for the contacts. The stability evaluation relies only on the contact points and contact normals together with the computed geometric center of mass assuming a homogeneous mass distribution.

Within real world experiments the developed algorithms are evaluated regarding a precise execution of various pick and place scenarios and different household objects. It is also discussed how many camera poses are required to be able not to use force information during the robot execution.

# INHALTSVERZEICHNIS

ZUSAMMENFASSUNG	i
ABSTRACT	iii
1 EINLEITUNG	1
1.1 Herausforderungen . . . . .	2
1.2 Aufgabenstellung und Abgrenzung . . . . .	3
1.3 Kapitelübersicht . . . . .	5
2 OBJEKT-REKONSTRUKTION	7
2.1 Anforderungen . . . . .	8
2.2 Stand der Forschung . . . . .	8
2.3 Algorithmus . . . . .	10
2.4 Experimente und Diskussion . . . . .	23
2.5 Schlussfolgerung . . . . .	32
3 KONTAKTPOSENBERECHNUNG	33
3.1 Anforderungen . . . . .	34
3.2 Stand der Forschung . . . . .	35
3.3 Ausgangssituation . . . . .	37
3.4 Lineare Kontaktposenberechnung . . . . .	38
3.5 Rotatorische Kontaktposenberechnung . . . . .	46
3.6 Experimente und Diskussion . . . . .	54
3.7 Schlussfolgerung . . . . .	62

4	GREIFPLANUNG	65
4.1	Stand der Technik . . . . .	65
4.2	Anforderungen . . . . .	71
4.3	Stand der Forschung . . . . .	72
4.4	Algorithmus . . . . .	78
4.5	Experimente und Diskussion . . . . .	101
4.6	Schlussfolgerungen . . . . .	113
5	ABLAGEPLANUNG	117
5.1	Taxonomie . . . . .	118
5.2	Anforderungen . . . . .	120
5.3	Stand der Forschung . . . . .	121
5.4	Algorithmus . . . . .	125
5.5	Experimente und Diskussion . . . . .	140
5.6	Schlussfolgerung . . . . .	149
6	PROTOTYP	151
6.1	Hardwareaufbau . . . . .	151
6.2	Systemarchitektur . . . . .	152
6.3	Experimente und Diskussion . . . . .	155
6.4	Schlussfolgerungen . . . . .	164
7	FAZIT	167
7.1	Zusammenfassung . . . . .	167
7.2	Ausblick . . . . .	169
	Literatur	171

# 1

## EINLEITUNG

Seit einigen Jahren rücken Roboter, die im Haushalt eingesetzt werden, immer mehr in den Fokus einer breiten Öffentlichkeit [AIS13]. Auf dem Markt befinden sich bereits zahlreiche Roboter verschiedener Firmen zum Reinigen von Fußböden, Rasen mähen und Pool reinigen, während Roboter zum Fenster wischen oder Überwachungsroboter bisher nur Nischenprodukte sind. Ein stark wachsender Markt sind die Unterhaltungs- und Spielzeugroboter. Ausgehend von diesen spezialisierten Helfern sollen sich *Personal Roboter* oder *Service Roboter* entwickeln. Eine Definition nach ISO 8373:2012 [ISO12] ist wie folgt:

**Definition 1.1.** A *service robot* is a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations.

Die Vision eines Personal Roboters im Haushalt lässt sich wohl am besten an einem Beispielszenario begreiflich machen (nach [AIS13]):

*Kathrin schlägt die Haustüre hinter sich zu, ihre Kinder hüpfen ins Auto. Sie sind spät dran für die Schule. In der Wohnung hinterlassen sie den Frühstückstisch in einem völligen Durcheinander. Im Wohnzimmer liegt Spielzeug der Kinder, Zeitschriften und Bücher herum. Der Fußboden im Bad ist nass vom Duschen und die Klamotten des letzten Tages hängen über der Badewanne. Der Tag geht zu Ende. Kathrin sperrt die Haustüre auf. Die Kinder stürmen jauchzend in die Wohnung und schleudern ihre Schulsachen in die Ecke. Aus der Küche duftet es nach Abendessen, der Tisch ist zum Abendessen gedeckt, die Wohnung aufgeräumt und geputzt. Nach dem Abendessen bleibt noch Zeit für die Kinder, dank des Personal Robots, der sich um den Haushalt gekümmert hat.*

Allerdings sind bis heute noch keine Roboter für Manipulationsaufgaben auf dem Konsumermarkt verfügbar, obwohl im industriellen Umfeld Roboter oftmals genau diese Aufgaben übernehmen. Wo sind die Unterschiede? Welche Funktionen

muss ein Personal Roboter im Bezug auf das Greifen und Ablegen von Objekten besitzen, um Aufgaben im Haushalt zu übernehmen?

### 1.1 HERAUSFORDERUNGEN

Das häusliche Umfeld ist eine besondere Herausforderung für Personal Roboter. Die Umgebung verändert sich von Tag zu Tag. Der Roboter kommt ständig mit *unbekannten Objekten* in Kontakt. Diese können deformierbar oder auch starr sein.

**Definition 1.2.** Ein Objekt ist unbekannt oder sensorisch-modelliert, wenn es ausschließlich durch Sensoren erfasst worden ist und aus diesen Sensorwerten rekonstruiert ist. Insbesondere ist über das Objekt keine Information über die Geometrie oder dessen Massenverteilung vorhanden.

Dennoch muss der Roboter mit diesen Objekten interagieren, sie greifen oder schieben, oder auf eine gewisse Art und Weise ablegen. Weiterführend sollte der Roboter in der Lage sein die unbekannt Objekte zu kategorisieren, um diese Einzelaktionen zu einer vollständigen Sequenz fusionieren zu können, wobei die Genauigkeit und Robustheit der Ausführung innerhalb der durch die Aufgabe definierten Toleranzen gewährleistet sein muss. Dabei sind die signifikanten Parameter bei der Manipulation von unbekannt Objekten die Größe der Objekte, deren geometrische Komplexität, Oberflächeneigenschaften und Dynamik [AIS14].

**Definition 1.3.** *Manipulation* bezeichnet den Prozess, Objekte geplant zu bewegen oder Objekte planvoll in der Umgebung zu positionieren. Für eine Manipulations- oder Handhabungsaufgabe besteht zwischen Roboter und Objekt ein physischer Kontakt [Kle+11].

Eine Grundvoraussetzung ist, dass der Roboter in der Lage ist, seine Umgebung und die unbekannt Objekte sensorisch wahrzunehmen. Das Ziel ist eine so exakte Rekonstruktion der unbekannt Objekte zu berechnen, dass schnelle und effiziente Greifplaner entwickelt werden können, die zusätzlich die Verwendung der Objektfunktionalität oder die gestellte Aufgabe mit berücksichtigen. Auf der anderen Seite sollen die Kosten möglichst gering gehalten werden. Dabei ist ein dominierender Faktor die Hardware-seitige Komplexität, welche klein bleiben

muss. Was wiederum bedeutet, dass die Anzahl der unterschiedlichen Sensoren und damit die verfügbaren Umweltinformationen begrenzt sind.

Auf Basis dieser ungenauen und limitierten sensorischen Informationen über die Objekte muss der Roboter die einzelnen Aktionen planen, um die gestellte Aufgabe zu erfüllen. Die Grundvoraussetzung, um eine Vielzahl von Aufgaben erfüllen zu können, ist, dass ein unbekanntes Objekt, nachdem es gegriffen worden ist, auch stabil abgelegt werden kann.

**Definition 1.4.** Ein Objekt ist stabil abgelegt, wenn es ohne äußere Krafteinwirkung, abgesehen von der Erdanziehungskraft, in Ruhe verharrt.

Dabei kann die Ablage beliebig kompliziert sein. Mit dieser Funktionalität wird es generell möglich Aufgaben wie das Einräumen einer Spülmaschine, das Aufräumen eines Tisches, oder das Ein- und Auspacken der Einkäufe zu realisieren.

Trotz der komplexen Anforderungen an die verschiedenen Funktionen eines Personal Roboters müssen die durchgeführten Berechnungen so effizient sein, dass der Roboter kontinuierlich seine Aufgaben durchführen kann, ohne dass er die Ausführung unterbrechen muss bis die verschiedenen Aktionsplanungen abgeschlossen sind.

Wünschenswert bezüglich der Verzahnung von Planung und Ausführung im Hinblick auf die Robustheit des Systems gegenüber Umwelteinflüssen ist außerdem, dass jederzeit zwischen diesen beiden Komponenten gewechselt werden kann. Damit könnten Pläne online basierend auf Umweltbeobachtungen aktualisiert, verfeinert und auf Fehler reagiert werden, und geplant werden wie der Roboter mehr über seine Umwelt erfahren kann.

## 1.2 AUFGABENSTELLUNG UND ABGRENZUNG

Um in Zukunft Personal Roboter zu entwickeln, die unbekannte Objekte (Definition 1.2) manipulieren können, wird in dieser Arbeit untersucht inwieweit es möglich ist Greif- und Ablageaufgaben ausschließlich unter Verwendung von Tiefenbildern einer kalibrierten Kamera zu realisieren. Insbesondere sollen keine Informationen aus Kraft-Momenten-Sensoren miteinbezogen werden. Die Einschränkung in Bezug auf die verwendeten Sensoren genügt dem Ziel mit möglichst wenigen Informatio-

nen auszukommen, um Hardware-seitig die Komplexität des Systems gering zu halten.

Daraus ergibt sich, dass sowohl die Greifplanung als auch die Ablageplanung ausschließlich mit den geometrischen Informationen arbeiten müssen, die mit Hilfe der gegebenen kalibrierten Tiefenkamera von der Umwelt gewonnen werden.

Um die Anzahl der Objekte, die manipuliert werden können, zu vergrößern, soll die Flexibilität eines Mehrfingergreifers ausgenutzt werden, damit möglichst viele unterschiedlich große und komplexe Geometrien manipuliert werden können und gleichzeitig die Anzahl möglicher Griffe pro Objekt vergrößert wird.

Um die Anzahl der realisierbaren Ablageaufgaben zu maximieren muss der Ablageplaner stabile Platzierungen der Objekte auf komplexen Ablagen, die insbesondere nicht eben sind, berechnen.

Die Ergebnisse der Planung sollen einfach für eine robuste Ausführung aufbereitet werden können und insbesondere dazu geeignet sein auf Störungen und sensorische Fehlinformationen reagieren zu können.

Wie in Kapitel 1.1 beschrieben besteht die generelle Anforderung, dass die Rechenzeit, die die Algorithmen benötigen, so gering ist, dass der Roboter die Ausführung der Aufgabe nicht unterbrechen muss, um auf die Ergebnisse der Planung zu warten.

**Definition 1.5.** Ein Planer ist *interaktiv*, wenn er ein verwertbares Ergebnis berechnet hat bis der Roboter dieses benötigt, um seine Bewegung nicht unterbrechen zu müssen.

Ein maßgeblicher Anteil an der Ausführungszeit einer Greif- und Ablageaufgabe ist das Erfassen der Umwelt. Wie schon erwähnt soll dies durch eine Tiefenkamera erfolgen. Damit ist die Anzahl der nötigen Aufnahmen ein direkter Zeitfaktor und muss gering gehalten werden.

Die obigen Anforderungen lassen sich in folgenden Fragestellungen zusammenfassen:

- H1 In wieweit lassen sich unvollständige geometrisch rekonstruierte Objekte konservativ vervollständigen und diese Informationen bei der Planung mit einbeziehen?



- H2 In wieweit lassen sich die Freiheitsgrade von modernen Mehrfingergreifern bei der Greifplanung von unbekanntem Objekten ausnutzen während die Planungszeit interaktiv bleibt?
- H3 In wieweit kann eine Ablage für unbekanntem Objekte ausschließlich basierend auf geometrischen Informationen auf (nicht-)ebenen Ablagen geplant und durchgeführt werden?
- H4 In wieweit lässt sich mit dem prototypischen Hardwareaufbau die Anzahl der benötigten Kamerabilder reduzieren, so dass die Greif- und Ablageaufgaben weiterhin erfüllbar bleiben?

**Definition 1.6.** Eine geometrische Rekonstruktion heißt *konservativ*, wenn aus den gegebenen Sensordaten ein geschlossenes Objektmodell rekonstruiert wird, welches das tatsächliche Objekt beinhaltet, unter der Annahme, dass der Sensor keine fehlerhaften Daten liefert.

Das verfügbare Vorwissen soll sich darauf beschränken, dass die Objekte auf einer horizontalen Fläche während der Rekonstruktion stehen und durch eine Segmentierung separiert werden können. Ebenso wird davon ausgegangen, dass die manipulierten Objekte starr sind. Weiter sei die Richtung der Gravitation bekannt. Ebenso wird davon ausgegangen, dass die Aufgabe bekannt ist und somit keine semantische Analyse oder grundsätzliche Ablaufplanung durchzuführen ist. Insbesondere ist eine sensorgestützte Ausführung nicht Bestandteil dieser Arbeit. Für die experimentelle Untersuchung der entwickelten Planer und der Rekonstruktion wird auf Standardansätze, wie eine Bahnplanung, zurückgegriffen.

### 1.3 KAPITELÜBERSICHT

In Kapitel 2 wird zunächst die Objektrekonstruktion (H1) vorgestellt und insbesondere eine einfache Möglichkeit, wie unvollständige Tiefenbilder vervollständigt werden können, damit sie in einem subtraktiven Rekonstruktionsverfahren eingesetzt werden können, und wie durch Sensordaten validierte und konservativ abgeschätzte Bereiche am Objektmodell unterschieden werden, um ein konservatives vervollständigtes dreidimensionales Modell zu erzeugen.

In Kapitel 3 wird ein Verfahren eingeführt wie man effizient Objekte kontinuierlich im Sinne eine kollisionsfreien Bewegung jedoch mit Kontakt zu einem anderen Objekt bewegen kann. Dieses Verfahren kommt in Kapitel 4 zum Einsatz, in dem der Greifplaner vorgestellt wird, der für Mehrfingergreifer variable Griffe unter Berücksichtigung der Unsicherheit der Rekonstruktion erzeugt (H2).

In Kapitel 5 wird der geometrische Ablageplaner eingeführt, der ausschließlich auf den rekonstruierten Objekten operiert. Es wird nicht nur eine heuristisch stabile Zielkonfiguration erzeugt, sondern auch ein Kontaktzustandsgraph unter Verwendung des Verfahrens aus Kapitel 3, der leicht in eine robuste Ausführung durch den Roboter überführt werden kann (H3).

In Kapitel 6 werden die Planer aus Kapitel 4 und 5 experimentell untersucht und die systemischen Abhängigkeiten zwischen den drei Komponenten Rekonstruktion, Greifplanung und Ablageplanung analysiert. Anhand des verwendeten Roboters und den Greif- und Ablageszenarien wird diskutiert wieviele Kameraperspektiven notwendig sind, um diese Szenarien durchführen zu können (H4).

In Kapitel 7 wird die Arbeit zusammengefasst, abschließende Folgerungen gezogen und ein Ausblick auf weiterführende Forschungsmöglichkeiten gegeben.

# 2

## OBJEKT-REKONSTRUKTION

Für die Manipulation von unbekanntem Objekten ist ihre Rekonstruktion der Ausgangspunkt. Die Rekonstruktion muss möglichst genau das reale Objekt modellieren und insbesondere sollten unvollständige Sensordaten die Qualität des Objektmodells nicht beeinflussen. Gerade Tiefenkameras liefern keine dichten Tiefenbilder.

Auf dem Gebiet der Oberflächenrekonstruktion basierend auf Tiefendaten oder Punktwolken hat in den letzten Jahren ein erheblicher Fortschritt stattgefunden. In [Ber+14] werden aktuelle Verfahren hinsichtlich der benötigten Eingaben, der Ausgabe und der Behandlung von Fehlern in der Eingabedaten (Abbildung 1) klassifiziert.

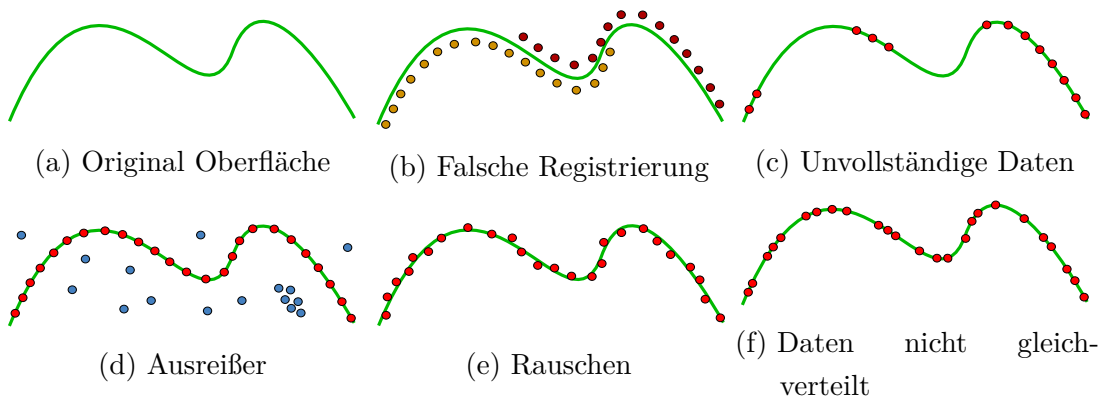


Abbildung 1: Illustration möglicher Fehlerquellen bei Eingabedaten aus [Ber+14]

Für die Objektrekonstruktion stehen unvollständige Tiefenbilder aus intrinsisch und extrinsisch kalibrierten Kameraposen zur Verfügung. Aus diesen Eingaben soll eine Menge von segmentierten Objekten rekonstruiert werden. Jedes Objektmodell soll durch eine Menge von Polygonen repräsentiert sein.

Das Kapitel gliedert sich wie folgt: Die Anforderungen an die Oberflächenrekonstruktion geeignet für eine Anwendung bei der Manipulation von unbekanntem

Objekten werden in Kapitel 2.1 formuliert. Anschließend stellt Kapitel 2.2 den Stand der Forschung auf diesem Gebiet vor. Kapitel 2.3 führt den entwickelten Algorithmus ein, der in Kapitel 2.4 experimentell untersucht wird. Abschließend werden in Kapitel 2.5 Schlussfolgerungen bezüglich des Rekonstruktionsalgorithmus gezogen.

## 2.1 ANFORDERUNGEN

Die gestellten Anforderungen an eine Oberflächenrekonstruktion zur Verwendung in Manipulationsplanungen sind: Bezüglich der Tiefenbilder soll das Verfahren insbesondere konservativ mit unvollständigen (Abbildung 1c), nicht gleich-verteilten (Abbildung 1f) und Sensordaten mit Ausreißern (Abbildung 1d) umgehen können und dennoch ein Oberflächenmodell in Echtzeit rekonstruieren (Kapitel 1.2). Gleichzeitig sollen die Abweichungen der Rekonstruktion im Bezug auf das tatsächliche Objekt gering bleiben.

Unsicherheiten bezüglich unvollständiger Tiefenbilder sollen explizit in dem rekonstruierten Oberflächenmodell enthalten sein, um auf diese während der Manipulationsplanung und Ausführung reagieren zu können.

Um die Abhängigkeit von dem Einsatzszenario zu reduzieren, soll die Anzahl der Parameter möglichst gering gehalten werden und deren Einfluss auf das Ergebnis der Rekonstruktion orthogonal sein, damit diese einfach angegeben werden können.

## 2.2 STAND DER FORSCHUNG

Aktuelle Ansätze zur Objektrekonstruktion auf Basis von Punktwolken und Tiefenbildern, die aus einer oder mehreren Kameraperspektiven eine Szene aufgenommen worden sind, werden in [Ber+14] zusammengefasst.

Oberflächenrekonstruktionen auf Basis von Punktwolken [Ame+01; Kaz+13; Kaz+06; Hop94] verwerfen wertvolle Informationen, die zusätzlich aus einem Tiefenbild extrahiert werden können. Darunter zählen die Strahlen ausgehend von der Kamera zu den dreidimensionalen Punkten, die zu einem Tiefenwert korrespondieren, und insbesondere die Information, dass auf dem Strahl bis zu diesem Punkt

freien Raum durchstreicht und auf dem verlängerten Strahl nach dem Punkt keine Information über den Freiraum extrahiert werden kann.

Ein Vergleich von Ansätzen zur Vervollständigung von Oberflächenmodellen ist in [Att+13] zu finden. Die Kombination aus einer klassischen Oberflächenrekonstruktion mit dem Ziel eine möglichst glatte Oberfläche zu bekommen und einem geeigneten Ansatz zum Schließen von Löchern ist als naive Lösung zur Erzeugung von Oberflächenmodellen aus unvollständigen Tiefenbildern zu sehen. Gerade die Rekonstruktionsalgorithmen besitzen eine Rechenzeit von mehreren Sekunden. Ein Rechenzeitvergleich ist in [Kaz+13] zu finden. Insbesondere ist mit diesem Vorgehen keine konservative Objektvervollständigung möglich.

Andere Verfahren, die heuristisch unvollständige Objektmodelle ergänzen, nehmen implizit die Vollständigkeit der Eingabedaten an [Kaz+06], verwenden geometrische Primitive [Li+11; Sch+09; Dun+13], wiederkehrende Muster in den Eingabedaten [Pau+08] oder Objektsymmetrien [Boh+11; Mar+10]. Diese Verfahren leisten keine Garantie bezüglich der Konservativität der Rekonstruktion.

Algorithmen, die von sich aus konservativ sind, sind Carving-Algorithmen. Sie schneiden iterativ aus einem definierten Ausgangsvolumen Bereiche weg, die nicht innerhalb des zu rekonstruierenden Objektes liegen. Diese Algorithmen basieren ursprünglich auf Farbkonsistenztests zwischen mehreren Farbbildern. Eine GPU-basierte Implementierung, die den geforderten Rechenzeiten entspricht, ist [Obe+14]. Carving-Algorithmen, die auch Tiefenbilder oder Punktwolken verarbeiten, sind [Sha+10; Lia+13; Kun+11]. In [Kli+14] wird ein *occupancy grid* erstellt und dieses mittels des Marching Cubes Algorithmus in ein Oberflächenmodell überführt. In [Gug+12] wird statt des *occupancy grids* eine *signed distance function* verwendet. Deren Ansatz benötigt zusätzlich die Normaleninformationen auf der Oberfläche und ist anfällig gegenüber unvollständigen Tiefenbildern. Ebenso wird in [Mul+10] eine Implizite Funktion aus Punktwolken erzeugt, die mittels einer Rekonstruktionsmethode basierend auf Isoflächen in ein Oberflächenmodell überführt werden kann. Diese Ansätze lassen sich vom prinzipiellen Vorgehen wie folgt zusammenfassen: Aus Punktwolken oder Tiefenbildern wird eine Volumenrepräsentation (in Form eines *occupancy grids* oder einer *signed distance function*) erstellt, die mittels einer Isoflächen-Rekonstruktion zu einem Oberflächenmodell transformiert wird. Keiner der vorgestellten Algorithmen verfügt über eine Rechenzeit von unter einer Minute. Das Erstellen der Volumenrekonstruktion ist dabei die zeit-intensivste

Komponente. Der in [Iza+11] vorgestellte GPU-basierte Ansatz ermöglicht die Rekonstruktion von Oberflächenmodellen in wenigen Millisekunden. Das Ausschlusskriterium für diesen Algorithmus ist jedoch, dass es auf Grund unvollständiger Tiefenbilder zu Auslöschungen kommt [Kai+12].

Um das Erstellen einer Volumenrekonstruktion zu beschleunigen, sind verschiedene Varianten einer Umsetzung auf Grafikkhardware [Soa+07; Big+09] beziehungsweise mit Hilfe eines Multi-Computer-Setups [Lad+08] vorhanden. Obgleich diese Ansätze nicht mit dem Ziel entwickelt worden sind ein Oberflächenmodell zu erstellen, reihen sie sich als Komponente in den oben beschriebenen Ansatz unter Zuhilfenahme einer Isoflächen-Rekonstruktion ein. Der Ansatz von [Wer+14] liefert selbst in einer reinen CPU-basierten Implementierung konkurrenzfähige Rechenzeiten mit weit höherer Auflösung des Rekonstruktionsvolumens. Daher wird im Folgenden dieser Ansatz verwendet und zu einer Oberflächenrekonstruktion vervollständigt.

### 2.3 ALGORITHMUS

Dieses Kapitel führt die geometrische Volumenrekonstruktion entwickelt von Werner und Henrich [Wer+14] basierend auf der Arbeit von Stefan Kuhn [Kuh12] ein und erweitert diese, so dass Oberflächenmodelle erzeugt werden. Sie unterstützt eine hierarchische, inkrementelle, echtzeitfähige, anytimefähige und konservative online Rekonstruktion unter Verwendung von segmentierten Bildern oder Tiefenbildern aus intrinsisch und extrinsisch kalibrierten Kameras. Als Eingabebilder werden in der vorliegenden Verwendung ausschließlich nicht segmentierte Tiefenbilder verwendet.

Für die Verwendung von realen Tiefenbildern und die Erzeugung von konservativen Oberflächenmodellen sind Ergänzungen nötig, die in diesem Kapitel zusätzlich betrachtet werden. Weiter wird die Volumenrekonstruktion in eine Oberflächenrekonstruktion transformiert, die die Besonderheit hat, dass die Oberflächensegmente basierend auf der Art und Weise ihrer Entstehung markiert werden, um diese Information direkt in die Planungsalgorithmen einfließen zu lassen.

Im Folgenden werden zuerst die Anforderungen und Annahmen für die Eingaben und resultierenden Objektmodelle benannt (Kapitel 2.3.1). Anschließend wird auf die Hauptkomponenten der Volumenrekonstruktion basierend auf Octrees und

Quadtrees eingegangen (Kapitel 2.3.2) und die benötigte Tiefenbildvervollständigung beschrieben (Kapitel 2.3.3). Im letzten Schritt wird mit Hilfe eines Marching-Cubes Algorithmus [Lor+87] aus der Octree-Repräsentation der segmentierten und verfeinerten Szene eine für die Planungsalgorithmen effizientes Oberflächenmodell und Kollisionsmodell pro Segment erstellt. Darin inbegriffen werden die erzeugten Dreiecke bezüglich der Label der Octree-Knoten klassifiziert (Kapitel 2.3.4).

Die Parameter für die Rekonstruktion sind die maximale Tiefe des Octrees und die Auflösung des regulären Gitters. Die Tiefe des Octrees bestimmt neben der Genauigkeit der Volumenrekonstruktion die minimale Distanz zwischen zwei separierbaren Objekten. Die Auflösung des regulären Gitters bestimmt die Anzahl der Dreiecke der Rekonstruktion und die maximale Abweichung der Oberflächenrekonstruktion von der Volumenrekonstruktion.

Sowohl die beiden hierarchischen Datenstrukturen (Octree und Quadtree) als auch der Marching-Cubes Algorithmus sind hinreichend studiert worden [Mea82; Wat99; Lor+87]. Deshalb beschränkt sich die Beschreibung in diesem Kapitel ausschließlich auf die für das Verfahren relevanten Aspekte.

### 2.3.1 Annahmen

Für eine multi-view Rekonstruktion des Arbeitsraumes eines Roboters werden folgende Annahmen getroffen. Der Arbeitsraum sei ein dreidimensionaler Quader  $V = [0, 1]^3 \subset \mathbb{R}^3$ . Teile des Arbeitsraumes sind durch a priori unbekannte Objekte belegt. Das Volumen  $V_{obj,t} \subseteq V$  repräsentiert das tatsächliche belegte Volumen innerhalb des Arbeitsraumes zu einem diskreten Zeitpunkt  $t \in \mathbb{N}$ . Gesucht ist eine konservative Rekonstruktion  $V_{rec,t}$  dieser Objekte damit  $V_{obj,t} \subseteq V_{rec,t} \subseteq V$  gilt.

Dazu beobachtet man den Arbeitsraum aus  $c$  Kameras beziehungsweise Kameraperspektiven. Jede Kamera  $i$  hat eine quadratische Auflösung  $r_i = 2^k$ ,  $k \in \mathbb{N}$  mit den Pixeln  $R_i = [1, r_i]^2 \subset \mathbb{N}^2$ . Eine nicht quadratische Kameraauflösung kann ohne Beschränkung der Allgemeinheit auf eine quadratische erweitert werden indem man die Randpixel dupliziert.

Zu jedem Zeitpunkt  $t$  stellt die Kamera  $i$  ein Tiefenbild als Abbildung  $\lambda_{t,i} : R_i \rightarrow D_{dep} \cup \{\emptyset\}$  mit  $D_{dep} = \mathbb{R}^+$  bereit. Unter der Annahme, dass die verwendete Kamera keine Detektionsfehler (beispielsweise durch spiegelnde Oberflächen) vorweist, existieren konstruktionsbedingt Pixel  $p \in R_i$ , für die gilt  $\lambda_{t,i}(p) = \emptyset$ . Um die

Konservativität der Rekonstruktion ( $V_{obj,t} \subseteq V_{rec,t}$ ) zu garantieren, werden diese Löcher konservativ im Tiefenbild  $\hat{\lambda}_{t,i} : R_i \rightarrow D_{dep}$  geschlossen (Kapitel 2.3.3), so dass gilt:  $\forall p \in R_i : \hat{\lambda}_{t,i}(p) \neq \emptyset$ .

Jede Kamera ist sowohl intrinsisch als auch extrinsisch kalibriert. Das bedeutet, dass eine Projektion  $\varphi_i : V \rightarrow R_i \times \mathbb{R}$  vom Arbeitsraum in die Tiefenkoordinaten für die Kamera  $i$  existiert. Normalerweise ziehen Verzerrungen des Kamerabildes rechenaufwendige pixel-exakte Projektionen nach sich. Dieser Mehraufwand wird vermieden, in dem die Projektion durch eine konservative Abschätzung  $\tilde{\varphi}_i : 2^V \rightarrow 2^{R_i \times \mathbb{R}}$  ersetzt wird<sup>1</sup>:

$$\tilde{\varphi}_i(\tilde{V}) = \left[ \left( \begin{array}{c} \min_{v \in \tilde{V}}(\varphi_{i,x}(v)) \\ \min_{v \in \tilde{V}}(\varphi_{i,y}(v)) \\ \min_{v \in \tilde{V}}(\varphi_{i,z}(v)) \end{array} \right), \left( \begin{array}{c} \max_{v \in \tilde{V}}(\varphi_{i,x}(v)) \\ \max_{v \in \tilde{V}}(\varphi_{i,y}(v)) \\ \max_{v \in \tilde{V}}(\varphi_{i,z}(v)) \end{array} \right) \right]$$

Lochkameraprojektionen erhalten konvexe Grenzen eines konvexen Polyeder innerhalb des Sichtkegels der Kamera. Eine solche konservative Projektionsfunktion berechnet die Boundingbox im Bildraum  $R_i$ . Komplexere Kameramodelle erfordern weitere Maßnahmen, um die Konservativität sicher zu stellen.

Abgesehen von den formalen Anforderungen soll die Differenz  $V_{rec,t} \setminus V_{obj,t}$ , bei einer Eingabebildaufösung äquivalent einer FullHD Auflösung, möglichst klein sein und dennoch den Echtzeit- und Anytimeanforderungen genügen.

Für die wissensbasierte Verfeinerung und Segmentierung der Rekonstruktion  $V_{rec,t}$  wird angenommen, dass die Objekte in der Szene vereinzelt auf einem Tisch stehen. Die Segmentierung  $\phi : 2^{V_{rec}} \rightarrow 2^{2^{V_{rec}}}$ ,  $\varphi(V_{rec,t}) = \{V_{rec,t}^k | V_{rec,t}^k \subseteq V_{rec,t}\}$  mit  $\bigcup_k V_{rec,t}^k = V_{rec,t}$  und  $V_{rec,t}^i \cap V_{rec,t}^j = \emptyset$  für  $i \neq j$  teilt die Rekonstruktion in Teilrekonstruktionen  $V_{rec,t}^k$  auf.

### 2.3.2 Volumenrekonstruktion

Dieser Abschnitt beschreibt die Volumenrekonstruktion reduziert auf die Verwendung von Tiefenkameras ohne vorsegmentierte Bilder.

Der generelle Ablauf der Rekonstruktion erfolgt in drei Schritten: Zuerst werden die konservativen Quadrees für alle Eingabebilder berechnet. Im zweiten Schritt wird inkrementell ein konservativer Octree der observierten Szene erstellt. Darin

<sup>1</sup>  $2^X$  bezeichnet die Potenzmenge von  $X$ .



inbegriffen ist die Projektion der Octree-Knoten in die Eingabebilder und das Markieren dieser Knoten basierend auf den Bildinformationen aus den Quadrees. Im dritten Schritt wird mit einer einfachen wissensbasierten Verfeinerung die Rekonstruktion verbessert und in einzelne Objekte zerlegt.

### Quatree-Aufbau

Im Folgenden wird der Aufbau der Quadrees  $q_{t,i}$  für die Tiefenbilder  $\hat{\lambda}_{t,i}$  von Kamera  $i$  zum Zeitpunkt  $t$  beschrieben.

Ein Quatree  $q_{t,i}$  für ein Bild  $\hat{\lambda}_{t,i}$  der Auflösung  $r_i^2$  hat  $h_i = \log_2(r_i) + 1$  Ebenen, wobei die unterste Ebene  $q_{t,i,0} : R_i \rightarrow D_{dep}, q_{t,i,0}(r) = \hat{\lambda}_{t,i}(r)$  entspricht. Die Quadrees werden ausgehend von den Bildern von unten nach oben aufgebaut. Dazu definiert man eine Mergefunktion  $m_{dep} : D_{dep}^4 \rightarrow D_{dep}$  um ein Quadrat aus vier Einträgen zu einem neuen Eintrag innerhalb der nächst höheren Ebene zusammenzufassen. Dies führt zu Quatree-Ebenen  $1 \leq j < h_i$ ,

$$q_{t,i,j} : \left[1, \frac{r_i}{2^j}\right]^2 \rightarrow D_{dep}$$

mit

$$q_{t,i,j}(r) = m_{dep}\left(q_{t,i,j-1}(2r), q_{t,i,j-1}(2r + (1, 0)^T), q_{t,i,j-1}(2r + (0, 1)^T), q_{t,i,j-1}(2r + (1, 1)^T)\right).$$

Für Tiefenbilder erhält eine Minimumfunktion als Mergefunktion schon die Objektgrößen. Zum Zweck späterer Optimierungen werden sowohl die minimalen als auch die maximalen Abstände zur Kamera innerhalb eines Quatree-Knotens gespeichert. Daher wird  $D_{dep} = \mathbb{R}^2$  verwendet. Für Octree-Knoten auf Ebene  $h_i$  sind beide Einträge gleich dem Pixeltiefenwert. Die Mergefunktion

$$m_{dep}\left(\left(\begin{pmatrix} d_{1,min} \\ d_{1,max} \end{pmatrix}, \dots, \begin{pmatrix} d_{4,min} \\ d_{4,max} \end{pmatrix}\right)\right) = \begin{pmatrix} \min(d_{1,min}, \dots, d_{4,min}) \\ \max(d_{1,max}, \dots, d_{4,max}) \end{pmatrix}$$

erzeugt alle weiteren Ebenen im Quatree.

Zuletzt werden Pixel in einem separaten Quatree  $q_{mod,t,i}$  markiert, die sich zum Zeitpunkt  $t$  im Bezug auf den Zeitpunkt  $t - 1$  verändert haben.

*Octree-Aufbau*

Aktualisierungen des Octrees zu einem Zeitpunkt  $t$  erfolgen sobald der Daten-Quadtree  $q_{t,i}$  und der Modifikations-Quadtree  $q_{mod,t,i}$  für jedes Eingabebild  $i$  aufgebaut sind.

Im Allgemeinen besteht ein dünn besetzter Octree aus einer Menge von Knoten  $N$ . Jeder Knoten  $n \in N$  belegt einen bestimmten Volumenbereich  $V_n = [n_{\min}, n_{\max}] \subset V$  mit  $n_{\min}, n_{\max} \in \mathbb{R}^3$ . Die Funktion

$$\sigma_t : N \rightarrow 2^N \tag{1}$$

definiert die Eltern-Kind-Beziehung zwischen den Knoten zu dem Zeitpunkt  $t$ . Im vorliegenden Fall speichert ein Knoten zusätzlich einen zeitabhängigen Zustand  $s_{t,n} \in S = \{full, empty, mixed\}$ . Voll und leer markierte Knoten müssen nicht weiter unterteilt werden, da sie einen homogenen Zustand darstellen. Diese Knoten bilden die Blattknoten des Octrees. Knoten, die als *mixed* markiert sind, werden weiter unterteilt, um die Genauigkeit der Rekonstruktion zu erhöhen. Im Startzustand besitzt der Octree einen einzigen Wurzelknoten  $n_{root}$  mit  $V_{n_{root}} = V$ ,  $\sigma_0(n_{root}) = \emptyset$  und  $s_{0,n_{root}} = mixed$ .

Für die inkrementellen Aktualisierungen kommuniziert der Octree mit jedem Quadtree  $i$  über eine Entscheidungsfunktion  $\tau_{dep,t,i} : N \rightarrow S$ . Diese Funktion identifiziert relevante Knoten im Quadtree und entscheidet welchen Zustand der Octreeknotten zugewiesen bekommt. Um die Konservativität zu erhalten muss ein Knoten  $n$  als *mixed* oder *full* markiert werden, wenn dessen Volumenbereich  $V_n$  von keiner Kamera erfasst wird oder durch andere näher an einer Kamera befindlichen Objekte verdeckt wird. Volumenbereiche, die nur teilweise von Kameras gesehen werden, werden als *mixed* markiert. Solche zusammenhängende Volumenbereiche werden als Scheinobjekte bezeichnet.

Zuerst wird eine Hilfsfunktion  $\hat{\tau}_{dep,t,i,j} : N \rightarrow S$  eingeführt, die auf einer einzigen Quadtreeebene  $j$  operiert. Das Volumen  $V_n$  eines Octree-Knotens wird gegen die minimale und maximale Tiefe im Quadtree getestet:

$$\hat{\tau}_{dep,t,i,j}(n) = \begin{cases} \text{empty} & \text{if } \forall (d_x, d_y, d_z) \in \tilde{\varphi}_i(V_n) : \\ & \max_{V_n} (\tilde{\varphi}_{i,z}(V_n)) < q_{t,i,j,min}((d_x, d_y)/2^j), \\ \text{full} & \text{if } \forall (d_x, d_y, d_z) \in \tilde{\varphi}_i(V_n) : \\ & \min_{V_n} (\tilde{\varphi}_{i,z}(V_n)) > q_{t,i,j,max}((d_x, d_y)/2^j), \\ \text{mixed} & \text{sonst,} \end{cases} \quad (2)$$

Die Werte  $q_{t,i,j,min}((d_x, d_y)/2^j)$  beziehungsweise  $q_{t,i,j,max}((d_x, d_y)/2^j)$  stehen für den minimalen oder maximalen Wert der Funktionsauswertung:

$$q_{t,i,j}((d_x, d_y)/2^j) = \begin{pmatrix} q_{t,i,j,min}((d_x, d_y)/2^j) \\ q_{t,i,j,max}((d_x, d_y)/2^j) \end{pmatrix}$$

Ausschließlich Tests auf der untersten Ebene des Quadrees können *mixed* zurückgeben. Die Funktion  $\tilde{\tau}_{t,i,j} : N \rightarrow S$ ,

$$\tilde{\tau}_{t,i,j}(n) = \begin{cases} \text{empty} & \text{if } j < h_i \wedge \tilde{\tau}_{t,i,j+1}(n) = \text{empty}, \\ \text{full} & \text{if } j < h_i \wedge \tilde{\tau}_{t,i,j+1}(n) = \text{full}, \\ \hat{\tau}_{dep,t,i,j}(n) & \text{sonst,} \end{cases}$$

modelliert diese optimierten Tests.

Um die Effizienz zu steigern wird die Entscheidungsfunktion nur auf Quadtree-Knoten angewendet, die sich zum aktuellen Zeitpunkt verändert haben. Damit ist die finale pro-Bild Entscheidungsfunktion  $\tau_{t,i}$  vollständig,

$$\tau_{0,i}(n) = \tilde{\tau}_{0,i,0}(n)$$

$$\tau_{t,i}(n) = \begin{cases} \tau_{t-1,i}(n) & \text{if } \exists j : \forall (d_x, d_y, d_z) \in \tilde{\varphi}_i(V_n) : \\ & q_{mod,t,i,j}((d_x, d_y)/2^j) = \text{unmodified}, \\ \tilde{\tau}_{t,i,0}(n) & \text{sonst.} \end{cases}$$

Die inkrementellen Aktualisierungen der Octreeknotten stoßen die Entscheidungsfunktion  $\tau_{t,i}$  pro Kamerabild an, um die Markierung für einen Knoten festzulegen.

Sobald eine Kamera garantiert, dass ein Knoten *empty* ist, stoppt der rekursive Abstieg und alle Kindknoten werden entfernt. Gleiches gilt falls alle Kameras einen Knoten als *full* markieren. Formal wird ein Knotenzustand wie folgt berechnet:

$$s_{t,n} = \begin{cases} \textit{empty} & \text{if } \exists i : \tau_{t,i}(n) = \textit{empty}, \\ \textit{full} & \text{if } \forall i : \tau_{t,i}(n) = \textit{full}, \\ \textit{mixed} & \text{sonst.} \end{cases}$$

Zu jedem Zeitpunkt der Rekonstruktion bilden alle Blattknoten des Octrees, die nicht leer sind, das rekonstruierte Volumen  $V_{rec,t}$ . Mit fortschreitender Unterteilung der gemischten Knoten wird die Rekonstruktion genauer.

Mit der Anwendung der Entscheidungsfunktionen wird deutlich, dass es unbedingt erforderlich ist, dass unvollständige Tiefenbilder mit möglichst geringere Differenz zur Realität gefüllt werden. Wenn Pixel ohne Tiefeninformation als *empty* markiert werden würden, ist in jedem Fall die Konservativität nicht mehr gewährleistet. Im Allgemeinen kommt es bei mehreren Kameras zu Auslöschungen, so dass im schlimmsten Fall keine Rekonstruktion existiert:  $V_{rec,t} = \emptyset$ . Falls alle Pixel ohne Tiefeninformation als *full* markiert werden, ist der Vorteil der Tiefeninformationen, dass Scheinobjekte nur hinter Verdeckungen entstehen können, negiert.

### *Segmentierung und Selektion*

Die Volumenrekonstruktion  $V_{rec,t}$  stellt eine konservative Abschätzung des realen eingenommen Volumens  $V_{obj,t}$  des Arbeitsraumes  $V$  dar. Die Differenz  $V_{rec,t} - V_{obj,t}$  enthält Scheinobjekte und die Ränder der realen Objekte, die auf Grund der diskretisierten Darstellung des Volumens enthalten sind. Mittels Informationen über den Inhalt der aufgenommenen Szene wird diese Differenz heuristisch verkleinert.

Im ersten Schritt wird dazu die Volumenrekonstruktion  $V_{rec,t}$  mittels der gegebenen Segmentierung  $\phi$  (Kapitel 2.1) in unabhängige Teilrekonstruktionen  $V_{rec,t}^k$  zerlegt.

Im Folgenden werden die verwendeten boolesche Entscheidungsfunktionen eingeführt, die eine Teilrekonstruktion auf Grund der Nichterfüllung eines Kriteriums verwerfen.

Es wird angenommen, dass alle Objekte auf einer ebenen Fläche repräsentiert durch eine Ebene  $E \subset \mathbb{R}^3$  stehen. Damit muss es einen Knoten  $n \in N_{V_{rec,t}^k}$  in den

Octreekonten von  $V_{rec,t}^k$  geben, der Kontakt mit dieser Ebene hat. Formal ist diese Entscheidungsfunktion wie folgt definiert:

$$\chi_E(V_{rec,t}^k) = \exists n \in N_{V_{rec,t}^k} : V_n \cap E \neq \emptyset \wedge s_{t,n} \neq \text{empty}$$

Wenn das Volumen eines Segments zu klein ist, wird es ebenso verworfen. Das ist in folgender Entscheidungsfunktion realisiert:

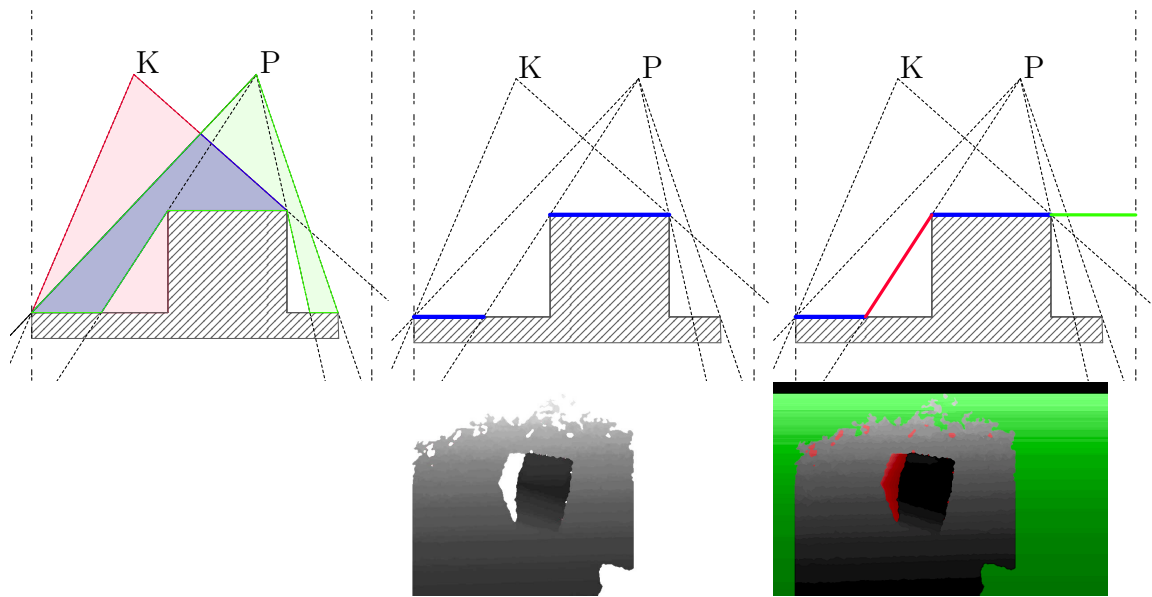
$$\chi_{\text{vol}}(V_{rec,t}^k) = \sum_{V_n \in V_{rec,t}^k} V_n > \epsilon_{\text{vol}}$$

### 2.3.3 Tiefenbildvervollständigung

Dieses Unterkapitel beschreibt eine konservative Vervollständigung der Rohtiefenbilder  $\lambda_{t,i}$ , die zu vollständigen Tiefenbildern  $\hat{\lambda}_{t,i}$  führt.

Die Funktionsweise von Tiefenkameras lässt sich in Verfahren basierend auf Triangulation oder der Messung der Signallaufzeit (time-of-light, kurz: ToF) aufteilen. Durch die Optik wird die Tiefe nicht parallel, sondern perspektivisch auf den Sensor und damit auf die Bildebene projiziert. Bei der Triangulation wird die Szene entweder durch zwei räumlich getrennte Kameras oder eine Kamera zusammen mit einem Projektor, der ein Muster auf die Szene projiziert, aufgenommen. Bei ToF Kameras besteht ebenso eine räumliche Trennung zwischen Lichtemitter und Lichtempfänger. Selbst wenn keine räumliche Trennung besteht, entstehen auf Grund der perspektivischen Projektion Verdeckungen. Im Folgenden wird ohne Einschränkung der Allgemeinheit angenommen, dass ein Projektor und eine Kamera verwendet werden, um ein Tiefenbild zu berechnen.

Diese räumliche Trennung hat Verdeckungen in der Szene zur Folge. Innerhalb einer Verdeckung kann kein Tiefenwert ausgerechnet werden, da entweder die Kamera diesen Bereich nicht einsehen kann oder der Projektor kein Muster an dieser Stelle erzeugt. In Abbildung 2a sind diese Verdeckungen exemplarisch dargestellt. Nur innerhalb des blauen Schnittvolumens zwischen dem Kamerasichtkegel (rot) und dem Projektionskegel (grün) können Tiefenwerte berechnet werden. In Abbildung 2b sind die Oberflächenbereiche blau eingefärbt, für die eine Tiefe im Bild berechnet werden kann. Diese blauen Bereiche sind nicht miteinander verbunden, was zur Folge hat, dass Pixel im Tiefenbild existieren, denen kein Wert zugewiesen werden kann.



- (a) Darstellung des Schnittvolumens (blau) der Sichtkegel des Projektors (grün) und der Kamera (rot)
- (b) Darstellung des Tiefenbildes  $\lambda_{t,i}$  (blau) das durch Triangulation erzeugt werden kann
- (c) Darstellung des vervollständigten Tiefenbildes  $\hat{\lambda}_{t,i}$  mit geschlossenen inneren Löchern (rot) und Randlöchern (grün)

Abbildung 2: Schematische Darstellung der Bauweise bedingten Verdeckungen bei der Aufnahme von Tiefenbildern mit Hilfe von Triangulation: Es können nur Tiefenwerte berechnet werden, wenn sich Sichtstrahlen von Projektor (P) und Kamera (K) auf der Oberfläche des aufgenommenen Objektes schneiden.

Um die Pixel  $p \in R_i, \lambda_{t,i}(p) = \emptyset$  des Tiefenbildes zu füllen, für die kein Wert auf Grund der Bauart des Sensors kein Wert berechnet werden kann, macht man sich zur Nutze, dass bei einer verzerrungsfreien Projektion alle Punkte auf einem Strahl vom Projektionszentrum der Kamera aus auf den gleichen Pixel abgebildet werden.

Im Folgenden sei angenommen, dass der Abstand zweier korrespondierender Bildpunkte bei der Triangulation ausschließlich die Differenz der x-Koordinaten in Pixel-Koordinaten ist. Demzufolge ist ein in die Kamera projizierter Strahl des

Projektors parallel zu der x-Achse des Bildes. Aus dieser Annahme folgt, dass man unvollständige Tiefenbilder zeilenweise bezüglich der x-Koordinate vervollständigen kann. Falls diese Annahme nicht zutrifft, muss anstatt entlang einer Bildzeile entlang einer Epipolarline vervollständigt werden.

Zur Umsetzung der Vervollständigung wird zwischen inneren Löchern und Randlöchern unterschieden. Löcher sind zusammenhängende Bereiche im Bild deren Pixel keine Werte besitzen.

**Definition 2.1.** Ein Pixel  $p \in R_i$  mit  $\lambda_{t,i}(p) = \emptyset$  liegt genau dann in einem inneren Loch, wenn es Pixel  $\underline{p} \in R_i$  und  $\bar{p} \in R_i$  mit folgenden Eigenschaften gibt:

- a)  $\lambda_{t,i}(\underline{p}) \neq \emptyset$
- b)  $\lambda_{t,i}(\bar{p}) \neq \emptyset$
- c)  $p_y = \underline{p}_y = \bar{p}_y \wedge \underline{p}_x < p_x < \bar{p}_x$
- d)  $\forall p' \in ]\underline{p}_x, \bar{p}_x[ \times \{p_y\} \subset R_i : \lambda_{t,i}(p') = \emptyset$

**Definition 2.2.** Ein Pixel  $p \in R_i$  mit  $\lambda_{t,i}(p) = \emptyset$  liegt genau dann in einem Randloch, wenn nur einer der beiden Pixel  $\underline{p} \in R_i$  und  $\bar{p} \in R_i$  aus Definition 2.1 existiert.

Seien die Funktionen

$$\underline{\rho}_{t,i} : R_i \rightarrow R_i \cup \{\emptyset\}, \quad \underline{\rho}_{t,i}(p) := \underline{p}$$

und

$$\bar{\rho}_{t,i} : R_i \rightarrow R_i \cup \{\emptyset\}, \quad \bar{\rho}_{t,i}(p) := \bar{p}$$

die beiden Auswahlfunktionen, die für einen gegebenen Pixel  $p \in R_i$  die Pixel  $\underline{p}$  und  $\bar{p}$  gemäß Definition 2.1 berechnen. Bei inneren Löchern werden die fehlenden Funktionswerte linear zwischen  $\lambda_{t,i}(\underline{p})$  und  $\lambda_{t,i}(\bar{p})$  interpoliert. Randlöchern werden zeilenweise mit dem letzten gültigen Tiefenwert gefüllt. Aus dieser Eigenschaft resultiert, dass die Funktionswerte beziehungsweise das vervollständigte Tiefenbild  $\hat{\lambda}_{t,i}$  für alle  $p \in R_i$  wie folgt berechnet wird:

$$\hat{\lambda}_{t,i}(p) := \begin{cases} \lambda_{t,i}(p), & \lambda_{t,i}(p) \neq \emptyset \\ \lambda_{t,i}(\underline{p}) + \frac{1 + \bar{p}_x - p_x}{\bar{p}_x - \underline{p}_x} (\lambda_{t,i}(\bar{p}) - \lambda_{t,i}(\underline{p})), & \lambda_{t,i}(p) = \emptyset \wedge \bar{\rho}_{t,i}(p) \neq \emptyset \wedge \underline{\rho}_{t,i}(p) \neq \emptyset \\ \lambda_{t,i}(\underline{p}), & \lambda_{t,i}(p) = \emptyset \wedge \bar{\rho}_{t,i}(p) = \emptyset \wedge \underline{\rho}_{t,i}(p) \neq \emptyset \\ \lambda_{t,i}(\bar{p}), & \lambda_{t,i}(p) = \emptyset \wedge \bar{\rho}_{t,i}(p) \neq \emptyset \wedge \underline{\rho}_{t,i}(p) = \emptyset \\ 0, & \text{sonst.} \end{cases}$$

In Abbildung 2c ist die konservative Vervollständigung des Tiefenbildes illustriert.

Die Interpolation erhält die Konservativität, da sie eine lokale visuelle Hülle berechnet. Wenn dies nicht der Fall wäre müsste ein Pixel  $\tilde{p} \in ]\underline{p}_x, \bar{p}_x[ \times \{p_y\}$  existieren mit  $\lambda_{t,i}(\tilde{p}) \neq \emptyset$ . Der zu  $\tilde{p}$  korrespondierende Punkt  $\varphi_i^{-1}(\tilde{p})$  im Arbeitsraum müsste aus der visuellen Hülle des Objektes herausragen, was nach Definition dieser Hülle nicht möglich ist [Lau94], da sie selbst eine konservative Abschätzung darstellt.

#### 2.3.4 Oberflächenrekonstruktion

In diesem Unterkapitel wird beschrieben wie eine selektierte Teilrekonstruktion  $V_{rec,t}^k$  in ein Oberflächenmodell transformiert wird, das für die Anwendung in den geometrischen Planungsalgorithmen effizienter als die bis jetzt verfügbare Volumenrekonstruktion oder allgemein Punktwolken verwendet werden kann.

Um eine Volumenrekonstruktion in ein Oberflächenmodell umzuwandeln stehen verschiedene Verfahren, die auf Isoflächen basieren, zur Verfügung. Bei diesen Verfahren wird eine Oberfläche innerhalb eines gewissen Raumbereichs bezüglich einer gegebenen Indikatorfunktion, die angibt ob ein Punkt innerhalb oder außerhalb des Objektes liegt, oder ein vorzeichenbehaftetes Distanzfeld erzeugt. Diese impliziten Funktionen operieren gewöhnlich auf einer regulären oder hierarchischen Raumunterteilung.

Die Verwendung des Marching-Cubes (MC) Alorithmus [Lor+87] bietet gegenüber anderen Verfahren verschiedene Vorteile. Seine Berechnungszeit und Genauigkeit hängt ausschließlich von der Anzahl der Gitterzellen ab und ist insbesondere unabhängig von der Tiefe des Octrees aus der Volumenrekonstruktion. Der Fehler der Rekonstruktion ist abschätzbar. Die Rekonstruktion ist hinreichend glatt für eine Greif- und Ablageplanung und das berechnete Oberflächenmodell ist intrinsisch konsistent und wasserdicht.

Für die Anwendung des Marching-Cubes Algorithmus muss das reguläre Gitter  $\mathcal{G} \subset V_{rec}$  mit der Menge von Gitterzellen  $G$  festgelegt werden. Eine Gitterzelle  $g \in G$  belegt einen Volumenbereich  $V_g = [g_{\min}, g_{\max}] \subset \mathcal{G}$ . Weiter sei  $I_G = [1, i_{G,x}] \times [1, i_{G,y}] \times [1, i_{G,z}] \subset \mathbb{N}^3$  die Indexmenge der Gitterzellen, mit der Indexfunktion  $\Gamma : I_G^k \rightarrow G$ . Der Algorithmus wird für eine Teilvolumenrekonstruktion  $V_{rec,t}^k$  wie folgt initialisiert. Das Volumen des regulären Gitters ist  $\mathcal{G}_{rec,t}^k = V_{rec,t}^k$  und



die Größen  $i_{G,x}, i_{G,y}, i_{G,z}$  der Indexmenge  $I_G^k$  sind abhängig von der gegebenen Anwendung vorgegeben.

Es wird über alle Gitterzellen iteriert. Die Initialisierungsfunktion  $\theta_I : \mathbb{R}^3 \rightarrow \{0, 1\}$  ist mit Hilfe des Octree bezüglich  $V_{rec,t}$  realisiert. Sie entscheidet, ob eine Ecke der Gitterzelle belegt oder nicht belegt ist. Auf die Ecken einer Gitterzelle wird über die Funktion  $\sigma_G : G \rightarrow \mathbb{R}^8$  mit  $\sigma_G(g) := (p_{g,1}, \dots, p_{g,8})$  zugegriffen. Um den Octree-Knoten zu einem gegebenen Punkt im  $\mathbb{R}^3$  zu bestimmen wird folgende Hilfsfunktion verwendet:  $\sigma_O : \mathbb{R}^3 \rightarrow N$  mit  $\sigma_O(p) = n$ . Mit diesen Hilfsfunktionen ist die Initialisierungsfunktion wie folgt definiert:

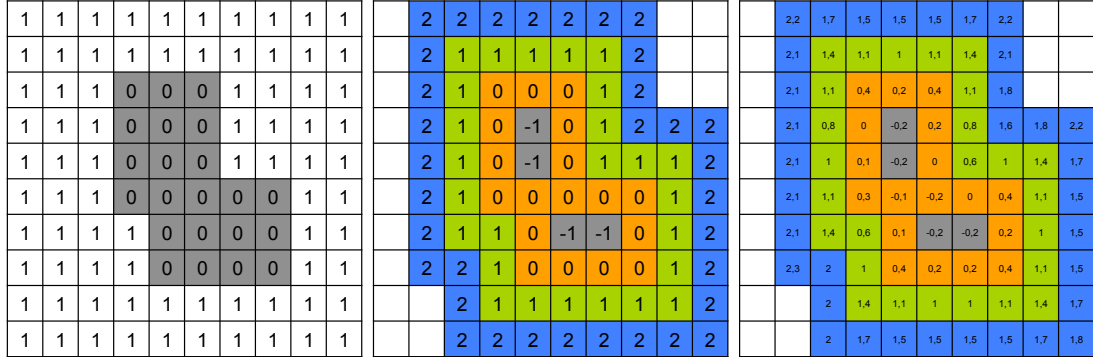
$$\theta_I(p) = \begin{cases} 0 & , s_{t,\sigma_O(p)} \neq \text{empty} \\ 1 & , \text{sonst} \end{cases}$$

Abbildung 3a zeigt das Resultat dieser Initialisierung an einem zweidimensionalen Beispiel. Die grau hinterlegten Gitterzellen sind als nicht *empty* markiert.

Ausgehend von den binär initialisierten Gitterzellen wird der *Brushfire* Algorithmus [Cho+05] angewendet. Freie Gitterzellen (mit dem Wert 1) werden mit diesem Algorithmus mit positiven diskreten Distanzwerten gefüllt. Belegte Gitterzellen (mit dem Wert 0) werden mit negativen diskreten Distanzwerten gefüllt. Gitterzellen die anfangs den Wert 0 und eine Nachbarzelle mit dem Wert 1 hatten, werden nicht verändert. In Abbildung 3b ist die resultierende Belegung der Gitterzellen inklusive der unveränderten Randgitterzellen (orange) beispielhaft illustriert.

Die diskreten Abstandswerte verhindern aktuell die Rekonstruktion einer glatten Oberfläche. Daher werden die Abstände mit einem einfachen Mittelwertfilter mit einer Filtermaske der Größe  $3 \times 3 \times 3$  geglättet. Durch diesen Filter kann sich der Nulldurchgang in das Innere des Objektes (siehe Abbildung 3c) verschieben. Die maximale Verschiebung lässt sich anhand der Filtergröße abschätzen. Bei einer Filtergröße von 3, wird der Mittelwert über 27 Zellen gebildet. Von diesen Zellen haben im schlechtesten Fall die mittlere Gitterzelle den Wert 0 und alle Anderen den Wert 1. Daraus ergibt sich der geglättete neue Zellenwert  $0 < \frac{26}{27} < 1$ . Um den Nulldurchgang nicht durch die Glättung zu verschieben wird vor der Anwendung des Filters ein zu wählender Wert subtrahiert. Nachdem dieser Wert abhängig von der zu rekonstruierenden Geometrie ist und die rekonstruierte Oberfläche möglichst

nahe an der tatsächlichen Oberfläche liegen soll, ist dieser Wert experimentell auf  $\frac{7}{9}$  festgelegt worden (siehe Kapitel 2.4).



(a) Binäre Initialisierung des Gitters basierend auf dem Belegungsstand des Octrees (b) Diskrete vorzeichenbehaftete Distanzfunktion (c) Geglättete Distanzfunktion

Abbildung 3: Beispielhafte Illustration der Initialisierung des Gitters für den Marching Cubes Algorithmus ausgehend von dem binären Belegungsstand des Octrees

Über den Belegungsstatus der Eckpunkte von  $g \in G_{I_g^k}$  entscheidet der MC Algorithmus, ob und wie viele Dreiecke innerhalb von  $g$  erzeugt werden [Lor+87].

Mit dem geglätteten Distanzfeld ist die Indikatorfunktion  $\theta : \mathbb{R}^3 \rightarrow \{full, empty\}$ , mit der der Belegungsstatus der Eckpunkte bezüglich der Nachbarzellen berechnet wird, ein einfacher Vorzeichentest.

Die Belegung für eine Gitterzelle  $g$  mit den Eckpunkten  $\sigma_{\mathcal{G}}(g) = (p_{g,1}, \dots, p_{g,8})$  ist demnach  $(\theta(p_{g,1}), \dots, \theta(p_{g,8}))$ . Die Position eines Eckpunktes eines Dreiecks auf der Kante zwischen zwei Eckpunkten einer Gitterzelle wird bezüglich der Distanzwerte an den Eckpunkten linear interpoliert. Damit sei  $D_g$  die Menge dieser Dreiecke innerhalb einer Gitterzelle  $g$ .

Eine Gitterzelle bekommt neben den zugewiesenen Dreiecken einen zusätzlichen Zustand  $z_g \in Z = \{interior, empty, visual, virtual\}$ . Der Zustand  $z_g$  einer Gitterzelle hängt von der Menge der Dreiecke und dem Resultat der Indikatorfunktion ab. Eine Gitterzelle ohne zugewiesene Dreiecke ist *empty* wenn sie außerhalb des belegten Volumens von  $V_{rec,t}^k$  liegt beziehungsweise als *interior* falls sie innerhalb

des belegten Volumens liegt. Die beiden Zustände *visual* oder *virtual* werden für eine Gitterzelle gesetzt, wenn dieser Dreiecke zugewiesen sind. Eine Gitterzelle ist *visual*, wenn innerhalb ihres Volumenbereiches Sensorinformationen von mindestens einer Kamera  $i$  vorhanden sind, sie also durch Sensordaten validiert ist, ansonsten wird sie als *virtual* gekennzeichnet.

Die Funktion  $\tau_Z : G \rightarrow Z$  entscheidet über den Zustand einer Gitterzelle,

$$\tau_Z(g) := \begin{cases} \textit{interior} & , \theta(g) = \textit{full} \wedge D_g \neq \emptyset \\ \textit{empty} & , \theta(g) = \textit{empty} \wedge D_g = \emptyset \\ \textit{visual} & , D_g \neq \emptyset \wedge \exists i : \exists p \in \sigma_G(g) : \tau_{dep,t,i}(\sigma_O(p)) = \textit{mixed} \\ \textit{virtual} & \text{sonst.} \end{cases} \quad (3)$$

Um approximativ die Größe der Kontaktfläche zwischen zwei Oberflächenmodellen robust berechnen zu können, werden die durch den MC Algorithmus erzeugten Dreiecke zusätzlich durch eine Menge gleichverteilter Kugeln repräsentiert. Die Menge der Kugeln innerhalb einer Gitterzelle sei  $K_g$ .

Demnach ist das Objektmodell  $\mathcal{O}$ , das im Weiteren verwendet wird, allgemein wie folgt definiert:

**Definition 2.3.** Ein Oberflächen-Objektmodell  $\mathcal{O}$  ist eine Menge von Polygonen die innerhalb eines quaderförmigen Volumens  $\mathcal{G} \subset V_{rec}$  mit der Menge an Gitterzellen  $G$  definiert sind. Jedes Polygon ist eindeutig einer Gitterzelle zugeordnet. Jede Gitterzelle  $g \in G$  hat einen Sensordaten-basierten Zustand  $z_g \in Z$ , die Menge der Dreiecke  $D_g$ , die zugehörige Kugelapproximation  $K_g$  und einen Volumenbereich  $V_g = [g_{min}, g_{max}] \subset \mathcal{G}$

## 2.4 EXPERIMENTE UND DISKUSSION

In diesem Abschnitt werden die Erweiterungen der Volumenrekonstruktion im Bezug auf eine Verwendung mit Tiefenbildern aus einer oder mehrer Kameraperspektiven inklusive der anschließenden Oberflächenrekonstruktion basierend auf Beispielen mit realen Eingabedaten diskutiert.

Abbildung 4 zeigt die Zellenmarkierung auf Basis der Tiefeninformationen aus einem Eingabebild. Das rekonstruierte quaderförmige Objekt ist aus unterschiedlichen Perspektiven zu sehen. An den Übergängen zwischen den validierten und

konservativ abgeschätzten Bereichen lässt sich beobachten, dass Dreiecke als validiert markiert werden, obwohl keine Tiefeninformationen verfügbar sind. Diese Übergänge entsprechen Tiefenwertsprüngen in den Eingabebildern. Dieser Fehler entsteht aus der konservativen Abschätzung der Projektion eines Octree-Knotens beziehungsweise Gitterzelle auf das Tiefenbild (vgl. Gleichungen 3 und 2). Dieser Knoten projiziert auf mehrere Pixel des Bildes. Sobald einer dieser Pixel einen validen Tiefenwert besitzt wird für die Gitterzelle als *visual* gekennzeichnet und somit als durch Tiefendaten validiert markiert.

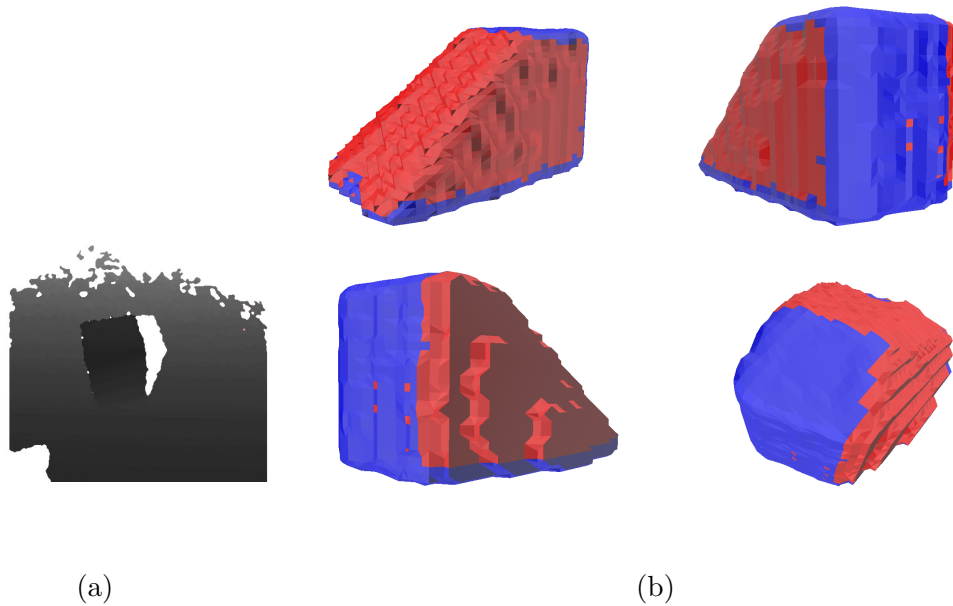


Abbildung 4: Illustration des der Markierung der Gitterzellen aus vier Perspektiven (b) im Bezug auf die Tiefendaten aus dem Kamerabild (a). Rote Dreiecke sind nicht durch Tiefendaten validiert und blaue sind validiert.

In Kapitel 2.3.3 ist eine minimale konservative Vervollständigung der fehlenden Tiefendaten in den Eingabebildern vorgestellt worden. Die Abbildung 5 setzt die resultierenden Rekonstruktionen basierend auf der naiven konservativen Vervollständigung (Abbildung 5a), der minimalen konservativen Vervollständigung (Abbildung 5b) und ohne Vervollständigung (Abbildung 5c) zueinander in Bezug. Die rekonstruierte Szene besteht aus einem quaderförmigen Objekt. Die Szene ist ein-

mal aus einer Kameraperspektive und einmal aus fünf Perspektiven aufgenommen worden.

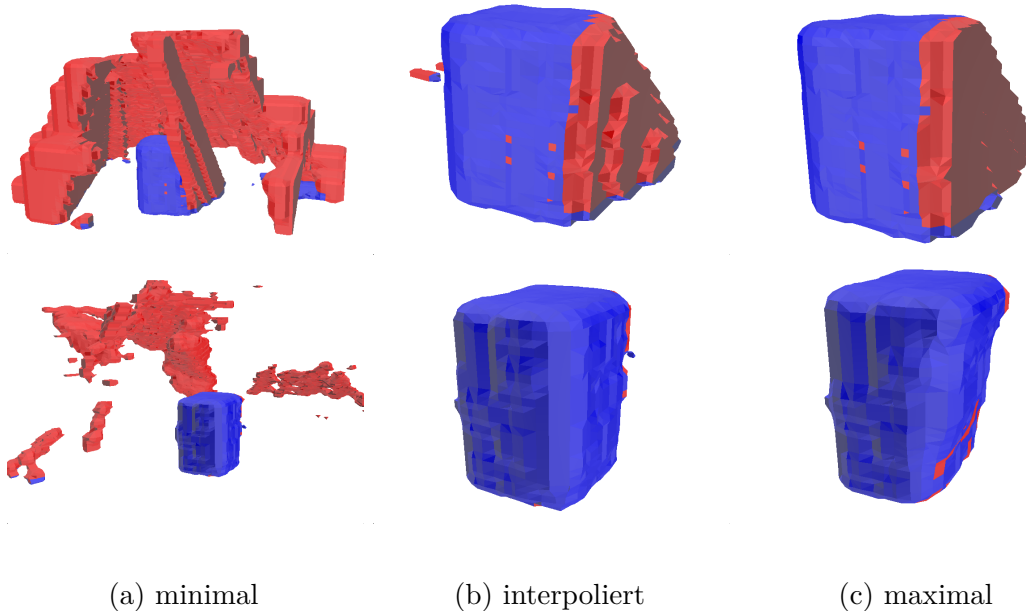


Abbildung 5: Vergleich zwischen den drei verschiedenen Ansätzen zur Vervollständigung fehlender Tiefendaten anhand eines Quaders. Die erste Reihe zeigt die Rekonstruktion mit einer und die zweite Reihe mit fünf Kameraperspektiven. Rote Dreiecke sind nicht durch Tiefendaten validiert und blaue sind validiert. In (a) entstehen Scheinobjekte und in (c) wird die untere rechte Ecke des Quaders weggeschnitten. Mit der interpolierten Vervollständigung in (c) wird das Objekt korrekt ohne Pseudoobjekte rekonstruiert.

Bei der naiven Vervollständigung entstehen große zusammenhängende Pseudoobjekte, die auch durch die wissensbasierten Selektionsfunktionen nicht eliminiert werden konnten. Diese werden mit zunehmender Anzahl der Kameraperspektiven weniger, können jedoch nicht gänzlich entfernt werden.

Bei der Verwendung der Roheingabebilder werden mit zunehmender Anzahl an Kameraperspektiven mehr Bereiche vom Objektmodell weggeschnitten, die schon zu dem realen Objekt zugehörig sind. In Abbildung 5c wird die vordere untere Ecke des Quaders so stark beschnitten, dass den erzeugten Dreiecken kein Tiefenwert

mehr zugeordnet werden kann, sie also mindestens um die minimale Seitenlänge einer Gitterzelle des Objektmodells zu weit im realen Objekt liegen.

Nachdem die Annahme, dass der Sensor keine Detektionsfehler besitzt, nicht der Realität entspricht, kommt es auch bei der vorgestellten Tiefenbildvervollständigung zu Auslöschungen. Diese fallen jedoch viel geringer ins Gewicht, da innere Löcher durch lineare Interpolation gefüllt werden. Wenn man die rechte vertikale Kante des Quaders in den Rekonstruktionen mit fünf Perspektiven aus Abbildung 5b und 5c vergleicht, wird diese Kante bei der Rekonstruktion mit den Rohbildern viel deutlicher abgeschnitten, so dass die Quaderform nur noch schwer zu erkennen ist.

Der Fehler der Rekonstruktion im Bezug auf das reale Objekt wird zunächst getrennt für die Volumenrekonstruktion und die Oberflächen analysiert, um anschließend aus beiden Fehlern eine obere  $\bar{e} \in \mathbb{R}$  und untere Schranke  $\underline{e} \in \mathbb{R}$  bezüglich des Abstandes zwischen der realen Szene und durch Sensordaten validierte Bereiche am Oberflächenmodell anzugeben.

Nachdem die Volumenrekonstruktion konservativ ist, ist die untere Fehlerschranke  $\underline{e}_V = 0$ . Für die obere Schranke ist die Größe des Volumenbereichs eines Octree-Knotens ausschlaggebend. Ein Knoten  $n \in N$  wird genau dann der Rekonstruktion  $V_{rec,t}$  zugeordnet, wenn dessen Volumen  $V_n = [n_{\min}, n_{\max}]$  das Volumen der Szene  $V_{obj,t}$  schneidet. Daraus ergibt sich die obere Schranke  $\bar{e}_V = \|n_{\max} - n_{\min}\|$ .

Bei der Oberflächenrekonstruktion mittels Marching-Cubes Algorithmus ist der Fehler von der Positionierung der Eckpunkte der Dreiecke abhängig. Ein Eckpunkt wird immer dann mittig zwischen einer als belegt und einer als frei markierten Ecke einer Gitterzelle erzeugt, wenn die verwendete Indikatorfunktion nur binär zwischen belegt und frei unterscheiden kann. Das ist genau dann der Fall, wenn das berechnete Distanzfeld nicht geglättet wurde. Ansonsten werden die Eckpunkte der erzeugten Dreiecke näher an die tatsächliche Oberfläche herangerückt. Mit der maximalen Seitenlänge  $a_g \in \mathbb{R}$  einer Gitterzelle  $g \in G$  ist die obere Fehlerschranke  $\bar{e}_{MC} = a_g$  beziehungsweise die untere Schranke  $\underline{e}_{MC} = -a_g$ . Daraus ergibt sich für als gesamte obere Schranke

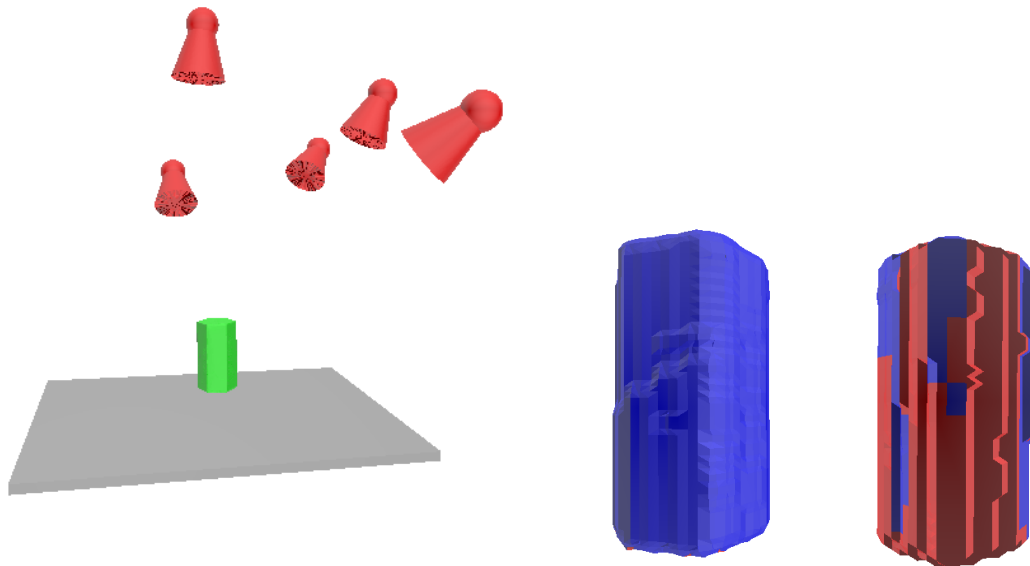
$$\bar{e} = \sqrt{\bar{e}_V^2 + a_g^2} + \bar{e}_V a_g$$

und als untere Schranke

$$\underline{e} = \underline{e}_V + \underline{e}_{MC} = -a_g.$$

Um die Abweichungen zwischen der Oberflächenrekonstruktion im Bezug auf die reale Oberfläche eines Objektes zu beurteilen, sind verschiedene Referenzobjektmodelle mit virtuellen Tiefenkameras aufgenommen worden, damit die genaue Pose des Referenzmodells bekannt sind, und keine Detektionsfehler durch den Sensor entstehen. Mit den virtuellen Kameraposen relativ zu dem jeweiligen Referenzobjekt aus Abbildung 6a, sind die Objekte rekonstruiert worden. Die Kameraposen erlauben es nicht, das Objekt aus allen Richtungen aufzunehmen. Dieses Szenario entspricht den möglichen Blickwinkeln, die eingenommen werden können, wenn eine Tiefenkamera auf dem Handgelenk eines Manipulators befestigt ist. Die Objekte sind mit einer Octreetiefe von sieben und einer Gitterzellenauflösung pro Seite von  $2^6$  beziehungsweise  $2^7$  rekonstruiert worden. Daraus resultieren Gitterzellengrößen von  $0,010 \times 0,008 \times 0,006$  m beziehungsweise  $0,005 \times 0,004 \times 0,003$  m bei dem verwendeten Rekonstruktionsvolumen. Eine weitere Vergrößerung der Auflösung ist nicht zielführend, da die Grenzen der möglichen Tiefenauflösung handelsüblicher Tiefenkameras erreicht sind. Die Filtergröße zur Glättung der Distanzfunktion ist  $3 \times 3 \times 3$ . Es sind 20 verschiedene Haushaltsgegenstände teilweise aus der *KIT ObjectModels Web Database* [Kas+12] verwendet worden. Die Datenbank stellt rekonstruierte Modelle der enthaltenen Objekte zur Verfügung. Mit dem der Offset zum Ausgleich der Verschiebung des Nulldurchgangs bestimmt worden ist. Dazu sind mit dem virtuellen Aufbau unterschiedliche Gitterzellenauflösungen und unterschiedliche Offsets getestet worden. In Abbildung 6b ist beispielhaft die Rekonstruktion mit den sensorisch validierten und konservativ abgeschätzten Oberflächenbereichen abgebildet. An der Vorderseite (der den Kameras zugewandten Seite) ist die Rekonstruktion sehr glatt. Auf der Rückseite (der den Kameras abgewandten Seite) ist die Rekonstruktion wellig, da schon der Octree an dieser Seite so treppenförmig ist, dass selbst die Glättung der Distanzfunktion keine optisch erkennbaren Verbesserungen bewirkt.

In Abbildung 7 ist beispielhaft der Vergleich zwischen Referenzmodell und Rekonstruktion des Objektes aus Abbildung 6b für die beiden Gitterzellenauflösungen illustriert. Die Distanzen sind von der Rekonstruktion zu dem Referenzmodell aufgetragen. Jeweils die Vorderseiten des Objekts sind glatt und dadurch mit einer gleichbleibenden Abweichung von  $0,005$  m beziehungsweise  $0,003$  m rekonstruiert. Bei der geringeren Gitterzellenauflösung sind die Abstände breiter gestreut, da die einzelnen Dreiecke größer sind als bei der höheren Gitterzellenauflösung.



(a) Anordnung der Kameraperspektiven (rot) und des Referenzmodells (grün) mit der minimalen Seitenlänge 0,003 m und maximalen Seitenlänge 0,005 m der Gitterzellen

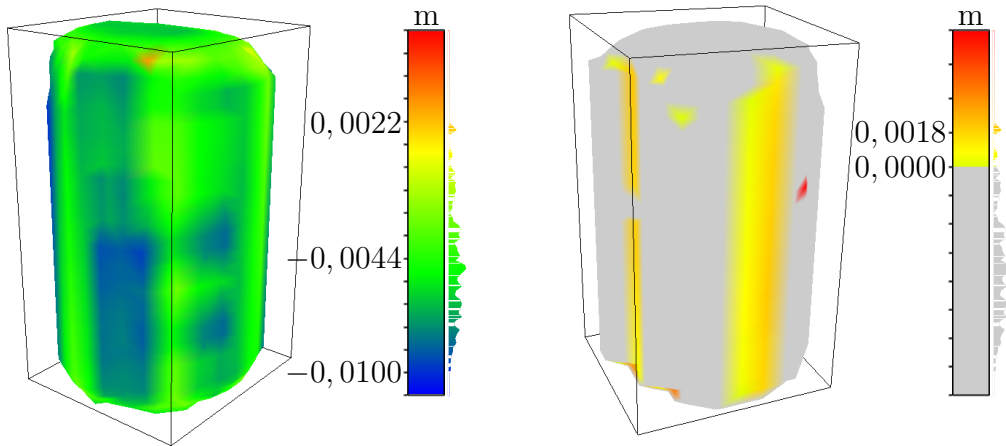
(b) Vorder- und Rückansicht des rekonstruierten Objektes mit sensorisch validierten (blau) und konservativ abgeschätzten (rot) Oberflächenbereichen

Abbildung 6: Grafische Darstellung der Abweichungen des rekonstruierten Objekts von dem Referenzmodell. Für das Referenzmodell sind virtuelle Tiefenbilder erzeugt worden, die als Eingabe verwendet werden.

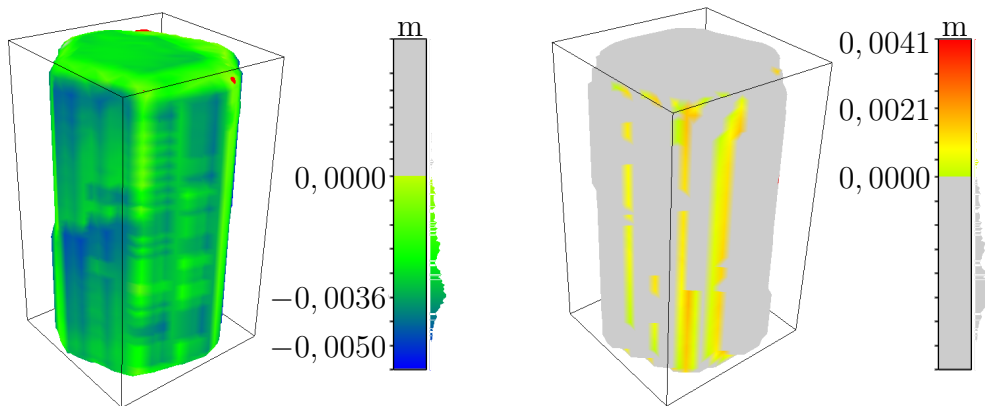
Die rechnerische untere Schranke der Rekonstruktion wird eingehalten. Die Messwerte unter dieser Schranke sind Messfehler. Durch das Übereinanderlegen beider Modelle sind alle Stellen überprüft worden, ob diese sich auch tatsächlich durchdringen. An den Stellen, an denen die untere Schranke verletzt wird, müsste das Referenzmodell die Rekonstruktion durchdringen, was allerdings nicht der Fall ist. An allen anderen Stellen in denen das Referenzmodell die Rekonstruktion durchdringt wird die untere Schranke eingehalten.

Nachdem die untere Schranke negativ ist, ist nach der Oberflächenrekonstruktion die Konservativität aus der Volumenrekonstruktion nicht mehr gegeben. Der maximale Fehler ist jedoch abschätzbar und kann so gewählt werden, dass dieser





(a) Linke Seite: Abweichungen bei denen die Rekonstruktion größer (maximal 0,01 m) als das Referenzmodell ist. Rechte Seite: Abweichungen bei denen die Rekonstruktion kleiner (maximal 0,002 m) als das Referenzmodell ist.



(b) Linke Seite: Abweichungen bei denen die Rekonstruktion größer (maximal 0,0057 m) als das Referenzmodell ist. Rechte Seite: Abweichungen bei denen die Rekonstruktion kleiner (maximal 0,0041 m) als das Referenzmodell ist.

Abbildung 7: Vorder- und Rückansicht der Abweichungen im Bezug auf das Referenzmodell

während der Manipulation eines Objektes, das mit der vorliegenden Rekonstruktion modelliert ist, durch den Roboter kompensiert werden kann. Beispielsweise durch eine passive Nachgiebigkeit des Roboterarms oder der Verwendung kraft geregelter Bewegungen. Es ist jedoch jeder Zeit möglich das rekonstruierte Ob-

jektmodell konservativ zu erstellen, wenn auf die Werte der geglätteten Distanzfunktion ein Wert größer eins addiert wird (siehe Kapitel 2.3.4). Daraus resultiert allerdings, dass das Objektmodell insgesamt größer und ungenauer wird.

Nachdem das Rekonstruktionsverfahren während den Experimenten für die Greif- (Kapitel 4.5) und Ablageplanung (Kapitel 5.5), sowie bei den Experimenten am realen Roboter (Kapitel 6.3) verwendet wird, wird hier auf die Darstellung weiterer rekonstruierter Objekte verzichtet.

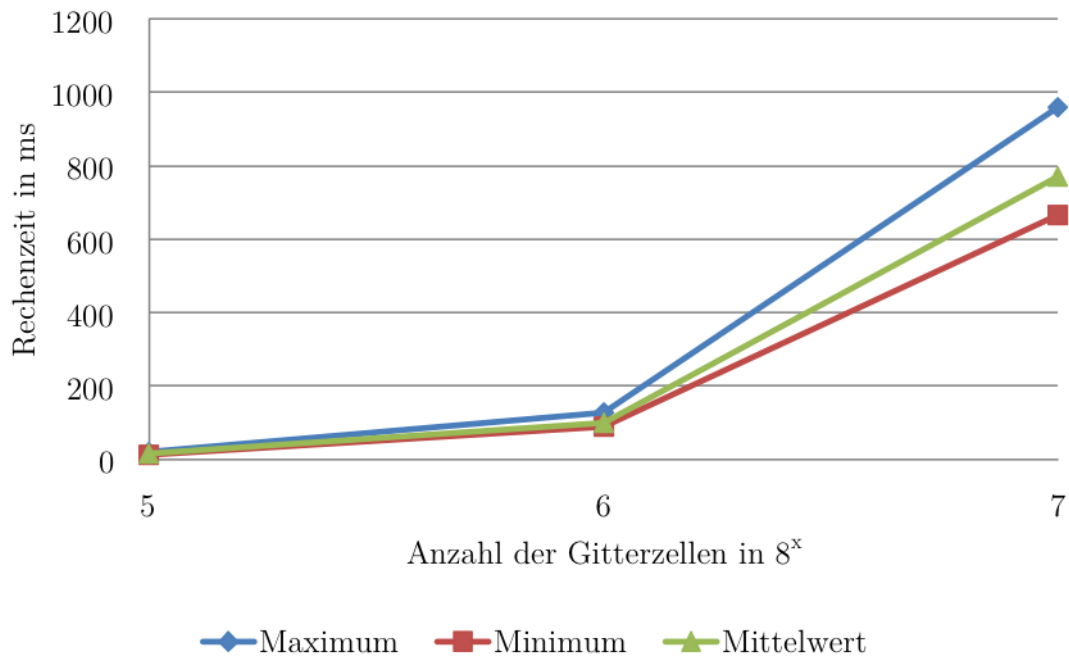


Abbildung 8: Rechenzeit der Oberflächenrekonstruktion für verschiedene Tiefen des Octrees pro Objekt

Die Rechenzeit<sup>2</sup> der Rekonstruktion, ermittelt aus 936 Testläufen mit 26 verschiedenen Objekten, ist maßgeblich von der Oberflächenrekonstruktion aus der Octree-Repräsentation abhängig. Die Rechenzeit einer sequenziellen Implementierung der Oberflächenrekonstruktion ist in Abbildung 8 aufgetragen. Die der Volumenrekonstruktion basierend auf dem oben beschriebenen Aufbau ist in Abbildung

<sup>2</sup> Die Experimente sind auf einem Computer mit einem 22 nm “Ivy Bridge” 2.9 GHz Intel “Core i7” Prozessor (3520M) und 8 GB RAM durchgeführt worden.

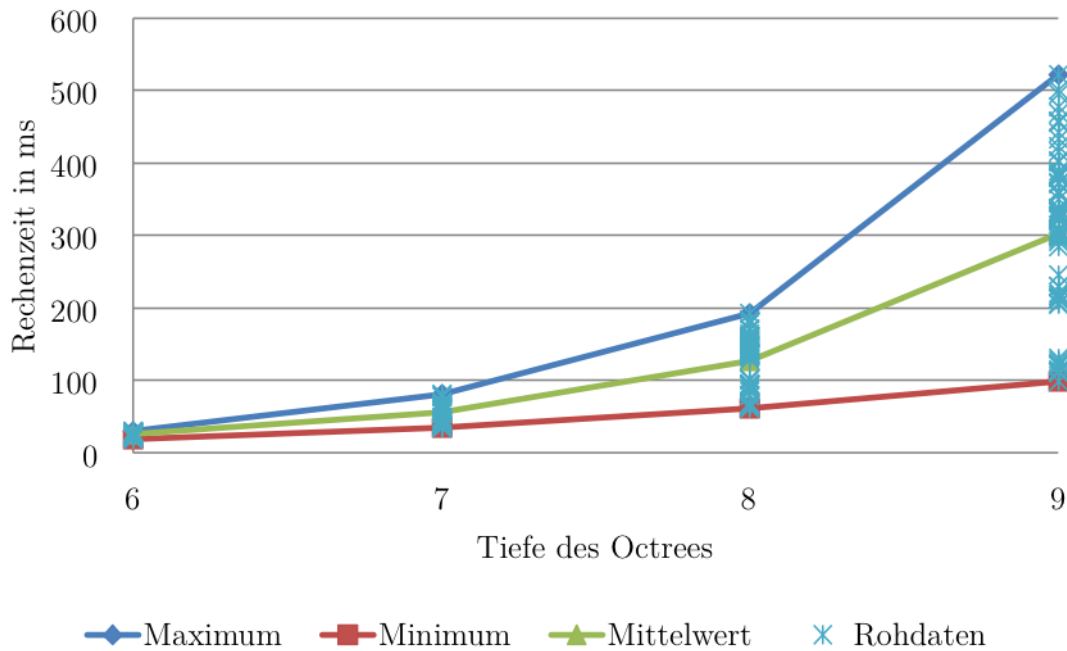


Abbildung 9: Rechenzeit der Volumenrekonstruktion für verschiedene Tiefen des Octrees

9 abhängig von der maximalen Tiefe des Octrees dargestellt. Bei Octree-Tiefen von acht oder mehr ist die Diskretisierung des Raumes durch den Octree feiner als die Tiefenauflösung von Tiefenkameras, wenn man den Arbeitsbereich des Manipulators erfassen und rekonstruieren möchte. Bei diesen Auflösungen ist eine Rechenzeit von unter einer halben Sekunde für die gesamte Rekonstruktion zu erwarten.

Die gesamte Rekonstruktion ist durch die nicht inkrementelle Oberflächenrekonstruktion nicht mehr ad-hoc anytime-fähig, die Echtzeitfähigkeit bleibt jedoch erhalten. Es lässt sich abhängig von der Auflösung des regulären Gitters eine obere Zeitschranke für den in der Anzahl der Gitterzellen linearen Marching-Cube Algorithmus angeben. Ebenso bleibt eine interaktive Rekonstruktionszeit erhalten.

Um die gesamte Rekonstruktion anytime-fähig zu gestalten, muss die Oberflächenrekonstruktion inkrementell erfolgen. Eine mögliche Realisierung ist, dass der Marching-Cubes Algorithmus inkrementell pro Octree-Ebene ausgeführt wird. Damit müssten nur Gitterzellen mit Dreiecken aktualisiert werden, deren Dreiecke bezüglich der Indikatorfunktion, repräsentiert durch den Octree, veraltet sind.

## 2.5 SCHLUSSFOLGERUNG

In diesem Kapitel ist ein zweistufiges Rekonstruktionsverfahren eingeführt worden, das als Eingabe ein oder mehrere unvollständige Tiefenbilder aus unterschiedlichen bekannten Kameraperspektiven verwendet. Die Eingabebilder werden unter der Annahme, dass sensorseitig keine Detektionsfehler auftreten, minimal konservativ vervollständigt. Mit diesen vor-verarbeiteten Tiefenbildern erfolgt eine Volumenrekonstruktion, auf deren Basis Objekte segmentiert werden. Jede segmentierte Teilvolumenrekonstruktion wird mit Hilfe eines erweiterten Marching-Cubes Algorithmus in ein Oberflächenmodell transformiert. Es wird nicht nur die Oberfläche erzeugt, sie wird auch noch klassifiziert, ob die rekonstruierten Oberflächenbereiche durch Tiefeninformationen aus den Eingabebildern validiert sind.

Die gesamte Rekonstruktion ist konservativ bis auf einen berechenbaren Fehler, der leicht während der Manipulation der Objekte durch den Roboter, durch beispielsweise eine passive Nachgiebigkeit, kompensiert werden kann. Weiter ist die Rekonstruktion echtzeitfähig und interaktiv.

Die sequentielle Rechenzeit des Algorithmus kann weiter verkürzt werden, in dem man eine Parallelisierung mittels mehrerer Octree Wurzelknoten wie in [Wer+14] verwendet und diese auch auf die Oberflächenrekonstruktion anwendet und die Implementierung auf der GPU vornimmt.

Insbesondere existieren nur zwei Parameter, die die Genauigkeit der Rekonstruktion beeinflussen. Diese Eigenschaft ermöglicht eine einfache Anpassung an unterschiedliche Anwendungsbereiche. Ebenso ist durch diese Rekonstruktion die Möglichkeit geschaffen, dass die Unsicherheit bezüglich einer konservativen Objektrekonstruktion direkt während der Manipulationsplanung berücksichtigt werden kann. Damit sind die Voraussetzungen zur Beantwortung der Fragestellung H1 aus Kapitel 1.2 geschaffen.

Die Information, ob Oberflächenbereiche sensorisch validiert sind, lässt sich zukünftig ausnutzen, um die Kameraperspektiven mittels eines *next-best-view* Algorithmus zu berechnen. Beispielsweise könnte die nächste Kameraperspektive so gewählt werden, dass die Fläche der sichtbaren Oberflächenbereiche, die nicht sensorisch validiert sind, maximiert wird. Mit dieser Erweiterung überwindet man das Problem, dass die Wahl der Kameraperspektiven anwendungsabhängig ist.

# 3

## KONTAKTPOSENBERECHNUNG

In diesem Kapitel wird beschrieben, wie man effizient durch eine kontinuierliche Bewegung ein Objekt von seiner aktuellen Pose in eine neue überführen kann.

**Definition 3.1.** Die *Pose*  $T$  eines Objektes ist nach ISO 8373:2012 [ISO12] seine Orientierung und Lage im Raum.

Die Grundlage für die Berechnung sind zwei gegebene dreidimensionale Oberflächenmodelle, deren Oberflächenstücke ausschließlich konvexe Polygone sind. Beide Objekte dürfen sich in ihrer Ausgangspose berühren. Eine solche Pose wird als Kontaktpose bezeichnet.

**Definition 3.2.** Die Pose eines Objekts heißt *Kontaktpose*  $T_K$ , falls es ein weiteres Objekt gibt, so dass beide Objekte einen gemeinsamen Punkt haben.

Gesucht ist eine neue Kontaktpose entlang einer gegebenen Bewegungsrichtung. Es wird zwischen zwei Bewegungsarten unterschieden, entlang einer gegebenen Richtung beziehungsweise um eine gegebene Rotationsachse. Kontaktposen, die aus diesen beiden Bewegungen entstehen, seien wie folgt definiert:

**Definition 3.3.** Sei eine *lineare Kontaktpose* eine Kontaktpose die durch die Bewegung entlang eines Vektors  $\vec{d}_L \in \mathbb{R}^3$  und  $\|\vec{d}_L\| = 1$  entsteht.

**Definition 3.4.** Sei eine *rotatorische Kontaktpose* eine Kontaktpose die durch die Rotation in mathematisch positiver Drehrichtung um die Achse  $A : p_R + \lambda \vec{d}_R$  mit  $p_R, \vec{d}_R \in \mathbb{R}^3$  und  $\|\vec{d}_R\| = 1$  entsteht.

Das eine Objekt  $\mathcal{O}_{\text{Bew}}$  wird entlang der gegebenen Richtung bewegt und für das andere Objekt  $\mathcal{O}_{\text{Sta}}$  wird angenommen, dass es in Ruhe verharrt<sup>1</sup>. In Abbildung 10 ist eine typische Ausgangssituation und die gesuchte neue Kontaktpose für das bewegte Objekt abgebildet.

---

<sup>1</sup> Das stellt keine Einschränkung der Allgemeinheit dar, da bei zwei Objekten immer die Bewegung relativ zueinander angegeben werden kann.

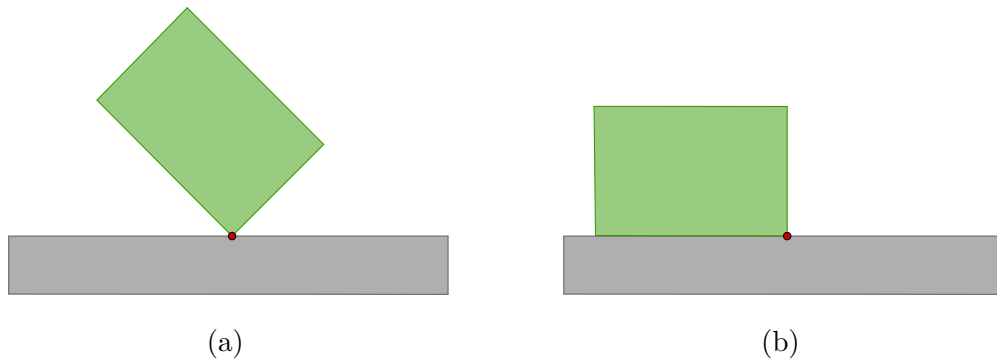


Abbildung 10: (a) Beispielhafte Ausgangssituation: Das grüne Objekt soll um den Kontaktpunkt mit dem grauen Objekt gedreht werden bis ein neuer Kontakt und damit eine neue Kontaktpose entsteht. (b) Gesuchte neue Kontaktpose für das grüne Objekt.

Basierend auf den Anforderungen (Kapitel 3.1) wird der Stand der Forschung (Kapitel 3.2) diskutiert und anschließend die Ausgangssituation (Kapitel 3.3) dargestellt. Darauf aufbauend werden die Verfahrensschritte zur Berechnung einer linearen (Kapitel 3.4) und rotatorischen Kontaktpose (Kapitel 3.5) vorgestellt und deren Komplexität und Berechnungszeit bezüglich eines alternativen Selektionsansatzes untersucht (Kapitel 3.6). Weiter wird gezeigt wie man mit Hilfe des oben beschriebenen regulären Gitters die Anzahl der eigentlichen Kollisionstests minimieren kann.

### 3.1 ANFORDERUNGEN

Das gesuchte Verfahren muss drei elementaren Anforderungen genügen, damit eine effiziente und robuste Posenberechnung möglich ist. Zum Einen muss das Verfahren die erste neu auftretende Kollision mit dem starren Objekt exakt berechnen, auch wenn die beiden involvierten Objekte in ihrer Ausgangspose schon in Kontakt sind (vergleiche Abbildung 10). Zum Anderen muss die Posenberechnung mit einer geringen Anzahl von Kollisionspunktberechnungen auskommen, um eine interaktive (Definition 1.5) Rechenzeit des Algorithmus zu ermöglichen. Nachdem die Objekte schon zum Start der Berechnung Kontakt haben dürfen, sind schnelle Vortests, wie zum Beispiel ein Hüllkörper-Kollisionstests, weitgehend unwirksam. Daher müssen

immer alle Kollisionspunkte berechnet und analysiert werden, um zu bestimmen, ob sich die Kontaktsituation zwischen beiden Objekten verändert hat. Die letzte Anforderung ist, dass das Verfahren parameterfrei ist, somit nicht auf zusätzliche dynamische Informationen wie die Masse des Objekts oder seine Geschwindigkeit angewiesen ist und die Bewegung nicht ausschließlich auf physikalischen Eigenschaften beruht, sondern die auszuführende Bewegungsbahn/-richtung vorgegeben werden kann (Kapitel 1.2).

### 3.2 STAND DER FORSCHUNG

Für eine Posenberechnung, gemäß der Anforderungen, existieren im wesentlichen zwei Ansätze. Der wohl am weitverbreitetste Ansatz ist die eine iterative Bewegung des Objekts auf Basis der Dynamik, so genannte Starrkörpersimulationen. Bekannte freie Softwarebibliotheken sind die Bullet Physics Library [Bul13] und die Open Dynamics Engine [Ope14]. In jeder Iteration wird eine neue Pose der simulierten Objekte berechnet und eine Kollisionserkennung durchgeführt. Bei falscher Wahl der Schrittweite kann *tunneling* auftreten. Tunneling bedeutet, dass die Iterationsschritte so groß sind, dass ein Objekt vollständig verfehlt wird.

Die Verfahren unterscheiden sich in der Schrittweite pro Iterationsberechnung und in der Kollisionsmodellierung, um unter anderem das tunneling zu vermeiden. Es existieren drei verschiedene Paradigmen [Erl05]. Die *Straf-basierte Methode* (engl.: penalty methods) [Bar+92; Moo+88] arbeitet auf festen Zeitschritten und modelliert Kollisionen durch virtuelle Federn. Die Strafkraften oder Federkräfte bei einer Kollision basieren oft auf dem Hook'schen Gesetz und sind proportional zu der Durchdringung beider kollidierender Objekte. Die Methode ist bei großen Durchdringungen zwischen Objekten numerisch instabil, daher muss die Schrittweite bei der Posenberechnung klein gewählt werden, das wiederum die Anzahl der Kollisionstests in die Höhe treibt und somit der Berechnungsaufwand steigt. Der *Impuls-basierte Ansatz* ist ungeeignet für vorgegebene Bewegungen, da zusätzliche Berechnung nötig sind, um die Stärke, Richtung und Angriffspunkt des Impulses zu berechnen. Dessen Stärken liegen besonders bei der Auflösung von Kollisionen zweier Objekte gemäß dem elastischen Stoß. Der *Nebenbedingung-basierte Ansatz* (engl.: constrained-based) [Ste00] löst alle Kontaktzustände gleichzeitig indem ein großes Gleichungssystem aufgestellt und gelöst wird. Für statische Objekte ist

dieser Ansatz besonders gut geeignet, skaliert jedoch für viele Kontaktzustände bezogen auf die Rechenzeit schlecht.

Alle drei Paradigmen benötigen zumindest elementare Informationen über die involvierten Objekte. Dazu gehören unter anderem deren Masse und Massenverteilung. Beide Informationen sind bei geometrisch rekonstruierten Objekten nicht a priori verfügbar und insbesondere ist die Massenverteilung auch schwer abzuschätzen, da ein Objekt häufig aus verschiedenen Materialien mit unterschiedlicher Dichte besteht. Ein weiterer Nachteil ist, dass es kein explizites Kriterium gibt, wann die Simulation terminiert. In der vorliegenden Anwendung würde nach der ersten Iteration, in der eine Kollision aufgetreten ist, die Prozedur beendet, da die Bedingung erfüllt ist, dass beide Objekte in Kontakt sind. Durch eine Analyse der Kontaktpunkte müsste entschieden werden, ob sich die Ausgangspose und die vorherrschende Kontaktpose ausreichend unähnlich sind. Die Parameter für diese Analyse müssten anwendungsabhängig experimentell bestimmt werden.

Der zweite Ansatz ist die kontinuierliche Betrachtung der Objektbewegung. Bei diesem Ansatz entfallen die zwischengeschalteten Kollisionstests. Die kontinuierliche Kollisionserkennung kann in vier generelle Ansätze unterteilt werden: algebraischer Gleichungen [Kim+03; Red+00; Cho+06; Cou13], *sweep-Volumen* [Xue+07; Täu+12], *adaptiver Bisektion* [Sch+04; Red+02], *Minkowski-Summen* [Ber04] und *conservative advancement* für nicht konvexe Objekte [Tan+14]. Einen allgemeine und gute Übersicht zu Kollisionstests und den verschiedenen betrachteten Problemstellungen ist in [Wel13] nachzulesen.

Die Ansätze basierend auf sweep-Volumen und der Minkowski-Summe arbeiten gewöhnlich auf approximierten Modellen, um die Rechenzeit zu verkürzen, und können somit keine exakte Kontaktpose im Bezug auf die gegebenen Oberflächenmodelle liefern. Verfahren denen conservative advancement zu Grunde liegt können nur Kontaktposen zwischen konvexen Objekten berechnen, ebenso wie das raycast-basierte Verfahren von van den Bergen [Ber04]. Bisektionsbasierte Verfahren berechnen die neue Kontaktpose nur bis zu einer gewissen Genauigkeit, die von der minimal zulässigen Intervallgröße abhängt.

Alle Ansätze der kontinuierlichen Kollisionserkennung terminieren sobald die beiden betrachteten Objekte Kontakt haben. Aus diesem Grund eignen sie sich nicht zur Lösung des beschriebenen Problems, das insbesondere zulässt, dass die



Ausgangspose des bewegten Objektes schon eine Kontaktpose ist, die mit dem statischen Objekt besteht.

Die entwickelte Methode basiert ebenfalls auf dem Ansatz einer kontinuierlichen Betrachtung der Bewegung, allerdings nicht basierend auf Volumenprimitiven, sondern auf konvexen, polygonalen Oberflächenprimitiven. Nachdem die Konvexität der Oberflächenprimitive vorausgesetzt ist, genügt es zu prüfen, ob ein Punkt-Ebenen-Kontakt oder ein Kante-Kante-Kontakt hergestellt ist.

### 3.3 AUSGANGSSITUATION

Die Ausgangssituation für die im Folgenden beschriebenen Berechnungen ist, dass zwei nicht notwendigerweise geschlossene Polygonnetze (gemäß Definition 2.3)  $\mathcal{O}_{\text{Bew}}$ ,  $\mathcal{O}_{\text{Sta}}$  im Raum gegeben sind. Insbesondere ist es nicht notwendig, dass topologische Informationen innerhalb der Polygonnetze verfügbar sind. Dabei ist  $\mathcal{O}_{\text{Bew}}$  das Polygonnetz, das bewegt werden soll. Für  $\mathcal{O}_{\text{Sta}}$  existiert ein reguläres 3D-Gitter gemäß Definition 2.3. Ein Polygon wird mit einer Gitterzelle assoziiert sobald sich beide schneiden. Die gewählte Gitterunterteilung ermöglicht es bei der kontinuierlichen Kollisionsberechnung effizient relevante von nicht relevanten Polygonen zu selektieren. Das ist neben verschiedenen Baumstrukturen, wie Octree und kd-Baum, eine der klassischen Unterteilungen. Für einzelne Objekte ist der Mehraufwand einer Baumstruktur in der Abfrage im Gegensatz zu der besseren Raumgruppierung nicht gerechtfertigt, da wenige große zusammenhängende Raumbereiche existieren, in denen sich keine Polygone befinden.

In der Computergrafik gibt es die Forschungsschwerpunkte des Raytracing und Raycasting. Bei diesen Ansätzen werden Strahlen kontinuierlich durch den Raum verfolgt. Dies entspricht gerade dem Punkt-Ebenen-Fall. Ein Vergleich von verschiedenen Traversierungsstrategien für solche Anwendungen ist in [San+12] nachzulesen.

## 3.4 LINEARE KONTAKTPOSENBERECHNUNG

Für die Berechnung der neuen linearen Kontaktpose (Definition 3.3) zwischen den beiden gegebenen Polygonnetzen (Definition 2.3)  $\mathcal{O}_{\text{Bew}}$ ,  $\mathcal{O}_{\text{Sta}}$  mit den jeweiligen regulären Gittern ist zusätzlich die Richtung  $\vec{d}_L \in \mathbb{R}^3$  der Bewegung mit  $\|\vec{d}_L\| = 1$  gegeben.

Gesucht ist die maximale Distanz  $\lambda_{\vec{d}_L} \in \mathbb{R}$  der Bewegung in Richtung  $\vec{d}_L$  ausgehend von  $\mathcal{O}_{\text{Bew}}$ , so dass die beiden Polygonnetze sich in dieser Richtung berühren jedoch nicht schneiden.

Der entwickelte Lösungsansatz bewegt jedes Polygon aus  $\mathcal{O}_{\text{Bew}}$  entlang  $\vec{d}_L$  unter Verwendung des regulären Gitters bis es auf  $\mathcal{O}_{\text{Sta}}$  trifft. Die minimale zurückgelegte Distanz über alle Polygone ist die gesuchte Translationsdistanz  $\lambda_{\vec{d}_L}$ , um die neue lineare Kontaktpose herzustellen.

Zuerst wird die effiziente Auswahl von geschnittenen Gitterzellen (Kapitel 3.4.1) beschrieben, für deren beinhalteten Polygone die eigentlichen Schnittberechnungen (Kapitel 3.4.2) durchgeführt werden müssen. Im Gegensatz zu dem Ansatz aus [Cou13], der anhand von Kugel-Sweeps die relevanten Polygone auswählt, arbeitet der hier entwickelte Ansatz mit einem regulären Gitter als Hilfsstruktur und benötigt keine weitere Oberflächenrepräsentation und ist insbesondere nicht von der Größe der Oberfläche des Objektes abhängig. Wenn die Objektmodelle mit dem Verfahren aus Kapitel 2 rekonstruiert sind ist dieses Gitter schon vorhanden, ansonsten ist der Aufwand es zu erstellen sehr gering. Zuletzt wird beschrieben, wie die geometrischen Schnittberechnung erfolgt.

3.4.1 *Traversierung des Regulären Gitters*

In diesem Unterkapitel wird zunächst der Traversierungsalgorithmus vorgestellt und anschließend wichtige Erweiterungen des regulären Gitters und Bedingungen an die Polygone aus  $\mathcal{O}_{\text{Bew}}$  diskutiert.

Für jedes Polygon  $P_{\mathcal{O}_{\text{Bew}}} \in \mathcal{O}_{\text{Bew}}$  mit der Eigenschaft, dass dessen Normale  $n_P$  in Richtung  $\vec{d}_L$  zeigt ( $\angle(\vec{d}_L, n_P) \leq 90^\circ$ ), werden die Zellen des Gitters von  $\mathcal{O}_{\text{Sta}}$  bestimmt, die bei der Bewegung entlang von  $\vec{d}_L$  geschnitten werden. Diese Berechnung erfolgt schrittweise bezüglich der geschnittenen Gitterzellen. Dazu sei  $\lambda_{P,m} \in \mathbb{R}$  die aktuelle kürzeste Translationsdistanz entlang  $\vec{d}_L$ .

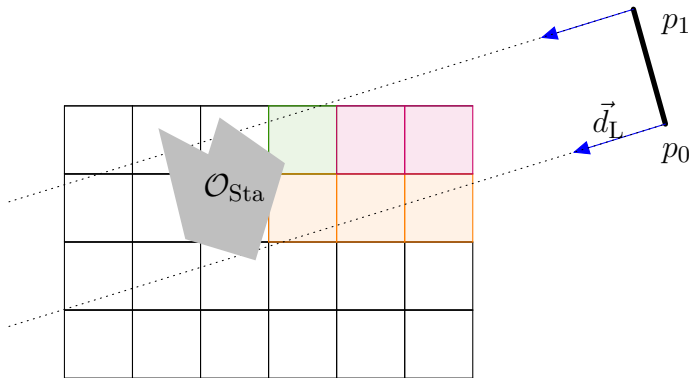


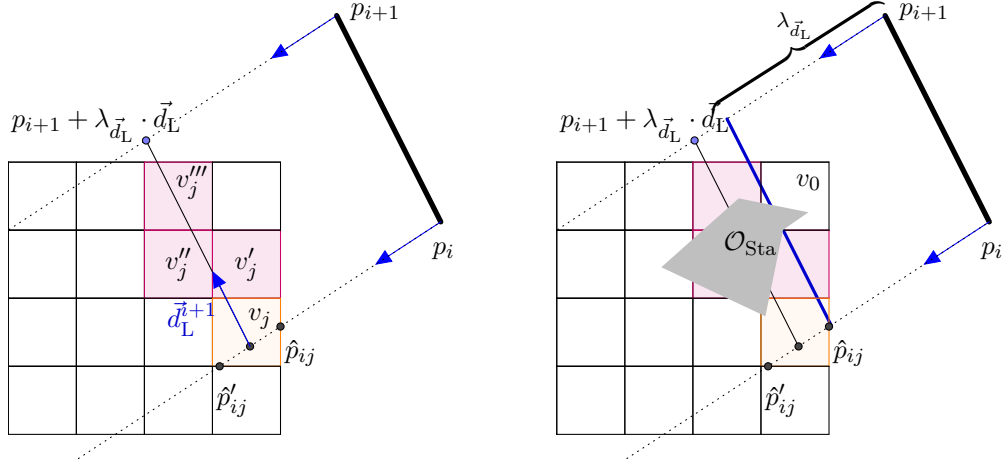
Abbildung 11: Illustration der linearen Traversierung einer Kante  $p_0, p_1$  entlang  $\vec{d}_L$ : Die orangen Gitterzellen werden von  $p_0$  traversiert und die grünen von  $p_1$ . Die roten Zellen werden nur von der Kante selbst durchstrichen.

Für die Berechnung der geschnittenen Gitterzellen für  $P_{\mathcal{O}_{\text{Bew}}}$  wird ein Raycast gleichzeitig für jeden Eckpunkt  $p_i \in P_{\mathcal{O}_{\text{Bew}}}$  in Richtung  $\vec{d}_L$  durchgeführt. Falls die Ausgangspose eine Kontaktpose ist und die  $\vec{d}_L$  senkrecht auf die Normale der Ebene in der alle Kontaktpunkte zwischen beiden Polygonnetzen liegt, werden Eckpunkte nicht betrachtet, die in einer Epsilonumgebung um diese Ebene liegen. Diese Situation tritt beispielsweise auf, wenn ein Objekt auf einer ebenen Fläche verschoben werden soll.

Die eigentliche Berechnung der durchlaufenen Gitterzellen erfolgt mit Hilfe des Algorithmus von John Amanatides und Andrew Woo [Ama+87]. Dieser berechnet iterativ alle Gitterzellen die von einem Strahl geschnitten werden ohne die eigentliche Schnittberechnung mit den Trennebenen zwischen den Zellen zu berechnen. Es wird ausgenutzt, dass der Strahl bezüglich der Gitterzellen eine konstante Steigung hat und somit die Trennebenen der Zellen immer in periodischen Abständen geschnitten werden. Somit reduziert sich die Berechnung auf einfache Kleiner-Relationen und Additionen von Konstanten. Der bekanntere Bresenham-Algorithmus [Bre65] zum Rastern von Linien ist nicht geeignet, da dieser nicht alle Pixel/Gitterzellen markiert, die von der zu rasternden Linie geschnitten werden.

In Abbildung 11 ist eine zweidimensionale Illustration der Traversierung des regulären Gitters für eine Kante  $(p_0, p_1) \in P_{\mathcal{O}_{\text{Bew}}}$  abgebildet. Die orange eingefärbten Gitterzellen werden von dem Strahl  $(p_0, \vec{d}_L)$  geschnitten und die grünen Zellen von dem Strahl ausgehend von  $p_1$ . Die rot markierten Gitterzellen werden ausschließlich von der translatierten Kante geschnitten. Damit muss die Traversie-

rung sicher stellen, dass diese Gitterzellen auch in der Reihenfolge des Auftretens eines Schnittes bei der Bewegung der Kante mit berücksichtigt werden.



(a) Raycast entlang der temporären Kante verschoben um  $\lambda_{\vec{d}_L}$  entlang  $\vec{d}_L$  ausgehend von dem Mittelpunkt zwischen  $\hat{p}_{ij}$  und  $\hat{p}'_{ij}$ . Reihenfolge der Gitterzellen:  $v_j, v'_j, v''_j, v'''_j$

(b) Beispiel warum die Beschränkung des regulären Gitters nicht ausreicht: Die obere rechte Gitterzelle wird nicht traversiert, obwohl in dieser die tatsächliche Kontaktpose zwischen der Kante  $(p_i, p_{i+1})$  und  $\mathcal{O}_{\text{Sta}}$  existiert. Die blaue Kante stellt die fehlerhafte Kontaktpose dar.

Abbildung 12: Illustrationen (a) des Raycasts entlang einer temporären Kante, die durch eine Translation um  $\lambda_{\vec{d}_L} := \frac{1}{2} \|d(p_i, \hat{p}_{ij}) + d(p_i, \hat{p}'_{ij})\|$  entlang  $\vec{d}_L$  ausgehend von der Kante  $(p_i, p_{i+1})$  definiert ist. (b) zeigt ein Beispiel für die fehlerhafte Traversierung bei der Beschränkung auf das reguläre Gitter.

Sei  $\hat{p}_{ij} \in \mathbb{R}^3$  der Eintrittspunkt und  $\hat{p}'_{ij} \in \mathbb{R}^3$  der Austrittspunkt des Strahls ausgehend von dem Eckpunkt  $p_i$  in und aus der Gitterzelle  $v_j$ . Wobei gilt, dass die 0-te Gitterzelle die Zelle ist, die die kürzeste Entfernung zu  $p_i$  hat:

$$v_0 := \operatorname{argmin}_{v \in V_{\mathcal{O}_{\text{Sta}}}} d_{\vec{d}_L}(v, p_i) \quad \forall p_i \in P_{\mathcal{O}_{\text{Bew}}}$$

mit  $d_{\vec{d}_L} : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$  der euklidischen Distanz entlang  $\vec{d}_L$ . Für jede von einem Eckpunkt geschnittene Gitterzelle  $v_j$  werden die beinhalteten Polygone auf

Punkt-Polygon-Kollision mit  $P_{\mathcal{O}_{\text{Bew}}}$  untersucht. Die daraus berechnete Translationsdistanz ausgehend von  $p_i$  bis zu dem berechneten Schnittpunkt entlang  $\vec{d}_L$  mit den Polygonen aus  $v_j$  sei  $\lambda_i^E$ . Weiter werden die beiden zugehörigen Kanten  $(p_i, p_{i+1}), (p_{i-1}, p_i) \in P_{\mathcal{O}_{\text{Bew}}}$  für jede von  $p_i$  traversierte Gitterzelle  $v_j$  entlang  $\vec{d}_L$  um  $\frac{1}{2}\|d(p_i, \hat{p}_{ij}) + d(p_i, \hat{p}'_{ij})\|$  verschoben. Jetzt wird wiederum ein Raycast entlang beider verschobener Kanten durchgeführt, um die Gitterzellen zu berechnen, die von der verschobenen Kante von  $P_{\mathcal{O}_{\text{Bew}}}$  geschnitten werden. Der Richtungsvektor für diesen Kanten-Raycast sei  $\vec{d}_L^{i+1} = p_{i+1} - p_i$  beziehungsweise  $\vec{d}_L^{i-1} = p_{i-1} - p_i$ . Die gleichzeitige Behandlung beider Kanten eines Eckpunktes ist notwendig, da die diskreten Verschiebungen der Kanten auf den Schnittpunkten  $\hat{p}_{ij}$  und  $\hat{p}'_{ij}$  der Strahlen ausgehend von dem Eckpunkt  $p_i$  basiert und jeder Eckpunkt nur einmal betrachtet wird. In Abbildung 12a ist der Prozess für eine Kante dargestellt. Mit den Polygonen aus diesen Gitterzellen (in Abbildung 12a sind das die rot unterlegten Zellen  $v'_j, v''_j$  und  $v'''_j$ ) werden die Kante-Kante-Kollisionen berechnet. Die aus diesen Tests resultierende Translationsdistanz sei  $\lambda_i^K$ .

Angenommen das statische Objekt  $\mathcal{O}_{\text{Sta}}$  liegt im regulären Gitter wie in Abbildung 12b, dann würde mit dem bis jetzt beschriebenen Vorgehen die obere rechte Gitterzelle  $v_0$  nicht traversiert werden. Gerade in diesem liegt jedoch das Polygon, mit dem der erste Kontakt entsteht. Somit ergibt sich eine fehlerhafte Translationsdistanz. In der Abbildung ist die resultierende translatierte Kante blau eingezeichnet. Die Folgerung aus diesem Beispiel ist, dass die Traversierung der Gitterzellen, die von einem Eckpunkt geschnitten werden, nicht auf das Volumen des regulären Gitters beschränkt werden darf, da sonst die Gitterzelle  $v_0$  aus der Abbildung nicht traversiert werden würde. Dazu wird die bei der Berechnung des Raycasts angenommen, dass das Gitter genau so groß ist, dass der Strahl innerhalb des Gitters liegt (vgl. Abbildung 13).

Die Traversierung für das aktuelle Polygon  $P_{\mathcal{O}_{\text{Bew}}}$  wird abgebrochen, falls

- für mindestens einen Eckpunkt  $p_i$  eine Translationsdistanz  $\lambda_i^E$  oder  $\lambda_i^K$  existiert.
- für  $\lambda_P := \min_{p_i \in P}(\lambda_i^E, \lambda_i^K)$  gilt  $\lambda_P < \lambda_{P,m}$  mit der bisherigen minimalen Translationsdistanz  $\lambda_{P,m}$ , dann ist eine neue temporäre minimale Translationsdistanz gefunden.

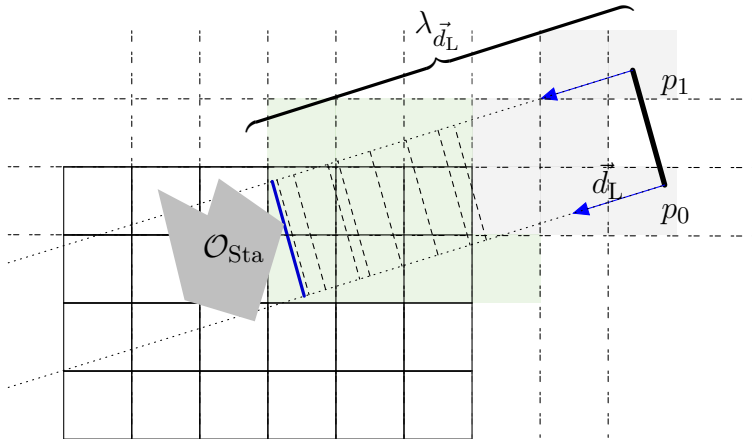


Abbildung 13: Illustration der linearen Traversierung einer Kante  $p_0, p_1$  entlang  $\vec{d}_L$  mit allen temporären translatierten Kanten (gestrichelt): Nur bei den grünen Gitterzellen ist ein Raycast entlang einer temporären Kante erfolgt.

- für keinen Eckpunkt beziehungsweise Kante eine Distanz berechnet werden konnte, da keine Schnitte mit den Polygonen innerhalb der aktuell betrachteten Gitterzellen existieren
- die aktuelle betrachteten Gitterzellen keine Polygone enthalten und gilt:  
 $d(p_i, \hat{p}_{ij}) > \lambda_{P,m}, \forall p_i$ .

Falls keine der Bedingungen zutrifft, wird die nächste Gitterzelle  $v_{j+1}$  entlang  $\vec{d}_L$  berechnet, dessen Abstand zu dem aktuellen Polygon  $P$  der nächst kleinere ist. Die beschriebene Berechnung wiederholt sich solange bis das aktuell betrachtete Polygon vollständig durch das reguläre Gitter hindurchbewegt worden ist oder eine Kollision mit  $\mathcal{O}_{\text{Sta}}$  berechnet worden ist.

Die Abbildung 13 zeigt eine zweidimensionale Illustration der vollständigen Traversierung der Kante  $(p_0, p_1)$  entlang der Bewegungsrichtung  $\vec{d}_L$  mit allen temporären translatierten Kanten und der berechneten Translationsdistanz  $\lambda_{\vec{d}_L}$ . Die betrachteten Gitterzellen sind farblich markiert.

Im zweidimensionalen Raum existiert eine Alternative zu dem Raycast entlang der translatierten Kante basierend auf dem Strahl  $(p_i + \lambda_{\vec{d}_L} \vec{d}_L, \vec{d}_L^{i+1})$ . Man kann abhängig von der Gitterzellengröße die maximale Länge einer bewegten Kante abschätzen und somit eine Kante, die länger als die minimale Seitenlänge einer Gitterzelle ist, unterteilen. Wenn die bewegte Kante kürzer als die minimale Seitenlänge einer Gitterzelle ist, werden alle Gitterzellen, die von der bewegten Kante

durchlaufen werden, von den Raycasts in Richtung  $\vec{d}_L$  ausgehend von den Eckpunkten der Kante berechnet.

Im dreidimensionalen Raum gilt diese Bedingung nicht. Abbildung 14 liefert die Begründung: Für die rote Schnittkante, die bei dem Schnitt der Ebene in der sich die Kante bewegt mit einer Gitterzelle, existiert keine untere Schranke bezüglich ihrer Länge in Abhängigkeit zu der Länge der bewegten Kante. Die Länge der Schnittkante kann beliebig kurz werden.

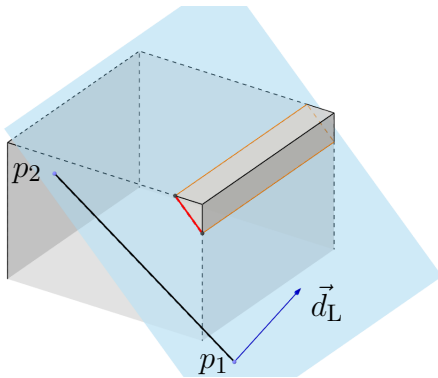


Abbildung 14: Die bewegte Kante  $(p_1, p_2)$  schneidet entlang der Bewegungsrichtung  $\vec{d}_L$  die dargestellte Gitterzelle (rote Strecke). Für diese Strecke existiert keine untere Schranke bezüglich ihrer Länge.

Nachdem bei der Traversierung nur die Kanten eines Polygons verwendet werden, muss noch diskutiert werden, ob eine obere Schranke bezüglich der Größe des Polygons abhängig von der Größe der Gitterzellen existiert, so dass garantiert werden kann, dass alle Gitterzellen betrachtet werden, die auch tatsächlich von dem Polygon durchlaufen werden. Falls der Kontakt zwischen den beiden Objektmodellen ein Punkt-Polygon-Kontakt ist, bei dem die involvierte Ecke zu dem statischen Objekt gehört, kann es abhängig von der Größe des involvierten Polygons aus dem bewegten Objektmodell zu einer fehlerhaften Berechnung der Translationslänge kommen. Nämlich genau dann, wenn bei der Traversierung der Kanten eine Gitterzelle ausgelassen wird, die ausschließlich von dem Inneren des Polygons entlang der Bewegungsrichtung durchstrichen wird. Dieser Fall kann auftreten, wenn eine der Zellenseiten, projiziert auf die Ebene des Polygons im Bezug auf  $\vec{d}_L$ , vollständig im Polygon enthalten ist. In Abbildung 15 ist dieser Fehlerfall konstruiert.

Falls solche Polygone existieren, kann dennoch die korrekte lineare Kontaktpose berechnet werden. Dazu sind zwei Möglichkeiten denkbar. Man führt die gesamte Berechnung zweimal mit getauschter Zuordnung zwischen bewegten und unbewegten Objekt aus. Oder man fügt in die Polygone, die das obige Kriterium verletzen,

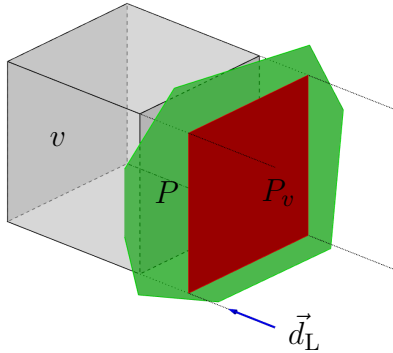


Abbildung 15: Illustration des Falls, dass eine Gitterzelle  $v$  nicht von einer Kante oder Eckpunkt des Polygons  $P$  (grün) durchstrichen wird, wenn eine projizierte Zellenseite  $P_v$  (rot) existiert, mit  $P \cap P_v = P_v$ .

einen zusätzlichen Eckpunkt ein, der ebenfalls traversiert wird. Eine gute heuristische Position dieser zusätzlichen Ecke ist der Schwerpunkt der projizierten Zellenseite.

### 3.4.2 Geometrische Berechnung

Eine Lösung für die lineare kontinuierliche Dreieck-Dreieck-Kollisionsberechnung ist in [Cou13] beschrieben. Dieser kann auf konvexe Polygone erweitert werden, da er auf Punkt-Flächen- und Kante-Kante-Schnitttests basiert. Es wird davon ausgegangen, dass man eine Start- und Zielpose gegeben hat und prüfen will, ob zwischen diesen beiden Posen eine Kollision auftritt. Diese Berechnung basiert auf der Lösung mehrerer kubischer Gleichungssysteme. Die resultierenden Nullstellen sind mögliche Lösungen, müssen jedoch noch durch Punkt-in-Dreieck- oder Punkt-auf-Kante-Tests verifiziert werden.

Nachdem die Traversierung aus Kapitel 3.4.1 die Eckpunkte beziehungsweise die Kanten eines Polygons verwendet, beruht die Kontaktposenberechnung auch auf den beiden Testkombinationen Eckpunkt (bewegt entlang  $\vec{d}_L$ ) mit Polygon und Kante (bewegt entlang  $\vec{d}_L$ ) mit Kante. Bei diesen Berechnungen ist eine nachträgliche Verifizierung von mehreren gefundenen Kontaktposen mittels eines Punkt-in-Polygon oder Punkt-auf-Kante nicht notwendig, da diese Bedingungen während der Berechnung erfüllt werden.



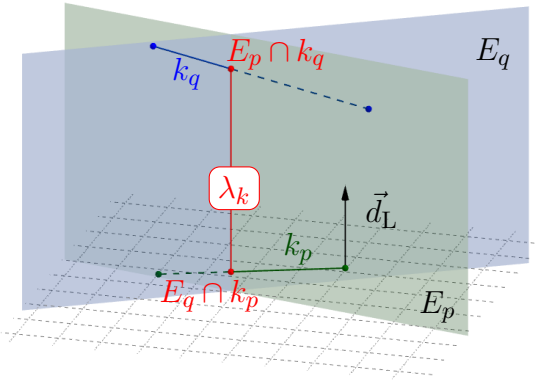


Abbildung 16: Illustration der linearen Kontaktposenberechnung entlang  $\vec{d}_L$  zwischen den Kanten  $k_p$  und  $k_q$  mit den dazugehörigen Hilfsebenen  $E_p$  und  $E_q$ .

**PUNKT-POLYGON** Seien der Punkt  $v \in \mathbb{R}^3$ , ein Polygon  $P$  mit den Eckpunkten  $p_1, \dots, p_n \in P$  und die Bewegungsrichtung  $\vec{d}_L \in \mathbb{R}^3$  mit  $\|\vec{d}_L\| = 1$  gegeben. Gesucht ist die kleinste Distanz  $\lambda^E$  von  $v$  nach  $v + \lambda \vec{d}_L \in P$ .

Sei  $n_P$  die Normale des Polygons. Dann ist

$$\lambda^E := \frac{p_1 \cdot n_P - v \cdot n_P}{n_P \cdot \vec{d}_L}$$

Falls  $v + \lambda^E \vec{d}_L \in P$  gilt, ist die gesuchte Distanz gefunden. Diese Bedingung kann man mit einem geeigneten Punkt-in-Polygon-Test [Tay94] prüfen. Falls zwei Distanzen basierend auf zwei Schnittpunkten existieren, wird muss die kleinere Distanz gewählt werden.

**KANTE-KANTE** Seien die beiden Kanten  $k_p := (p_1, p_2)$  und  $k_q := (q_1, q_2)$ ,  $p_i, q_i \in \mathbb{R}^3$  und die Bewegungsrichtung  $\vec{d}_L \in \mathbb{R}^3$  mit  $\|\vec{d}_L\| = 1$  gegeben. Gesucht ist die Distanz  $\lambda^K$ , so dass  $(p_1 + \lambda^K \vec{d}_L, p_2 + \lambda^K \vec{d}_L) \cap (q_1, q_2) \neq \emptyset$ .

Zur Berechnung von  $\lambda^K$  werden die Hilfsebenen  $E_p : (p - p_1) \cdot n_p = 0$  und  $E_q : (p - q_1) \cdot n_q = 0$  mit

$$n_p := \frac{1}{\|p_1 - p_2\|} (p_1 - p_2) \times \vec{d}_L \quad \text{und}$$

$$n_q \quad \text{analog}$$

konstruiert.

Damit ist die gesuchte Distanz  $\lambda^K := \|E_p \cap k_q - E_q \cap k_p\|$ . In Abbildung 16 ist die Berechnung von  $\lambda^K$  visualisiert. Die Distanz existiert nicht, falls für eine der beiden Kanten gilt, dass sie vollständig auf einer Seite der Hilfsebene, definiert durch die andere Kante, liegt. Wenn somit gilt:  $k_p \cap E_q = \emptyset$  oder  $k_q \cap E_p = \emptyset$ .

## 3.5 ROTATORISCHE KONTAKTPOSENBERECHNUNG

Für die Berechnung der rotatorischen Kontaktpose (Definition 3.4) zwischen den beiden gegebenen Polygonnetzen (Definition 3.3)  $\mathcal{O}_{\text{Bew}}$ ,  $\mathcal{O}_{\text{Sta}}$  mit dem regulären Gitter für  $\mathcal{O}_{\text{Sta}}$  ist zusätzlich die Rotationsachse  $A : p_A + \lambda \vec{d}_A$  gegeben. Sie ist durch einen Stützpunkt  $p_A \in \mathbb{R}^3$  und einen Richtungsvektor  $\vec{d}_A \in \mathbb{R}^3$  definiert. Die Drehrichtung ist durch die Orientierung der Achse  $A$  gegeben. Wie anfangs gefordert, können die beiden Polygonnetze schon einen gemeinsamen Kontakt haben.

Gesucht ist der kleinste Winkel  $\alpha_A$  mit dem das Polygonnetz  $\mathcal{O}_{\text{Bew}}$  um die Achse  $A$  gedreht werden muss, so dass beide Polygonnetze sich mindestens in einem neuen Kontaktpunkt in Richtung der Rotation berühren jedoch nicht durchdringen. Durch die Drehung um den berechneten Winkel entsteht die neue rotatorische Kontaktpose.

Die Herangehensweise ist grundsätzlich gleich zu der Berechnung der linearen Kontaktpose (Unterkapitel 3.4). Die geometrischen Kollisionsberechnungen (Kapitel 3.5.2) basieren auf der Betrachtung von Punkt-Polygon- und Kante-Kante-Kontakten. Die effiziente Auswahl der Polygone des statischen Objektes erfolgt ebenso unter Verwendung des regulären Gitters und basiert auch auf einem Raycast-Ansatz (Kapitel 3.5.1).

3.5.1 *Traversierung des regulären Gitters*

In diesem Kapitel wird das Traversierungsverfahren eines regulären Gitters für Polygonnetze auf einer Kreisbahn beschrieben. Im Kapitel 3.4.1 ist schon das Traversierungsverfahren für eine lineare Bewegung eingeführt worden. Grundsätzlich ist das Vorgehen analog zu diesem.

Die formale Ausgangssituation ist: Sei  $P \in \mathcal{O}_{\text{Bew}}$  das gedrehte Polygon mit den Eckpunkten  $p_1, \dots, p_n \in P$  den Kanten  $k_1 := (p_1, p_2), \dots, k_n := (p_n, p_1)$  und mit der Normalen  $n_P \in \mathbb{R}^3$ , die vom Objekt weg zeigt. Sei weiter  $d_x, d_y, d_z \in \mathbb{R}$  die achsenparallelen Ausdehnungen einer Gitterzelle und  $m_x, m_y, m_z \in \mathbb{N}$  die Anzahl der Gitterzellen in Koordinatenachsenrichtung. Sei die Achse  $A : p_A + \lambda \vec{d}_A$  durch einen Stützpunkt  $p_A \in \mathbb{R}^3$  und einen Richtungsvektor  $\vec{d}_A \in \mathbb{R}^3$  definiert. Die

Drehrichtung ist durch den Richtungsvektor im mathematische positivem Sinn gegeben.

Der grundsätzliche Ablauf der Traversierung ist folgender: Für jedes Polygon des bewegten Polygonnetzes  $\mathcal{O}_{\text{Bew}}$  werden schrittweise die Schnittpunkte mit den Gitterzellen berechnet, die entstehen, wenn man die Ecken des Polygons auf der gegebenen Kreisbahn bewegt. Die zugehörigen Polygonkanten werden zu diesen Schnittpunkten verschoben und die Gitterzellen berechnet, die von dieser transformierten Kante geschnitten werden. Für alle so berechneten Zellen werden die geometrischen Kollisionstest mit den beinhalteten Polygonen  $P \in \mathcal{O}_{\text{Sta}}$  durchgeführt.

Die Berechnung der Schnittzellen mit einer transformierten Kante ist äquivalent zu der aus Kapitel 3.4.1. Die Berechnung der Gitterzellen, die von einer bewegten Ecke geschnitten werden, differiert hingegen. Es kann nicht auf bekannte Raycasting-Verfahren zurückgegriffen werden, da die Steigung der Ein- und Austrittspunkte der Kreisbahn nicht konstant ist.

Im Folgenden wird die Auswahl der bewegten Polygone und die Traversierung der Ecken durch das reguläre Gitter beschreiben. Ein Polygon  $P$  wird nur verwendet, um das Gitter zu traversieren, wenn dessen Normale  $n_P$  in Drehrichtung der Achse  $A$  zeigt. Das heißt wenn gilt:

$$\exists p_i \in P : n_P \cdot \left( (p_i - p_A) \times \vec{d}_A \right) < 0$$

Nur in diesem Fall kann das Polygon mit seiner Vorderseite (in Richtung der Normalen) ein anderes Polygon berühren. Andernfalls wäre nur eine Kantenberührung möglich. Eine solche wird aber von einem benachbarten Polygon schon behandelt, wenn man davon ausgeht, dass ein geschlossenes Polygonnetz vorliegt.

Jeder Eckpunkt  $p_i$  des Polygons und die Achse  $A$  definieren eine Kreisbahn  $K_i$ :

$$\begin{aligned} r_{i,A} &:= d(p_i, A) \\ K_i &:= \left\{ p \in \mathbb{R}^3 \mid d(p, A) = r_{i,A} \wedge (p - p_i) \cdot \vec{d}_R = 0 \right\} \end{aligned}$$

Diese Kreisbahnen werden mit den Trennebenen des Gitters geschnitten. Die  $T_j^x$ ,  $T_j^y$ ,  $T_j^z$  sind wie folgt definiert:

$$\begin{aligned} T_j^x &:= \{(x, y, z)^T \in \mathbb{R}^3 \mid x = j \cdot d_x\} \\ T_j^y &:= \{(x, y, z)^T \in \mathbb{R}^3 \mid y = j \cdot d_y\} \\ T_j^z &:= \{(x, y, z)^T \in \mathbb{R}^3 \mid z = j \cdot d_z\} \end{aligned}$$

mit  $j \in \mathbb{N}$ .

Die Trennebenenschnitte werden ausgehend von der Gitterzelle, in der  $p_i$  liegt, berechnet. Das heißt, wenn  $p_i$  bezüglich der x-Achse in der  $j$ -ten Gitterzelle liegt, werden die Trennebenen in der Reihenfolge  $T_j^x, T_{j+1}^x, \dots, T_{m_{\max}^x}^x$  und  $T_j^x, T_{j-1}^x, \dots, T_{m_{\min}^x}^x$  mit  $K_i$  geschnitten. Dabei ist zu beachten, dass  $m_{\max}^x > m_x$  und  $m_{\min}^x < 0$  erlaubt ist. Sobald für einen Schnitt  $T_l^x \cap K_i = \emptyset$  gilt werden die weiteren Trennebenen  $T_{l+1}^x$  beziehungsweise  $T_{l-1}^x$  und alle Folgenden nicht mehr betrachtet, da kein Schnitt mit  $K_i$  mehr möglich ist. Für die Trennebenen  $T_j^y$  und  $T_j^z$  wird äquivalent verfahren. Die Menge der Schnittpunkte  $S_i$  für einen Eckpunkt  $p_i \in P$  sei wie folgt definiert:

$$S_i := \left\{ q \in T_j^k \cap K_i \mid m_{\min}^k < j < m_{\max}^k; k \in \{x, y, z\}; \angle_A(p_i, q) < \alpha_{\text{cur}} \right\}$$

mit dem zu dem aktuell kleinsten berechneten Drehwinkel  $\alpha_{\text{cur}}$  und der Winkel-funktion  $\angle_A : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$ , die den Winkel zwischen zwei Punkten im Bezug auf die Achse  $A$  berechnet. Durch die Berücksichtigung von  $\alpha_{\text{cur}}$  beinhaltet  $S_i$  keine Schnittpunkte, die nur erreicht werden, falls man über den aktuell kleinsten Drehwinkel hinaus rotieren würde. Für alle berechneten Schnittpunkte eines jeden Eckpunktes  $p_i$  wird der Drehwinkel in mathematisch positiver Richtung definiert durch  $\vec{d}_A$  zwischen  $p_i$  und  $q_j \in S_i$  um die Achse  $A$  berechnet. Diese Winkel zusammen mit den jeweiligen Schnittpunkten bilden das Tupel

$$\begin{aligned} W_i &:= \left( \frac{\beta_1}{2}, \frac{\beta_2 - \beta_1}{2}, \dots, \frac{\beta_{|S_i|} - \beta_{|S_i|-1}}{2} \right) \\ &\quad \beta_j = \angle_A(p_i, q_j) \\ &\quad q_j \in S_i \\ &\quad \beta_j < \beta_{j+1}, \forall j = 1, \dots, |S_i| \end{aligned}$$

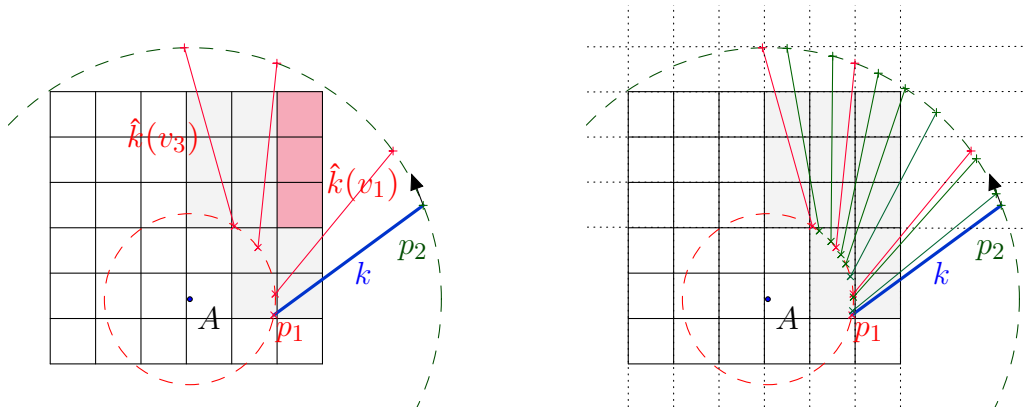
aufsteigend nach den Winkeln sortiert. Falls man direkt die  $\beta_j$  verwendet, kann es auf Grund numerischer Fehler dazu kommen, dass Gitterzellen bei den Berechnungen fälschlicherweise nicht berücksichtigt werden.

Jetzt wird für jede Kante  $k_i$  das Tupel  $W_{k_i} = W_i \cup W_{i+1}$  der zuvor berechneten Schnittwinkel mit den Trennebenen und dazugehöriger Schnittpunkte gebildet. Dieses  $n$ -Tupel ist wiederum nach den Schnittwinkeln sortiert. Algorithmisch lassen sich die Elemente von  $W_{k_i}$  sukzessiv nacheinander berechnen, da die Trennebenen monotone Folgen bezüglich  $j \cdot d_k, k \in \{x, y, z\}$  sind. Damit werden ebenso keine unnötigen Schnittberechnungen mit den Trennebenen  $T_j^x, T_j^y$  und  $T_j^z$  durchgeführt und der Sortieraufwand für die  $W_i$ 's entfällt.

Anschließend wird für jedes Element von  $\beta \in W_{k_i}$  beginnend bei dem ersten Element die Kante um die Achse  $A$  mit dem Winkel  $\beta$  gedreht. Die so konstruierte Kante sei  $\hat{k}_i := (\hat{p}_i, \hat{p}_{i+1})$ . Die weiteren Berechnungen zur Bestimmung der Gitterzellen, die von der transformierten Kante  $\hat{k}_i$  geschnitten werden, sind ausführlich in Kapitel 3.4.1 beschreiben. Für alle Gitterzellen, die von  $\hat{k}_i$  mittels Raycast zwischen den Endpunkten  $\hat{p}_i$  und  $\hat{p}_{i+1}$  geschnitten werden, wird getestet, ob eine kontinuierliche Kollision zwischen den in der Gitterzelle enthaltenen Polygonen und dem bewegten Polygon besteht. Für die Start- und Endzelle wird zusätzlich zum Test zwischen zwei Kanten auch noch der Punkt-Polygon-Test (Kapitel 3.5.2) durchgeführt. Pro Kante wird eine Gitterzelle nur einmal untersucht. Falls eine solche Kollision besteht, existiert ein neuer vorläufiger gesuchter Rotationswinkel  $\alpha_{\text{cur}}$ , um die beiden Objekte in Kontakt über eine Drehbewegung zu bringen. Falls dieser kleiner ist als ein zuvor berechneter wird der kleinere als neues vorläufiges Ergebnis verwendet. Als Konsequenz kann die Traversierung dieser Kante beendet werden. Ein weiteres Abbruchkriterium ist, wenn der gerade betrachtete Schnittwinkel  $\beta_k$  größer ist als ein zuvor gefundenes minimales  $\alpha_{\text{cur}}$ .

Wenn alle Polygone aus  $\mathcal{O}_{\text{Bew}}$  durch das reguläre Gitter bewegt worden sind, entspricht  $\alpha_{\text{cur}}$  dem gesuchten minimalen Rotationswinkel  $\alpha_A$ . Zusammen mit der Achse  $A$  ist die Drehmatrix definiert, die zu der neuen gesuchten rotatorischen Kontaktpose führt.

Falls man nicht zulässt, dass bei der Konstruktion der Mengen  $S_i$  die Grenzen des regulären Gitters überschritten werden, kann bei dem späteren Raycast zwischen den Endpunkten einer Kante nicht garantiert werden, dass alle Gitterzellen, die auch von der Kante bei der Rotation geschnitten werden gefunden werden. In



- (a) Nachdem der Eckpunkt  $p_2$  das reguläre Gitter nicht schneidet, werden die rot markierten Gitterzellen nicht traversiert.
- (b) Nachdem die rotierten Kanten durch die Schnittpunkte mit den zusätzlichen Trennebenen (gestrichelt) konstruiert sind, werden keine Gitterzellen bei der Traversierung ausgelassen.

Abbildung 17: Diese zweidimensionale Illustration zeigt die Traversierung beschränkt auf das reguläre Gitter (a) und die beschriebene erweiterte Variante (b). Die blaue Kante  $k := (p_1, p_2)$  wird um den schwarzen Rotationspunkt  $A$  gedreht. Die Spuren der Eckpunkte sind gestrichelt dargestellt. Die roten Kanten sind rotierte Kanten bezüglich des roten Eckpunktes  $p_1$  und die grünen Kanten sind rotierte Kanten bezüglich des grünen Eckpunktes  $p_2$ .

Abbildung 17 ist eine solche Situation dargestellt. Der äußere Eckpunkt der Ausgangskante (blau) schneidet das reguläre Gitter erst sehr spät und deshalb werden die rot markierten Gitterzellen nicht durch einen Raycast entlang der gedrehten Kanten (schwarz) getroffen. Die in Kapitel 3.4.1 formulierten Bedingungen an die einzelnen Polygone im Bezug auf die Zellengröße sind für die rotatorische Traversierung äquivalent.

### 3.5.2 Geometrische Berechnung

In diesem Unterkapitel wird die Berechnung des Rotationswinkels, um den ein Punkt oder eine Kante gedreht werden kann bis Kontakt mit einem Polygon oder einer anderen Kante besteht, beschrieben.

**PUNKT-POLYGON** Seien der Punkt  $v \in \mathbb{R}^3$ , ein Polygon  $P$  mit  $n$  Eckpunkten  $p_1, \dots, p_n \in \mathbb{R}^3$  und die Rotationsachse  $A : p_A + \lambda_A \vec{d}_R$  gegeben (Abbildung 18a). Gesucht ist der Rotationswinkel um  $A$  der  $v$  auf  $P$  rotiert.

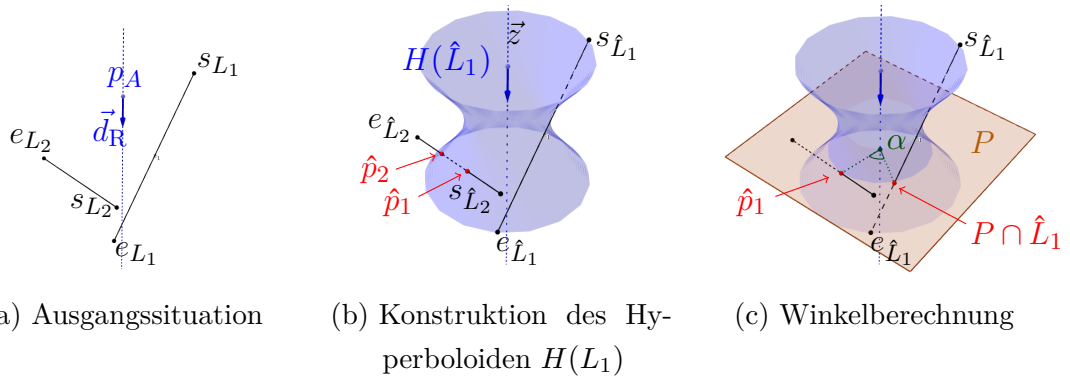
Dafür sei  $E_P \subset \mathbb{R}^3$  die Ebene die durch  $P$  definiert wird. Weiter sei  $K(v, A) \in \mathbb{R}^3$  der Behelfskreis wie folgt definiert:

$$K(v, A) := \left\{ p \in \mathbb{R}^3 \mid (p - p_A) \cdot \vec{d}_R = 0 \right. \\ \left. \begin{array}{l} \exists p_A \in A : \min d(p_A, v), \\ d(p, A) = d(p_A, v) \end{array} \right\}$$

Der Kreis  $K(v, A)$  schneidet die Ebene  $E_P$  in bis zu zwei Punkten  $p_1, p_2 \in E_P \cap K(v, A)$ . Für beide Punkte ist zu prüfen, ob sie innerhalb von  $P$  liegen. Der gesuchte Rotationswinkel ist gerade der Winkel der zwischen  $p_1$  oder  $p_2$  und  $v$  im Bezug auf die Achse  $A$ . Falls beide Punkte innerhalb des Polygons liegen, wird der kleinere von beiden Winkeln bezüglich der gewählten Drehrichtung gewählt.

**KANTE-KANTE** Gegeben seien zwei windschiefe Geradenstücke  $L_1 : s_{L_1} + \lambda_1(e_{L_1} - s_{L_1}) \in \mathbb{R}^3$ ,  $L_2 : s_{L_2} + \lambda_2(e_{L_2} - s_{L_2}) \in \mathbb{R}^3$  und eine Rotationsachse  $A : s_A + \lambda_A \vec{d}_R$  mit den Startpunkten  $s_{L_i} \in \mathbb{R}^3$ , den Endpunkten  $e_{L_i} \in \mathbb{R}^3$ , der Richtung der Achse  $\vec{d}_R \in \mathbb{R}^3$  und  $\lambda_i \in [0; 1]$ . Der Winkel  $\alpha$  rotiert  $L_1$  um  $A$ . Gesucht ist  $\alpha$ , so dass nach der Rotation  $L_1$  und  $L_2$  sich schneiden.

Die Berechnung des gesuchten Winkels umfasst im Wesentlichen vier Schritte: (1) Zuerst wird überprüft, ob der Winkel existiert. (2) Die Winkelberechnung erfolgt mit Hilfe einer allgemeinen Quadrik. Um deren Konstruktion zu erleichtern werden die beiden Geradenstücke und die Achse in ein geeignetes Koordinatensystem transformiert, so dass die Berechnung der Parameter der Quadrik stark vereinfacht ist und insbesondere keine Mischterme entstehen. (3) Zuletzt erfolgt die Berechnung der Parameter der Quadrik und des Rotationswinkels (4) durch den



(a) Ausgangssituation      (b) Konstruktion des Hyperboloiden  $H(L_1)$       (c) Winkelberechnung

Abbildung 18: Illustration der wesentlichen Berechnungsschritte für die rotatorische Kontaktposenberechnung zwischen den Kanten  $L_1$  und  $L_2$  um die Achse  $A : s_A + \lambda \vec{d}_R$

Schnitt zwischen der Quadrik und dem zweiten Geradenstück. Die Abbildung 18 illustriert die beschriebenen Schritte grafisch.

(1) Existenztests: Um sicherzustellen, dass die beiden Geradenstücke sich bezüglich der Rotation auch schneiden müssen die Abstände der Endpunkte bezüglich des Abstands zur Rotationsachse alternieren. Sei ohne Einschränkung der Allgemeinheit die Distanz von  $L_1$  zu  $A$  kleiner als die zwischen  $L_2$  und  $A$ . Dann gilt: Falls

$$\min_{p \in \{s_{L_2}, e_{L_2}\}} d(p, A) > \max_{p \in \{s_{L_1}, e_{L_1}\}} d(p, A)$$

gilt, existiert kein möglicher Schnitt bezüglich der Drehung um die Achse  $A$ .

Die zweite Bedingung für die Existenz des gesuchten Winkels ist, dass die auf die Achse  $A$  projizierten Endpunkte von  $L_1$  und  $L_2$  entlang  $\vec{d}_R$  alternieren müssen. Das heißt, dass keine Ebene  $(x - t) \cdot \vec{d}_R = 0$  mit  $t \in A$  existierend darf, die beide Geradenstrücke separiert.

(2) Wahl eines geeigneten Koordinatensystems: Um die spätere Konstruktion des einschaligen Hyperboloiden durch die Drehung von  $L_1$  um  $A$  zu vereinfachen und insbesondere das Auftreten von Mischtermen bei der geschlossenen Darstellung des Hyperboloiden zu vermeiden, werden  $L_1$  und  $L_2$  so transformiert, dass

1.  $\vec{d}_R$  die z-Achse des neuen Koordinatensystem ist



2.  $t_0 := \operatorname{argmin}_{t \in A} \{d(t, L_1)\}$  der Punkt mit der kürzesten Distanz zu  $L_1$  im Ursprung des neuen Koordinatensystems liegt.

Falls  $t_0$  nicht eindeutig ist (z. B.  $L_1 \parallel A$ ), wird ein beliebiger Punkt auf  $A$  gewählt.

Von nun an sei  $\hat{L}_1$  und  $\hat{L}_2$  die in das konsturierte Koordinatensystem transformierten Geradenstücke  $L_1$  und  $L_2$ .

Nachdem die angewendete Transformation affin ist, werden die Winkel erhalten und man muss den berechneten Winkel nicht zurück in den Ursprungsraum transformieren.

(3) Geometrische Konstruktion: Um den einschaligen Hyperboloiden  $H$  zu konstruieren wird  $\hat{L}_1$  als Generatrix verwendet der um die z-Achse  $\vec{z}$  gedreht wird.

Falls eine Ebene existiert, die koplanar zu der x-y-Ebene ist und  $s_{\hat{L}_1}$  und  $e_{\hat{L}_1}$  enthält, entsteht ein Kreisring. Durch eine Verschiebung von  $e_{\hat{L}_1}$  um  $(0, 0, \epsilon)^T$  aus dieser Ebene, lässt sich dieser Ausnahmefall leicht abfangen.

In allen anderen Lagen von  $\hat{L}_1$  bezüglich  $\vec{z}$  entsteht entweder ein Zylinder, Kegel oder allgemein ein einschaliges Hyperboloid. Alle diese Formen können als generalisierte Quadrik der Form

$$H : \frac{x^2 + y^2}{a^2} - k \cdot \frac{z^2}{b^2} = c.$$

geschrieben werden. Um die Parameter  $k$ ,  $c$ ,  $a^2$  und  $b^2$  zu mit Hilfe von  $\hat{L}_1$  und  $\hat{L}_2$  zu berechnen, werden folgende Hilfspunkte definiert:

- $p_1 := \operatorname{argmin}_{p \in \hat{L}_1} \{d(p, \vec{z})\}$  und
- $p_2 \in \hat{L}_1, p_2 \neq p_1$  und  $d(p_2, \vec{z}) \neq 0$ .

In Tabelle 1 ist die Berechnung der Parameter mit Hilfe von  $p_1$  und  $p_2$  für alle drei möglichen geometrischen Formen abzulesen. Die Parameter  $k$  und  $c$  definieren die Form von  $H$ . Falls  $\hat{L}_1 \parallel \vec{z}$  gilt, entsteht ein Zylinder mit dem Radius  $d(p_2, \vec{z})$ . Falls eine Ebene existiert die sowohl  $\hat{L}_1$  als auch  $\vec{z}$  enthält, entsteht ein Kegel mit dem Steigungsdreieck definiert durch  $d(p_2, \vec{z})$  und  $p_{2,z}$ . Im letzten Fall stehen  $\hat{L}_1$  und  $\vec{z}$  windschief zueinander. Der Radius des Schnittkreises mit einer Ebene parallel zur x-y-Ebene ist  $d(p_1, \vec{z})$  und der Faktor  $\frac{d(p_2, \vec{z})^2}{d(p_1, \vec{z})^2} - 1$  lässt sich als Verdrehung der Geraden auf der Oberfläche von  $H$  im Bezug zu den Geraden auf der Oberfläche eines Kegels interpretieren.

$\frac{x^2+y^2}{a^2} - k \cdot \frac{z^2}{b^2} = c$	k	c	$a^2$	$b^2$
Zylinder	0	1	$d(p_2, \vec{z})^2$	-
Kegel	1	0	$d(p_2, \vec{z})^2$	$p_{2,z}^2$
Einschaliges Hyperboloid	1	1	$d(p_1, \vec{z})^2$	$\frac{p_{2,z}^2}{\frac{d(p_2, \vec{z})^2}{d(p_1, \vec{z})^2} - 1}$

Tabelle 1: Parametrisierung der verschiedenen Rotationskörper durch eine allgemeine Quadrik mit Hilfe von  $p_1 := \operatorname{argmin}_{p \in \hat{L}_1} \{d(p, \vec{z})\}$  und  $p_2 \in \hat{L}_1, p_2 \neq p_1$

(4) Winkelberechnung: Bei dem Schnitt zwischen dem zweiten Geradenstück  $\hat{L}_2$  und der Quadrik  $H$  entstehen im Allgemeinen zwei Schnittpunkte:  $\{\hat{p}_1, \hat{p}_2\} = H \cap \hat{L}_2$ . Falls  $H$  so degeneriert ist, dass es vollständig in einer Ebene liegt kann nur ein Schnittpunkt entstehen.

Mit Hilfe von jedem Schnittpunkt wird folgende Ebene definiert:  $P : (x - \hat{p}_1) \cdot \vec{z}$ . Diese Ebene ist genau die, in der der gesuchte Rotationswinkel existiert. Der Wert des gesuchten Rotationswinkels ist:  $\alpha := \angle_{\vec{z}}(\hat{p}_1, P \cap \hat{L}_1)$ . Im Fall von zwei Schnittpunkten zwischen  $H$  und  $\hat{L}_2$  ist der kleinere Winkel bezüglich der Drehrichtung der gesuchte Winkel.

### 3.6 EXPERIMENTE UND DISKUSSION

In diesem Unterkapitel werden Eigenschaften der Konatkposenberechnung und die beschriebenen Einschränkungen bezüglich des Objektmodells diskutiert.

Bei der Bewertung des Verfahrens zur Kontaktposenberechnung ist die Rechenzeit ein maßgebliches Kriterium, da es immer in Konkurrenz zu klassischen Kollisionstests (Kapitel 3.2) steht. Grundsätzlich ist zu diskutieren, welche und wie viele Polygone des bewegten Polygonnetzes betrachtet werden, wieviele Polygone des statischen Objektes während der Berechnung abgefragt werden und ob es eine obere beziehungsweise untere Schranke bezüglich der Bewegungslänge gibt, die ausgenutzt werden kann.

Nachdem nicht vorausgesetzt ist, dass das bewegte Polygonnetz konvex ist, die Berechnung dennoch exakt sein soll, müssen grundsätzlich alle Polygone des bewegten Polygonnetzes betrachtet werden. Es können nur solche ausgeschlossen werden, die aufgrund ihrer Orientierung bezüglich der Bewegungsrichtung nicht kollidieren können.

Bei der Auswahl der Polygone aus dem statischen Polygonnetz arbeitet das vorgestellte Raycast-basierte Verfahren mit einem regulären Gitter als Raumunterteilung, hierarchische oder nicht reguläre Raumunterteilungen sind auch denkbar. Wie schon in der Einleitung des Kapitels erwähnt, eröffnen gerade die hierarchischen Strukturen ihre Vorteile erst, wenn durch sie ganze Szenen dargestellt werden. Nachdem die Kontaktposenberechnung drauf ausgelegt ist zwei isolierte Objekte zu betrachten stellt eine hierarchische Struktur nur zusätzlichen Aufwand dar, da auf oberen Hierarchiestufen bei einzelnen Objekten selten große zusammenhängende Raumbereiche vorhanden sind, die keine Polygone enthalten.

Die Komplexität und Rechenzeit der Kontaktposenberechnung hängt maßgeblich von der Anzahl der geometrischen Kollisionsberechnungen, da gerade die rotatorischen Kollisionsberechnungen aus Kapitel 3.5.2 sehr rechenintensiv sind.

Wenn man die Traversierung eines einzelnen Polygons betrachtet, werden sequentiell Gitterzellen berechnet, die entlang der Bewegungsrichtung geschnitten werden bis eine Kollision zwischen dem bewegten Polygon und dem unbewegten Polygonnetz auftritt oder die obere Schranke bezüglich der Bewegungslänge überschritten ist. Der häufigste Fall ist, dass, sobald eine Gitterzelle mit enthaltenen Polygonen betrachtet wird, die geometrischen Kollisionstests eine Bewegungslänge berechnen. Damit werden für dieses bewegte Polygon nur so viele geometrische Berechnungen durchgeführt wie Polygone mit einer Gitterzelle assoziiert sind. Der Fall, dass mehrere Gitterzellen mit assoziierten Polygonen betrachtet werden, tritt nur auf wenn das bewegte Polygon nahe am statischen Polygonnetz vorbeibewegt wird oder in einen kleinen nicht konvexen Teil bewegt wird. Die Häufigkeit dieses Falles hängt neben der grundsätzlichen Geometrie des statischen Polygonnetzes mit der Größe der Gitterzellen zusammen, je größer die Zellen, desto häufiger tritt dieser auf. Ein weiterer Faktor, der häufige geometrische Kollisionstests begünstigt, ist die Reihenfolge in der die Polygone des bewegten Polygonnetzes betrachtet werden.

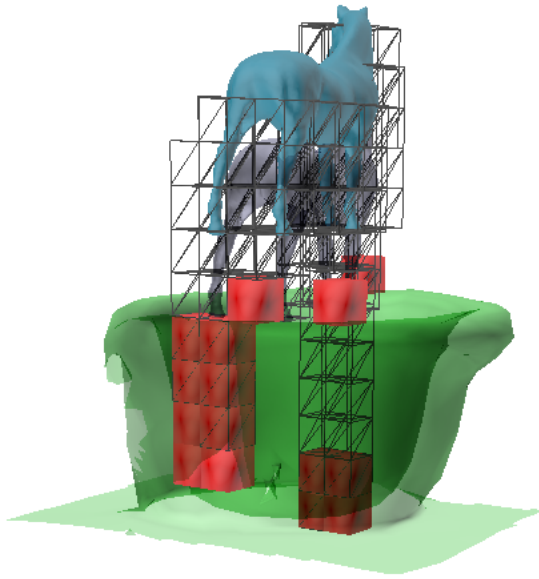


Abbildung 19: Visualisierung der traversierten Gitterzellen: Translation des bewegten Polygonnetzes mit  $t$  Polygonen in Richtung des statischen. Davon sind die Gitterzellen in Drahtdarstellung von  $t_b$  Polygonen traversiert worden, davon enthalten die  $v$  roten Gitterzellen Polygone.

	$v/t_b$	$t_b/t$
Maximum	2,89	0,56
Mittelwert	0,72	0,45
Standardabweichung	0,61	0,06

Unter der Annahme, dass der Quotient betrachteter Gitterzellen mit Polygonen zu der Anzahl an Kanten des bewegten Polygonnetzes klein ist, ist die Rechenzeit unabhängig von der Anzahl der Polygone im statischen Polygonnetz. In Abbildung 19 und 20 sind alle besuchten Gitterzellen markiert. Sowohl bei der linearen als auch bei der rotatorischen Kontaktposenberechnung ist der Faktor<sup>2</sup> zwischen besuchten Gitterzellen mit Polygonen und den betrachteten bewegten Polygonen maximal 2,89 mit einem maximalen Mittelwert von 0,91. Die vergleichsweise hohe Standardabweichung ist durch die Auswahl der Polygone aus dem bewegten Polygonnetz zu erklären. Falls zu einem frühen Zeitpunkt der Berechnung ein gutes Zwischenergebnis gefunden wird, bricht die Traversierung für viele Polygone ab bevor es dazu kommt, dass eine Gitterzelle betrachtet wird, die Polygone enthält. Diese experimentellen Ergebnisse bekräftigen, dass die Anzahl der traversierten Gitterzellen mit assoziierten Polygonen klein ist und als konstant angesehen werden kann.

<sup>2</sup> Zur Bestimmung dieser Faktoren sind zehn verschiedene bewegte Polygonnetze in unterschiedlichen Ausgangsposen mit vier verschiedenen statischen Polygonnetzen verwendet worden. Insgesamt sind 92 Kombinationen evaluiert worden.

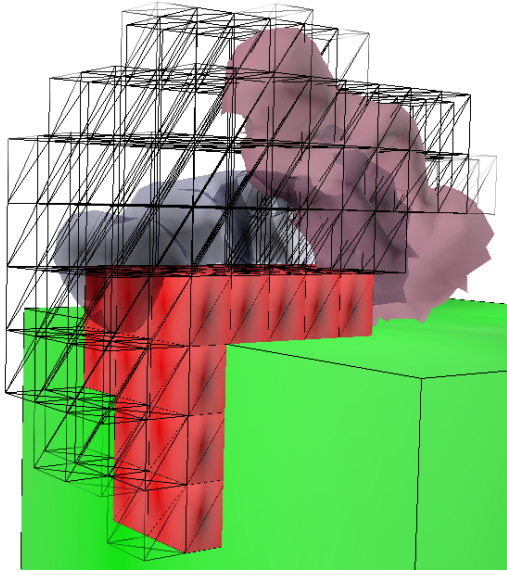


Abbildung 20: Visualisierung der traversierten Gitterzellen: Rotation des bewegten Polygonnetzes mit  $t$  Polygonen in Richtung des Statischen. Davon sind die Gitterzellen in Drahtdarstellung von  $t_b$  Polygonen traversiert worden, davon enthalten die  $v$  roten Gitterzellen Polygone.

	$v/t_b$	$t_b/t$
Maximum	1,89	0,69
Mittelwert	0,91	0,55
Standardabweichung	0,48	0,06

Aus den Tabellen aus Abbildung 19 und 20 geht weiter hervor, dass im Mittel bei kleiner Standardabweichung die Hälfte aller Polygone des bewegten Polygonnetzes für die Berechnungen verwendet werden.

Die Rechenzeit<sup>3</sup> der Berechnung abhängig von der Anzahl der Polygone in beiden Polygonnetzen ist in Abbildung 21 und 22 aufgetragen. Für die Rechenzeitmessung sind Quader mit unterschiedlich vielen Polygonen verwendet worden. Aus den Diagrammen ist zu entnehmen, dass die Rechenzeit erwartungsgemäß linear mit der Anzahl der Polygone des bewegten Objektmodells steigt und die Anzahl der Polygone im statischen Polygonnetz nur einen minimalen Einfluss auf die Rechenzeit hat. Daraus ergibt sich, dass immer das Polygonnetz mit der kleineren Anzahl an

<sup>3</sup> Die Experimente sind auf einem Computer mit einem 22 nm "Ivy Bridge" 2.9 GHz Intel "Core i7" Prozessor (3520M) und 8 GB RAM durchgeführt worden. Das vorgestellte Verfahren ist nicht mit allgemeinen Polygonen implementiert, stattdessen arbeitet es auf Dreiecksmodellen. Der Grund dafür ist, dass das Verfahren insbesondere in dem Greif- und Ablageplaner mit unbekanntem Objekten verwendet werden soll. Diese unbekanntem Objekte werden mit dem Verfahren aus Kapitel 2 rekonstruiert das unter Anderem den Marching Cubes Algorithmus verwendet. Die Rechenzeitexperimente sind pro Polygonnetzpaarung zehnmals mit randomisierten Ausgangsposen begonnen worden. Die Bewegungsrichtung für jeden Testlauf ist so gewählt, dass eine resultierende Kontaktpose garantiert ist.

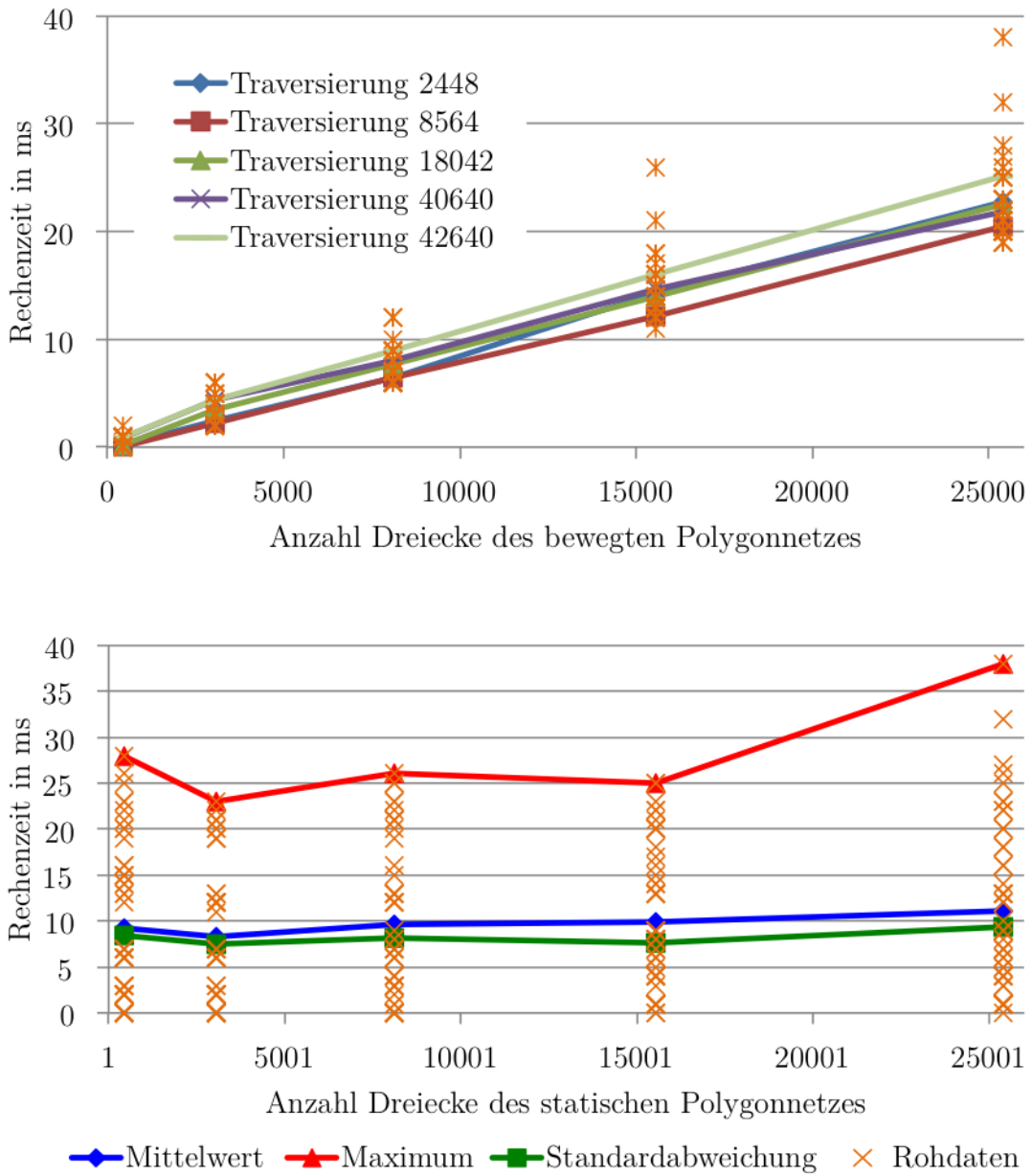


Abbildung 21: Rechenzeit der linearen Kontaktposenberechnung mit unterschiedlicher Anzahl an Polygonen pro Polygonnetz und fester Auflösung des regulären Gitters von  $16 \times 16 \times 16$ . Jede der Kombinationen mit zufälligen Ausgangsposen ist zehnmals ausgeführt worden.

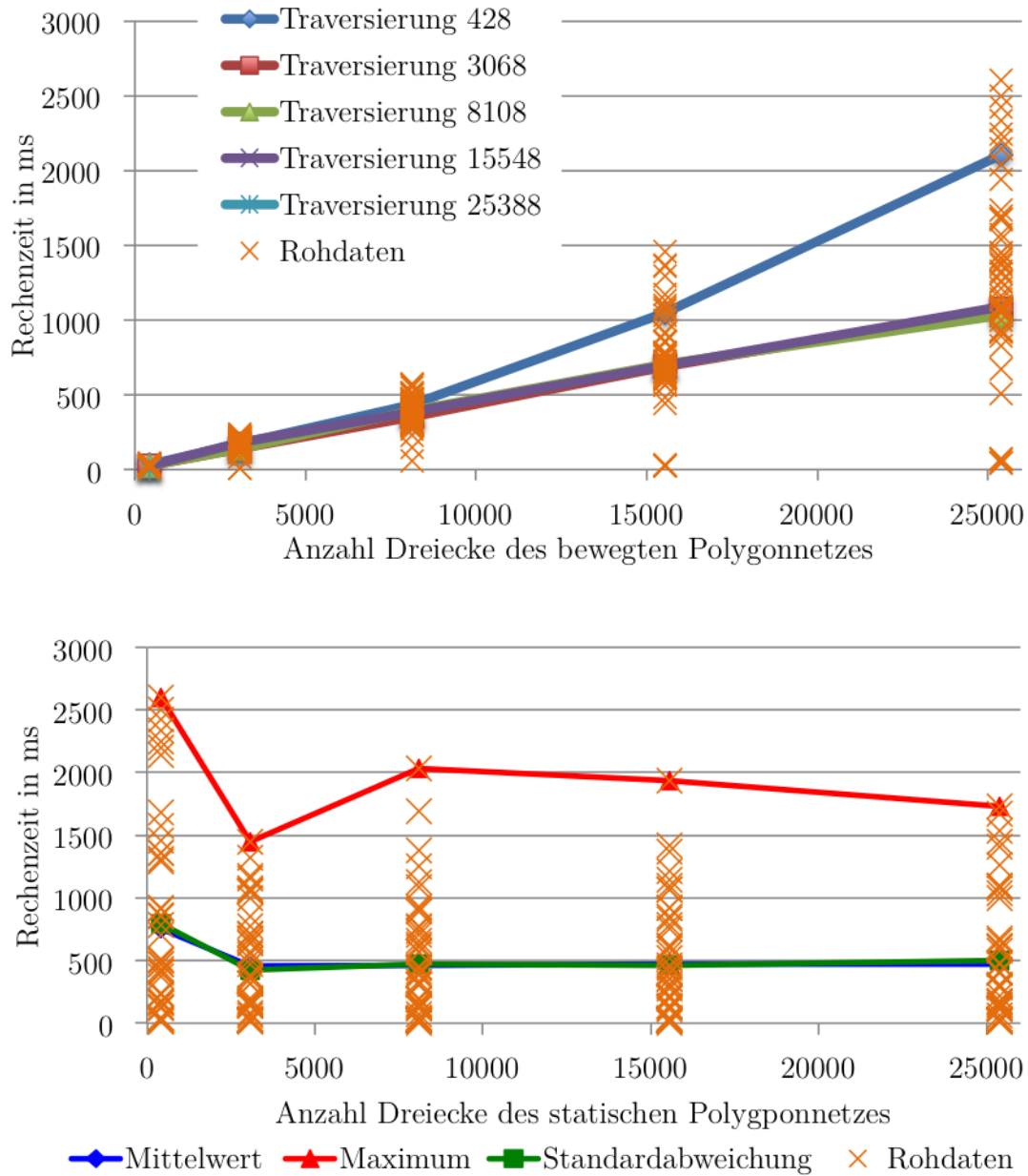


Abbildung 22: Rechenzeit der rotatorischen Kontaktposenberechnung mit unterschiedlicher Anzahl an Polygonen pro Polygonnetz und fester Auflösung des regulären Gitters von  $16 \times 16 \times 16$ . Jede der Kombinationen mit zufälligen Ausgangsposen ist zehnmals ausgeführt worden.

Polygonen bewegt werden muss und die Auflösung des regulären Gitters passend gewählt sein muss, um die Rechenzeit zu minimieren.

Der zweite Aspekt, der die Komplexität und Rechenzeit beeinflusst, ist die Granularität des regulären Gitters. Die Berechnung der nächsten Gitterzelle entlang eines Strahls ist konstant. Somit ist die Komplexität der Traversierung abhängig von der Anzahl der Gitterzellen, die traversiert werden.

Wenn das reguläre Gitter zu hoch aufgelöst ist, steigt die Rechenzeit der Traversierung an und die Anzahl der geometrischen Kollisionstests sinkt, da weniger Polygone einer Gitterzelle zugeordnet sind. Bei niedriger Auflösung des Gitters verhält sich diese Abhängigkeit invers. In Abbildung 23 ist die Rechenzeit<sup>4</sup> abhängig von der Zellenseitenlänge für die rotatorische Kontaktposenberechnung aufgetragen. Daraus ist abzulesen, dass bei einer Seitenlänge von 2 cm die kürzeste Rechenzeit (im Mittel 7 ms) erreicht wird und die Abweichungen der Rechenzeit in den unterschiedlichen Testläufen gering ist. Diese Seitenlänge stellt einen Kompromiss aus der Anzahl der traversierten Gitterzellen und der geometrischen Berechnungen dar.

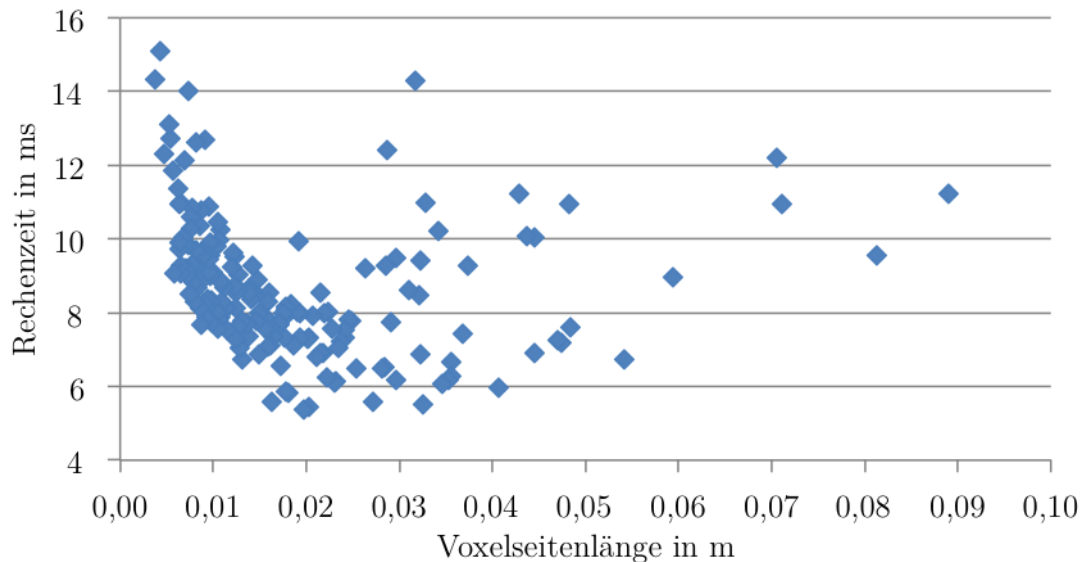


Abbildung 23: Rechenzeit abhängig von der Gitterzellengröße

<sup>4</sup> Für die Analyse sind die gleichen Objektmodelle wie in Abbildung 20 verwendet worden.



In der eigenen Veröffentlichung [Bau+13] basiert die Kontaktposenberechnung noch auf einer Bereichsabfrage auf einem zugrundeliegenden Octree. Dazu wird die Boundingbox des Strahls beziehungsweise der rotierten Kante berechnet und mit dieser Boundingbox alle Knoten des Octrees bestimmt, die innerhalb dieser Boundingbox liegen. Mit den Polygonen auf den selektierten Knoten werden die geometrischen Berechnungen durchgeführt. Durch das Verfahren ist schon offensichtlich, dass eine sehr hohe Anzahl an geometrischen Berechnungen angestoßen werden. In Abbildung 24 wird dieser klassische Ansatz dennoch mit dem Raycast-basierten gegenübergestellt, um zu verdeutlichen, dass die Minimierung der geometrischen Berechnungen schon bei der linearen Kontaktposenberechnung einen starken Verkürzung der Rechenzeit bewirkt. Selbst der Raycast-basierte Ansatz mit über 25000 Polygonen für das statische Polygonnetz berechnet die gleiche lineare Kontaktpose mehr als doppelt so schnell wie die Octree-basierte Bereichsabfrage mit 2448 Polygonen für das gleiche Objekt.

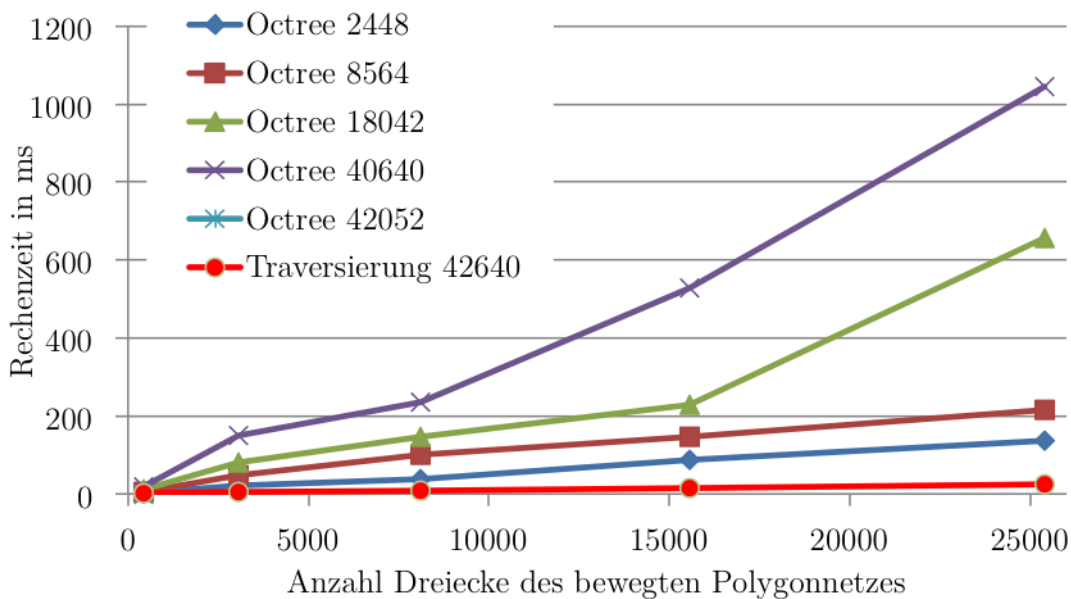


Abbildung 24: Vergleich zwischen Bereichsabfrage mittels Octree aus [Bau+13] zu Raycast-basierter Traversierung aus Kapitel 3.4: Die Polygonnetze sind ebenso konstruiert wie in Abbildung 21.

Falls gilt, dass die Ausgangspose beider Polygonnetze keine Kontaktpose ist, können bekannte Verfahren, wie sie auch in [Tan+14] verwendet werden, und Ab-

schätzungen bezüglich der Bewegungslänge vorgenommen werden. Wenn die Ausgangspose eine Kontaktpose ist, werden schnelle globale Vortests scheitern und deren Berechnung irrelevant. Gerade für diesen Fall ist diese Kontaktposenberechnung ausgelegt.

Wenn weiter gilt, dass die Kantenlängen der Polygone so klein sind, dass der Fall aus Abbildungen 17a und 12b nicht auftritt, kann die Traversierung erst im ursprünglichen reguläre Gitter begonnen werden. Diese Situation ist insbesondere gegeben, wenn beide Objektmodelle mit dem Verfahren aus Kapitel 2 rekonstruiert sind.

In Kapitel 3.3 sind einige Eigenschaften des Polygonnetzes formuliert worden. Diese Annahmen werden im Folgenden nochmals aufgegriffen und diskutiert. Für das Verfahren müssen keinerlei topologische Informationen bezüglich des Polygonnetzes vorliegen. Falls diese dennoch verfügbar sind, können sie ausgenutzt werden, um die Berechnungen für jeden Eckpunkt des Polygonnetzes nur einmal vorzunehmen auch wenn dieser zu mehreren Polygonen gehört. Gleiches gilt auch für die Kanten.

Die einzelnen Polygone des Objektmodells können nicht nur nicht konvex sein, sondern auch Löcher enthalten. Diese Eigenschaft eröffnet neue Möglichkeiten für Objektrepräsentationen, die sich an einer effizienten CAD-ähnlichen Oberflächenmodellierung orientieren. Mit solchen Modellen lassen sich Objekte exakt repräsentieren ohne dabei die Oberfläche durch unverhältnismäßig viele Polygone darzustellen. Der Nachteil ist, dass der Punkt im Polygontest aufwendiger und auch die Berechnung des maximalen Rechtecks innerhalb eines solchen Polygons, um die Traversierung aller Gitterzellen zu gewährleisten, aufwendiger ist. Daher muss abgewägt werden, ob wenige nicht konvexe Polygone verwendet werden oder eine höhere Anzahl konvexer Polygone zur Repräsentation eines Objektes verwendet werden.

### 3.7 SCHLUSSFOLGERUNG

Dieses Kapitel stellt ein Verfahren vor, das es ermöglicht eine neue Kontaktpose für zwei polygonale Oberflächenmodelle zu berechnen. Die Oberflächenmodelle müssen nicht geschlossen sein, es bedarf keiner topologischer Informationen und

sie müssen nicht konvex sein. Die Besonderheit bei dem Verfahren ist es, dass die beiden Modelle schon in Kontakt stehen dürfen und das Verfahren trotzdem in der Lage ist entlang einer gegebenen linearen oder rotatorischen Bewegungsrichtung die nächste Kontaktpose zu berechnen. Dabei wird ein Raycast-basierter Ansatz verfolgt, der sowohl die Eckpunkte als auch die Kanten der Polygone entlang der Bewegungsrichtung kontinuierlich bewegt und somit parameterfrei den Kontaktpunkt beziehungsweise den Abstand entlang der Bewegungsrichtung berechnet.

Der Umstand, dass die Objektmodelle sich schon beim Start des Verfahrens berühren dürfen, ist ein Alleinstellungsmerkmal. Die Rechenzeit des Algorithmus ist linear in der Anzahl der Polygone des bewegten Objektes und unabhängig von der Anzahl der Polygone des statischen Objektmodells, wenn man davon ausgeht, dass die Anzahl der Polygone, die einer Zelle des regulären Gitters zugeordnet sind, vernachlässigbar ist. Die Rechenzeit liegt mit optimaler Parametrisierung zwischen 7 – 12 ms für die rotatorische Kontaktposenberechnung.

Das Verfahren ist speziell für zwei isolierte Objekte ausgelegt und verwendet daher keine hierarchische Struktur, um Freiraum von belegtem Raum zu unterscheiden, da bei vielen Objekten der Freiraum innerhalb des umgebenen Boundingbox nicht groß genug ist, um diesen effizient zu modellieren und damit diese Raumbereiche nicht zu betrachten. Wenn man das Verfahren allerdings darauf ausweiten möchte, dass man es mit einem einzelnen Objekt und einer statischen Szene mit mehreren Objekten verwenden kann, ist eine hierarchische Raumunterteilung wie beispielsweise ein Octree eine sinnvolle Erweiterung, um die Anzahl der traversierten Gitterzellen beziehungsweise Blattknoten zu reduzieren. Die vorgestellten Vorgehensweisen zur Traversierung des regulären Gitters lassen sich leicht auf einen Octree erweitern.

Nachdem der vorgestellte Algorithmus datenparallel bezüglich der Polygone des bewegten Objektmodells ist, lässt sich die Rechenzeit durch eine Parallelisierung auf der CPU oder GPU mit geringem Aufwand verbessern.



# 4

## GREIFPLANUNG

Ein wichtiger Aspekt im Hinblick auf die neuen Einsatzgebiete von Personal Robotern außerhalb der klassischen industriellen Massenfertigung ist die Einsetzbarkeit in unbekanntem Umgebungen. Dabei bestimmt der verwendete Greifer maßgeblich die Manipulationsfähigkeit des gesamten Robotersystems. Dieses Kapitel stellt den Greifplaner vor, der motiviert aus den aktuellen technischen Entwicklungen auf dem Gebiet der Greiferhardware entwickelt worden ist.

In Kapitel 4.1 werden unterschiedliche Greifertypen vorgestellt und eine Entscheidung getroffen, welcher grundsätzliche kinematische Aufbau dem Greifer zu Grunde liegen soll, auf dessen Basis der entwickelte Greifplaner arbeiten soll. Die resultierenden Anforderungen werden in Kapitel 4.2 formuliert und der Stand der Forschung unter Berücksichtigung dieser Anforderungen in Kapitel 4.3 vorgestellt. Anschließend wird in Kapitel 4.4 der Algorithmus beschrieben, der in Kapitel 4.5 experimentell untersucht wird. In Kapitel 4.6 werden die experimentellen Ergebnisse interpretiert und der entwickelte Greifplaner insgesamt bewertet.

### 4.1 STAND DER TECHNIK

Die technischen Entwicklungen der letzten Jahre hat zu einer signifikanten Erweiterung des verfügbaren Portfolios an unterschiedlichen Kinematiken geführt. Aktuelle Greiferhardware lässt sich zum Einen anhand der möglichen Griffotypen und zum Anderen anhand ihrer Kinematik unterscheiden.

Klassenübergreifend ist das Verhältnis von *aktiven* und *passiven Freiheitsgraden* von Bedeutung. Ein Freiheitsgrad wird als aktiv bezeichnet, sobald dieser direkt von einem Aktor angetrieben ist. Ein Freiheitsgrad wird als passiv bezeichnet, sobald er ausschließlich abhängig von anderen Freiheitsgraden angesteuert werden kann. Greifer mit passiven Freiheitsgraden werden auch als *unteraktuierte Greifer* bezeichnet. Je mehr aktive Freiheitsgrade zur Verfügung stehen, desto besser ist

eine Inhand-Manipulation von Objekten möglich. Je mehr passive Freiheitsgrade zur Verfügung stehen, desto einfacher und kostengünstiger ist der kinematische Aufbau der Hardware.

Greifer lassen sich kinematisch danach einteilen, ob Finger/Backen zum Fixieren der Objekte verwendet werden oder nicht. Beispiele für Greifer ohne Finger sind der „Kaffee-Greifer“ [Bro+10] oder Vakuumbreifer [SCH15a]. Eine mögliche Klassifikation der Greiferhardware, die Finger oder Backen besitzen, ist folgende:

**BACKENGREIFER** besitzen einen Freiheitsgrad, um die Greiferbacken zu schließen. Die Anzahl der Backen variiert von zwei bis hin zu vier. Sie werden oft für einfache Manipulationsaufgaben im industriellen Umfeld eingesetzt.

**MEHRFINGERGREIFER** bestehen aus zwei oder drei Fingern mit mehreren Freiheitsgraden pro Finger. Sie sind nicht primär der menschlichen Hand nachempfunden, verfügen aber über ähnliche Manipulationsfähigkeiten. Das vorrangige Designkriterium ist „form follows function“.

**ANTHROPOMORPHE HÄNDE** sind sowohl bezüglich der Größe als auch bezüglich der Kinematik der menschlichen Hand nachempfunden. Beim Design dieser Hände wird besonderen Wert auf eine menschenähnliche Größe und Manipulationsfähigkeit gelegt. Durch ihre im Vergleich zu den Mehrfingergreifer höhere Anzahl an Freiheitsgraden ist eine Inhand-Manipulation von Objekten leichter möglich. Jedoch ist die Ansteuerung dieser Hände bezogen auf Mehrfingergreifer viel komplexer.

Neben den Backengreifern bieten Firmen wie Schunk GmbH & Co. KG, Robotiq, Festo AG & Co. KG oder Barrett Technology Inc. Mehrfingergreifer für den industriellen Einsatz an. Sie sind im Gegensatz zu Anthropomorphe Händen nicht ausschließlich als Forschungsprototypen verfügbar, sondern für eine ständige Benutzung über einen langen Zeitraum entwickelt.

Die Kinematik dieser Greifer ist in der Regel sehr ähnlich aufgebaut. Jeder Finger verfügt über mehrere Gelenke, die es ermöglichen den Greifer flexibel an ein Objekt anzupassen. Die Gelenke werden im Folgenden als *Hauptgelenke* und dementsprechend die zugehörigen Achsen als *Hauptgelenkachsen* bezeichnet, da diese Gelenke direkt verwendet werden, um ein Objekt mit dem Greifer zu fixieren (Abbildung 25).

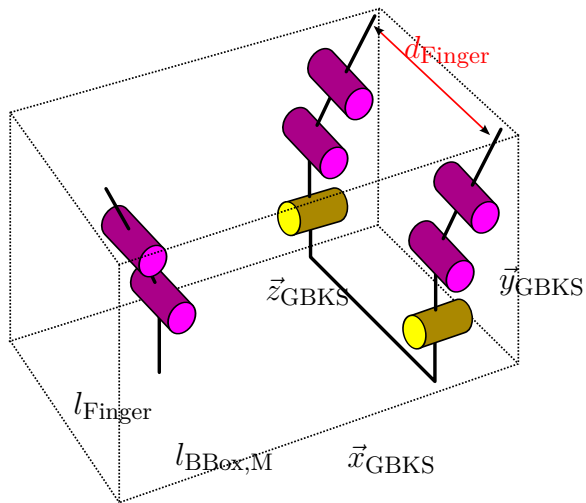


Abbildung 25: Illustration der grundsätzlichen Kinematik eines Mehrfingergrifers (Hauptgelenke (lila) und Nebengelenke (gelb)) und der verwendeten Kenngrößen: Achsen des Greiferbasiskoordinatensystem  $\vec{x}_{\text{GBKS}}, \vec{y}_{\text{GBKS}}, \vec{z}_{\text{GBKS}}$ , Distanz zwischen den Fingern  $d_{\text{Finger}}$  parallel zu  $\vec{y}_{\text{GBKS}}$ , maximale Länge der Finger  $l_{\text{Finger}}$  und Abstand zwischen den Fingern mit Nebengelenken  $l_{\text{BBox},M}$

Oftmals verfügen diese Greifer über einen weiteren Freiheitsgrad im Fingergrundgelenk. Mit diesem Gelenk kann nicht nur der Finger in Richtung Greiferbasis geschlossen werden, sondern die Finger zusätzlich, gespreizt oder gedreht werden. Demzufolge lässt sich die relative Position eines ganzen Fingers bezogen auf die Greiferbasis verändern. Dieses Gelenk wird vornehmlich nicht dazu verwendet, um ein Objekt im Greifer zu fixieren, daher wird dieser Gelenktyp im folgenden als *Nebengelenk* bezeichnet. Greifer, die über ein solches Gelenk in den Fingern verfügen, erlauben eine deutlich höhere Variabilität an möglichen Griffen als gewöhnliche Backengreifer.

Die Nebenachsen lassen sich bezüglich der Veränderung der relativen Pose des Fingers im Bezug auf die Greiferbasis klassifizieren. Dazu sei das Basiskoordinatensystem wie folgt definiert:

**Definition 4.1.** Das Greiferbasiskoordinatensystem (GBKS)  $\mathcal{B}_{\text{GBKS}}$  sei durch eine affine Transformationsmatrix mit den Achsen  $\vec{x}_{\text{GBKS}}, \vec{y}_{\text{GBKS}}, \vec{z}_{\text{GBKS}} \in \mathbb{R}^3$  im Bezug auf das tool-center-Koordinatensystem des Manipulators definiert, dessen Ursprung so gewählt ist, dass dieser mit dem tool-center-Punkt des Manipulators übereinstimmt, wenn der Greifer am Manipulator montiert ist. Die z-Achse (*approach axis*) ist normal zum Flansch des Roboters ausgerichtet und zeigt von diesem weg. Die y-Achse (*sliding axis*) ist die Bewegungsrichtung der Finger. Diese

sind in positiver und negativer Richtung bezüglich der  $y$ -Achse an der Greiferbasis montiert.

Für Drei-Finger-Greifer wird angenommen, dass zwei Finger in Richtung der positiven  $y$ -Achse des Greiferbasiskoordinatensystems und symmetrisch bezüglich dessen  $z$ - $y$ -Ebene an der Greiferbasis montiert sind. Die Distanz zwischen diesen beiden Fingern gemessen an deren Grundgelenk sei  $d_{\text{Finger}} \in \mathbb{R}$ . Der verbleibende Finger ist in negativer  $y$ -Richtung montiert, so dass die Gelenkursprünge des Fingers innerhalb der  $z$ - $y$ -Ebene liegen. In der Grundstellung des Greifers sollen die Achsen aller Fingergelenke normal zu der  $z$ - $y$ -Ebene und parallel zur  $x$ - $z$ -Ebene beziehungsweise  $x$ - $y$ -Ebene des GBKS sein. Abbildung 25 setzt die eingeführten Begriffe Greiferbasis, Hauptgelenk und Nebengelenk nochmals in Bezug zueinander.

**Definition 4.2.** Die Greiferkinematik sei  $G := (d_{\text{Finger}}, l_{\text{Finger}}, l_{\text{Box},M})$  mit Abstand der Finger  $l_{\text{Box},M} \in \mathbb{R}$  parallel zur  $y$ -Achse des GBKS, der maximalen Länge der Greiferfinger  $l_{\text{Finger}} \in \mathbb{R}$  und der Distanz  $d_{\text{Finger}} \in \mathbb{R}$  zwischen den Fingern mit Nebengelenken.

Anhand der Orientierungsänderung der Hauptgelenkachsen lassen sich die Nebenachsen in zwei Typen klassifizieren.

**SCHERENACHSEN** kippen den Finger, so dass dessen Hauptachsen nicht mehr senkrecht auf die  $y$ - $z$ -Ebene und nicht parallel zur  $x$ - $y$ -Ebene des GBKS jedoch weiterhin parallel zur  $x$ - $z$ -Ebene stehen.

**DREHACHSEN** drehen den Finger, so dass dessen weiterhin parallel auf die  $x$ - $y$ -Ebene aber nicht mehr senkrecht auf die  $y$ - $z$ -Ebene des GBKS stehen.

In Tabelle 2 ist für einige Mehrfingergreifer die Anzahl der Freiheitsgrade, sowie die korrespondierende Anzahl von aktiven Freiheitsgraden sortiert nach dem Typ des Nebengelenks aufgelistet. Insbesondere in der Klasse der Mehrfingergreifer ist die Anzahl an passiven Freiheitsgraden besonders hoch. In [Kru+10] werden weitere unteraktuierte Greiferkinematiken beschrieben.

Die Vorzüge eines unteraktuierten Mehrfingergreifers lassen sich folgendermaßen zusammenfassen:



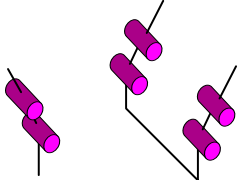
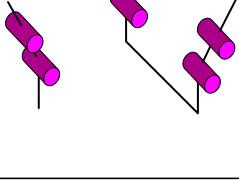
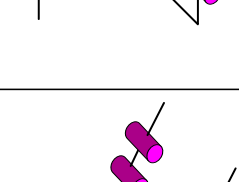
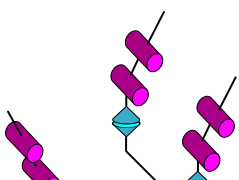
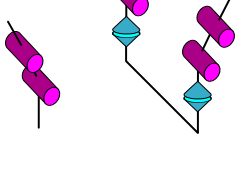
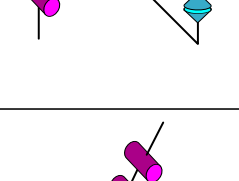
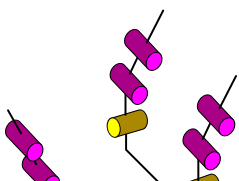
Nebengelenk	Finger	Nebenachsen	Hauptachsen	Aktive Freiheitsgrade	Modell
	2	-	4	2	Willow Garage Velo Gripper
	2	-	4	1	Robotiq Adaptive Gripper 2
	3	-	6	3	Jaco 3 Finger Hand
	3	1	?	?	Schunk SDH Hand
	3	1	7	4	i-HY Hand
	3	1	6	4	Barrett Hand
	3	1	9	4	Robotiq Adaptive Gripper 3

Tabelle 2: Auswahl an verbreiteten Mehrfingergrifern

- Durch die Mehrgelenkigkeit eines Fingers ist der Greifer hardwareseitig in der Lage Unsicherheiten bei der Objektrekonstruktion auszugleichen („Shape adaption by underactuated fingers“[Kru+10]), da sich der Finger um das Objekt herum legen kann.

- Durch die hohe Anzahl von Freiheitsgraden können viele Griffe, die auch ein Mensch realisieren kann, umgesetzt werden. Nachdem der Greifplaner in Domänen, die der Mensch für seine Bedürfnisse geschaffen hat, eingesetzt werden soll ist diese Funktionalität besonders wichtig (Kapitel 1.2).
- Durch die geringe Anzahl der aktiven Gelenke bewegt sich der Planungsaufwand in einem akzeptablen Rahmen und die Komplexität der nötigen Berechnungen ist aufgrund der verringerten Dimensionalität begrenzt.

Es lässt sich feststellen, dass Mehrfingergreifer die aktuell die beste Balance zwischen der hardwareseitigen Variabilität, technologischem Reifegrad und Verfügbarkeit in sich vereinen.

Die *Greifervorkonfiguration* legt das Greiferbasiskoordinatensystem im Bezug auf das Koordinatensystem des Objektmodells und die Gelenkstellung der Nebengelenke fest. Je nach Vorkonfiguration ergeben sich unterschiedliche Grifftypen. Wenn die Greiferbasis Kontakt mit dem Objekt hat, handelt es sich um einen *Kraftgriff* [Nap56]. Die Finger des Greifers drücken das Objekt gegen die Greiferbasis und ermöglichen damit einen besonders stabilen Griff. Wenn das Objekt nur mit den äußersten Segmenten der Finger fixiert wird, handelt es sich um einen *Präzisionsgriff*. Im Vergleich zu Backengreifern ist mit Mehrfingergreifern leichter ein formschlüssiger Griff möglich, da die Finger bereits das Objekt umschließen. Bei Backengreifern kann im Allgemeinen nur ein Kraftschluss hergestellt werden [Kru+10].

Die Greifervorkonfiguration zusammen mit den Gelenkwinkeln der Hauptgelenke führt zu der im Folgenden verwendeten Definition einer Greifer(vor)konfiguration:

**Definition 4.3.** *Greiferkonfiguration* für einen  $n$ -Fingergreifer ist ein  $(n + 2)$ -Tupel  $G := (\mathcal{B}_{\text{GBKS}}, \alpha_{\text{N}}, \mathcal{F}_1, \dots, \mathcal{F}_n)$  mit der Pose des Greiferbasiskoordinatensystems  $\mathcal{B}_{\text{GBKS}}$ , dem Gelenkwinkel der Nebengelenke  $\alpha_{\text{N}}$  und den Gelenkwinkeln der Hauptgelenke pro  $i$ -ten Finger  $\mathcal{F}_i := \{\alpha_1, \dots, \alpha_{m(i)}\}$  mit  $m(i)$  Hauptgelenken. Die dazugehörige *Greifervorkonfiguration* ist  $G_{\text{VK}} : (\mathcal{B}_{\text{GBKS}}, \alpha_{\text{N}})$ .

## 4.2 ANFORDERUNGEN

Aus dem Ziel (Kapitel 1.1) möglichst wenig Informationen bei einer möglichst universellen Funktionalität zu erreichen und der verfügbaren Hardware (Kapitel 4.1) ergeben sich folgende konkrete Anforderungen für den Greifplaner:

**MINIMALE APRIORI INFORMATIONEN:** Bei sensorisch erfassten Objekten (Definition 1.2) ist es schwierig sicherzustellen, ob die Oberfläche eines rekonstruierten Objekts vollständig durch Sensordaten validiert ist. Der Greifplaner soll diese Unsicherheit aktiv berücksichtigen und damit nicht nur ein Qualitätsmaß für eine Greiferkonfiguration liefern, sondern auch ein Unsicherheitsmaß basierend auf der Rekonstruktion bereitstellen.

**GROSSER EINSATZBEREICH:** Der Greifplaner soll möglichst viele verschiedene Objektgeometrien greifen können. Wenn möglich soll ein Griff auch aufgabenabhängig oder sogar unter Berücksichtigung von der Funktionalität des Objektes geplant werden.

**AUSNUTZEN DER HARDWAREVARIABILITÄT,** um die Stabilität der Griffe zu verbessern: Moderne Mehrfingergreifer verfügen über Nebengelenke, die so konfiguriert werden können, dass der Griff eines Objektes stabiler wird. Diese Variabilität soll ausgenutzt werden. Des Weiteren soll automatisch entschieden werden ob sich ein Kraftgriff oder ein Präzisionsgriff angewendet wird.

**ROBUSTE AUSFÜHRBARKEIT:** Die Planung soll insbesondere bei Kraftgriffen eine Greiferkonfiguration (Definition 4.3) berechnen, bei deren Ausführung nur minimale Objektbewegungen relativ zum Greifer zu erwarten sind. Diese Anforderung ist besonders wichtig, wenn kein System zur Verfügung steht, dass eine nachträgliche Evaluation der Objektpose relativ zum Greifer ermöglicht und das Objekt nach dem Griff weiter präzise manipuliert werden soll, beispielsweise wieder abgelegt werden soll.

**INTERAKTIVE PLANUNGSZEIT:** Nachdem mit sensorisch erfassten Objekten (Definition 1.2) gearbeitet werden soll, muss die Planung online erfolgen. Damit verbunden ist, dass ein Griff für ein rekonstruiertes Objekt in einer interaktiven Rechenzeit (Definition 1.5) erreicht werden kann. Durch die hohe

Anzahl der Freiheitsgrade eines Greifers ist der Konfigurationsraum so groß, dass eine vollständige Suche über diesen Raum nicht in kurzer Zeit möglich ist.

### 4.3 STAND DER FORSCHUNG

Die Greifplanung ist eines der wichtigsten Forschungsbereiche auf dem Gebiet der Robotik. In den vergangenen Jahren ist die Fragestellung nach der bestmöglichen Griffkonfiguration aus unterschiedlichen Blickrichtungen und mit diversen Anforderungen untersucht worden. In [Sah+12] werden bestehende Algorithmen in analytische und empirische Ansätze unterteilt. Die analytischen Ansätze sind von [Bic+00] gegliedert worden und wurden von [Shi96] mit besonderem Augenmerk auf Mehrfingergreifer klassifiziert. Diese Ansätze schließen Annahmen im Bezug auf das Material, Gewicht und Massenträgheit des Objekts mit ein und benötigen Objektmodelle die die Qualität eines CAD-Modells haben. Mit diesen Nebenbedingung wird ein Optimierungsproblem formuliert. In dieser Arbeit soll die Planung für unbekannte Objekte erfolgen für die diese detaillierten Information nur teilweise approximiert werden können. Gerade die klassischen Metriken aus der Literatur, die für die Optimierung verwendet werden, haben den Nachteil, dass es bisher in Frage steht, ob eine gute Bewertung basierend auf einer dieser Metriken auch einen guten realen Griff zur Folge hat [Boh+14].

In [Boh+14] werden empirische Ansätze anhand der verfügbaren Informationen bezogen auf die Objekte gegliedert. Grundsätzlich lassen sich die empirischen Ansätze in drei Kategorien unterteilen: Greifplaner die auf *bekanntem*, *ähnlichen* und *unbekanntem* Objekten operieren. Greifplaner [Han+14; Roa+12; Sau+12], die für bekannte Objekte entworfen sind, eignen sich besonders für unterschiedliche Greifer an geometrisch komplexen Objekten zu finden. Dabei sind sie typischerweise für die offline Anwendung konzipiert. In der Kategorie der ähnlichen Objekte sind typischerweise geplante Griffe für wenige verschiedene (Teil-)Objekte offline verfügbar [Det+12; Gol+11; Gol+07; Mil+03]. Der Planungsalgorithmus sucht basierend auf einem Ähnlichkeitsmaß bezogen auf die Objekte den am besten passenden Griff. In diese Kategorie fallen zusätzlich alle Verfahren denen ein Lernverfahren zu Grunde liegt [Sax+08; Gor+09; Her+12; Fis+12]. Das grundsätzliche Problem bei der Adaption von bekannten Griffen auf bekannten Objekten

auf ähnliche Objekte liegt bei der Ausführung der Griffe. Wie in Kapitel 4.2 formuliert soll das Objekt während des Greifvorgangs möglichst nicht seine relative Pose im Bezug auf das Greiferkoordinatensystem verändern. Durch die Adaption auf das vorliegende unbekannte Objekt ist die Abweichung zwischen geplanten und ausgeführten Griff schwierig abzuschätzen. Bei dem vorgeschlagenen hardwareseitigen Aufbau (Kapitel 1) ist eine nachträgliche Überprüfung dieser relativen Pose des Objektes nur eingeschränkt möglich, da der Sensor auf dem Handgelenk des Roboters durch das Objekt selbst verdeckt ist und damit durch den Sensor keine Validierung stattfinden kann.

Greifplaner die mit unbekanntem Objekten arbeiten können, sehen sich mit besonderen Herausforderungen konfrontiert. Für die Berechnungen sind apriori keinerlei Informationen über die Objekte verfügbar. Sie werden zusammen mit Sensoren verwendet, die online Informationen über das Objekt zur Verfügung stellen. Hierbei handelt es sich um taktile Sensordaten [Fel+09] oder wesentlich häufiger um visuelle Sensordaten. Gemäß der Aufgabenstellung werden Ansätze für unbekannte Objekte, die mittels einer Kamera observiert werden, näher betrachtet.

Neben den verfügbaren Informationen über die Objekte können Greifplaner auch bezüglich der Eigenschaften der Objekte klassifiziert werden. Es existieren spezialisierte Planer für deformierbare oder weiche zerbrechliche Objekte [Mai+10], für besonders große Objekte [Vah+11] oder kleine flache Objekte [Epp+15]. Gerade bei zerbrechlichen und kleinen Objekten liegt die Herausforderung in der Regelung der Hardware oder im Einbeziehen der Umgebung. Viele Objekte, die im privaten oder beruflichen Umfeld vorkommen sind starr und sind so robust, dass eine "normale" Handhabung keine besonderen Vorsichtsmaßnahmen voraussetzt. Deshalb wird im Folgenden gemäß Kapitel 1.2 angenommen, dass es sich um starre unzerbrechliche Objekte handelt.

Ein weiteres Merkmal nach dem Greifplaner klassifiziert werden können ist die Greiferhardware mit der die geplanten Griffe ausgeführt werden können. Für unbekannte Objekte werden häufig Backengreifer [Ric+08; Rao+10; Bon+08; Kli+11; Lee+14; Hsi+10; Cio+14; Kru+14] verwendet, da ihre Kinematik besonders einfach ist und damit die Komplexität hardwareseitig abnimmt. Der Nachteil der Beschränkung auf diesen Greifertyp ist, dass ausschließlich Kraftgriffe möglich sind und wegen der fehlenden Hardwarevariabilität geometrisch komplexe Objek-

te schwerer und instabiler zu greifen sind [Shi96]. Wegen der hohen Anzahl an Freiheitsgraden von Mehrfingergreifern ist die Greifplanung basierend auf unbekanntem Objekten mit diesen noch eine offene Fragestellung, insbesondere besteht die Herausforderung genauso wie bei Planern für bekannte Objekte die Variabilität bezüglich der Fingervorkonfiguration auszunutzen.

Im Folgenden werden Greifplaner für unbekannte Objekte und Mehrfingergreifer gemäß den Anforderungen aus Kapitel 4.2 differenzierter betrachtet. Insbesondere wird auf die Unterschiede die Modellierung der unbekanntem Objekte, die Ausnutzung der Hardwarevariabilität und der Genauigkeit des Planungsergebnisses im Bezug auf die tatsächliche Ausführung des Griffes durch den Roboter eingegangen. Die Anforderung, dass die Planung interaktiv im Bezug auf die Planungszeit sein soll, erfüllen alle diese Planer. Ansonsten wäre deren Verwendung in einer Online-Anwendung nicht zielführend, da sie gerade dann eingesetzt werden, wenn durch einen offline-Planer vorhandene Griffe nicht verwendet werden können und der Roboter dennoch seine Aufgabe erfüllen soll.

#### *Modellierung der unbekanntem Objekte*

Die Objektrekonstruktion für den in diesem Kapitel vorgestellten Greifplaner ist schon in Kapitel 2 eingeführt worden. Nachdem bei unbekanntem Objekten die Modellierung des Objektes eng mit dem Greifplaner verwoben ist und sich daraus besondere Eigenschaften für den Greifplaner ergeben, wird an dieser Stelle dieser Teil detailliert beleuchtet. Die Fragestellungen sind: Wie werden Objekte repräsentiert? Wie wird mit unvollständigen Objekten umgegangen? Wird das Objekt global oder lokal betrachtet?

Die Geometrie der Objekte wird gewöhnlich als Punktwolke oder Oberflächenmodell repräsentiert. Nachdem Punktwolken die Geometrie eines Objektes am ineffizientesten im Bezug auf die Datenmenge darstellen, werden diese Rohdaten gewöhnlich in ein Oberflächenmodell überführt. Neben der effizienteren Modellierung der Objekte ist die Vervollständigung der unvollständigen Geometrieinformationen ein der wesentlichen Herausforderungen. Es existieren Ansätze, die das Objektmodell heuristisch oder konservativ vervollständigen.

Bei den heuristischen Ansätzen existieren Verfahren die sich Symmetrieeigenschaften der Objekte zu Nutzen machen. In [Mar+10] werden Rotationssymmetrien identifiziert und in [Boh+11] werden unvollständige Objektgeometrien durch eine Spiegelung der Punktwolke an einer Symmetrieebene vervollständigt. In [Rao+10] wird darauf verzichtet das Objekt zu vervollständigen, es wird aber angenommen, dass ein Punkt auf dem Objekt an dem der Greifer angreift gespiegelt werden kann um somit den zweiten Finger eines Backengreifers zu positionieren. Eine weitere Herangehensweise ist, die Objekte durch Primitive [Mar+10] oder allgemeiner durch Quadriken [Dun+08; Uck+12; Dun+13] zu approximieren. Mit diesem globalen Ansatz lassen sich nur Objekte vervollständigen, deren Geometrie annähernd konvex ist. In [Lip+13] wird eine das Objekt umgebende Ellipse solange geschrumpft und lokal deformiert bis diese die Geometrie des Objektes ausreichend approximiert. Ein weiterer Ansatz ist die Sensordaten durch ein Wahrscheinlichkeitsmodell zu repräsentieren [Mal+10] und damit implizit die sensorisch nicht erfassten Regionen vom Objekt darzustellen.

Bone et al.[Bon+08] rekonstruiert das Objekt in einem zweistufigen Verfahren konservativ (Definition 1.6). Zuerst wird die visuelle Hülle des Objektes rekonstruiert und im zweiten Schritt diese via Space Carving basierend auf Tiefeninformationen verfeinert. Dieser zweistufige Ansatz ist besonders rechenaufwendig, da der Carving-Algorithmus auf dem Oberflächenmodell der visuellen Hülle operiert.

Alle Verfahren haben gemein, dass sie keine Unterscheidung zwischen Oberflächenbereichen die ausschließlich abgeschätzt sind und solchen die durch Sensordaten validiert sind ermöglichen. Der in der vorliegenden Arbeit entwickelte Ansatz verwendet gerade diese Information, da es für die Verifikation und Bewertung einer Greifkonfiguration von enormen Vorteil ist. Dadurch kann man bei der Bewertung des Griffes diese Information mit einbeziehen und zusätzlich neben einem Qualitätsmaß auch eine Unsicherheit bezüglich der Kontaktflächen auf nicht validierten Oberflächenbereichen des Objektes berechnen. In [Chr+07] wird ein solches Unsicherheitsmaß für eine zweidimensionale Bild-basierte Planung vorgestellt, die sowohl die Unsicherheit der Objektrekonstruktion als auch die der Fingerplatzierung in das Qualitätsmaß mit einbezieht.

*Berechnung der Greifervorkonfiguration*

Um mit der hohen Anzahl an Freiheitsgraden und damit mit der hohen Dimensionalität des Konfigurationsraumes des Greifers in einer akzeptablen Planungszeit umzugehen, existieren unter den empirischen Planern vier generelle Ansätze.

Im ersten objektzentrierten Vorgehen werden so genannte *Greifregionen* auf dem Objektmodell gesucht. Eine Greifregion ist ein Bereich an dem der Finger Kontakt mit dem Objekt herstellen soll. Dementsprechend müssen Tupel von Greifregionen gefunden werden deren Mächtigkeit der Anzahl der Finger entspricht, die verwendet werden sollen. Basierend auf diesem Tupel wird die letztendliche Greifkonfiguration mittels der inversen Kinematik des Greifers berechnet. Die Vorteile dieses Vorgehens sind, dass die finale Greiferkonfiguration innerhalb des Nullraumes<sup>1</sup> variabel ist und der Suchraum auf die Oberfläche des dreidimensionalen Objektes beschränkt ist. Gängige Repräsentationen der Greifregionen sind Punkte [Ric+08; Rao+10; Kru14], Kanten [Mor+06; Bau+12] und Flächen [Bon+08]. Bei einfachen Kinematiken wie Backengreifern oder Kinematiken deren Gelenke alle aktiv gesteuert werden können ist dieser Ansatz sehr effizient. Sobald die Kinematik auch passive Gelenke aufweist kann die inverse Kinematik nicht mehr unabhängig vom Objekt berechnet werden. Daher ist der Ansatz für solche Greifer ungeeignet.

Der zweite Ansatz geht davon aus, dass es für eine gegebene Greifervorkonfiguration eine korrespondierende Geometrie existiert, die besonders gut gegriffen werden kann. Zwischen dieser Geometrie und dem Objekt wird ein Matching-Verfahren ausgeführt. Die Pose der Geometrie korrespondiert direkt die Pose des Greifers. Die Fingerkonfiguration ist durch die Geometrie festgelegt. Bei Planern für bekannte vollständige Objekte existieren verschiedene Ansätze die das Objekt (hierarchisch) mit Primitiven wie Quadern [Hue+09; Mil+03], Superquadriken [Uto+13; Gol+07] oder Kugeln [Prz+10] approximieren. Für unbekannte Objekte ist dieses Vorgehen besonders bei Backengreifern verbreitet. Die korrespondierende Geometrie kann ein Quader [Lee+14], ein Rechteck [Buc+13], rotationsymmetrischer Kernel [Buc+14] oder eine extrudierte Parabel [Kli+11] sein. Eppner und Brock [Epp+13] erweitern das Konzept auf Mehrfingergreifer indem sie verschiedene geometrische Primitive mittel eines RANSAC-Algorithmus dem Objekt anpassen. In

---

<sup>1</sup> Der *Nullraum* einer Greiferkonfiguration ist genau der Unterraum des Konfigurationsraums des Greifers, bei dem die Posen der Fingerspitzen für alle Finger identisch sind.



[Pop+11; Koo+12] verwenden Kanten beziehungsweise Flächen Anordnungen die mit einer Fingervorkonfiguration assoziiert sind.

In [Lip+13] wird ein deformierbarer Ellipsoid, auf dem die Positionen der Finger definiert sind, mittels lokaler Deformation an das eigentliche Objekt angepasst. Dabei werden die Positionen der Finger auf dieser Geometrie mit verändert. Der Vorteil ist, dass die Fingervorkonfiguration nicht mehr diskret betrachtet werden muss. Von Nachteil ist, dass dieser Ansatz wiederum die inverse Kinematik des Greifers verwenden muss, um zu ermitteln, ob der Griff möglich ist. Damit ist im ersten Ansatz die Verwendung von Greifern mit passiven Gelenken problematisch. In [Mal+10] wird die Fingerkonfiguration unter der Annahme, dass das Objekt auf einem Tisch liegt, bestimmt. Auch hier ist die Konfiguration der Nebengelenke kontinuierlich. Nachdem das Objekt global betrachtet wird, kann mit diesem Ansatz kein Objekt gegriffen werden, das nicht größer ist als der das maximale Volumen das mit dem Greifer umschlossen werden kann (beispielsweise eine Schüssel).

Der dritte Ansatz verwendet die Kinematik des Greifers. Im Gegensatz zum ersten Ansatz kommt ausschließlich die Vorwärtskinematik zum Einsatz. Es werden Annäherungsrichtungen basierend auf Eigenschaften des Objektes berechnet. Ausgehend von diesen Annäherungsrichtungen wird der Greifer auf das Objekt zubewegt und die Finger basierend auf der Kinematik des Greifers geschlossen. Mit Hilfe der Kontaktpunkte zwischen Greifer und Objekt wird der Griff bewertet. Die Berechnung der Annäherungsrichtung erfolgt basierend auf der Boundingbox [Dia10], den Hauptachsen [Rom+12; Uck+12], dem Schwerpunkt [Boh+11], Kantenfeatures der dreidimensionalen Objektgeometrie [Pop+11] oder den Normalen der Objektfläche [Kru+14; Bau+14a]. Die Fingerbewegungen bis zum Kontakt vom Objekt werden mittels Simulationssoftware realisiert, beispielsweise mit OpenRAVE [Dia10] oder GraspIt! [Mil01], oder mittel eines Optimierungsalgorithmus realisiert [Kru+14]. Die Arbeit von Krug basiert auf dem Optimierungsalgorithmus von [Cio+09]. Dieser verwendet sogenannte *Eigengrasps* um die Dimensionalität des Konfigurationsraums des Greifers auf dem operiert wird einzuschränken, der Berechnungsaufwand der möglichen Greiferposen ist dennoch ohne weitere Einschränkung wie in [Kru+14] erheblich. Die diskreten Annäherungsrichtungen werden in [Kru+15] durch Annäherungsintervalle ersetzt, die direkt als Nebenbedingungen während der Optimierung berücksichtigt werden. Diese Intervalle sind abhängig von der Geometrie des zu greifenden Objektes. Der schwierigen Defini-

tion der Intervalle wird durch einen speziellen Backengreifer vereinfacht, der es ermöglicht die Objekte in den Greifer hineinzuziehen.

Der in dieser Arbeit entwickelte Ansatz verwendet sowohl Annäherungsrichtungen, um deren Vorteil auszunutzen, dass der Greifer nur aus Richtungen an das Objekt herangeführt wird aus denen es die Greiferkinematik und die Umwelt zulässt. Gleichzeitig wird eine Geometrie zur Approximation der Greifervorkonfiguration verwendet, um die lokale Geometrie des Teilobjekts, das der Greifer aus einer Annäherungsrichtung umschließt, zu approximieren. Allerdings wird eine parametrisierte Geometrie verwendet. Dadurch wird der Nachteil eliminiert, dass man diskrete Fingervorkonfigurationen mit festen Geometrien verknüpft. Aus der Parametrisierung der Geometrie wird anschließend der Griffotyp, die Ausrichtung des Greiferbasiskoordinatensystems und die Fingervorkonfiguration berechnet. Durch die Möglichkeit die Fingervorkonfiguration kontinuierlich anzugeben, wird die Kontaktfläche zwischen Greifer und Objekt vergrößert was die Fehleranfälligkeit im Bezug auf externe Kräfteinwirkungen reduziert. Ein weiterer Vorteil ist, dass basierend auf der parametrisierten Geometrie unterschiedliche Greiferkinematiken vorkonfiguriert werden können.

#### 4.4 ALGORITHMUS

Für die Planung stehen dem Algorithmus die Kinematik des Greifers zusammen mit dessen Geometrie als polygonales Oberflächenmodell zur Verfügung. Das zu greifende Objektmodell ist mit der Rekonstruktion aus Kapitel 2 erstellt und unterscheidet validierte und konservativ abgeschätzte Oberflächenbereiche. Um möglichst frühzeitig ungeeignete Greiferkonfigurationen auszuschließen, wird die Kinematik des Roboters und die rekonstruierte Umwelt verwendet.

Gesucht ist eine Menge an bewerteten Greiferkonfigurationen für das Objekt, die vom Roboter kollisionsfrei erreichbar und für eine nachfolgende Manipulation geeignet sind. Dabei soll der Planer automatisch den Griffotyp sowie die Greifervorkonfiguration mit der Konfiguration der Nebengelenke berechnen und die Greiferkonfiguration hinsichtlich der Unsicherheit basierend auf den konservativ abgeschätzten Bereichen und der Qualität basierend auf den validierten Bereichen des Objektmodells bewerten.

Der Ablauf der Greifplanung unterscheidet sich von dem gängigen sampling-basierten Ansätzen für unbekannte Objekte. Diese legen als ersten Schritt der Planung die Gelenkstellungen der Nebengelenke fest. Im vorliegenden Ansatz werden die Gelenkwinkel der Nebengelenke erst nach der Positionierung der Greiferbasis berechnet. Generell lässt sich dieser sampling-basierte Ansatz in mehrere Schritte unterteilen, die jeweils in einem Abschnitt beschrieben werden.

Im ersten Schritt (Kapitel 4.4.1) werden Annäherungsrichtungen unter Berücksichtigung der Erreichbarkeit durch den Roboter und der restlichen Umwelt generiert. Basierend auf diesen Annäherungsrichtungen wird das vom Greifer erreichbare Teilobjektmodell extrahiert, welches durch eine parametrisierbare Geometrie approximiert wird. Basierend auf dieser Geometrie wird der Grifftyp, die Pose der Greiferbasis und die Gelenkwinkel der Nebengelenke berechnet. Im letzten Schritt (Kapitel 4.4.2) werden die Gelenkwinkel der Hauptgelenke bestimmt und die jetzt vollständige Greiferkonfiguration bewertet.

#### 4.4.1 *Greifervorkonfiguration*

Ist eine Greifervorkonfiguration gegeben, bevorzugt diese eine bestimmte Menge von Objektgeometrien. Das Problem ist allerdings invers zu betrachten. Man hat ein Objekt gegeben, für das man eine geeignete Greiferkonfiguration und damit eine Greifervorkonfiguration sucht. Die Idee ist, eine parametrisierbare Geometrie zu verwenden, die möglichst gut die unterschiedlichen geometrischen Formen darstellt, die von den möglichen Greifer- und Fingervorkonfigurationen gegriffen werden kann.

##### *Lokale Approximation der Objektgeometrie*

Eine besonders gut geeignete geometrische Form zur Approximation der verschiedenen zu einer Greifervorkonfiguration korrespondierenden Geometrie sind Superellipsoiden [Bar84], eine Spezialisierung der Superquadriken. Sie lassen sich besonders gut mittels eines Optimierungsverfahrens an eine gegebene Geometrie fitten, da man sie durch eine implizite Funktion darstellen kann. Für die vorliegende Anwendung werden konvexe Superellipsoiden mit einem eingeschränkten Parame-

terraum definiert, damit rotationsinvariant betrachtet jede Geometrie basierend auf den Parametern eindeutig ist:

**Definition 4.4.** Ein *konvexes Superellipsoid*  $\mathcal{S}$  ist durch die Parameter  $0 < \epsilon_1, \epsilon_2 \leq 1$  und  $a_1, a_2, a_3 \in \mathbb{R}^3$  definiert. Er kann in einem kartesischen Koordinatensystem implizit als die Menge aller Punkte  $(x, y, z) \in \mathbb{R}^3$  beschrieben werden, für die folgende Eigenschaft gilt:

$$\mathcal{S} : \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1$$

Die Parameter  $a_1, a_2, a_3$  sind Skalierungen entlang der drei Koordinatenachsen. Damit definieren sie auch die achsen-parallele Boundingbox des Superellipsoids mit den Seitenlängen  $2a_i$ . Die Parameter  $\epsilon_1$  und  $\epsilon_2$  beeinflussen die Krümmung des Superellipsoids:

$\epsilon_1$  - die Krümmung des Schnitts zwischen dem Superellipsoiden und einer Ebene orthogonal zur x-y-Ebene, die die z-Achse enthält.

$\epsilon_2$  - die Krümmung des Schnitts zwischen dem Superellipsoid und einer Ebene parallel zur x-y-Ebene.

Sowohl  $\epsilon_1$  als auch  $\epsilon_2$  definieren eine Lamésche Kurve (auch Superellipse) innerhalb einer der jeweiligen Ebenen. Durch die Variation beider  $\epsilon_i$  entstehen unter anderem die speziellen Körper Würfel, Zylinder, Kugel, Zylinder, Doppelkegel, Steinmetzkörper und Oktaeder (Abbildung 26). Darüber hinaus sind alle Zwischenformen möglich.

In der Definition 4.4 ist nur der konvexe Superellipsoid definiert. Wenn  $\epsilon_i < 0$  erlaubt wird, sind auch Formen möglich die nicht konvex sind. Die Einschränkung des Parameterraumes bezüglich der  $\epsilon_i$  vereinfacht das Fitting des Superellipsoids an ein (Teil-)Objektmodell. Problematisch ist allerdings, dass diese geometrische Form stets drei Symmetrieebenen aufweist. Durch diese Eigenschaft lässt sich beispielsweise ein Ei schlecht durch einen Superellipsoid approximieren. Deshalb wird auf den Superellipsoid noch eine globale Deformation, das *Tapering*, [Bar84] angewendet. Hierfür werden die zwei *lineare Verjüngungsparameter*  $k_x, k_y$  eingeführt.

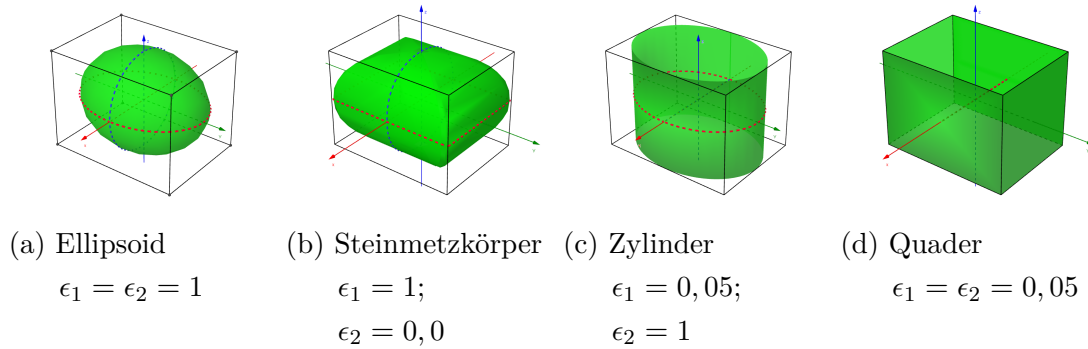


Abbildung 26: Illustration ausgewählter Parametrisierungen eines Superellipsoids dessen Geometrie einem speziellen Körper entspricht: Für alle Superellipsoiden ist  $a_1 = a_3 = 0.3$  und  $a_2 = 0.4$ . Die Werte für  $\epsilon_1$  und  $\epsilon_2$  sind jeweils unter der Darstellung angegeben.

Sie variieren die Steigung der linearen Funktion, die die Skalierung der  $x$ - bzw.  $y$ -Koordinate eines Punktes im Bezug auf dessen  $z$ -Koordinate verändert:

$$s_x(z) := \left( k_x \frac{z}{a_3} + 1 \right)$$

$$s_y(z) := \left( k_y \frac{z}{a_3} + 1 \right)$$

mit  $-1 \leq k_x, k_y \leq 1$ . Diese Deformation entspricht einer Skalierung mit variablen Skalierungsparametern  $(s_x(z), s_y(z), 1)$ . Die Abbildung 27 zeigt zwei beispielhafte Deformationen eines Ellipsoids.

Zusätzlich zu den Parametern des Superellipsoids und der für die globale Deformation muss die Pose (Definition 3.1) des Superellipsoids berechnet werden. Diese Pose wird durch die Parameter  $(p_x, p_y, p_z, \alpha, \beta, \gamma)$  mit der Translation  $(p_x, p_y, p_z)$  und den Eulerwinkeln  $(\alpha, \beta, \gamma)$  repräsentiert. Somit ist das vollständige Parametertupel für einen Superellipsoid  $\mathcal{S}$  in allgemeiner Lage definiert:

$$\Lambda := (a_1, a_2, a_3, e_1, e_2, k_x, k_y, p_x, p_y, p_z, \alpha, \beta, \gamma)$$

Mit dieser geometrischen Form, parametrisiert durch  $\Lambda$  lässt sich der Arbeitsraum eines Mehrfingergrifiers approximieren. Somit ist die Grundvoraussetzung gegeben, dass aus einer Teilgeometrie eines Objektmodells eine Greifervorkonfiguration berechnet werden kann.

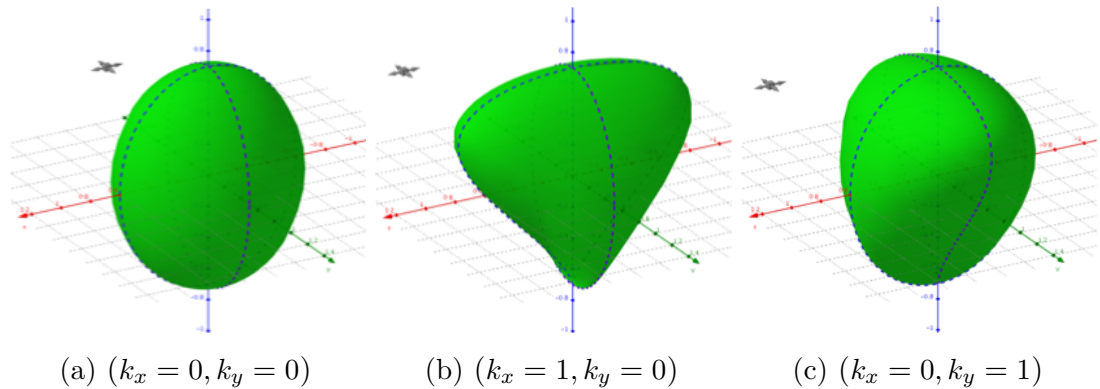


Abbildung 27: Illustration zweier Anwendungen der Verjüngung als globale Deformation auf einen Ellipsoid ( $\epsilon_1 = 1, \epsilon_2 = 1$ ): In (a) ist die undeformierte Ellipse abgebildet und in (b),(c) das Resultat der Deformation mit jeweils den angegebenen Parametern.

Die Berechnung der Greifervorkonfiguration läuft in folgenden Schritten ab. Ausgehend von initialen Annäherungsrichtungen werden die Schnitte zwischen dem maximalen Arbeitsraum des Greifers und dem Objekt berechnet. An diesen Schnittgeometrien wird jeweils ein Superellipsoid mittels des Levenberg-Marquardt Verfahrens angepasst. Anhand der Parametrisierung der Superellipsoiden zusammen mit der Greiferkinematik werden die vollständigen Greifervorkonfiguration berechnet. An dieser Stelle ist zu bemerken, dass ein Superellipsoid nicht alle Teilgeometrien mit der nötigen Genauigkeit approximieren kann, so dass der daraus resultierende Griff mit dem geplanten Griff übereinstimmt. Bei einer schlechten Approximation kann es während des Schließens der Finger dazu kommen, dass das Objekt relativ zur Greiferbasis verrutscht, insbesondere dann, wenn ein Kraftgriff ausgeführt wird. Um einen solchen Fehler möglichst auszuschließen wird gegebenenfalls während der Berechnung des Greiferbasiskoordinatensystems die Pose der Greiferbasis durch ein Kippen optimiert.

Es wird ein zweistufige Vorgehensweise angewandt, um ein Teilobjektmodell zu bestimmen, welches zur Berechnung der Greiferkonfiguration verwendet wird. Im ersten Schritt wird basierend auf einer Annäherungsrichtung ein Teil des Objektmodells identifiziert, der verwendet wird um die letztendliche Annäherungsrichtung zu berechnen.

Zur Bestimmung des Teilobjektes werden drei Vorgehensweisen zur Berechnung der initialen Annäherungsrichtung eingeführt. Deren Ziel ist es eine möglichst vollständige aber überschneidungsfreie Abdeckung des Objektmodells durch die ausgewählten Teilobjekte zu garantieren. Die Annäherungsrichtungen basieren auf der (1) Boundingbox des Objektmodells, der (2) umschließenden Kugel des Objektmodells und einer (3) zufälligen Auswahl der Annäherungsrichtungen.

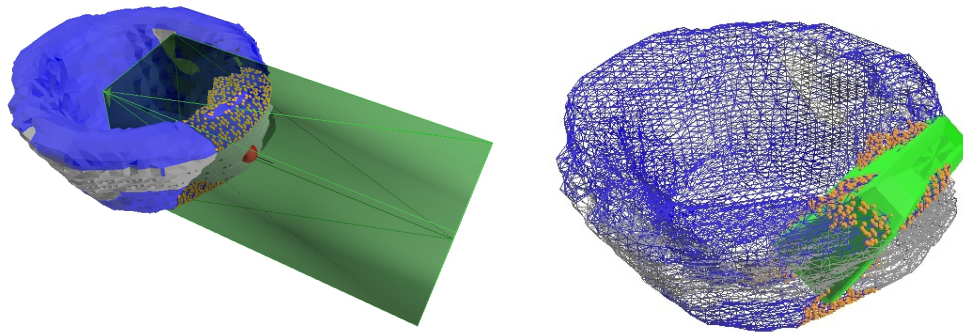
Alle Vorgehensweisen haben ein gemeinsam genutztes Verfahren zur Berechnung des Teilobjektes. In Abbildung 28a sind die im Folgenden beschriebenen Schritte illustriert. Es wird ein einfacher Quader als Hilfsgeometrie berechnet, dessen Größe auf dem Arbeitsraum der Greiferfinger basiert. Dieser ist als achsenparallele Boundingbox in einem eigenen Koordinatensystem repräsentiert. Zur Berechnung des Quaders seien ein Punkt  $p_{\mathcal{O}} \in \mathbb{R}^3$  auf der Oberfläche des Objekts zusammen mit einer Richtung  $\vec{r}_{\mathcal{O}} \in \mathbb{R}^3$ , die in Richtung der Normalen im Punkt  $p_{\mathcal{O}}$  zeigt, gegeben.

Das Koordinatensystem  $T_{\text{AR}}$  des Quaders definiert sich aus den gegebenen Information wie folgt: Die Richtungsvektoren sind durch die Drehmatrix gegeben, die die  $z$ -Achse des kartesischen Koordinatensystems so dreht, dass die  $z$ -Achse identisch mit  $\vec{r}_{\mathcal{O}}$  ist. Die Position des Ursprungs ist  $p_{\mathcal{O}} + d \cdot \vec{r}_{\mathcal{O}}$ . Die Wahl von  $d$  muss größer sein als die Ausdehnung des Arbeitsraums der Greiferfinger in  $z$ -Richtung des Greiferbasiskoordinatensystems (GBKS). Die achsenparallele Boundingbox  $B_{\text{AR}}$  innerhalb dieses Koordinatensystems ist damit wie folgt definiert:  $B_{\text{AR}} := (p_{\text{m}}, p_{\text{M}})$  mit

$$p_{\text{m}} := \frac{1}{2}(l_{\text{BBox,M}}, l_{\text{BBox,M}}, 0)^{\text{T}}$$

und  $p_{\text{M}} := p_{\text{m}} + (l_{\text{BBox,M}}, l_{\text{BBox,M}}, d + l_{\text{Finger}})^{\text{T}}$  mit  $l_{\text{Finger}}$  und  $l_{\text{BBox,M}}$  aus Definition 4.2. Alle Polygone  $P$  des Objektmodells  $\mathcal{O}$ , die innerhalb von  $B_{\text{AR}}$  liegen und innerhalb einer Gitterzelle mit dem Label *visual* (Kapitel 2.3.4), bilden das gesuchte Teilobjekt  $\mathcal{O}_{\text{AR}} \subseteq \mathcal{O}$ , das innerhalb des approximierten Arbeitsraumes des Greifers liegt.

Zur Berechnung der Tupel  $(p_{\mathcal{O}}, \vec{r}_{\mathcal{O}})$  für die initialen Annäherungsrichtungen werden die oben genannten drei verschiedene Varianten eingeführt und verglichen. Jedes dieser Tupel stellt eine neue Annäherungsrichtung dar. Die probabilistische Möglichkeit ist zufällig  $n$  Polygone aus dem Objektmodell zu ziehen. Diese Polygone müssen innerhalb einer Gitterzelle, die das Label *visual* (siehe Kapitel 2)



(a) Berechnung der Schnittgeometrie basierend auf einer Annäherungsrichtung (b) Darstellung des angepassten Superellipsoids

Abbildung 28: Illustration zur Auswahl des Teilobjekts für das Fitting-Verfahren: Das Objektmodell zeigt Sensor-validierte Oberflächenbereiche in blau und nicht validierte Oberflächenbereiche in grau. In grün ist der auf dem rot eingezeichneten Oberflächenpunkt berechnete Auswahlquader eingezeichnet. Die gelben Oberflächenpunkte sind die Schwerpunkte der Sensor-validierten Oberflächenpatches, die innerhalb eines Auswahlquaders liegen. In hellgrün ist der an die Schwerpunkte angepasste Superellipsoid eingezeichnet.

hat, liegen. Der Schwerpunkt des Polygons und dessen Normale definieren zusammen ein Annäherungstupel. Die anderen beiden Verfahren greifen auf die Boundingbox bzw. eine das Objektmodell umschließende Kugel, deren Ursprung im Schwerpunkt des Objektes liegt, zurück. Es wird ausgehend von Punkten auf diesen beiden Oberflächen mittels eines Raycasts (Kapitel 3) in Richtung des Schwerpunktes des Objekts der Schnittpunkt mit dem Objektmodell berechnet. Die inverse Richtung des Strahls und dieser Schnittpunkt ergeben ein Annäherungstupel.

In Abbildung 29 ist ein Vergleich der drei Varianten zur Berechnung der initialen Annäherungsrichtung gezeigt. Bei jeder Variante sind sechs Annäherungsrichtungen berechnet worden. Das verwendete Objekt illustriert die unterschiedenen Vor- und Nachteile, da das Objekt höher als die Ausdehnung des Arbeitsraumes der



Greiferfinger ist und nicht vom Greifer umschlossen werden kann. Die Abdeckung der Oberfläche des Objektmodells ist bei der Boundingbox- und Kugel-Variante annähernd identisch. Hingegen ist die Abdeckung bei der zufälligen Auswahl der Annäherungsrichtung gerade bei Objektmodellen mit vielen Oberflächenpolygone deutlich geringer. Nachdem das Objektmodell höher ist als die Ausdehnung des Arbeitsraumes der Greiferfinger liegen die Polygone außerhalb der Auswahlquader. Die Auswahl basierend auf der umschließenden Kugel berücksichtigt hingegen diese Polygone. Bei Objekten, die vom Greifer umschlossen werden können, sind zwischen den drei Verfahren keine signifikanten Unterschiede bezüglich der Abdeckung der Objektoberfläche festzustellen. Letztendlich liefert die Berechnung der Annäherungsrichtung basierend auf der umschließenden Kugel theoretisch die besten Ergebnisse, in Praxistests ist allerdings kein Unterschied zu dem Ansatz basierend auf der Boundingbox festzustellen. Basierend auf der Einfachheit der Berechnung ist die Boundingbox-Variante als Standard-Ansatz gewählt worden.

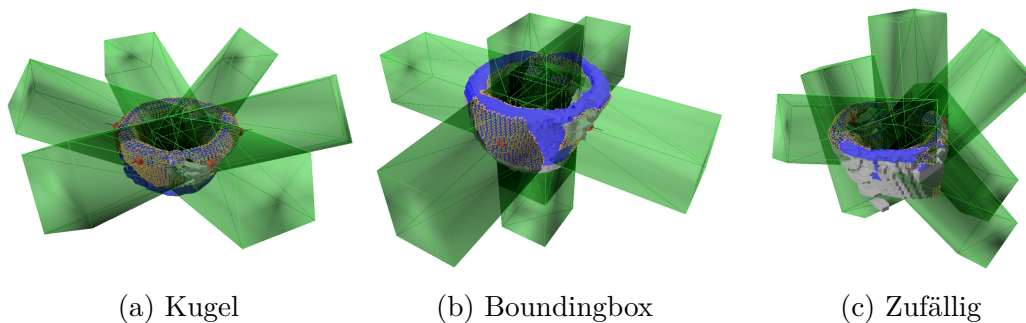


Abbildung 29: Illustration dreier Varianten zur Auswahl der Teilobjekte für das Fitting-Verfahren: Das Objektmodell ist in blau für Sensorvalidierte Oberflächenbereiche und in grau für nicht validierte Oberflächenbereiche eingezeichnet. In grün sind jeweils die sechs, auf den rot eingezeichneten Oberflächenpunkten, berechneten Auswahlquader eingezeichnet. Die gelben Oberflächenpunkte sind die Schwerpunkte der Sensor-validierten Oberflächenpatches, die innerhalb eines Auswahlquaders liegen.

Das selektierte Teilobjekt  $\mathcal{O}_{AR}$  wird im nächsten Schritt durch ein Superellipsoid mit der Parametrisierung  $\Lambda_{AR}$  approximiert. Als Eingabe dienen die Schwerpunkte

der in  $\mathcal{O}_{\text{AR}}$  enthaltenen Polygone. Dazu wird ein um Nebenbedingungen erweitertes Levenberg-Marquardt-Verfahren [Mor78] verwendet. In der Literatur findet man verschiedene Arbeiten [Zha03; Coc+12; Dun+13; Bie+07; Sol+90], die ebenfalls auf dieses Optimierungsverfahren zurückgreifen. Sie unterscheiden sich hauptsächlich in der Wahl der Zielfunktion und der Verwendung von Nebenbedingungen. In [Zha03] werden verschiedene Zielfunktionen diskutiert. Die in [Sol+90] vorgeschlagene Zielfunktion wird ebenfalls hier verwendet. Basierend auf der impliziten Gleichung einer Superellipse wird folgende Funktion definiert:

$$F(x, y, z) := \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} \quad (4)$$

Falls  $F(x, y, z) = 1$  liegt der Punkt  $(x, y, z)$  auf der Hülle des Superellipsoids, falls  $F(x, y, z) < 1$  liegt er innerhalb und falls  $F(x, y, z) > 1$  liegt er außerhalb bezogen auf die Hülle des Superellipsoids.  $F$  ist bisher nur im lokalen Koordinatensystem des Superellipsoids definiert. Nachdem diese parametrisierte Form eine beliebige Pose im Bezug auf das Objektkoordinatensystem haben soll und zusätzlich noch mittels Tapering deformiert sein soll, muss ein Punkt des Objektmodells noch in das Bezugskordinatensystem mit  $T^{-1}(p_x, p_y, p_z, \alpha, \beta, \gamma)$  transformiert werden und mit  $S^{-1}(k_x, k_y)$  undeformiert werden, damit sich  $F$  aus Gleichung (4) anwenden lässt. Damit ist die vollständige Bewertungsfunktion  $F(\Lambda; x, y, z)$  wie folgt definiert:

$$F(\Lambda; x, y, z) = \left( F \circ S^{-1}(k_x, k_y) \circ T^{-1}(p_x, p_y, p_z, \alpha, \beta, \gamma) \right) (x, y, z)$$

Nun kann die Zielfunktion für  $n$  Punkte formuliert werden:

$$\min_i \sum_{i=0}^n \left( \sqrt{a_1 a_2 a_3} (F(\Lambda; x_i, y_i, z_i)^{\epsilon_1} - 1) \right)^2 \quad (5)$$

wobei der Faktor  $\sqrt{a_1 a_2 a_3}$  garantiert, dass die Superellipse die minimale Größe hat. Das Potenzieren mit  $\epsilon_1$  sorgt für eine schnellere Konvergenz und für eine bessere numerische Stabilität, da durch  $\epsilon_1$  im Exponenten von  $F(x, y, z)$  dividiert wird [Sol+90].

Die Initialisierung des Levenberg-Marquardt Verfahrens ist besonders wichtig für die Konvergenz [Zha03]. Eine gute grobe Startparametrisierung für  $\Lambda$  sind

die Eigenvektoren des Teilobjekts  $\mathcal{O}_{\text{AR}}$  als Basiskoordinatensystem des Superellipsoids und die halbe Ausdehnung der minimalen Boundingbox für die  $a_i$ . Die verbleibenden Parameter sind  $\epsilon_1 = \epsilon_2 = 0,5$  und  $k_x = k_y = 0$ .

Das Fehlermaß zur Beurteilung der Güte ist durch

$$\sum_{i=0}^n (F(\Lambda; x_i, y_i, z_i)^{\epsilon_1} - 1)^2 \quad (6)$$

gegeben. Damit entspricht es bis auf den Faktor  $\sqrt{a_1 a_2 a_3}$  der Zielfunktion (5). Dieses Fehlermaß fließt in die Bewertung des Griffes mit ein, nachdem es ein Maß ist, wie gut  $\mathcal{O}_{\text{AR}}$  durch den Superellipsoid approximiert ist. Das Resultat einer solchen Approximation ist in Abbildung 28b illustriert.

Mit den berechneten Parametern  $\Lambda_{\text{AR}}$  wird die Pose des GBKS festgelegt, der Griffityp bestimmt und die Winkel der Nebengelenke des Greifers berechnet.

#### *Berechnung des Greiferbasiskoordinatensystems*

Ohne aufgabenbedingte, roboterabhängige oder szenenabhängige Nebenbedingungen existieren für jede Achse des lokalen Koordinatensystems  $K(\Lambda_{\text{AR}})$  des Superellipsoids vier Orientierungen für das GBKS. Damit ergeben sich theoretisch 24 Konfigurationen für des GBKS. Eine Konfiguration ist: Die z-Achse des GBKS  $\vec{z}_{\text{GBKS}}$  identisch mit der x-Achse  $\vec{x}_{K(\Lambda_{\text{AR}})}$ , die x-Achse des GBKS parallel zur z-Achse von  $K(\Lambda_{\text{AR}})$  und der Ursprung des GBKS mit den Koordinaten  $(0, 0, -d_{\text{GBKS}})$  bezogen auf  $K(\Lambda_{\text{AR}})$ . Nach dem gleichen Schema lassen sich alle anderen Konfigurationen für das GBKS bestimmen. Es sind jedoch nicht alle diese Konfigurationen zielführend, weshalb ihre Auswahl an Nebenbedingungen geknüpft ist, die im Folgenden dargelegt werden. Weiter wird darauf eingegangen wie sich  $d_{\text{GBKS}}$  berechnet.

Falls  $|k_x(\Lambda_{\text{AR}})| \gg |k_y(\Lambda_{\text{AR}})|$  gilt ist der Superellipsoid stark unsymmetrisch bezüglich der x-y-Ebene von  $K(\Lambda_{\text{AR}})$  (vgl. Abbildung 27b). Folglich werden alle Konfigurationen für das GBKS ausgeschlossen, bei denen gelten würde  $\vec{z}_{\text{GBKS}} \parallel \pm \vec{x}_{K(\Lambda_{\text{AR}})}$ . Gleiches gilt für  $|k_y(\Lambda_{\text{AR}})| \gg |k_x(\Lambda_{\text{AR}})|$  bezüglich der y-Achse von  $K(\Lambda_{\text{AR}})$ .

Eine weitere Bedingung für die Existenz einer bestimmten Konfiguration des GBKS ist die axiale Ausdehnung  $a_i$  von  $\mathcal{S}(\Lambda_{\text{AR}})$ . Falls gilt  $a_i > 0,5l_{\text{BBox},M}$  (Definition 4.2) passt der Superellipsoid nicht zwischen die ausgestreckten Finger des

Greifers. Wenn beispielsweise  $a_1 > 0,5d_{\text{BBox},M}$  gilt, werden alle Konfigurationen mit der Ausrichtung  $\vec{x}_{\text{GBKS}} \parallel \pm \vec{x}_{K(\Lambda_{\text{AR}})}$  verworfen. Gleiches gilt für alle  $a_i$ . Alle Bedingungen und daraus resultierenden Konfigurationen sind in Tabelle 3 eingetragen. In Abbildung 30 sind alle auf der Grundlage der Annäherungsrichtungen aus Abbildung 29b resultierenden GBKS eingezeichnet.

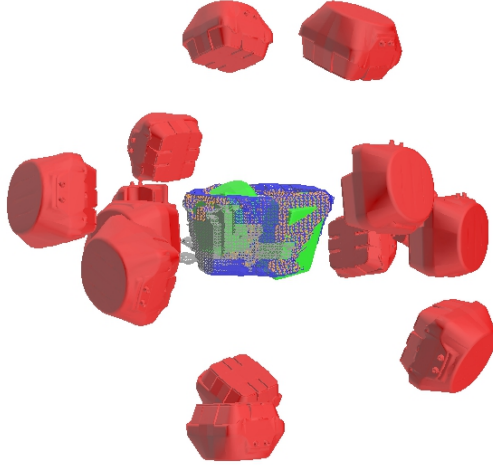


Abbildung 30: Illustration der Posen der Greiferbasis (rot) basierend auf den angepassten Superellipsoiden (grün) mit den Annäherungsrichtungen aus Abbildung 29b

Um die Berechnung von weiteren Greifervorkonfigurationen, die eine gegebene Nebenbedingung verletzen, abubrechen, muss zuerst das vollständige GBKS definiert sein, bisher ist ausschließlich dessen Orientierung festgelegt. Für eine vollständige Definition fehlt noch die Position des Ursprungs. Diese ist abhängig vom Grifftyp. Dazu muss zuerst dieser ermittelt werden. Hierbei wird wiederum auf die Parametrisierung von  $\mathcal{S}$  zurückgegriffen.

Ein ein Präzisionsgriff wird angestrebt, wenn die Ausdehnung  $2a_i$  entlang der ausgewählten Achse, die parallel zu  $\vec{z}_{\text{GBKS}}$  ist, kleiner als die halbe Fingerlänge  $\frac{1}{2}l_{F,M}$  (Definition 4.2) des Greifers ist. Es sei erwähnt, dass ein Präzisionsgriff nicht garantiert werden kann, da nicht sicher gestellt wird, dass nur das letzte Segment eines Fingers Kontakt mit dem Objekt hat. Die Position des Ursprungs des schon orientierten GBKS bezüglich  $K(\Lambda_{\text{AR}})$  ist in diesem Fall:  $(p_x, p_y, p_z)_{\Lambda_{\text{AR}}} - (l_{F,M} - a_i) \vec{z}_{\text{GBKS}}$ .

Wenn allerdings ein Kraftgriff angestrebt wird, kann die Konfiguration des GBKS nicht direkt aus  $\Lambda_{\text{AR}}$  abgeleitet werden, sondern muss berechnet werden. Um der Anforderung nachzukommen, dass bei Kraftgriffen das Objekt seine Lage relativ

$a_i$	$k_i$	$\vec{z}_{\text{GBKS}} \parallel$	$\vec{x}_{\text{GBKS}} \parallel$
$a_2 > a_3$	$k_y > 0$	$\pm \vec{x}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{z}_{K(\Lambda_{\text{AR}})}$
	$k_y < 0$	$\mp \vec{x}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{z}_{K(\Lambda_{\text{AR}})}$
$a_2 < a_3$	$ k_y  < 0.4$	$\vec{x}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{y}_{K(\Lambda_{\text{AR}})}$
$a_1 < a_3$	$ k_x  < 0.4$	$\vec{y}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{x}_{K(\Lambda_{\text{AR}})}$
$a_1 < a_2$		$\pm \vec{z}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{x}_{K(\Lambda_{\text{AR}})}$
$a_1 > a_2$		$\pm \vec{z}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{y}_{K(\Lambda_{\text{AR}})}$
$a_1 > a_2$	$k_x > 0$	$\pm \vec{y}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{z}_{K(\Lambda_{\text{AR}})}$
	$k_x < 0$	$\mp \vec{y}_{K(\Lambda_{\text{AR}})}$	$\pm \vec{z}_{K(\Lambda_{\text{AR}})}$

Tabelle 3: Festlegung der Orientierung des GBKS bezüglich der Parametrisierung  $\Lambda_{\text{AR}}$  des Superellipsoids

zum Greifer während des Zugreifens und weiteren Roboterbewegungen möglichst nicht ändert, muss besonderen Wert auf die Positionierung der Greiferbasis gelegt werden. Da das Objekt während dem Schließen der Finger gegen die Greiferbasis gedrückt wird, muss der Kontaktzustand zwischen Greiferbasisgeometrie und Objekt ausgeglichen sein. Das bedeutet, dass die Distanz zwischen Schwerpunkt der Kontaktfläche und  $\vec{z}_{\text{GBKS}}$  klein sein soll.

Um das GBKS im Fall eines Kraftgriffes zu berechnen werden folgende Schritte durchgeführt. Zuerst initialisiert man das GBKS so als würde ein Präzisionsgriff angestrebt werden. Diese Vorkonfiguration wird verwendet, um den geöffneten Greifer mittels der linearen Kontaktposenberechnung aus Kapitel 3.4 in Richtung  $\vec{z}_{\text{GBKS}}$  an das Objekt zu bewegen. Mit dieser kontinuierlichen Berechnung ist gleichzeitig sicher gestellt, dass diese Annäherungsbewegung kollisionsfrei ist. Falls gleichzeitig die direkte Umgebung mit berücksichtigt wird, kann auch eine kollisionsfreie Bewegung in der gesamten Szene garantiert werden. Dieses Vorgehen ist in [Bau+14a] veröffentlicht worden.

Basierend auf dem Kollisionsobjektmodell aus Kapitel 2.3.4 berechnet man die Kontaktpunkte  $\mathcal{K}$  zwischen Objekt und Greifergeometrie. Die Lage des Schwerpunktes  $c_{\mathcal{K}} := (\mathcal{K}_{\mathcal{K},x}, \mathcal{K}_{\mathcal{K},y}, \mathcal{K}_{\mathcal{K},z})^T$  und der Größe der Kontaktfläche in der x-y-Ebene des GBKS sind ein Maß für die Stabilität des Kontaktes zwischen geöffneten

Greifer und Objekt. Wenn mit dem Schwellwert  $\sigma_{\mathcal{K}}$  gilt  $d(c_{\mathcal{K}}, \vec{z}_{\text{GBKS}}) > \sigma_{\mathcal{K}}$ , ist anzunehmen, dass während des Schließens der Finger sich das Objekt relativ zum GBKS bewegen wird.

Wenn  $\sigma_{\mathcal{K}}$  verletzt ist, wird das GBKS noch einmal angepasst. Dazu soll der Greifer mittels der rotatorischen Kontaktosenberechnung (Kapitel 3.5) so gekippt werden, dass  $d(c_{\mathcal{K}}, \vec{z}_{\text{GBKS}})$  kleiner wird. Hierfür muss noch die Rotationsachse  $A : s_A + \lambda \vec{d}_A$  und darin inbegriffen die Rotationsrichtung bestimmt werden. Die Fehlerrichtung  $\vec{d}_{\text{FR}} := (-p_{\text{KG},x}, -p_{\text{KG},y}, 0)^T$  zeigt von  $c_{\mathcal{K}}$  Richtung  $\vec{z}_{\text{GBKS}}$ . Der Aufpunkt  $s_A$  der Rotationsachse sei der Kontaktpunkt in  $\mathcal{K}$ , der am weitesten entfernt in Richtung  $\vec{d}_{\text{FR}}$  bezüglich des Ursprungs vom GBKS liegt. Die resultierende Rotationsachse wird wie folgt berechnet:  $\vec{d}_A := \vec{d}_{\text{FR}} \times \vec{z}_{\text{GBKS}}$ . Mit dieser Rotationsachse wird die Greifergeometrie an das Objekt heran gekippt. Diese Kippbewegung wird maximal zweimal ausgeführt. Dieser Fall tritt genau dann ein, wenn der erste Kontakt zwischen Greifer und Objekt ein Punktkontakt gewesen ist und beim zweiten Kontakt alle Kontaktpunkte auf einer Geraden in der x-y-Ebene verortet sind.

Diese optimierte Pose der Greiferbasis wird insbesondere bei Objekten benötigt, deren Geometrie sich nicht durch ein einzelnes Primitiv modellieren lässt. Abbildung 31 zeigt ein Beispiel eines solchen Objektmodells inklusive initialer Pose der Greiferbasis (rot) und der optimierten Pose (grün).

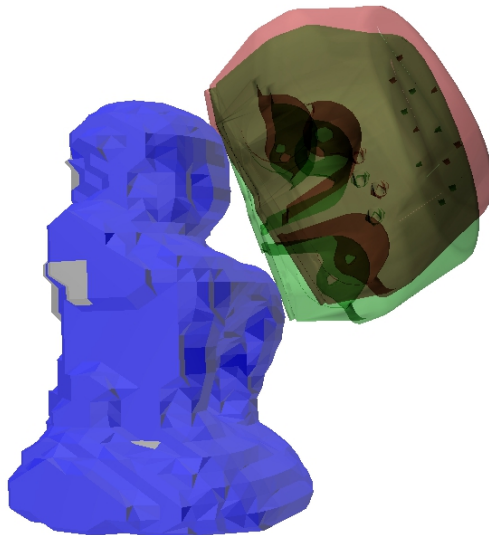


Abbildung 31: Beispielsituation für eine optimierte Pose (grün) der Greiferbasis ausgehend von der initialen Pose (rot), berechnet auf Basis der Parametrisierung des Superellipsoids

Durch die kontinuierliche Bewegung des Greifers (Kapitel 3.4) im Bezug auf das Objektmodell ist diese Optimierung effizient berechenbar. Nachdem die Kippbewegungen abgeschlossen sind ist das finale GBKS für einen Kraftgriff gefunden.

Zur vollständigen Definition der Greifervorkonfiguration müssen, falls vorhanden, zusätzlich die Winkel der Nebenachsen berechnet werden.

### *Berechnung der Gelenkwinkel der Nebengelenke*

In Kapitel 4.1 sind die Nebengelenke in Scherengelenke und Drehgelenke klassifiziert worden. Die Berechnung des korrespondierenden Gelenkwinkels läuft unterschiedlich für beide Gelenke ab, mit dem Ziel die Kontaktfläche zwischen Greiferfingern und Objekt zu vergrößern, um den Griff gegenüber äußeren Momenten zu stabilisieren und die Anzahl valider berechenbarer Greiferkonfigurationen zu erhöhen.

**DREHACHSEN** Bei rotatorischen Gelenken ist eine mögliche Antwort vergleichsweise einfach: Die Finger sollen so orientiert werden, dass die Kontaktfläche zwischen Objekt und Greifer bei gegebenem GBKS maximiert wird. Das heißt, dass die Oberfläche des Fingers tangential an das Teilobjekt angelegt werden muss. Nachdem das Teilobjekt, das innerhalb des Arbeitsraumes des Greifers liegt, durch ein Superellipsoid approximiert ist, lässt sich die gesuchte Orientierung des Fingers basierend auf dieser Form berechnen.

Nachdem ein Finger mehrere Segmente hat, die sich alle die gleiche Bewegungsebene teilen, und die Greiferbasis an den Hauptachsen des Superellipsoids ausgerichtet ist, kann man jeweils einen der beiden Krümmungsparameter außer Acht lassen. Diese Behauptung wird durch folgenden Umstand verdeutlicht: Die Krümmungsparameter  $\epsilon_1$  und  $\epsilon_2$  wirken in zueinander orthogonalen Ebenen,  $\epsilon_2$  in allen Ebenen parallel zur x-y-Ebene und  $\epsilon_1$  in allen Ebenen orthogonal zur x-y-Ebene, die die z-Achse enthalten, des lokalen Koordinatensystemes des Superellipsoids. Somit ist die Lamésche Kurve, die den Schnitt des Superellipsoids mit der x-y-Ebene beschreibt, abhängig von den Parametern  $a_1, a_2, \epsilon_1 \in \Lambda$ . Für die anderen Hauptebenen werden analog nur  $\epsilon_2, a_3$  und  $a_1$  oder  $a_2$  benötigt.

Diese zweidimensionale Lamésche Kurve sei für einen Punkt  $(x, y) \in \mathbb{R}^2$  wie folgt definiert:

$$\mathcal{S}_{2D} : \left(\frac{x}{a}\right)^{\frac{2}{e}} + \left(\frac{y}{b}\right)^{\frac{2}{e}} = 1. \quad (7)$$

Um die Tangente an  $\mathcal{S}_{2D}$  bezüglich eines Punktes  $(x_p, y_p)$  zu berechnen, muss der Punkt auf  $\mathcal{S}_{2D}$  berechnet werden, der am nächsten an  $(x_p, y_p)$  liegt, um die Normale an dieser Stelle berechnen zu können. Diese Berechnung ist ein Optimierungsproblem [Bre+98], wenn man den euklidischen Abstand als Maß verwendet. Diese aufwendige Berechnung soll vermieden werden. Eine gute Approximation der Tangente ist ausreichend, da man davon ausgehen kann, dass das zugrunde liegende Teilobjekt nicht exakt durch  $\mathcal{S}$  repräsentiert ist. Daher wird an dieser Stelle eine approximative Berechnung der Normalen beziehungsweise der Tangente vorgeschlagen, die auf der radialen euklidischen Distanz  $d_r$  basiert. Alle folgenden Berechnungsschritte sind in Abbildung 32 exemplarisch dargestellt. Diese Distanz  $d_r$  ist wie folgt definiert:

$$d_r := \sqrt{(x_p - x_{\mathcal{S},r})^2 + (y_p - y_{\mathcal{S},r})^2}$$

Der Punkt  $p_{\mathcal{S},r} = (x_{\mathcal{S},r}, y_{\mathcal{S},r})$  ist der Schnittpunkt zwischen dem Ortsvektor von  $(x_p, y_p) \in \mathbb{R}^2$  und  $\mathcal{S}_{2D}$ . Dessen Berechnung folgt aus (7):

$$p_{\mathcal{S},r} := \begin{pmatrix} x_{\mathcal{S},r} \\ y_{\mathcal{S},r} \end{pmatrix} = \begin{pmatrix} \left[ \frac{1}{\frac{1}{a^{\frac{2}{e}} + \left(\frac{y_p}{x_p b}\right)^{\frac{2}{e}}} \right]^{\frac{e}{2}} \\ x_{\mathcal{S},r} \left(\frac{y_p}{x_p}\right) \end{pmatrix}$$

Der Winkel von  $p_{\mathcal{S},r}$  zur x-Achse ist durch

$$\theta := \arccos \left( \frac{x_{\mathcal{S},r}}{\|p_{\mathcal{S},r}\|} \right) \quad (8)$$

gegeben.

Wenn  $\mathcal{S}_{2D}$  wie in Abbildung 33a als Kreis parametrisiert ist, ist der euklidische Abstand gleich  $d_r$ . Falls die Krümmung  $e$  kleiner wird verschiebt sich  $p_{\mathcal{S},r}$  und ist nicht mehr identisch mit dem nächsten Punkt auf  $\mathcal{S}_{2D}$  bezüglich  $(x_p, y_p)$ . Falls  $\theta > \frac{\pi}{4}$  verschiebt sich  $p_{\mathcal{S},r}$  in Richtung der y-Achse, andernfalls erfolgt eine Verschiebung in Richtung der x-Achse. Diese Verschiebung erfolgt analog abhängig



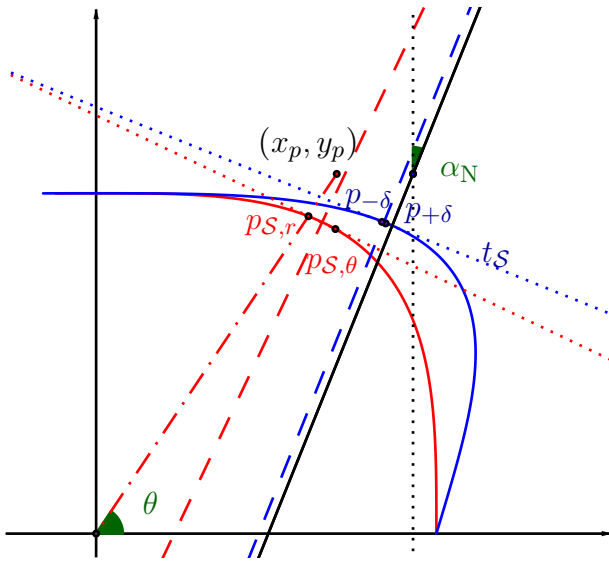


Abbildung 32: Illustration der Winkelberechnung  $\alpha_N$  für ein Drehneubengelenk basierend auf der Superellipse  $\mathcal{S}$  ohne tapering in rot und der originalen Superellipse in blau mit Winkel  $\theta$  zugehörig zu dem nächsten Punkt  $p_{S,r}$  basierend auf der radialen euklidischen Distanz und dem Punkt  $p_{S,\theta}$  definiert durch den Winkel  $\theta$

von den Parametern  $a, b$  von  $\mathcal{S}_{2D}$ . In Abbildung 33 ist diese Verschiebung visuell verdeutlicht. Es sind Liniensegmente vom Ursprung mit äquidistanten Winkeln  $\theta$  bezüglich eines Kreises jeweils mit variierter Parametrisierung eingezeichnet. Es ist zu beobachten, dass diese Liniensegmente in Richtung der stärksten Krümmung verschoben werden und sich dort bündeln. Daraus resultiert, dass der nächste Punkt basierend auf der radialen euklidischen Distanz weiter entfernt ist als der nächste Punkt basierend auf der tatsächlichen Euklidischen Distanz bezüglich des Punktes auf der Oberfläche von  $\mathcal{S}_{2D}$  mit der stärksten Krümmung.

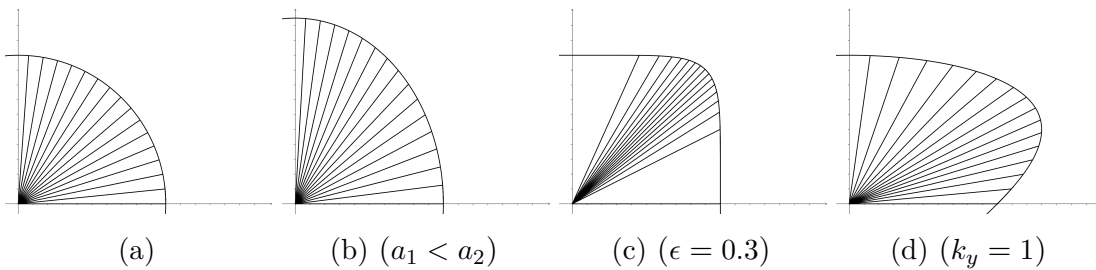


Abbildung 33: Illustration der Auswirkungen der einzelnen Parameter in  $\Lambda$  auf die Bogenlänge ausgehend von einem Kreis (a) mit  $(a_1 = a_2, k_y = 0, \epsilon = 1)$ ,  $\theta$  in äquidistanten Schritten von  $0, 1$

Diese Eigenschaft wird im Folgenden ausgenutzt um einen weiteren Punkt  $p_{S,\theta}$  auf  $\mathcal{S}_{2D}$  zu berechnen, so dass die Gerade durch  $p_{S,r}$  und  $p_{S,\theta}$  die gesuchte Tangente

bezüglich  $(x_p, y_p)$  annähert. Der Punkt  $p_{\mathcal{S},\theta}$  soll die Eigenschaft haben, dass er näher an Punkt auf  $\mathcal{S}_{2D}$  mit der stärksten lokalen Krümmung liegt als  $p_{\mathcal{S},r}$ . Dies wird erreicht indem man  $\theta$  aus (8) in die Parameterform von  $\mathcal{S}_{2D}$  einsetzt:

$$p_{\mathcal{S},\theta} := \begin{pmatrix} a \cos^e(\theta) \\ b \sin^e(\theta) \end{pmatrix}.$$

Die angenäherte Tangentenrichtung bezüglich einer Laméschen Kurve ohne globale Deformation ist  $\vec{d}_{r,\theta} = p_{\mathcal{S},\theta} - p_{\mathcal{S},r}$ .

Nachdem bei der Approximation des Teilobjektes durch einen Superellipsoid auch Tappinging eingesetzt wird, muss diese globale Deformation  $f(y) := \frac{k}{a}y + 1$  mit berücksichtigt werden. Wobei  $k, a$  wiederum passend aus den Parametern von  $\Lambda_{AR}$  gewählt werden müssen. Dazu wird die inverse Deformation auf den Punkt  $(x_p, y_p)$  angewendet und die obigen Berechnungen durchgeführt, deren Resultate  $p_{\mathcal{S},\theta}, p_{\mathcal{S},r}$  und  $\vec{d}_{r,\theta}$  sind. Nachdem  $p_{\mathcal{S},\theta}$  und  $p_{\mathcal{S},r}$  bei kleinem  $e$  oder großer Differenz zwischen  $a$  und  $b$  aus der Parametrisierung von  $\mathcal{S}_{2D}$  weit auseinander liegen, werden zwei Punkte  $p_{+\delta}, p_{-\delta}$  mit kleinem Abstand  $\delta > 0$  um den Mittelpunkt zwischen  $p_{\mathcal{S},\theta}$  und  $p_{\mathcal{S},r}$  bezüglich der Richtung  $\vec{d}_{r,\theta}$  berechnet. Auf diese Punkte wendet man die Deformation  $f(y)$  an und berechnet die angenäherte Tangente  $\vec{t}_{\mathcal{S}}$ :

$$\begin{aligned} p_{-\delta} &:= \begin{pmatrix} x_{-\delta} \\ y_{-\delta} \end{pmatrix} = p_{\mathcal{S},r} + \left(\frac{1}{2} - \delta\right) \vec{d}_{r,\theta} \\ p_{+\delta} &:= \begin{pmatrix} x_{+\delta} \\ y_{+\delta} \end{pmatrix} = p_{\mathcal{S},r} + \left(\frac{1}{2} + \delta\right) \vec{d}_{r,\theta} \\ \vec{t}_{\mathcal{S}} &:= \begin{bmatrix} f(y_{-\delta}) & 0 \\ 0 & 1 \end{bmatrix} p_{-\delta} - \begin{bmatrix} f(y_{+\delta}) & 0 \\ 0 & 1 \end{bmatrix} p_{+\delta} \end{aligned}$$

Aus Tabelle 3 ergeben sich die Zuweisungen der Parameter aus  $\Lambda_{AR}$  auf die zwei-dimensionale Superellipse. Für den Punkt  $(x_p, y_p)$  aus den obigen Berechnungen wird der auf die Ebene projizierte Ursprung des Nebengelenkes eingesetzt. Diese Ebene hängt von der Ausrichtung von  $\vec{z}_{GBKS}$  an den Achsen des Superellipsoids ab.

In Tabelle 4 sind die Parametrisierungen von  $\mathcal{S}_{2D}$  basierend auf der Ausrichtung des GBKS bezüglich dem Koordinatensystems des angepassten Superellipsoids aufgeführt.

	y-z-Ebene	x-z-Ebene	x-y-Ebene
$a =$	$a_2$	$a_1$	$a_1$
$b =$	$a_3$	$a_3$	$a_2$
$e =$	$\epsilon_1$	$\epsilon_1$	$\epsilon_2$
$f(y) =$	$s_y(y)$	$s_x(y)$	$y$
$y_p =$	$z$	$z$	$y$
$x_p =$	$\frac{y}{f(y_p)}$	$\frac{x}{f(y_p)}$	$\frac{x}{f(y_p)}$

Tabelle 4: Parametrisierung von  $\mathcal{S}_{2D}$  basierend auf der Parametrisierung  $\Lambda_{AR}$  des Superellipsoids und der Ausrichtung des Greiferbasiskoordinatensystems

Der gesuchte Winkel für das Nebengelenk kann jetzt aus der Tangente  $\vec{t}_S$  und der rotatorischen Ausrichtung bezüglich  $\vec{z}_{GBKS}$  des GBKS an den Achsen des lokalen Koordinatensystems des Superellipsoids ausgerechnet werden. Sei die Achse  $\vec{x}_{GBKS}$  des Greifers an der Achse  $\vec{e}$  von  $S_{2D}$  ausgerichtet. Dann ist der gesuchte Winkel für rotatorische Nebenachsen:

$$\cos(\alpha_N) = \frac{\langle \vec{e}, \begin{pmatrix} y_{\vec{t}_S} \\ x_{\vec{t}_S} \end{pmatrix} \rangle}{\|\vec{t}_S\|}$$

In Abbildung 34 sind Beispiele für die resultierende Gelenkstellung  $\alpha_N$  einer rotatorischen Nebenachse für den Schunk SDH Greifer dargestellt.

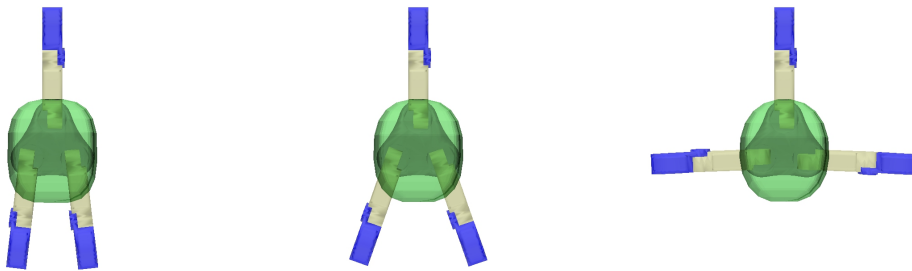


Abbildung 34: Verschiedene Nebengelenkstellungen basierend auf  $\Lambda_{AR}$  für einen Greifer mit Drehnebenelenk

**SCHERENACHSEN** Bei Nebengelenken die über eine Scherenachse verfügen ist ein identisches Vorgehen wie bei den Drehgelenken nicht möglich, da sich die Finger nicht so drehen lassen, dass sie tangential an den Superellipsoid anliegen. Man kann allerdings andere Abhängigkeiten bezüglich der Parameter  $\Lambda_{AR}$  finden, die zu einer funktionalen Abhängigkeit zwischen den Parametern des Superellipsoids und dem Gelenkwinkel des Nebengelenks führen. Es gilt weiterhin wie bei den Drehgelenken, dass die Berechnung des Gelenkwinkels des Nebengelenkes auf der zweidimensionalen Superellipse, der den Schnitt zwischen einer Ebene aufgespannt von den Hauptachsen des lokalen Koordinatensystems definiert durch  $\Lambda_{AR}$  und  $\mathcal{S}$  darstellt, erfolgt.

Angenommen das GBKS ist wie folgt ausgerichtet:  $\vec{z}_{GBKS} \parallel \pm \vec{z}_{K(\Lambda_{AR})}$  und  $\vec{x}_{GBKS} \parallel \pm \vec{y}_{K(\Lambda_{AR})}$ . Damit der Finger den Superellipsoid gut umschließen kann, muss der Gelenkwinkel  $\alpha_N$  abhängig von dem Krümmungsparameter  $\epsilon_2$  angegeben werden. Falls  $\epsilon_2 \rightarrow 0$  ist, muss ebenso  $\alpha_N \rightarrow 0$  gelten. Das bedeutet, falls keine Krümmung vorhanden ist, soll der Finger so ausgerichtet sein, dass die Achsen der Hauptgelenke der Finger parallel zu rechteckigen Oberfläche des Superellipsoids sind. Für den Tapering Parameter  $k_y$  gilt folgende Abhängigkeit: Falls  $k_y$  klein ist, soll  $\alpha_N$  groß werden. Nachdem der Finger sich an das Teilobjekt anlegen soll und nicht darüberhinaus greifen soll, ist  $\alpha_N$  noch von der Länge des Fingers  $l_{Finger}$  und der Ausdehnung des Superellipsoids in x-Richtung  $a_1$  abhängig.

Wenn der Abstand des Ursprungs des Nebengelenks zu dem Ursprung des lokalen Koordinatensystems des Superellipsoids  $d_{Finger}$  kleiner als  $a_1$  ist, soll der Finger von dem Zentrum der Greiferbasis weg gespreizt werden, anderenfalls soll sich der Finger in Richtung des Greiferzentrums bewegen.

Diese Bedingungen führen zu folgender Heuristik für  $\alpha_N$ :

$$\Delta = a_1 - d_{Finger}$$

$$\alpha_N := \begin{cases} + \arctan \left( (1 - k_y) \epsilon_2 \frac{d_{f,N}}{l_{f,M}} \right) & , \Delta < 0 \\ - \arctan \left( (1 - k_y) \epsilon_2 \frac{\Delta}{l_{f,M}} \right) & , \Delta > 0 \end{cases}$$

mit  $a_1, \epsilon_1, k_y \in \Lambda$

Für alle anderen Annäherungsrichtungen und Orientierungen des GBKS sind nur die zu verwendenden Parameter aus  $\Lambda_{AR}$  auszutauschen. Ansonsten erfolgt

die Berechnung analog. Nachdem die Berechnung nur im ersten Quadranten des Koordinatensystems erfolgt, müssen bei bestimmten Orientierungen des GBKS die Parameter aus  $\Lambda_{AR}$  angepasst werden. Die Tapering Parameter sind die einzigen, die dafür sorgen, dass eine Asymmetrie bezüglich einer Koordinatenachse entsteht. Falls der Ursprung des Nebengelenkkoordinatensystems im dritten oder vierten Quadranten des zweidimensionalen Koordinatensystems der Superellipse liegt, muss das Vorzeichen des relevanten Tapering Parameters geändert werden. In Abbildung 35 sind Beispiele zusehen, die die Konfiguration des Scherenneben- gelenks des Robotiq Model S Greifers anhand der Parametrisierung des Superel- lipsoids zeigen.

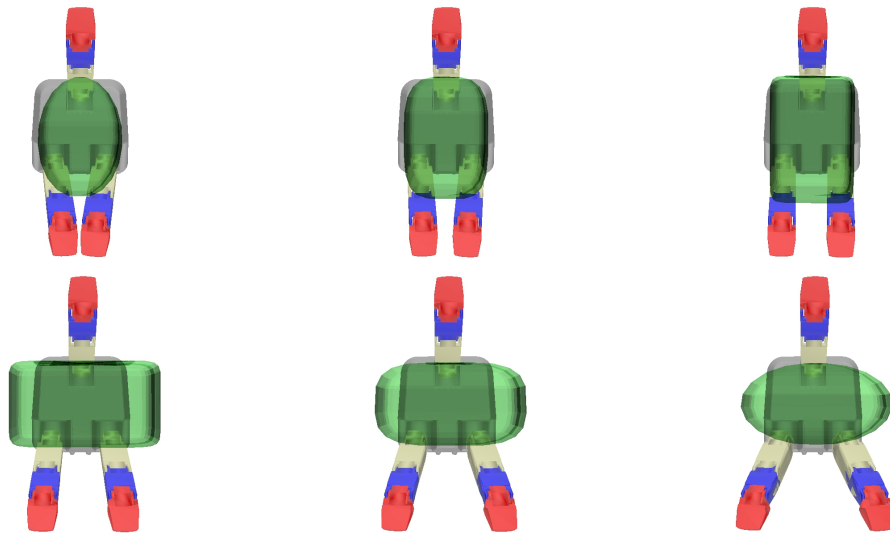


Abbildung 35: Verschiedene Nebengelenkstellungen basierend auf  $\Lambda_{AR}$  für einen Greifer mit Scherenneben- gelenk

Zusammenfassend ist festzustellen, dass die Berechnung der Greifervorkonfigura- tion mit Hilfe einer Approximation des Teilobjektes durch einen Superellipsoid er- folgt. Sowohl die Position und Orientierung des Greifers und der Griffotyp als auch die Gelenkwinkel der Nebengelenke werden abhängig von der Parametrisierung  $\Lambda_{AR}$  des Superellipsoids berechnet. Damit wird die verfügbare hardwareseitige Vari- abilität des Greifers mittels Heuristiken ausgenutzt, um die Stabilität eines Griffes im Vergleich zu einer Greifervorkonfiguraiton mit fester Fingervorkonfiguration und festgelegtem Griffotyp zu verbessern und die Vielfalt valider Greiferkonfigura-

tionen zu erhöhen. Darüber hinaus wird durch die Festlegung der Annäherungsrichtung basierend auf den durch Sensordaten verifizierten Bereichen des Objektmodells die Wahrscheinlichkeit erhöht, dass das Objekt seine Pose während des Zugreifens nicht verändert. Nachdem bei der Bestimmung der Annäherungsrichtung zusätzlich Umweltinformationen mit berücksichtigt werden, wird die Anzahl der zu überprüfenden Greiferkonfigurationen zusätzlich frühzeitig verringert.

#### 4.4.2 Greiferkonfiguration und Bewertung

Im letzten Unterkapitel ist die Berechnung der Greifervorkonfiguration vorgestellt worden. Zu der vollständigen Definition der Greiferkonfiguration fehlen jedoch die Gelenkwinkel der Hauptgelenke. Diese werden benötigt, damit ein Griff noch mit Hilfe von Qualitätsmetriken bewertet werden kann. Zusätzlich soll eine Greiferkonfiguration mit einem Unsicherheitsmaß im Bezug auf die Kontakte mit Bereichen am Objektmodell, die nicht sensorisch validiert sind, belegt werden.

**STRATEGIE ZUM SCHLIESSEN DER FINGER** Eine Strategie zur Berechnung der Gelenkwinkel der Hauptachsen des Greifers ist notwendig, da gerade bei Greifern mit passiven Hauptgelenken die Stellung der Gelenkwinkel von dem zu greifenden Objekt abhängig ist. Wie in Kapitel 4.1 festgestellt, haben Fingerkinematiken typischerweise folgenden Aufbau: Das Fingergrundgelenk ist ein aktives Gelenk und kann direkt gesteuert werden. Alle folgenden Gelenke sind passiv und umschließen sequentiell ausgehend von dem Grundgelenk das Objekt. Damit ist die Fingerkonfiguration direkt von der Geometrie des Objektes innerhalb des Arbeitsraumes des Fingers abhängig. Durch diesen Umstand muss eine Strategie angegeben werden, wie die Fingergelenke geschlossen werden.

Für Greiferkinematiken mit passiven Fingergelenken bietet sich eine Strategie zum Schließen der Finger an, die auf der Vorwärtskinematik beruht. Neben dem Vorteil, dass keine inverse Kinematik berechnet werden muss, entspricht diese Vorgehensweise auch weitestgehend der einzig möglichen realen Bewegung der Finger. Für Finger mit mehreren aktiven Gelenken kann die gleiche Bewegung ausgeführt werden, obgleich diese Finger auch andere komplexere Bewegungen ausführen könnten, um Kontakt mit dem Objekt herzustellen. Die Strategie entspricht einer Fingerkinematik bei der nur das Fingerbasisgelenk aktiv ist und alle anderen Gelenke

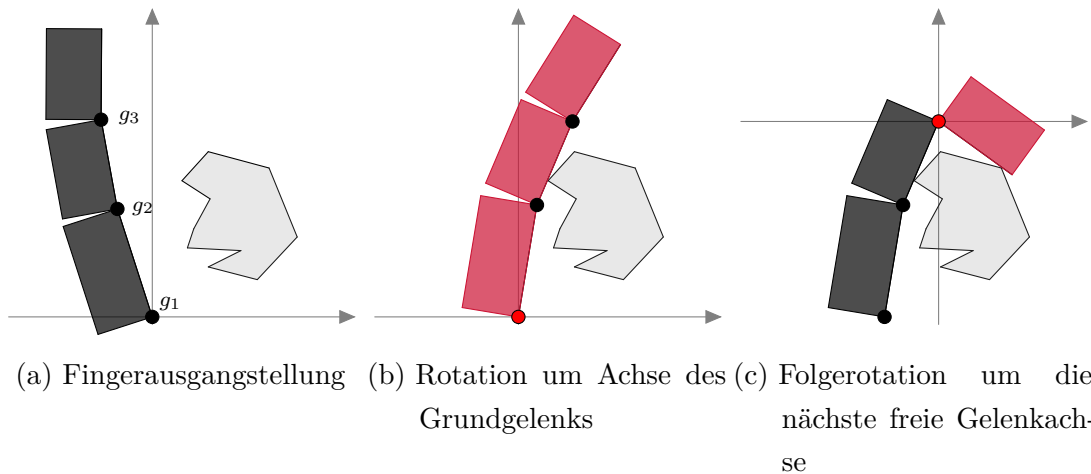


Abbildung 36: Reihenfolge zum Schließen der Fingergelenke

passiv sind. Die in Kapitel 4.1 vorgestellten Mehrfingergreifer sind bis auf eine Ausnahme so konstruiert.

Von nun an wird angenommen, dass ein Finger  $\mathcal{F} := (g_1, \dots, g_n)$  (Definition 4.3) mit den Gelenken  $g_i$  gegeben ist. Die Gelenke seien wie folgt nummeriert:  $g_1$  sei das Fingerbasisgelenk und die weiteren Gelenknummierungen entsprechen der Reihenfolge ausgehend vom Basisgelenk bis hin zur Fingerspitze (vgl. Abbildung 36a). Die dazugehörigen Segmentgeometrien  $\mathcal{O}_i$  seien entsprechend nummeriert.

Für jeden Finger werden die Gelenkwinkel sequentiell beginnend bei  $g_1$  berechnet (vgl. Abbildung 36b). Für die Berechnung des Gelenkwinkels werden alle  $\mathcal{O}_i, \dots, \mathcal{O}_n$  um die Achse von  $g_i$  rotiert bis sie Kontakt mit dem Objekt  $\mathcal{O}$  haben. Der kleinste berechnete Winkel ist der gesuchte Gelenkwinkel von  $g_i$ . Das nächste Gelenk für das ein Gelenkwinkel berechnet werden muss, hat den Index  $j = 1 + \arg \max_k (\hat{\mathcal{O}}_k \cap \mathcal{O} \neq \emptyset)$  mit den um  $g_i$  gedrehten Segmentgeometrien  $\hat{\mathcal{O}}$ . Das ist gerade das Gelenk dessen Segmentgeometrie und alle folgenden Segmentgeometrien keinen Kontakt mit  $\mathcal{O}$  haben (vgl. Abbildung 36c). Mit dem Abschluss der Winkelberechnungen für die Hauptgelenke aller Finger ist der Griff vollständig bestimmt.

Für die Berechnung eines Rotationswinkels wird abermals die rotatorische Kontaktposenberechnung aus Kapitel 3 verwendet. Allerdings sei an dieser Stelle erwähnt, dass in diesem Fall auch andere kontinuierliche Kollisionserkennungsverfahren eingesetzt werden können, da die zu drehende Segmentgeometrie noch keinen

Kontakt mit dem Objekt hat. Nachdem allerdings die Segmentgeometrie typischerweise nur aus einigen wenigen Polygonen besteht die in Drehrichtung zeigen, bieten reine Kollisionserkennungsverfahren keinen signifikanten Vorteil bezüglich der Rechenzeit.

**BEWERTUNG DER GRIFFE** In der Veröffentlichung von J. Bohg et al. [Boh+14] wird das bekannte Problem aufgegriffen, dass die über die Jahre entwickelten analytischen Qualitätsmetriken zur Bewertung von Griffen nicht besonders gut mit der Realität übereinstimmen. Gut oder als statisch im Gleichgewicht bewertete Griffe scheitern dennoch bei der Ausführung. Man kann nicht von Stabilität sprechen, da diese keine statische Eigenschaft ist, sondern unter der Berücksichtigung der Objektdynamik betrachtet werden müsste. In [Leo+14] werden häufig verwendete Maße klassifiziert und analysiert welche Metriken miteinander korrelieren. In [Suá+06] ist eine weitere Übersicht zu Qualitätsmetriken und eine kurze Einführung in die theoretischen Grundlagen zu finden.

Nach [Leo+14] werden Maße nach algebraischen Eigenschaften der *grasp matrix* [Kaj+08], der Verteilung der Kontaktpunkte zwischen Greifer und Objekt und der Konfiguration des Greifers selbst eingeteilt. Mit Hilfe einer Korrelationsanalyse sind unabhängige Maße identifiziert und eine Aussage formuliert worden wie die einzelnen Maße zu interpretieren sind.

Zur Bewertung der Greifkonfigurationen werden verschiedene Maße vorgestellt, die den Experimenten an realen rekonstruierten Objekten bewertet werden. Basierend auf der Analyse von [Leo+14] ist das Maß basierend auf den Gelenkwinkeln der Hauptfingergelenke  $\alpha_i$  mit deren minimalen  $\alpha_{m,i}$  und maximalen  $\alpha_{M,i}$  Gelenkwinkeln [Dia+08]

$$Q_{\text{Gelenk}}(\mathcal{G}) = \frac{1}{n_q} \sum_{i=1}^{n_q} \left( \frac{\alpha_i - \sigma_i}{d_i} \right)^2$$

$$\sigma_i = \frac{\alpha_{M,i} - \alpha_{m,i}}{2}$$

$$d_i = \begin{cases} \alpha_i - \alpha_{m,i}, & \text{falls } \alpha_i < \sigma_i \\ \alpha_{M,i} - \alpha_i, & \text{sonst} \end{cases}$$

$$\alpha_i \in [\alpha_{m,i}, \alpha_{M,i}]$$



und die Distanz zwischen dem Schwerpunkt der Kontaktpunkte  $c_{\mathcal{K}}$  und dem des Objekts  $c_{\mathcal{O}}$  aus [Dan+01]

$$Q_{\text{Distanz}}(\mathcal{O}, \mathcal{K}) = d(c_{\mathcal{O}}, c_{\mathcal{K}})$$

ausgewählt worden. Zusätzlich zu den beiden Metriken wird als weiteres Maß die Summe über alle Kontaktflächen

$$Q_{\text{Kontakt}}(\mathcal{O}, \mathcal{G}) = \frac{1}{\sum_{i=1}^{n_q} |\mathcal{O} \cap P_{\mathcal{G},i}|}$$

und die Qualitätsbewertung der Superellipsoid-Anpassung aus (6)

$$Q_{\text{Quadratic}}(\mathcal{O}, \Lambda) = \sum_{i=0}^n (F(\Lambda; x_i, y_i, z_i)^{\epsilon_1} - 1)^2$$

verwendet.

In allen Maßen in denen Informationen über die Kontaktpunkte benötigt werden, werden nur solche Punkte verwendet, die auf sensorvalidierten Bereichen des Objektmodells liegen (Kapitel 2.3.4). Nur bei diesen Punkten kann davon ausgegangen werden, dass diese tatsächlich nahe an der Oberfläche des realen Objektes liegen. Darüber hinaus bewerten alle vier Maße gut geeignete Griffe mit niedrigen Werten.

In den Experimenten wird die Zusammensetzung der Kombinationen aus den einzelnen Maßen bestimmt.

#### 4.5 EXPERIMENTE UND DISKUSSION

In diesem Unterkapitel wird der vorgestellte Greifplaner experimentell untersucht. Dazu werden zwei Greiferkinematiken verwendet, die beide mit drei Fingern ausgestattet sind. Der Robotiq Adaptive Gripper [Rob14] besitzt Scheren-Nebengelenke und je drei Hauptgelenke pro Finger. Der Schunk SDH 2 Greifer [SCH15b] besitzt Drehnebengelenke und je zwei Finger-Hauptgelenke. Die verwendeten Objekte sind mit synthetischen oder realen Tiefenbildern durch das Verfahren aus Kapitel 2 rekonstruiert. Die synthetischen Tiefenbilder werden von Modellen aus der *KIT ObjectModels Web Database* [Kas+12] erzeugt. Tiefenbilder der Objekte, die nicht in der Datenbank vorhanden sind, werden mit dem Hardwareaufbau aus Kapitel 6 erzeugt. Zur Rekonstruktion werden immer zwei Kameraperspektiven verwendet.

Auf Basis der Experimente aus Kapitel 2 und Kapitel 3 sind die Parameter der Rekonstruktion folgende: Der Octree hat eine Maximaltiefe von sieben. Für die Anwendung der Oberflächenrekonstruktion ist die Anzahl der Gitterzellen pro Achse auf  $2^6$  festgelegt. Beide Teilschritte sind in dem gleichen Ausgangsvolumen ausgeführt worden, was zu einer Gitterzellengröße von  $0,010 \times 0,008 \times 0,005$  m führt. Nachdem die Kontaktposenberechnung aus Kapitel 2 einen wesentlichen Anteil der Berechnung ausmacht, werden benachbarte Gitterzellen zu einer zusammengefasst, so dass die Seitenlänge der Gitterzellen etwa 0,02 m beträgt. Die rekonstruierten Dreiecke werden mit diesem Vorgehen einer eindeutigen neuen Gitterzelle zugewiesen, der Detailgrad der Rekonstruktion bleibt unverändert.

In Abbildung 37 ist eine Auswahl der verwendeten Testobjekte dargestellt. Diese lassen sich grob in folgende Kategorien einteilen:

**PRIMITIV** (Abbildung 37a) Die gesamte Geometrie des Objekts entspricht einem einzigen Superellipsoid.

**PRIMITIV MIT ÖFFNUNG** (Abbildung 37b) Das Objekt kann durch einen einzigen Superellipsoid eingehüllt werden. Auf einer Seite der Approximation existiert keine korrespondierende Oberfläche des Objektes, bei einer Schüssel ist beispielsweise die Oberseite offen und hat somit dort keine korrespondierende Oberfläche mit dem Objekt.

**KOMPLEX** (Abbildung 37c) Die Geometrie des Objekts kann nicht durch einen einzigen Superellipsoid repräsentiert werden, ohne dass der Superellipsoid stark von der tatsächlichen Geometrie des Objekts abweicht.



(a) Primitiv

(b) Primitiv mit Öffnung

(c) Komplex

Abbildung 37: Präsentation einer Auswahl der Testobjekte

Unabhängig von den Kategorien bezüglich der Geometrie der Objekte spielt die Größe relativ zu dem Greifer eine besondere Rolle, insbesondere im Hinblick auf die Konfiguration der Nebengelenke. Deshalb sind in jeder Klasse *kleine Objekte* (beispielsweise die Ente aus Abbildung 37c), die sich dadurch auszeichnen, dass sie kleiner als die approximierten Abmessungen des Arbeitsraums der Greiferfinger sind. Neben den kleinen Objekten sind unter den Testobjekten auch *große Objekte* (beispielsweise die Schüssel aus Abbildung 37b), die sich dadurch auszeichnen, dass sie viel größer als der Arbeitsraum der Greiferfinger sind und deshalb nicht vom Greifer umschlossen werden können.

Kapitel 4.4.2 stellt vier verschiedene Bewertungsmaße für eine Greifkonfiguration vor. Diese Maße werden in Kapitel 4.5.1 bezüglich ihres Einflusses auf die Güte eines Griffes bewertet. Konkret wird untersucht, welche Kombination aus diesen Maßen die Unterscheidbarkeit und Ordnung der berechneten Greiferkonfigurationen liefert. Es soll weiter eine Schranke abgeschätzt werden, ab welcher Qualitätsbewertung ausreichend gute Konfigurationen gefunden sind, damit die Greifplanung vorzeitig beendet werden kann.

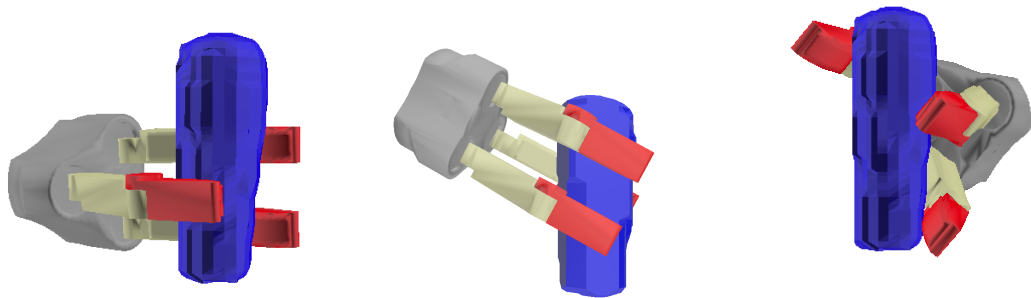
Um die Anforderung der Interaktivität aus Kapitel 1.2 zu überprüfen, werden in Kapitel 4.5.2 sowohl die Rechenzeit der einzelnen Komponenten des Greifplaners als auch die Gesamtrechenzeit untersucht.

Abschließend werden in Kapitel 4.5.3 die Resultate des entwickelten Greifplaners anhand zweier Greiferkinematiken diskutiert.

#### 4.5.1 *Bewertung der Qualitätsmaße*

Die vier ausgewählten Qualitätsmaße sind basierend auf ihren in [Leo+14] verglichenen theoretischen Eigenschaften und motiviert aus den Teilberechnungen dieses Greifplaners ausgewählt worden. Durch die experimentelle Evaluation soll eine Kombination aus diesen gefunden werden, um für den entwickelten Greifplaner ein geeignetes Qualitätsmaß zu finden, welches Griffe gut bewertet, die präzise ausgeführt werden können. Weiter soll eruiert werden, wie die neu eingeführten Maße mit den Bekannten korrelieren.

Das Maß bezüglich der Gelenkwinkel hat den offensichtlichsten Einfluss auf die Güte einer Greiferkonfiguration. Insbesondere werden Konfigurationen schlecht bewertet, bei denen mindestens ein Finger nahezu nicht oder vollständig geschlossen



(a) Griff mit gutem Fit (b) Griff mit schlechtem Fit (c) Griff mit schlechtem Fit  
und guter Greifpose und schlechter Greifpose

Abbildung 38: Vergleich dreier Greiferkonfigurationen basierend auf einem guten und schlechten Superellipsoid Anpassung

ist (Abbildung 38c). Diese beiden Fälle treten auf, wenn dieser Finger am Objekt vorbeigreift, das Objekt so groß ist, dass es nur knapp zwischen die geöffneten Greiferfinger passt oder wenn die Greiferbasis schlecht positioniert ist. Im zweiten Fall führen Ungenauigkeiten bei der Ausführung des Griffs dazu, dass das Objekt touchiert und von seiner Ausgangspose wegbewegt wird. Falls das Objekt dennoch gegriffen wird, weicht die tatsächliche relative Pose im Bezug auf den Greifer von der geplanten ab. Diese Abweichung führt bei nachfolgenden Manipulationen zu nicht berücksichtgbaren Ungenauigkeiten, die nur durch die Verwendung zusätzlicher Sensorik kompensiert werden können.

Durch die Approximation eines Teiles des Objektmodells, das vom Greifer umschlossen werden könnte, mit einem Superellipsoid werden per Konstruktion viele unmögliche Greiferkonfigurationen vorzeitig ausgeschlossen. Nachdem der Superellipsoid verschiedene geometrische Formen annehmen kann, die alle sehr gut von Mehrfingergrifern umschlossen werden können, ist die Güte des angepassten Superellipsoids ein gut geeignetes Kriterium. Abbildungen 38a und 38b zeigen exemplarisch zwei Greiferkonfigurationen basierend auf einer gut und einer schlecht bewerteten Anpassung. Insbesondere bei Präzisionsgriffen ist der Einfluss der Güte der Anpassung auf den Erfolg des Griffes besonders hoch, da die Greiferfinger nicht das Objekt umschließen. Gerade bei dünnen Objekten, wie Flaschen, ist der häufigste Fall, dass ein oder mehrere Finger am Objekt vorbei greifen, insbesondere

weil die Konfiguration der Nebengelenke fehlerhaft im Bezug auf die tatsächliche Geometrie des Objekts ist.

Bezüglich des Maßes für die Kollisionspunkte gilt folgender einfache Zusammenhang: Je größer die Kontaktfläche zwischen zwei Objekten ist, desto größere Kräfte können aufgebracht werden, bevor sich beide Objekte gegeneinander verschieben oder verdrehen. Diese Eigenschaft ist bei Greif- und Ablage-Aufgaben besonders wichtig, da während des Ablegens gerade solche Momente durch den Kontakt zwischen Objekt und Ablage auf das Objekt wirken. Daher muss der verwendete Griff möglichst große externe Kräfte und Momente kompensieren können.

Der Abstand zwischen dem Schwerpunkt des Objektes und dem Mittelpunkt der Kollisionspunkte bewertet ebenso die Momente, die von dem Griff kompensiert werden können. Je kleiner dieser Abstand ist, desto kleiner sind die Momente die durch das Eigengewicht des Objektes auf den Griff wirken. Bei der Berechnung der Greiferkonfigurationen basierend auf einem angepassten Superellipsoid leistet dieses Maß einen vernachlässigbaren Beitrag zur Bewertung einer Greiferkonfiguration. Bei primitiven Objekten ist der Ursprung des Koordinatensystems in dem der Superellipsoid definiert ist nahezu identisch mit dem Schwerpunkt (bei angenommener homogener Massenverteilung des Objekts) und damit ist durch die Positionierung des Greifers basierend auf dem Superellipsoid der Abstand zu dem Mittelpunkt der Kontaktpunkte minimal. Bei großen Objekten ist es oft nicht möglich nahe am Schwerpunkt des Objektes zu greifen. Das dies jedoch für viele Griffe gleichermaßen gilt, leistet dieses Maß für große Objekte einen untergeordneten Beitrag zur Bewertung eines Griffes. Folglich wird das Maß bezüglich des Schwerpunktes nicht verwendet.

	$Q_{\text{Quadric}}$	$Q_{\text{Kontakt}}$	$Q_{\text{Gelenk}}$	$Q_{\text{Distanz}}$
$Q_{\text{Quadric}}$	1	0,141	-0,036	0,110
$Q_{\text{Kontakt}}$	0,142	1	0,191	0,302
$Q_{\text{Gelenk}}$	-0,035	0,190	1	0,073
$Q_{\text{Distanz}}$	0,107	0,304	0,075	1

Tabelle 5: Korrelation zwischen den Qualitätsmaßen auf Basis von 1257 Greiferkonfigurationen erzeugt mit zwei Greiferkinematiken und 23 verschiedenen Objekten

Tabelle 5 zeigt die Korrelationen zwischen den verschiedenen Maßen. Hervorzuheben ist die Korrelation der beiden Maße  $Q_{\text{Kontakt}}$  und  $Q_{\text{Distanz}}$ . Sie ist am stärksten unter den paarweise verglichenen Maßen ausgeprägt, was die Argumentation der obigen Diskussion unterstützt. Gerade primitive Objekte können gut von Mehrfingergrifern umschlossen werden, damit ist auch die Distanz zwischen dem Schwerpunkt des Objektes und dem der Kontaktpunkte klein. Gleichzeitig ist die Kontaktfläche zwischen Greifer und Objekt hoch. Der Greifplaner positioniert, auf Grund der Art und Weise wie die Greifervorkonfiguration berechnet wird, die Finger, so dass die Greifflächen der Finger möglichst tangential zu der Oberfläche des Objektes sind (Abbildung 38a) und die Greiferbasis am Koordinatensystem des angepassten Superellipsoiden ausgerichtet ist. Der Ursprung dieses Koordinatensystems entspricht bei einer optimalen Anpassung genau dem geometrischen Schwerpunkt primitiver Objekte.

Nachdem die drei verbliebenen Maße unterschiedliche Eigenschaften eines Griffes bewerten und die Maße untereinander nur schwach korrelieren, wird als Gesamtmaß  $Q_{\text{Summe}}$  die Summe über diese verwendet:

$$Q_{\text{Summe}} = Q_{\text{Gelenk}} + Q_{\text{Kontakt}} + Q_{\text{Quadic}}.$$

In Abbildung 39 ist das Histogramm des Gesamtmaßes aufgetragen. Durch manuelle Klassifikation aller Griffe aus den 1257 berechneten Griffen ergibt sich ein Grenzwert von 0,55 für eine Qualitätsbewertung. Falls  $Q_{\text{Summe}} < 0,55$  wird die Greiferkonfiguration als instabil bewertet und darf nicht verwendet werden, um einen Griff des Objektes durchzuführen.

Die Ergebnisse der Greifplanung für die Rechenzeitanalyse und den qualitativen Vergleich aus diesem Kapitel und aus Kapitels 6 werden mit diesem kombinierten Maß bewertet.

#### 4.5.2 *Rechenzeitanalyse*

Um die Rechenzeit des Greifplaners zu analysieren, werden zuerst der Einfluss der Berechnung der Greifervorkonfiguration und der Fingerkonfiguration mit der beinhalteten Qualitätsbewertung auf die Rechenzeit diskutiert. Zusätzlich müssen die Parameter untersucht werden, die potentiell Einfluss auf die Rechenzeit haben.

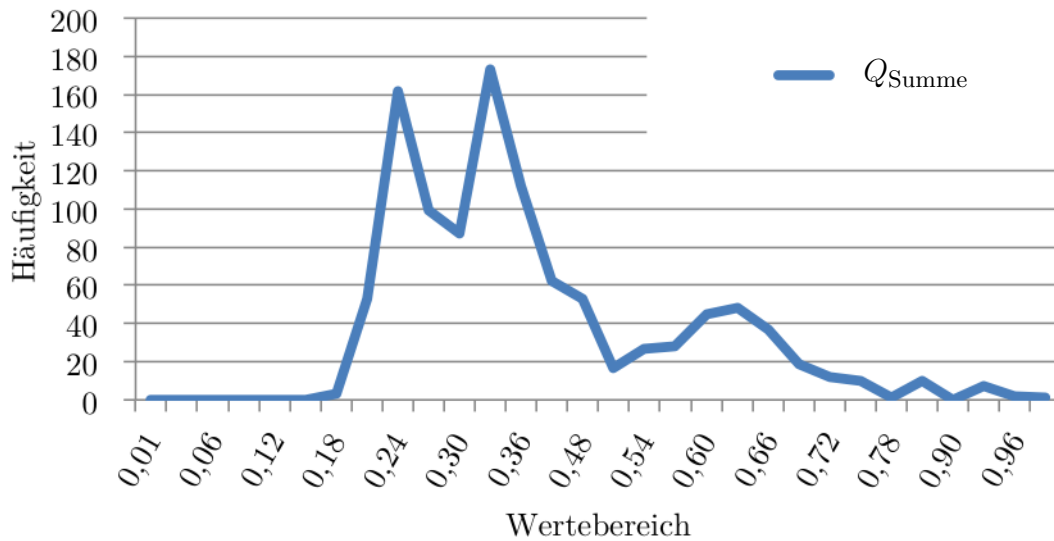


Abbildung 39: Histogramm des kombinierten Qualitätsmaßes über alle Greiferkonfigurationen aus Tabelle 5

Aus den Experimenten aus Kapitel 3 geht hervor, dass die Anzahl der Dreiecke des Objektmodells keine Auswirkung auf die Rechenzeit haben. Vielmehr spielt die Auflösung des regulären Gitters in den das Objektmodell eingebettet ist eine signifikante Rolle für die Rechenzeit. Aus den Experimenten der Kontaktposenberechnung resultiert eine optimale Auflösung bezüglich der Seitenlänge einer einzelnen Gitterzelle. Aus diesem Grund muss dieser Parameter für die Greifplanung nicht weiter untersucht werden.

Die Parameter, die zur Regulierung der Rechenzeit verbleiben, sind die Anzahl der Annäherungsrichtungen zur Auswahl des Teilobjekts an den der Superellipsoid angepasst wird und die Anzahl der Dreiecke des Oberflächenmodells des Greifers. Die Anzahl der zu testeten Greifervorkonfigurationen wird durch die Parameter der Superellipsoiden festgelegt.

Die Anzahl der Annäherungsrichtungen, die benötigt werden, um möglichst die gesamte Oberfläche des Objektmodells zu betrachten, hängt stark von der Größe des Objektes ab, konkret, ob es vom Greifer umschlossen werden kann oder nicht.

Bei primitiven Objekten reicht eine Annäherungsrichtung aus. Die Objekte dieser Klasse sind annähernd konvex und lassen sich insbesondere durch einen ein-

zelen Superellipsoid approximieren. Das heißt, wenn das Objekt zwischen die geöffneten Greiferfinger passt, kann es auch gegriffen werden. Das selektierte Teilobjekt umfasst damit annähern das gesamte Objektmodell. Die Argumentation gilt ebenso für kleine primitive Objekte mit Öffnung.

Für große primitive Objekte mit Öffnung werden mehrere Annäherungsrichtungen benötigt, damit die Oberfläche des Objektmodells durch die Teilobjekte abgedeckt ist. Werden zu viele Annäherungsrichtungen verwendet, wird das Objektmodell durch die angepassten Superellipsoiden überrepräsentiert. Für die verwendeten Testobjekte haben vier Annäherungsrichtungen basierend auf den Boundingboxseiten (siehe Abbildung 29b) ausgereicht, die Oberfläche des Objektes nahezu vollständig zu betrachten.

Eine höhere Anzahl an Annäherungsrichtungen führt dazu, dass ein Superellipsoid mit der nahezu identischen Parametrisierung mehrmals berechnet wird. Damit werden identische Greifervorkonfigurationen mehrmals erzeugt und somit auch identische Greiferkonfigurationen.

Für komplexe Objekte ergeben sich dieselben Schlussfolgerungen bezüglich der Annäherungsrichtungen wie auch für die anderen Objektklassen.

In Abbildung 40 sind die minimale, maximale und durchschnittliche Rechenzeiten<sup>2</sup> für verschiedene Objekte aus den verschiedenen Objektklassen aufgetragen. Dafür sind zwei Annäherungsrichtungen verwendet worden, einmal aus x-Richtung und einmal aus y-Richtung. Die Rechenzeiten beruhen auf insgesamt 200 Planungsdurchläufen je 10 pro Objekt, damit sind über 120000 Greiferkonfigurationen ausgerechnet worden. Für die Rechenzeitexperimente sind keinerlei Nebenbedingungen verwendet worden, die die Berechnung einer Konfiguration vorzeitig abbrechen. Weiter ist in der Abbildung die Anzahl der als stabil bewerteten Greiferkonfigurationen aufgetragen. Nachdem keine Nebenbedingungen verwendet werden, um Greiferkonfigurationen zu verwerfen, ist die Anzahl der resultierenden Konfigurationen von den Parametern der angepassten Superellipsoiden abhängig. Dabei sind dessen Größe und Symetrie die maßgeblichen Faktoren (Tabelle 3).

Die Berechnung der Greifervorkonfiguration benötigt pro Annäherungsrichtung durchschnittlich 30 ms. Im Schnitt verstreichen weitere 70 ms, um die Optimierung der Greiferbasis bei Kraftgriffen vorzunehmen, die Konfiguration der Hauptfinger-

<sup>2</sup> Die Experimente sind auf einem Computer mit einem 22 nm "Ivy Bridge" 2.9 GHz Intel "Core i7" Prozessor (3520M) und 8 GB RAM durchgeführt worden.



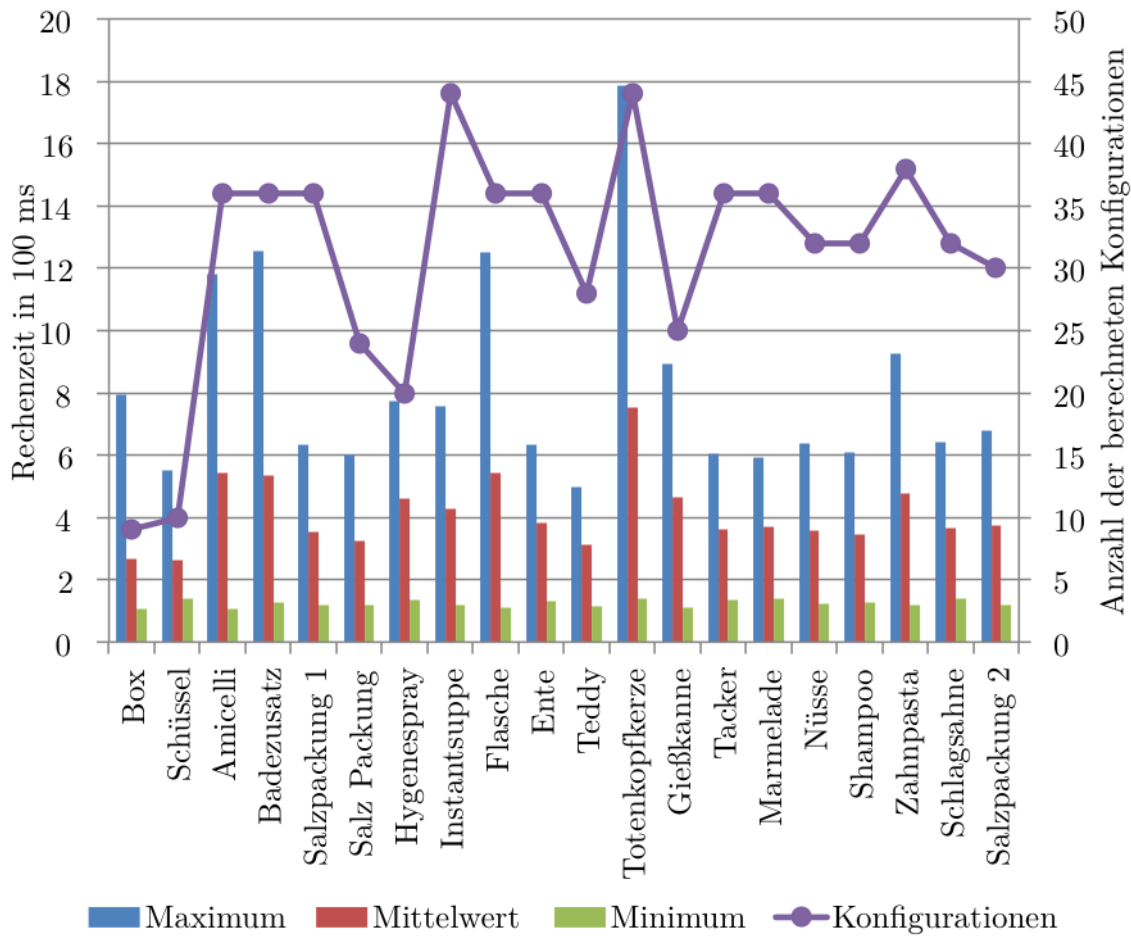


Abbildung 40: Rechenzeit abhängig von unterschiedlichen Objektgeometrien inklusive der Anzahl an stabilen Greiferkonfigurationen eines Planungsvorgangs

gelenke zu berechnen und das Qualitätsmaß zu berechnen. Daraus ergibt sich, dass nach etwa 105 ms die ersten Ergebnisse von der Greifplanung vorliegen. Die mittlere Rechenzeit über alle Testobjekte beträgt 433 ms, wobei in dieser Zeit im Mittel 31 Greiferkonfigurationen berechnet werden. Nachdem pro Superellipsoid mehrere Greifervorkonfigurationen erzeugt werden, ist die Auswertung der mittleren Rechenzeit für eine Konfiguration nicht aussagekräftig (diese würde etwa 13 ms ergeben), da Sockelrechenzeit zum Fitten des Superellipsoids durch die Mittlung auf die einzelnen Konfigurationen aufgeteilt werden würde.

Die Schlussfolgerungen aus der Rechenzeitanalyse sind: Der Detailgrad des Objektmodells beeinflusst nicht die Rechenzeit der Greifplanung. Die Anzahl der Annäherungsrichtungen und der Detailgrad des Greifermodells sind die Variablen, die die Rechenzeit des Planungsvorganges beeinflussen. Mit der Verwendung von Nebenbedingung reduziert sich die durchschnittliche und maximale Rechenzeit weiter, da nicht mehr alle möglichen Konfigurationen vollständig berechnet werden. Diese Rechenzeitverkürzung ist allerdings stark anwendungsabhängig.

#### 4.5.3 Qualitativer Vergleich

In diesem Abschnitt werden die Ergebnisse des Greifplaners für unterschiedliche Objekte aus den drei eingeführten Klassen und zwei Greiferkinematiken vorgestellt.

Durch die Berücksichtigung der Nebengelenke bei der Berechnung der Greiferkonfiguration ist im Gegensatz zu einer im Vorfeld festgelegten Nebengelenkstellung die Anzahl an möglichen Griffen deutlich höher. Die Abbildung 41 zeigt zwei Konfigurationen pro Objekt, die sich gegenseitig im Bezug auf die Nebengelenkstellung ausschließen. Im Fall einer festen Nebengelenkkonfiguration gemäß einer der beiden Abbildungen wäre die andere Konfiguration nicht möglich. Bei der Kinematik mit den Scherennebenengelenken trifft diese Aussage ebenso zu. In den Abbildungen 42 und 43 sind weitere Ergebnisse der Greifplanung für beide Kinematiken abgebildet.

In Abbildung 40 ist neben der Rechenzeit auch die Anzahl der stabilen Greiferkonfigurationen abzulesen. Daraus lässt sich schließen, dass der Greifplaner für *primitive Objekte* für beide Kinematiken mindestens acht augenscheinlich stabile Griffe berechnet. Bei den Griffen für die komplexen Objekte ist zu beobachten, dass die angepassten Superellipsoide teilweise schlecht die Geometrie des Objektes approximieren. Daraus ergeben sich dennoch stabile Griffe. Bei der Ausführung ist allerdings zu erwarten, dass das Objekt während des Zugreifens leicht verdreht wird und damit die tatsächliche Pose des Objektes nicht mit der aus der Planung übereinstimmt. Die Relevanz dieses Fehlers hängt von der nachfolgenden Manipulationsaufgabe mit diesem gegriffenen Objekt ab. Falls die Präzision des Griffs nur eine untergeordnete Rolle spielt, da das Objekt beispielsweise lediglich in eine Kiste geworfen und nicht gelegt werden soll, ist auch ein solcher Griff als Erfolg

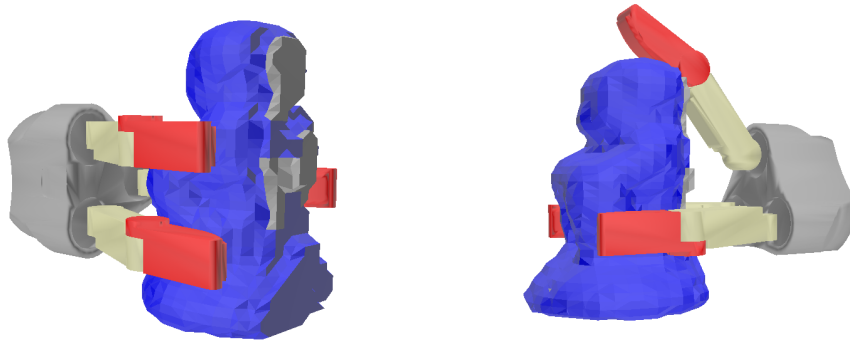


Abbildung 41: Greiferkonfigurationen, die sich bei fester Nebengelenkstellung ausschließen

zu werten. Wenn das Objekt nachfolgend präzise abgelegt werden soll, ist dieser Fehler von großer Bedeutung und der Griff kann nicht als Erfolg gewertet werden.

Objekte aus der Klasse *Primitiv mit Öffnung* stellen die größte Herausforderung für den Greifplaner dar. Die Rührschüssel ist eines der wohl komplexesten Objekte für diesen Greifplaner. Sie ist so groß, dass der Greifer sie nicht umschließen kann. Der Rand der Schüssel ist zusätzlich so gekrümmt, dass der Superellipsoid mit der verwendeten Parametrisierung nicht optimal angepasst werden kann. Dennoch impliziert das Resultat (Abbildung 43a), dass die Schüssel erfolgreich gegriffen werden kann und zusätzlich noch so exakt, dass sie während dem Schließen der Greiferfinger nicht verrutscht.

Festzuhalten ist, dass die Berücksichtigung aller Freiheitsgrade der Greiferfinger die Anzahl der möglichen Greiferkonfigurationen erhöht. Dadurch ist eine bessere Auswahl an gut bewerteten Griffen für anschließende Manipulationsaufgaben vorhanden. Obwohl die Positionierung des Greifers von den angepassten Superellipsoiden abhängt, erzeugt der Greifplaner Konfigurationen und Griffen für Objekte, deren Geometrie nicht gut von einem Superellipsoid approximiert werden kann. Diese Fähigkeit ist insbesondere bei Greiferkinematiken gegeben, die eine hohe Anzahl an Hauptgelenken pro Finger haben. Bei Kraftgriffen sorgt das zusätzliche Kippen der Greiferbasis bei schlecht verteilten Kontaktpunkten dafür, dass schlechte Superellipsoid-Parametrisierungen kompensiert werden können.

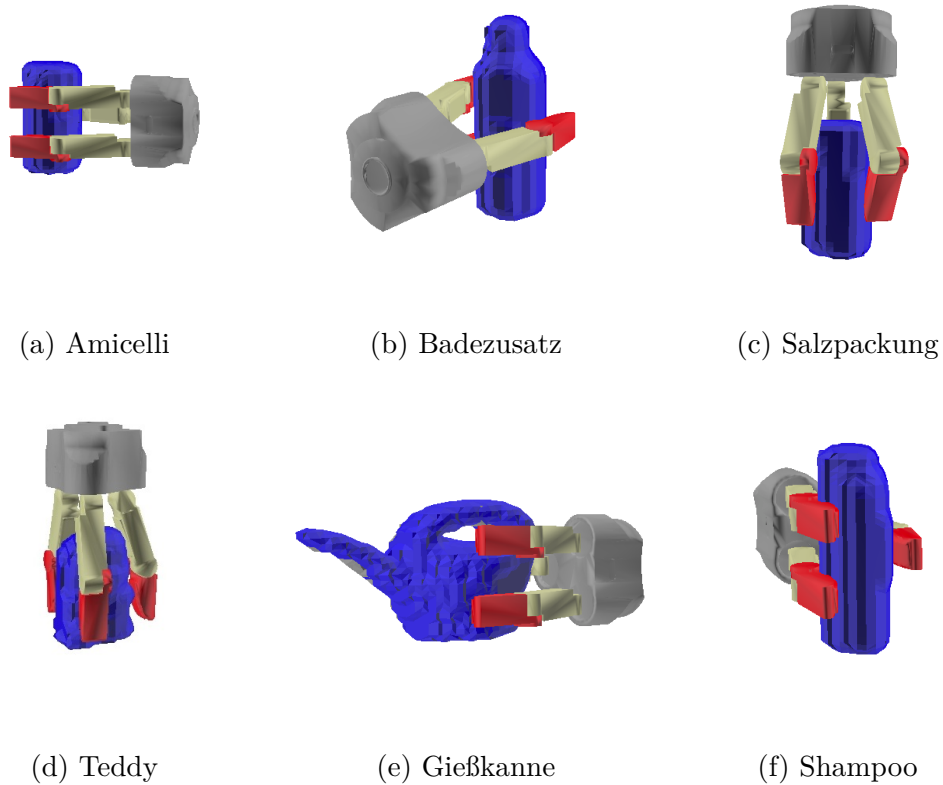


Abbildung 42: Ergebnisse einer Auswahl an Testobjekten erzielt mit dem Schunk SDH 2 Greifer

Bezüglich der Anzahl der Kameraperspektiven ist zu beobachten, dass eine Perspektive im Allgemeinen nicht ausreicht. Wie schon in Kapitel 2 festgestellt ist die Kameraperspektive anwendungsabhängig. Wenn die relative Pose zwischen Kamera und Objekt es nicht erlaubt genügend Information über die Geometrie des Objektes zu erfassen, ist es dem Levenberg-Marquardt Verfahren nicht möglich verwertbare Lösungen zu erzeugen. Dies ist beispielsweise der Fall, wenn ein Objekt wie eine Pfanne gegriffen werden soll und die Kameraperspektive so gewählt ist, dass der Griff nicht aufgenommen wird.

Die Annahme aus Kapitel 2.3.1, dass die Objekte auf einem Tisch vereinzelt stehen, ermöglicht es mit zwei gegenüberliegenden Kameraperspektiven ein ausreichend gutes Objektmodell zu rekonstruieren, dass der Greifplaner bei allen Testob-

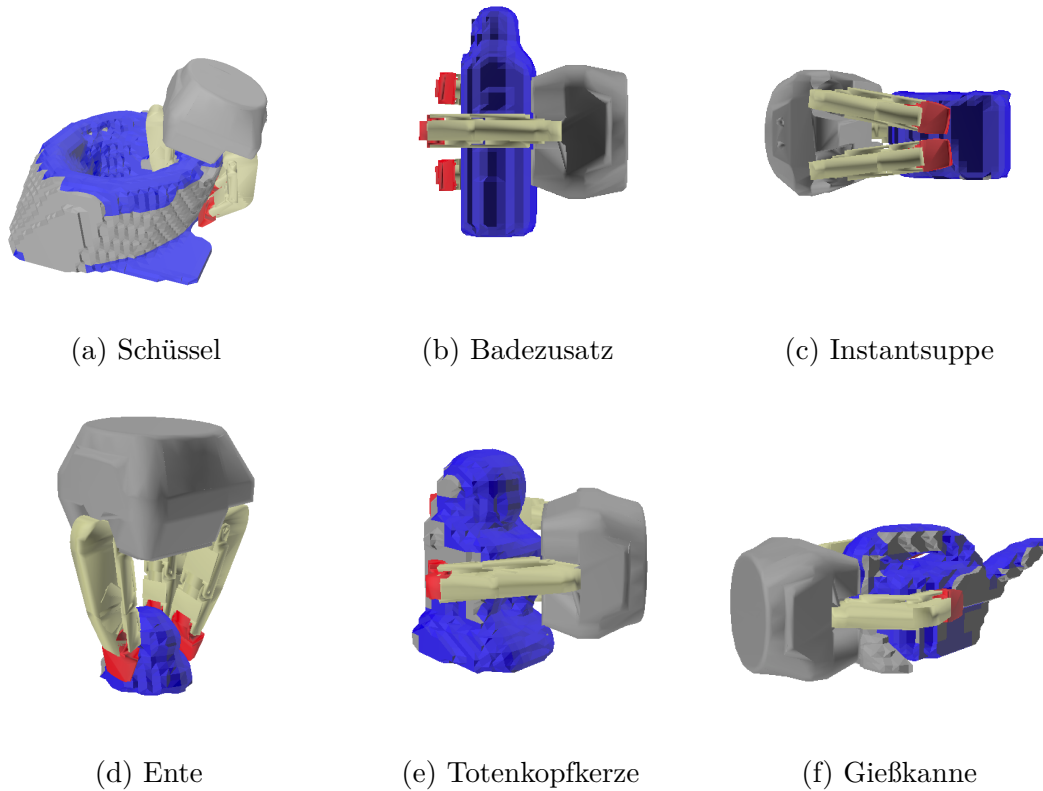


Abbildung 43: Ergebnisse einer Auswahl an Testobjekten für den Robotiq Adaptive Gripper

jekten valide Griffe berechnen kann. Anzumerken ist, dass die Positionierung der Kameraperspektiven im Bezug auf das Objekt ausschlaggebend für die Güte der Rekonstruktion und damit auch auf die nachfolgenden Algorithmen ist, die auf der Rekonstruktion arbeiten. In Kapitel 2.5 wird der Ausblick auf eine Erweiterung vorgestellt mit der die nächste Kameraperspektive auf Basis der schon verfügbaren geometrischen Informationen berechnet werden kann.

#### 4.6 SCHLUSSFOLGERUNGEN

In diesem Kapitel ist der entwickelte Greifplaner auf Basis einer lokalen Approximation von Teilbereichen des Objektes durch einen Superellipsoid vorgestellt

worden. Die verschiedenen Parametrisierungen dieser geschlossenen Form repräsentieren die möglichen Geometrien der Arbeitsräume von Mehrfingergreifern mit unterschiedlichen Fingervorkonfigurationen. Diese Abhängigkeit wird genutzt, um aus der Parametrisierung eines angepassten Superellipsoids die Greifervorkonfiguration zu berechnen. Durch die variable Greifervorkonfiguration erhöht sich die Anzahl an validen Greiferkonfigurationen aus denen unter Berücksichtigung der Anwendung die beste gewählt werden kann. Durch die Berechnung der Greifervorkonfiguration in Abhängigkeit von der Geometrie des Objektes werden Objekte präziser gegriffen und die Anzahl der validen Griffe erhöht sich deutlich. Die Unsicherheit aus der Rekonstruktion unbekannter Objekte wird während der Greifplanung und bei der Bewertung berechneter Griffe mit berücksichtigt.

Der Planer ist mit zwei verschiedenen Mehrfingergreifern experimentell untersucht worden. Für Objekte, deren Geometrie einer einfachen primitiven Form genügt, werden stets valide Griffe berechnet, die zusätzlich robust ausgeführt werden können. Durch die Verwendung der lokalen Objektgeometrie ist der Greifplaner fähig Griffe an Objekten, die nicht vom Greifer umschlossen werden können, wie beispielsweise eine Rührschüssel, zu planen.

Die Rechenzeit die der Planungsvorgang benötigt liegt im Bereich von einer Sekunde und ist insbesondere unabhängig von dem Detailgrad der Oberfläche des Objektmodells auf dem die Greifplanung durchgeführt wird. Diese Eigenschaft ist besonders vorteilhaft, da bei der Rekonstruktion der unbekannt Objekte deren Oberfläche durch Oberflächenpatches überrepräsentiert wird.

Die verwendete Parametrisierung des Superellipsoids ermöglicht es, auch gekrümmte Objekte wie eine Banane oder den Rand einer Rührschüssel so zu approximieren, dass die Greiferkinematik mit drei Gelenken pro Finger in der Lage den Fehler aus der Anpassung zu kompensieren. Bei Greiferkinematiken mit weniger Freiheitsgraden pro Finger ist diese Kompensation nur bedingt möglich. Um auch solche Objekte besser greifen zu können und nicht auf die Kompensationsfähigkeit der Kinematik angewiesen zu sein kann man zusätzlich eine globale rotatorische Transformation bei dem Anpassen des Superellipsoids verwenden.

Die Auswahl des Teilobjektes basierend auf generischen Annäherungsrichtungen an das Objekt berücksichtigt keine Objekteigenschaften. Der vorgestellte Algorithmus ermöglicht allerdings die Verwendung einer Auswahl an Teilobjekten, um die Objekteigenschaften bei der Greifplanung mitzuberechnen. Wenn ein Verfah-

ren existiert, das beispielsweise Henkel oder Griffe an einem Objektmodell identifiziert kann diese Teilgeometrie direkt in dem Greifplaner verwendet werden ohne die generischen Annäherungsrichtungen zu verwenden.





# 5

## ABLAGEPLANUNG

In Kapitel 1.1 wird festgestellt, dass ein Personal Roboter mit verschiedenen Strategien ausgestattet sein muss, damit er autonom unterschiedliche Aufgabenstellungen lösen kann. Um den Einsatzbereich eines solchen Roboters zu erweitern, muss er fähig sein ein gegriffenes Objekt abzulegen, auch wenn er nur geometrische Informationen zur Verfügung hat. Insbesondere ist die Ablagepose des Objektes nicht gegeben.

Die Herausforderung für einen solchen Algorithmus besteht darin, ausschließlich mit Hilfe der rekonstruierten Geometrie des Objektes und der Ablage eine geeignete Ablagepose im Vorfeld der Ausführung zu berechnen, damit eine Bewertung der berechneten Ablage basierend auf der gestellten Aufgabe möglich ist. Die Bewertung muss sicher stellen, dass das Objekt nach dem Lösen des Griffs durch den Roboter in Ruhe verharrt (Definition 1.4).

Bei der Greifplanung lässt sich die Variabilität des Planungsalgorithmus an den Greiferkinematiken festmachen, für die der Algorithmus ausgelegt ist. Bei der Ablageplanung ist die ausschlaggebende Komponente die Ablage, um deren Leistungsfähigkeit zu beurteilen. Dazu werden in Kapitel 5.1 die verschiedenen Ablagen und Lagebeziehungen zwischen Objekten kategorisiert. Anschließend werden in Kapitel 5.2 die daraus resultierenden Anforderungen formuliert, nach denen in Kapitel 5.3 der Stand der Forschung unterteilt wird. Anschließend führt Kapitel 5.4 den entwickelten Planer ein. Der Ablageplaner und die eingeführten Heuristiken werden in Kapitel 5.5 experimentell untersucht. In Kapitel 5.6 werden die Ergebnisse aus diesem Kapitel zusammengefasst und auf die Leistungsfähigkeit des Planers geschlussfolgert.

## 5.1 TAXONOMIE

Um Ablageplaner zu klassifizieren wird eine Taxonomie für die Lagebeziehung zwischen Objekten und für die Geometrie der Ablage eingeführt, anhand derer die unterschiedlichen Ausprägungen und Disziplinen der Ablageplaner definiert werden können.

Abbildung 44 zeigt die verwendete Taxonomie, um die Lage zwischen Objekten zu klassifizieren. Darin werden vier Lagebeziehungen unterschieden: *ineinander*, *hängend*, *aufeinander* und *nebeneinander*. Dabei kann ein Objekt auf ein oder mehrere unterstützende Objekte (Ablagen) gelegt werden.

Objekte liegen *ineinander*, wenn das abgelegte Objekt durch eine Öffnung in dem Inneren des unterstützenden Objektes platziert wird und somit von dem unterstützenden Objekt umgeben ist. Beispiele aus dem menschlichen Umfeld sind ein Teelöffel, der in eine Tasse gestellt wird, ein Brief der in einen Umschlag gesteckt wird, oder eingekaufte Gegenstände in einer Tragebox. Diese Anordnung wird im industriellen Umfeld als *peg in hole* Problem bezeichnet [Whi82].

Ein Objekt *hängt* an dem unterstützenden Objekt, wenn es sich in mindestens eine Richtung wie ein Pendel bewegen kann. Ein Pendel ist ein Objekt, das an einer Achse oder einem Punkt außerhalb seines Massenmittelpunktes drehbar gelagert ist und um seine eigene Ruheposition schwingen kann. Ein Beispiel ist ein Kochlöffel, der an einem Haken über dem Herd hängt.

Ein Objekt *liegt* auf einem unterstützenden Objekt, wenn keine Bewegungsrichtung für das unterstützende Objekt existiert, so dass das abgelegte Objekt in Ruhe bezüglich eines festen Raumpunktes verharrt. Diese Lagebeziehung lässt sich noch feiner untergliedern. Wenn mehrere Objekte aufeinander liegen, wird diese Lagebeziehung folgend als gestapelt bezeichnet. Beispiele sind ein Stapel Bücher, deren Schwerpunkte alle auf einer senkrechten Achse liegen oder Teller, die aufgetürmt sind. Im Gegensatz dazu spricht man von gepackten oder geschütteten Objekten, wenn ein Objekt auf mehreren unterstützenden Objekten gleichzeitig liegt. Das Packen von Objekten hat in der Logistik besondere Relevanz, daher ist das vorrangige kombinatorische Optimierungsproblem, wie quaderförmige Objekte dicht angeordnet werden können [Lod+02; Ort10]. Es existieren vereinzelt Arbeiten, die sich mit irregulären Objekten wie das Beladen eines Containers mit Möbelstücken [Ege08] beschäftigen.

Die Anordnung von Objekten *nebeneinander* unterscheidet sich von den anderen Lagebeziehungen darin, dass es noch ein unterstützendes Objekt, die Ablage, geben muss. Diese Beziehung stammt aus dem Fachgebiet der Innenraumgestaltung [Koh03]. Eine der wesentlichen Fragen ist, wie Objekte ästhetisch zueinander angeordnet werden können. In Bezug auf die Ablageplanung lassen sich *nebeneinander* angeordnete Objekte weiter unterscheiden. Sie können sich berühren, beispielsweise Tassen auf einem Tablett oder Bücher in einem Regal. Sie können vereinzelt stehen ohne eine erkennbare Struktur, beispielsweise Objekte auf einem unaufgeräumten Tisch. Als letzte Möglichkeit können die Objekte nach einer gewissen Struktur angeordnet werden. Diese kann auf der menschlichen Ästhetik oder der Objektfunktion basieren [Jia+13].

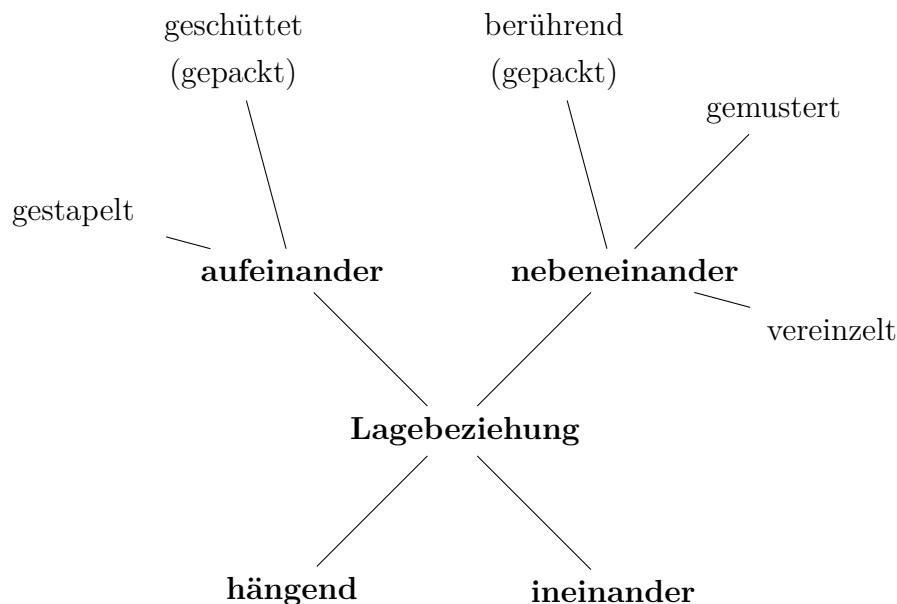


Abbildung 44: Taxonomie für Lagebeziehungen zwischen Objekten

Neben der Lagebeziehung zwischen Objekten ist die Geometrie der unterstützenden Objekte ein weiteres Unterscheidungsmerkmal. Es lassen sich grob ebene und nicht ebene Ablagen unterscheiden.

Ebene Ablagen lassen sich weiter bezüglich ihrer Orientierung in Bezug auf die Richtung der Gravitation klassifizieren. Man unterscheidet zwischen horizontalen und schiefen ebenen Ablagen.

Für nicht ebene Ablagen ist die weitere Klassifikation nicht so eindeutig wie im ebenen Fall. An dieser Stelle sei unterschieden zwischen Ablagen, die die Form einer Kiste haben, oder Ablagen deren Geometrie eine ebene Fläche besitzt, die annähernd horizontal ist und angrenzend an diese Ebene einige seitliche Begrenzungen existieren, als Beispiel sei ein Bücherregal genannt. Alle anderen nicht ebenen Ablagen werden als komplex bezeichnet. Ein Beispiel zur Verdeutlichung ist, wenn man einen teilweise befüllten Einkaufskorb als ein unterstützendes Objekt ansieht, in das man gerade ein neues Objekt legen will.

## 5.2 ANFORDERUNGEN

Die Ablageplanung befasst sich mit der Zielpose eines Objektes auf einem unterstützenden Objekt. Dabei unterscheidet sie sich wesentlich von dem in der Robotik zentralen Forschungsbereich der Montageplanung (engl. assembly planning). Während in der Montageplanung die Zielpose vorgegeben ist und der Fokus auf der Reihenfolge der Manipulation der einzelnen Objekte und der Planung robuster Roboterbewegungen inklusive Feinbewegungen liegt, ist die Ablageplanung vor allem dadurch gekennzeichnet, dass die Zielpose für das zu manipulierende Objekt nicht gegeben ist.

Aus der Unbekanntheit der Zielpose entsteht die Herausforderung, dass diese Pose zusätzlich berechnet werden muss. Eine berechnete Zielpose kann nach verschiedenen Kriterien bewertet werden, von denen die Stabilität (Definition 1.4) des Objektes in der Zielpose das Wichtigste ist.

Weitere Kriterien ergeben sich basierend auf den Anforderungen der gestellten Aufgabe. Bei der Aufgabenstellung Objekte möglichst dicht in eine Kiste zu packen, muss ein Kriterium verwendet werden, das bewertet, ob ein Objekt dicht gepackt ist. Wenn die Aufgabe darin besteht, wie zum Beispiel beim Decken eines Tisches, mehrere Objekte auf einer Fläche anzuordnen, muss die Zielpose basierend auf der geforderten funktionsabhängigen Anordnung bewertet werden.

Wie bei der Montageplanung muss neben der Zielpose immer auch die zugehörige Feinbewegung geplant werden.

Im Fall der Ablageplanung von unbekanntem Objekten kommt eine weitere Anforderung hinzu. Nachdem ein direkter physischer Kontakt zwischen Objekten her-

gestellt werden muss, muss die Unsicherheit im Bezug auf die rekonstruierten Objektmodelle der involvierten realen Objekte berücksichtigt werden.

Die Zeit, die für eine Ablageplanung eines Objektes benötigt wird, soll der Anforderung aus Kapitel 1.2 einer interaktiven Planungszeit (Definition 1.5) genügen.

Der Algorithmus soll eine inkrementelle Planung ermöglichen. Das heißt, dass Planung und Ausführung nicht nur nacheinander anwendbar sind, sondern diese sich auch abwechseln können sollen.

Bezüglich der Geometrie der Ablagen beziehungsweise der unterstützenden Objekte soll der Planer möglichst alle Typen aus Kapitel 5.1 unterstützen. Die Fragestellung (H3) aus Kapitel 1.2, inwieweit die Ablageplanung von unbekanntem Objekten möglich ist, bezieht sich ebenfalls auf die Frage, welche der eingeführten Lagebeziehungen realisierbar sind.

### 5.3 STAND DER FORSCHUNG

Lozano-Pérez [Loz+89] hat eine der ersten Problemdefinitionen für Greif- und Ablageaufgaben formuliert. Seither liegt der Fokus der wissenschaftlichen Arbeiten auf der Greifplanung (Kapitel 4.3) für unbekanntem Objekte. Erst in den letzten Jahren erfährt die Ablageplanung zunehmend mehr Aufmerksamkeit.

An dieser Stelle wird nicht auf die Arbeiten eingegangen, die sich mit der klassischen Manipulationsplanung beschäftigen, bei der sowohl die Start- als auch Zielpositionen bekannt sind. Beispiele für diese Art von Problemstellung, sind: In [Sim+04] wird eine geometrische Lösung für dieses Problem vorgeschlagen, in [Sun+10; Kwa+11] wird ein Kontaktzustandsgraph basierend auf Dreiecken berechnet. Die Definition von Kontaktzuständen zwischen Objekt und Ablage wird in [Noz+13] auf Objekt-Roboter Kontaktzustände erweitert, um auch das Greifen der Objekte robuster zu gestalten. Eine Generalisierung der häufig verwendeten *skill primitives* wird in [Tho+03] eingeführt, ebenfalls mit dem Ziel die Ausführung robuster zu machen.

Im Folgenden werden verschiedene Ausprägungen der Ablageplanung vorgestellt. Angefangen mit dem geplanten Verschieben (engl. push planning) von Objekten, über das Anordnen von Objekten bis hin zu Ansätzen, die sich der generellen Ablageplanung widmen.

**VERSCHIEBEN** Beim push planning ist eine horizontale ebene Ablage gegeben, auf der Objekte stehen. Die Problemstellung ist diese Objekte neu anzuordnen ohne sie dabei anzuheben. In [Kro+14] wird eine Lösung vorgestellt, bei der mehrere Objekte in ein geometrisches Muster gebracht werden. Es werden ausschließlich gleiche Objekte manipuliert und nur die schon platzierten Objekte verwendet. Diese beiden Einschränkungen sind in [Mol+13] nicht gegeben. Es wird ein generelles Konzept zur Adaption einzelner Demonstrationen am Beispiel vom push planning zur Ablage eines neuen Objektes vorgestellt.

In [Cos+11] wird ein push-Planer eingeführt, der es ermöglicht mehrere Objekte gleichzeitig zu bewegen. Das ist genau dann der Fall, wenn das aktiv verschobene Objekt andere berührt und diese mit verschiebt.

Um Objekte greifen zu können deren Pose nicht exakt bekannt ist, führt Dogar in seiner Dissertation [Dog13] den so genannten *push grasp* ein. Das zu greifende Objekt wird so verschoben, dass es zentriert im Greifer liegt. Zusätzlich wird geplant wie andere Objekte verschoben werden müssen, damit das zu greifende Objekt erreicht werden kann.

Diese Ansätze zum Verschieben von Objekten operieren ausschließlich auf bekannten Objekten und Ablagen.

**ANORDNEN** Die Problemstellung eine Menge von Objekten zueinander anzuordnen kann an der Eigenschaft unterteilt werden, ob zusätzliche semantische Informationen über die Objekte verfügbar sind.

Ein Ansatz ohne zusätzliche semantische Informationen wird in [Bay+08] vorgestellt. Die Aufgabe ist Objekte auf dem Meeresboden in Form von geometrischen Mustern anzuordnen. In [Mos+11] wird ein Planer vorgestellt, der für typische Haushaltsszenarien Objektanordnungen erzeugt. Dabei wird die Durchführbarkeit mit einem Roboter unter Verwendung einer physikalischen Simulation mit berücksichtigt, wobei nur die Zielpose des Objektes auf Erreichbarkeit geprüft wird. Ein Entscheidungsprozess führt die verschiedenen Informationen zusammen. In [Hav+14] wird eine weitere Variante der Problemstellung ähnlich zu der des push planning betrachtet. Die Aufgabe ist Objekt effizient neu anzuordnen, damit ein neues Objekt platziert werden kann. Im Unterschied zum push planning werden Objekte angehoben und ineinander gelegt.

In [Gup+11] werden Ablagen basierend auf einer ähnlichen Taxonomie wie in Kapitel 5.1 mit bekannten Objekten geplant, wobei die Objekte auf einem einzelnen Ablageobjekt so platziert werden, dass ein Mensch diese verwenden beziehungsweise erreichen kann. Ein weiterer Ansatz, der es ermöglicht Objekte basierend auf ihrer Funktionalität in einer Szene anzuordnen wird in [Fis+11] vorgestellt. Allerdings wird nur das Problem der Auswahl des zu platzierenden Objektes basierend auf einem Ähnlichkeitsmaß und strukturellen Abhängigkeiten zwischen Objekten gelöst. Durch die strukturellen Abhängigkeiten wird implizit berücksichtigt, wie ein Mensch diese Objekte anordnen würde.

Die bisher genannten Ansätze sind beschränkt auf horizontale Kontakte zwischen dem Objekt und der Ablage und benötigen bekannte Objekte, von denen die unterschiedlichsten Zusatzinformationen verfügbar sein müssen. Die Bewertung, ob eine Ablagepose stabil ist, wird implizit angenommen und nicht verifiziert, da die Objekte auf horizontalen Ablagen platziert werden.

Eine konzeptionelle Erweiterung ist in [Jia+13] zu finden. Hier werden Objekte, basierend auf einem Lernverfahren und der Information über die Funktionalität der Objekte, in einer Szene positioniert, so dass ein Mensch diese verwenden kann. Das vorrangige Problem, das behandelt wird, ist wie beispielsweise ein vollständiges Zimmer mit Objekten ausgestattet werden kann. Durch das Lernverfahren ist es prinzipiell möglich unbekannte Objekte zu manipulieren. In deren experimentellen Analyse ist nachzulesen, dass erwartungsgemäß der Ansatz schlechtere Resultate liefert je größer die Abweichung des unbekanntes Objektes zu einem aus der Trainingsmenge ist.

**ABLEGEN** Bisher ist eine Ablagepose nicht auf ihre Stabilität (Definition 1.4) beurteilt worden. Im Folgenden werden Verfahren eingeführt, die diese explizit thematisieren.

Einfache Verfahren verwenden einen einzelnen Ablagepunkt (keine vollständige Pose) auf einem horizontalen unterstützenden Objekt und führen den Ablagevorgang mittels einer sensorüberwachten Bewegung [Jai+10] oder einer zusätzlichen passiven Nachgiebigkeit des Manipulators [Eds+06] aus. Durch diese Einschränkungen wird davon ausgegangen, dass das Objekt nach dem Lösen des Griffs durch den Roboter in Ruhe verharrt. In [Fin+03] werden ebenfalls sensorüberwachte Bewegungen verwendet, die durch skill primitives repräsentiert sind. Allerdings ist

die Lage des Objektes in Bezug auf die Sensorik unbekannt und die Ablage ist nicht ausschließlich eben. In einem Experiment wird gezeigt wie ein Akku in ein Mobiltelefon eingesetzt wird.

Die unbedingte Notwendigkeit von sensorischen Informationen während des Ablagevorgangs steht den Anforderungen aus Kapitel 5.2 entgegen. Daher werden im Folgenden ausschließlich Verfahren vorgestellt, die ausschließlich auf geometrischen Informationen operieren.

In [Fu+08] wird ein Lernverfahren vorgestellt, das die Vorzugsorientierung von Objekten berechnet. Das Verfahren ist ursprünglich dazu entwickelt worden automatisch synthetische Produktbilder zu erstellen. Dazu muss die aufrechte Orientierung bestimmt werden. Damit ist das Verfahren geeignet Objekte so zu orientieren, dass sie auf horizontalen Ablagen platziert werden können. Das Problem eine horizontale Ablagefläche auf einer unbekanntem Ablage zu finden wird in [Sch+10] thematisiert, allerdings ohne einen expliziten Ablageplaner vorzustellen. In [Tri+10] wird ein Algorithmus vorgestellt, der es ermöglichen soll unbekannte Objekte zu stapeln. Allerdings bleibt unklar wie sichergestellt wird, dass sich die Objekte stapeln lassen und wie die Ablagepose des zu stapelnden Objektes berechnet wird.

Ein Verfahren, das explizit Instabilität berücksichtigt, wird in [Hol+13] eingeführt. Die Ablageplanung wird als inverses Bewegungsplanungsproblem betrachtet, wobei instabile Ablageposen unter Verwendung eines zweiten Roboterarms abgestützt werden. Wie auch bei den bisherigen Verfahren sind nur ebene horizontale unterstützende Objekte als Ablagen möglich. Darüber hinaus ist das abzulegende Objekt bekannt.

In [Har+12; Har+14] wird ein Ablageplaner vorgestellt, der auch auf nicht ebenen Ablagen operieren kann. Die Oberfläche der bekannten Objekte und der unbekanntem Ablage werden in planare Cluster verlegt. Basierend auf der Nachbarschaftsbeziehung wird ein Puzzleproblem gelöst um zu ermitteln welcher Cluster des Objektes auf welchen der Ablage passt. Die Stabilität einer Ablage wird mittels der Analyse der Kontaktfläche realisiert. Dabei besteht die Einschränkung, dass diese Kontaktfläche nur soweit von der Horizontalen abweicht, dass das Objekt nicht ins Rutschen kommt.

Der einzige bekannte Ansatz, der ausschließlich basierend auf den geometrischen Informationen von Objekt und Ablage eine Ablageplanung durchführt ist in [Jia+12] nachzulesen. Das vorgestellte Lernverfahren generiert während des Ler-



nens mittels einer physikalischen Simulation basierend auf der Änderung der kinetischen Energie stabile Ablageposen. Unter Berücksichtigung von über 100 Eigenschaften wird für ein abzulegendes Objekt eine der simulierten Ablagen ausgewählt. Das Verfahren wird auch mit unbekanntem Objekten getestet. Es zeigt sich, dass die Resultate umso schlechter werden je größer die Abweichung der Objekte von der Trainingsmenge ist.

Alle vorgestellten Ansätze ermöglichen es nicht für unbekannte Objekte und Ablagen online und ohne verfügbares Vorwissen Ablagen zu planen. Wie schon in Kapitel 3 festgestellt, ist eine der Herausforderungen Kontaktposen ohne Informationen bezüglich der Objektdynamik zu berechnen.

#### 5.4 ALGORITHMUS

Der entwickelte Algorithmus benötigt ein Objektmodell (Definition 2.3) das abzulegen ist und ein Objektemodell, das als Ablage dient. Dabei muss die Ablage nicht aus einem einzigen realen Objekt bestehen. Vielmehr kann es auch eine halbgefüllte Kiste oder eine Ansammlung mehrerer Objekte sein. Diese Geometrien seien mit dem Verfahren aus Kapitel 2 rekonstruiert. Zusätzlich zu den Objektgeometrien sei die Richtung der Gravitation  $\vec{g} \in \mathbb{R}^3$  und die angestrebte Lagebeziehung (Kapitel 5.1) zwischen Ablage und Objekt bekannt. Der hier entwickelte Algorithmus basiert auf einem Sampling-Ansatz. Daher muss noch die Anzahl der Startposen, von denen der Planer ausgehen soll, festgelegt werden.

Gesucht ist eine statisch stabile Ablagepose (Definition 3.2) zwischen Objekt und Ablage. Neben der Ablagepose ist außerdem der Kontaktzustandsgraph gesucht, mit dem sich die Ausführung der Feinbewegungen direkt realisieren lässt. Ein Kontaktzustand sei wie folgt definiert:

**Definition 5.1.** Ein Kontaktzustand  $\mathcal{C} = (\mathbf{P}, \mathcal{K})$  ist ein Tupel mit der Kontaktpose  $\mathbf{P}$  (Definition 3.2) und der Menge der Kontaktpunkte mit korrespondierenden Normalen  $\mathcal{K} = \{(p_1, \vec{n}_1), \dots, (p_m, \vec{n}_m) \mid p_i, \vec{n}_i \in \mathbb{R}^3, \|\vec{n}_i\| = 1\}$ .

Bezüglich der Reibung zwischen den Oberflächen der involvierten Objekte wird angenommen, dass diese einem Reibungskoeffizienten  $\mu = 0, 2$  entspricht [Mül10]. Weiter wird angenommen, dass die Objekte starr und nicht deformierbar sind. Die Notwendigkeit beider Annahmen resultiert aus dem Umstand, dass diese Informa-

tionen nicht belastbar aus der Oberflächen- beziehungsweise Volumenrekonstruktion der Objekte und Ablagen gewonnen werden können.

Der Ablageplaner führt folgende grundsätzliche Schritte aus: Zuerst wird das Objekt basierend auf der geforderten Lagebeziehung in die Ausgangspose transformiert. Ausgehend davon wird es anhand von translatorischen und rotatorischen Bewegungen mit der Ablage in Kontakt gebracht. Basierend auf dem Kontaktzustand wird entschieden, welche Bewegung folgt und ob der aktuelle Kontaktzustand stabil ist. Im Falle einer stabilen Kontaktpose ist eine mögliche Lösung gefunden, die mit einem Stabilitätsmaß bewertet wird. Wenn anschließend der Kontaktzustand es zulässt und die geforderte Lagebeziehung eine Anordnung relativ zu weiteren Objekten fordert, wird eine stabile Kontaktpose nochmals angepasst.

Im Folgenden wird zuerst die Stabilitätsberechnung (Kapitel 5.4.1) als zentraler Bestandteil des Ablageplaners eingeführt. Anschließend wird der Aufbau des Kontaktzustandsgraphen angefangen bei der Berechnung der Startpose (Kapitel 5.4.2), über die Berechnung der benachbarten Kontaktzustände (Kapitel 5.4.3), bis hin zu der nachträglich möglichen Anpassung an die geforderte Lagebeziehung (Kapitel 5.5) erläutert.

#### 5.4.1 Stabilitätsberechnung

In diesem Unterkapitel wird Verfahren vorgestellt, das entscheidet, ob ein Oberflächenobjekt (Definition 2.3) in einem gegebenen Kontaktzustand  $\mathcal{C}$  (Definition 5.1) in Ruhe verharrt (Definition 1.4). Falls das nicht gegeben ist, soll die Versagensrichtung bestimmt werden, in die sich das Objekt bewegen würde. Zusätzlich soll im Fall der statischen Stabilität ein Qualitätsmaß für diese Kontaktpose angegeben werden können.

Dazu sind die Kontaktpunkte  $\mathcal{K} = \{(p_1, \vec{n}_1), \dots, (p_m, \vec{n}_m) \mid p_i, \vec{n}_i \in \mathbb{R}^3, \|\vec{n}_i\| = 1\}$  mit den korrespondierenden Kontaktnormalen, sowie die Richtung der Gravitation  $\vec{g} \in \mathbb{R}^3$  gegeben.

An einem Kontaktpunkt  $p_i$  wirken die Normalkraft  $\vec{F}_{N,i} = f_{N,i}\vec{n}_i$  und zwei Tangentialkräfte  $\vec{F}_{T1,i} = f_{T1,i}\vec{t}_{1,i}$  und  $\vec{F}_{T2,i} = f_{T2,i}\vec{t}_{2,i}$ , die die Haftreibung aufnehmen. Die Vektoren  $\vec{t}_{1,i}$  und  $\vec{t}_{2,i}$  sein so gewählt, dass sie aufeinander und auf  $\vec{n}_i$  orthogonal sind. Die skalaren Kraftwerte  $f_{N,i}, f_{T1,i}, f_{T2,i} \in \mathbb{R}$  sind nicht gegeben.

Ebenso lassen sich pro Kontaktpunkt die Momente bezüglich eines festen Punktes  $q \in \mathbb{R}^3$  angeben:

$$\begin{aligned}\vec{M}_{N,i}^q &= (p_i - q) \times \vec{F}_{N,i} \\ \vec{M}_{T1,i}^q &= (p_i - q) \times \vec{F}_{T1,i} \\ \vec{M}_{T2,i}^q &= (p_i - q) \times \vec{F}_{T2,i}\end{aligned}$$

Unter der Annahme, dass die Dichte des Objektes konstant ist, lässt sich der Oberflächenschwerpunkt  $c_{\mathcal{O}} \in \mathbb{R}^3$  für das Objekt bestimmen:

$$c_{\mathcal{O}} = \frac{\sum_{i=1}^{l_{\mathcal{O}}} (c_{P,i} \cdot A_{P,i})}{\sum_{i=1}^{l_{\mathcal{O}}} A_{P,i}}$$

mit den Schwerpunkten  $c_{P,i}$  und Flächeninhalten  $A_{P,i}$  der einzelnen  $l_{\mathcal{O}}$  Oberflächenpolygone.

Nach der Definition aus der Mechanik verharrt ein Objekt in Ruhe, sobald ein Kräfte- und Momentengleichgewicht besteht:

$$\begin{aligned}\vec{0} &= \sum_{i=1}^m (\vec{M}_{N,i}^q + \vec{M}_{T1,i}^q + \vec{M}_{T2,i}^q) \\ \vec{0} &= m_{\mathcal{O}} \vec{g} + \sum_{i=1}^m (\vec{F}_{N,i} + \vec{F}_{T1,i} + \vec{F}_{T2,i})\end{aligned}$$

mit der Masse des Objekts  $m_{\mathcal{O}}$ . Die direkte Lösung dieser Gleichgewichtsbedingungen ist nicht zielführend, da schon bei einfachen Kontaktzuständen die Verteilung der Kräfte auf die Kontaktpunkte mit ihren Normalen nicht eindeutig ist. Ein einfaches Beispiel ist ein quadratischer Tisch mit vier Beinen auf einer horizontalen ebenen Fläche stehend. Die zwei Tischbeine, die diagonal gegenüber liegen, reichen aus, um die Gewichtskraft des Tisches zu kompensieren und die Kraft- und Momenten-Gleichgewichtsbedingungen zu erfüllen. Den verbleibenden Tischbeinen können beliebige Kraftwerte zugewiesen werden, solange diese sich additiv auslöschen.

In [Mil+01] wird eine Lösung zur Berechnung der Kontaktkräfte für konvexe Geometrien vorgestellt. Nachdem in dieser Arbeit diese Annahme nicht getroffen wird, ist dieser Ansatz nicht geeignet, da die Zerlegung einer Geometrie in konvexe Teilgeometrien besonders rechenaufwendig ist. Eine solche weitverbreitete Biblio-

thek ist V-HACD<sup>1</sup> einer Weiterentwicklung von [Mam+09], deren Rechenzeit selbst bei der Berechnung auf der Grafikkarte noch mehrere Sekunden ist.

Die Finite Elemente Analyse (beispielsweise [Bra13]) stellt eine weitere Möglichkeit dar die Kontaktkräfte zu berechnen. Hierzu werden alle Kontaktkräfte als Lager angenommen. Allerdings werden auch in diesem Verfahren spezielle Anforderungen an die Modellierung des Objektes gestellt. Es muss als Volumenmodell bestehend aus finiten Elementen vorliegen. Die Güte der Lösung und die Lösbarkeit selbst hängen maßgeblich an der Qualität des erzeugten Modells. Erstrebenswert ist eine möglichst homogene Anordnung der finiten Elemente. Eine robuste automatische Erzeugung dieses Modells ist für allgemeine Oberflächengeometrien sehr rechenaufwendig ebenso wie die Finite Elemente Analyse selbst.

Der Lineare-Komplementaritätsproblem-Ansatz aus [Bar94] arbeitet auch mit nicht konvexen Geometrien. Allerdings ist eine Berücksichtigung von Haftreibung bei diesem Ansatz problematisch, da das aufgestellte Gleichungssystem in dem Fall Singularitäten enthält und damit numerisch instabil wird. In [Bar94] wird zusätzlich festgestellt, dass die Berechnung der Kontaktkräfte bei gegebenen Kontaktpunkten inklusive Kontaktnormalen NP-hart ist. Folglich wird eine Heuristik vorgestellt, die ohne explizite Berechnung der Kontaktkräfte auskommt und eine akzeptable Rechenzeit aufweist, damit die Forderung bezüglich der Rechenzeit des Algorithmus (Kapitel 1.2) eingehalten werden kann.

### *Ebener Fall*

Wenn in Abbildung 45 die rote Kontaktfläche nicht von der Geraden  $(c_{\mathcal{O}}, \vec{g})$  durch den Schwerpunkt  $c_{\mathcal{O}}$  des Objektes und parallel zur Gravitationsrichtung  $\vec{g}$  geschnitten wird, ist der Gleichgewichtszustand instabil. Wenn der Schnittpunkt auf dem Rand der Kontaktfläche liegt, ist der Gleichgewichtszustand labil. Liegt der Schnittpunkt innerhalb der Kontaktfläche, ist der Gleichgewichtszustand stabil.

Angenommen alle Kontaktpunkte  $\mathcal{K}$  liegen in einer gemeinsamen Ebene  $E_{\mathcal{K}}$  mit identischen Kontaktnormalen, dann kann die konvexe Hülle

$$\Omega_{\mathcal{K}} = \left\{ \sum_{i=1}^{|\mathcal{K}|} \lambda_i p_i \mid (\forall i : \lambda_i \geq 0) \wedge \sum_{i=1}^{|\mathcal{K}|} \lambda_i = 1 \wedge p_i \in \mathcal{K} \right\}$$

<sup>1</sup> <https://code.google.com/p/v-hacd/>

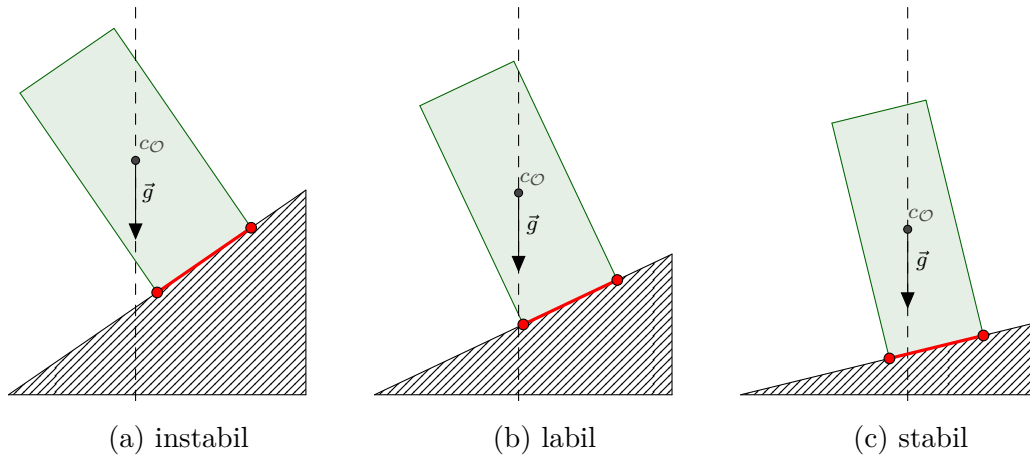


Abbildung 45: Zweidimensionale Illustration der Gleichgewichtszustände mit der Kontaktfläche (rot), dem Schwerpunkt  $c_O$  und der Gravitationsrichtung  $\vec{g}$

der Kontaktpunkte berechnet werden und der Kontaktzustand als Flächenkontakt zwischen Objekt und Ablage angenommen werden. Falls für den Schnittpunkt  $\hat{c}_O = (c_O, \vec{g}) \cap E_K$  gilt  $\hat{c}_O \in \Omega_K$ , ist die Kontaktpose stabil unter der Bedingung, dass die Reibung so groß ist, dass das Objekt nicht ins Rutschen gerät. In Abbildung 45 sind dieser und die weiteren Gleichgewichtszustände illustriert.

Die Güte ihrer Stabilität hängt von der euklidischen Distanz  $d_{\hat{c}_O, \Omega_K}$  zwischen  $\hat{c}_O$  und der Hülle von  $\Omega_K$ , der euklidischen Distanz  $d_{c_O, \Omega_K}$  zwischen dem Schwerpunkt des Objekts  $c_O$  und  $\Omega_K$  und der Distanz  $d_{c_O, c_{\Omega_K}}$  zwischen dem Schwerpunkt der konvexen Hülle  $c_{\Omega_K}$  und  $\hat{c}_O$  ab. Daraus berechnet sich das Qualitätsmaß

$$Q := 1 - \frac{d_{c_O, \Omega_K}}{2A_{\Omega_K}} - \frac{d_{\hat{c}_O, c_{\Omega_K}}}{2d_{\hat{c}_O, \Omega_K}}$$

Nachdem die Reibung und damit die Haftkraft nicht flächenabhängig ist, reicht es aus zu testen, ob der Winkel  $\alpha_K$  zwischen der Normalen  $\vec{n}_K$  der Ebene  $E_K$  und  $\vec{g}$  kleiner ist als der Winkel  $\alpha_\mu = \arctan(\mu)$ , der durch den Reibungskoeffizienten  $\mu$  definiert ist.

Falls  $\hat{c}_O$  außerhalb  $\Omega_K$  liegt oder  $\alpha_K > \alpha_\mu$ , ist die Kontaktpose instabil. In diesem Fall ist die Versagensrichtung  $\vec{v}_{\text{Vers}}$  folgendermaßen definiert:

**Definition 5.2.** Die *Versagensrichtung* ist  $\vec{v}_{\text{Vers}} = \hat{c}_{\mathcal{O}} - c_{\Omega_{\mathcal{K}}}$  mit dem dreidimensionalen Schwerpunkt der konvexen Hülle  $c_{\Omega_{\mathcal{K}}} \in \Omega_{\mathcal{K}}$  und dem auf die Ebene der Kontaktpunkte projizierten Schwerpunkt  $\hat{c}_{\mathcal{O}}$ .

In diese Richtung wird das Objekt rutschen beziehungsweise kippen. Damit ist der Drehpunkt, um den das Objekt kippen wird, wie folgt definiert:

**Definition 5.3.** Der *Drehpunkt*  $p_{\text{Vers}} \in \Omega_{\mathcal{K}}$ , an dem das Kippmoment anliegt, ist der Schnittpunkt mit dem Rand von  $\Omega_{\mathcal{K}}$  und der Strecke zwischen  $\hat{c}_{\mathcal{O}}$  und  $p_{\Omega_{\mathcal{K}}}$ . Die Achse, um die das Objekt kippt, steht demnach normal auf der Versagensrichtung und der Ebenennormalen, die gleichzeitig die Kontaktnormale aller Punkte ist.

#### *Nicht-ebener Fall*

Wenn nicht alle Kontaktpunkte  $\mathcal{K}$  auf einer gemeinsamen Ebene liegen (Abbildung 46a), besteht nicht die Möglichkeit direkt eine Vereinfachung wie im Ebenen Fall vorzunehmen, da die skalaren Kraftwerte nicht gegeben sind. Ansonsten ließe sich das Verfahren aus [Mil+99] anwenden, das die Stabilitätsberechnung mit Hilfe der konvexen Hülle im Schraubenraum durchführt.

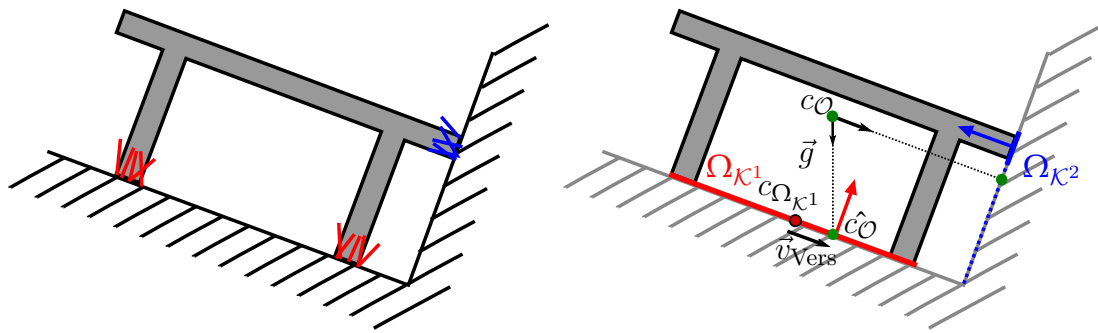
Die hier entwickelte Heuristik basiert darauf, dass der ebene Fall iterativ angewendet wird. Dazu werden disjunkte Teilmengen  $\mathcal{K}^k \subseteq \mathcal{K}$  der Kontaktpunktmenge  $\mathcal{K}$  betrachtet. Sei  $M_{\text{T}}$  die Menge aller erzeugten Teilmengen. Die Kontaktpunkte einer Teilmenge genügen der Bedingung, dass ihre korrespondierenden Normalen innerhalb einer gewählten Toleranz in die gleiche Richtung zeigen (Abbildung 46a). Diese gemeinsame Kontaktnormale sei  $\vec{n}_{\mathcal{K}}^k = \frac{1}{|\mathcal{K}^k|} \sum_{i=1}^{|\mathcal{K}^k|} \vec{n}_i$ .

Alle Kontaktpunkte innerhalb einer Teilmenge stützen das Objekt in der identischen Kontaktkraftrichtung. Nachdem das Objekt als starr angenommen ist, kann deren Einfluss auf das Kräftegleichgewicht gemeinsam betrachtet werden.

Die Reihenfolge in der die Teilmengen betrachtet werden, ist aus deren Einfluss auf das Kräftegleichgewicht motiviert. Nachdem die Gewichtskraft des Objektes in Richtung der Gravitation wirkt, wird zuerst die Teilmenge untersucht mit der Eigenschaft

$$\mathcal{K}^1 := \arg \max_{\mathcal{K}^k \in M_{\text{T}}} \left( \angle(\vec{g}, \vec{n}_{\mathcal{K}}^k) \right),$$

da sie den größten Beitrag zur Kompensation der Gewichtskraft leistet (Abbildung 46b).



(a) Ausgangssituation

(b) Reihenfolge der Behandlung und Berechnung der Versagensrichtung  $\vec{v}_{\text{Vers}}$  der ersten Teilmenge.

Abbildung 46: Zweidimensionale Illustration der Stabilitätsberechnung anhand der Kontaktpunktteilmenge  $\mathcal{K}^k$ , mit deren konvexen Hüllen  $\Omega_{\mathcal{K}^k}$ , deren Schwerpunkten  $c_{\Omega_{\mathcal{K}^k}}$ , und dem Objektschwerpunkt  $c_O$  und dessen Projektion  $\hat{c}_O$  entlang  $\vec{g}$

Um die Gleichgewichtsbedingung als ebenen Fall zu behandeln, muss die Ebene noch vollständig definiert werden. Dazu wird der Mittelpunkt der Kontaktpunkte der Teilmenge als Stützpunkt der Ebene verwendet und die Punkte in  $\mathcal{K}^1$  auf diese Ebene entlang  $\vec{n}_{\mathcal{K}}^1$  projiziert.

Aus der Anwendung des ebenen Falls ergibt sich, ob die vorliegende Kontaktpose bezüglich dieser Teilmenge der Kontakte schon stabil ist. In diesem Fall endet die iterative Berechnung. Andernfalls muss aus den verbleibenden Teilmengen eine Weitere ausgewählt werden, die das Objekt in Richtung des Versagens stützt.

Basierend auf  $\mathcal{K}^1$  wird die Gewichtskraft in Richtung  $\vec{g}$  durch Kräfte in Richtung der Normalen  $\vec{n}_{\mathcal{K}}^1$  und der negativen Versagensrichtung  $-\vec{v}_{\text{vers}}^1$  (Definition 5.2), die als Richtung für die Haftkraft angenommen wird, gestützt. Das bedeutet, dass alle Teilmengen deren Normale durch eine Linearkombination aus diesen drei Vektoren mit positiven Faktoren, nur einen untergeordneten Beitrag zur Etablierung des Gleichgewichtszustandes haben. Wenn man die Reibung außer Acht lässt und damit  $\vec{v}_{\text{vers}}^1$  nicht berücksichtigt, kompensiert die Teilmenge das Versagen am besten, deren Normale  $\vec{n}_{\mathcal{K}}^k$  die verfügbaren Kraftrichtungen  $\vec{g}$  und  $\vec{n}_{\mathcal{K}}^1$  additiv auslöscht.

Daraus ergibt sich motiviert durch das Kräftegleichgewicht die Bedingung für die Auswahl der nächsten Teilmenge:

$$\arg \min_{M_{\text{PK}}^k \in M_{\text{T}}} \left( \angle(\vec{r}, \vec{n}_{\mathcal{K}}^k) \right) \wedge \\ \angle(-\vec{n}_{\mathcal{K}}^1, \vec{n}_{\mathcal{K}}^k) \leq \pi/2$$

mit  $\vec{r} = -\vec{n}_{\mathcal{K}}^1 - \vec{v}_{\text{vers}}^1$ . Um zusätzlich die durch die erste Kontaktfläche auftretenden Momente zu kompensieren, muss die Drehrichtung des Momentes durch die zu wählende Kontaktfläche entgegen dem aktuellen wirken. Das impliziert, dass der Schwerpunkt der zu bestimmenden nächsten Teilmenge in der positiven Halbebene in Bezug auf die Ebene liegen muss, die durch den Schwerpunkt des Objekts  $\mathcal{O}$  und der Normalen  $\vec{n}_{\mathcal{K}}^1$  definiert wird.

Sei die so ausgewählte nächste Teilmenge  $\mathcal{K}^2$ . Für diese Kontaktpunkte soll nun wiederum der ebene Fall angewendet werden. Dazu wird die Richtung der Gravitation durch  $-\vec{n}_{\mathcal{K}}^2$  ersetzt. Nachdem die Haftreibung der Kontaktebene von  $\mathcal{K}^1$  die Stabilität in Richtung  $\vec{n}_{\mathcal{K}}^2$  mit beeinflusst, werden zu den Kontaktpunkten aus  $\mathcal{K}^2$  die Kontaktpunkte aus  $\mathcal{K}^1$  exklusive derer Kontaktnormalen mit verwendet. Diese zusätzlichen Kontaktpunkte werden entlang  $\vec{n}_{\mathcal{K}}^2$  auf die für die Berechnung des ebenen Falls verwendete Ebene projiziert und bei der Berechnung der konvexen Hülle mitberücksichtigt. Angenommen es sind schon  $l$  Teilmengen betrachtet worden, dann werden die Kontaktpunkte aller vorherigen  $l - 1$  Teilmengen projiziert und mitberücksichtigt.

Dieses Vorgehen wird solange fortgeführt bis keine Teilmenge mehr ausgewählt werden kann oder die Lage als stabil eingeschätzt wird. Falls die vorliegende Kontaktpose als instabil bewertet wird, wird die Versagensrichtung und der Drehpunkt aus der Berechnung der letzten Teilmenge verwendet.

Falls die vorliegende Kontaktpose als stabil eingestuft wird, muss noch untersucht werden, ob die Haftreibung an einem Kontaktpunkt ausreicht beziehungsweise durch die verwendeten Kontaktnormalen aus den Teilmengen kompensiert werden kann. Dieses Versagen wird wiederum anhand der verwendeten Teilmengen geprüft. Falls eine verwendete Teilmenge  $\mathcal{K}^f$  existiert, bei der der Winkel zwischen berechnetem Drehpunkt (Definition 5.3) und Schwerpunkt des Objekts größer ist als der Winkel des Reibungskegels, wird angenommen, dass das Objekt in Richtung der negativen Versagensrichtung  $-\vec{v}_{\text{vers}}^f$  abrutscht. Falls zusätzlich aus den übrigen



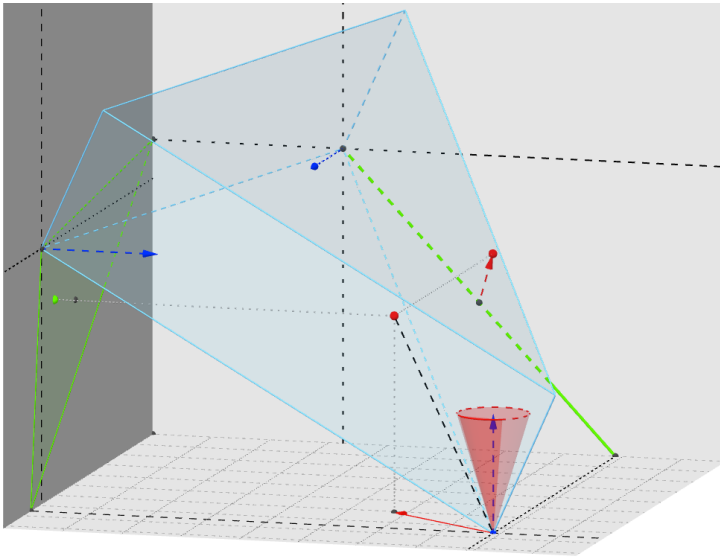


Abbildung 47: Beispiel für eine Kontaktpose mit versagender Reibung am horizontalen Kontaktpunkt. Bedeutung der Farben: Objekt hellblau, Kontaktpunkte und Kontaktnormalen blau, Konvexe Hüllen grün, Versagensrichtungen und horizontaler Reibungskegel rot.

verwendeten Teilmengen keine Linearkombination der Kontaktnormalen mit positiven Faktoren existiert, die diese negative Versagensrichtung kompensiert, wird die Kontaktpose als instabil bewertet. In diesem Fall werden die Versagensrichtung und der Drehpunkt aus der Berechnung des ebenen Falls mit  $\mathcal{K}^f$  verwendet. In Abbildung 47 ist beispielhaft eine solche versagende Kontaktpose dargestellt.

Wenn ein Kontaktzustand als stabil bewertet ist, wird das Qualitätsmaß basierend auf dem zuletzt betrachteten Teilmenge als Bewertung des gesamten Kontaktzustands verwendet.

#### 5.4.2 Startpose

Die resultierende Ablage hängt maßgeblich von der initialen Pose des Objektes ab. Aus diesem Grund wird eine einfache Heuristik eingeführt, die basierend auf den geometrischen Eigenschaften des Objektes und der geforderten Lagebeziehung entscheidet, wie das Objekt vororientiert wird und wie die initiale Position relativ zur Ablage berechnet wird.

Wenn man die Hauptachsen  $\vec{v}_1, \vec{v}_2, \vec{v}_3 \in \mathbb{R}^3$  mit den korrespondierenden Eigenwerten  $e_1 \geq e_2 \geq e_3, e_i \in \mathbb{R}$  und die daraus resultierende minimale Boundingbox eines Objektes heranzieht, dann lassen sich daraus verschiedene Annahmen bezüglich der bevorzugten Orientierung des Objektes formulieren. Diese Annahmen

werden verwendet um das Objekt bezüglich der Ablage heuristisch zu platzieren. Nachdem die Ablage von einem Roboter durchgeführt werden soll, der das Objekt in einer bestimmten Greiferkonfiguration gegriffen hat, muss auch sichergestellt werden, dass jede der Platzierungen mit dem vorliegenden Griff ausgeführt werden kann. Ansonsten müsste der Roboter das Objekt umgreifen. Ein solches Umgreifen ist nur möglich, wenn sich das Objekt während der Platzierung temporär in einer stabilen Pose befindet.

Grundsätzlich lässt sich folgende Aussage treffen: Je größer die Auflagefläche des Objektes mit der Ablage ist und je näher der Schwerpunkt des Objektes an der Oberfläche der Ablage liegt, desto stabiler ist der Kontaktzustand (vergleiche Stabilitätsbewertung Kapitel 5.4.1).

An folgendem Beispiel lässt sich die Idee veranschaulichen: Wenn ein Löffel in eine Kaffeetasse gelegt werden soll, wird wohl niemand versuchen den Löffel nicht hochkant in die Tasse zustellen, weil er anders orientiert nicht in die Tasse passt. Wenn der Löffel dagegen auf einen Tisch gelegt werden soll, ist es selbstverständlich, dass man versucht ihn flach auf den Tisch zu legen.

Wie lässt sich diese Beobachtung anhand der Hauptachsen beschreiben? Angenommen die zu den Hauptachsen des Löffels korrespondierenden Eigenwerte  $e_1, e_2, e_3 \in \mathbb{R}$  genügen folgenden Relationen:  $e_1 \gg e_2$  und  $e_2 - e_3 < \epsilon$ . Dann ist die Folgerung daraus, dass wenn die geforderte Lagebeziehung *ineinander* ist und das Objekt eine dominante Hauptachse aufweist, das Objektmodell so gedreht wird, dass der Eigenvektor  $\vec{v}_1$  zu dem Eigenwert  $e_1$  parallel zur Gravitationsrichtung ist.

Wenn man das Beispiel aus dem Blickwinkel der Ablage analysiert, stellt man fest, dass man versucht Objekte möglichst weit nach unten in der Ablage zu legen. Diese Strategie lässt sich auch beim Packen von Kisten beobachten. Wenn man hingegen ein Objekt auf ein anderes stapeln soll, versucht man es möglichst weit oben abzulegen.

Um daraus eine gute Startpose für das Objekt zu berechnen muss noch eine Position relativ zu der Ablage gefunden werden. Dazu wird das Objektmodell der Ablage abgetastet. In Abbildung 48 ist dieser Prozess illustriert. Ohne Einschränkung der Allgemeinheit wird angenommen, dass die Richtung der Gravitation  $\vec{g}$  gegengleich zur z-Achse des Koordinatensystems der Ablage ist. Sei die achsenparallele Boundingbox  $B$  des Ablagemodells durch die Punkte  $p_{\max}, p_{\min} \in \mathbb{R}^3$  definiert. Dann werden auf der Oberseite (liegt in der Ebene  $(p - p_{\max}) \cdot \vec{g} = 0$ )

der Boundingbox gleich-verteilt Punkte generiert (Abbildung 48a). Diese Punkte definieren die Menge  $\mathcal{F}'$ . Der Abstand der Punkte in  $\mathcal{F}'$  ist abhängig von der Größe des Objektes das abgelegt werden soll. Wenn die Punkte zu eng beieinander liegen wird der Ablageplaner für jeden Startpose sehr ähnliche stabile Posen finden. Daher wird der Abstand der Punkte zueinander heuristisch auf die halbe Länge der Boundingbox-Diagonalen des Objektes festgelegt.

Jeder dieser Punkte in  $\mathcal{F}'$  wird entlang  $\vec{g}$  auf das Oberflächenmodell der Ablage projiziert. Das ergibt die Kandidatenmenge für die initiale Positionierung des Objektes bezüglich der Ablage  $\mathcal{F} = \{p = (p', \vec{g}) \cap \mathcal{O}_{\text{Sta}} | p' \in \mathcal{F}'\}$ .

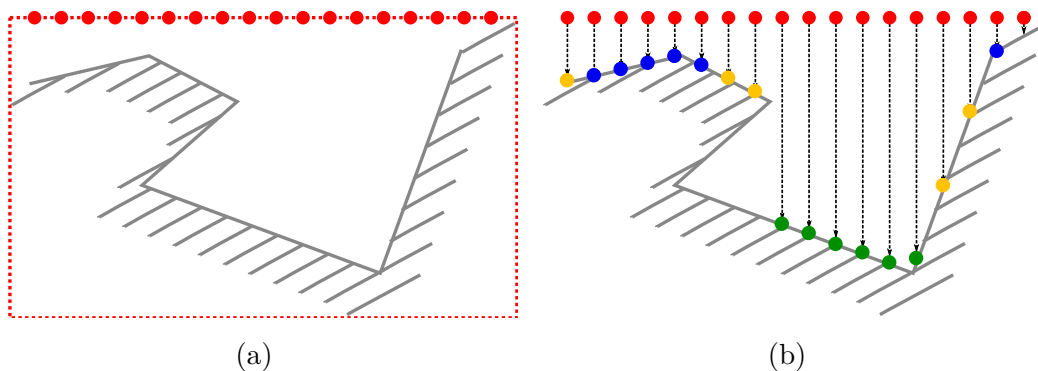


Abbildung 48: Berechnung der Startpunkte für die Ablageplanung: (a) Illustration der Berechnung von  $\mathcal{F}'$  auf Basis der Boundingbox (rot) der Ablage, (b) Illustration der Berechnung von  $\mathcal{F}$  inklusiver farblicher Markierung der sechs untersten (grün) und sechs obersten (blau) Startpunkte

Die Kandidatenmenge  $\mathcal{F}$  wird abhängig von der gegebenen Lagebeziehung weiter ausgedünnt. Wenn die Lagebeziehung *ineinander* sein soll, werden die  $n$  untersten Punkte basierend auf  $\vec{g}$  aus der Kandidatenmenge verwendet. In den anderen Fällen werden die  $n$  obersten Punkte verwendet. Wie Eingang des Kapitels 5.4 beschrieben entsprechen diese  $n$  Punkte genau der Anzahl der Startposen die der Algorithmus verwenden soll, um stabile Ablagen für das Objekt zu berechnen. Die Vororientierung des Objektes wird nur verändert, wenn die geforderte Lagebeziehung *ineinander* ist. Dann wird es so gedreht, dass die dominante Hauptachse parallel zur Richtung der Gravitation ist. Die hier vorgestellte Vorgehensweise garantiert nicht, dass die angestrebte Lagebeziehung während des Planungsvorgangs

erreicht wird. Durch die gezielte Auswahl der Punkte aus der Kandidatenmenge wird es lediglich wahrscheinlicher, dass diese erreicht wird, da das Objekt an eine günstige Startpose bezüglich der Ablage transformiert wird. Um die Lagebeziehung zu garantieren müsste eine stabile Ablagepose bezüglich dieses Kriteriums bewertet werden.

Nach dem Ausdünnen der Kandidatenmenge enthält sie genau  $n = \|\mathcal{F}\|$  Punkte. Für jeden Punkt  $p_{\mathcal{F}} \in \mathcal{F}$  kann jetzt zusammen mit der Lagebeziehung eine Startpose des Objektes im Bezug auf die Ablage berechnet werden. Die nicht festgelegten Freiheitsgrade bei der Orientierung werden frei gewählt. Die Position des Objektes wird bezüglich dessen Schwerpunktes  $c_{\mathcal{O}}$  festgelegt als  $c_{\mathcal{O}} = p_{\mathcal{F}} + (0, 0, d)^T$  mit der Höhe  $d$ , so dass sich die Boundingboxen der Ablage und des Objektes nicht berühren.

### 5.4.3 Kontaktzustandsgraph

In Kapitel 5.4.2 wird die Berechnung der Startposen für eine gegebene Objekt-Ablage-Paarung abhängig von der gegebenen Lagebeziehung eingeführt. Deren Anzahl ist, wie eingangs von Kapitel 5.4 beschrieben, fest vorgegeben. In diesem Kapitel wird darauf aufbauend das Verfahren vorgestellt, wie man von einer Startpose zu stabilen Kontaktzuständen gelangt. Für jede Startpose  $p_{\mathcal{F}} \in \mathcal{F}$  wird gleichzeitig ein Kontaktzustandsbaum  $T_{\mathcal{C}}(p_{\mathcal{F}})$  berechnet.

Abbildung 49 bildet den logischen Ablauf der Ablageplanung für eine Startpose ab. Im Folgenden werden zuerst die Funktionen definiert, die darin verwendet werden. Anschließend wird das Vorgehen erläutert und diskutiert.

Im ersten Schritt berechnet die Funktion aus der Menge aller Kontaktzustände  $M_{\mathcal{C}}$

$$\begin{aligned} \iota_{\mathcal{O}_{\text{Bew}}, \mathcal{O}_{\text{Sta}}} : \mathcal{F} &\rightarrow M_{\mathcal{C}}, \\ \iota_{\mathcal{O}_{\text{Bew}}, \mathcal{O}_{\text{Sta}}}(p_{\mathcal{F}}) &\mapsto \mathcal{C}_0, \end{aligned}$$

gemäß Kapitel 5.4.2 pro generierter Startpose auf der Ablage den korrespondierenden Startkontaktzustand  $\mathcal{C}_0$  des bewegten Objektes  $\mathcal{O}_{\text{Bew}}$  relativ zu dem statischen Objekt  $\mathcal{O}_{\text{Sta}}$ . Bei diesem  $\mathcal{C}_0$  ist nach Definition der Startpose die Menge der Kontaktpunkte  $\mathcal{K}_0 \in \mathcal{C}_0$  leer.

Als zweiter Schritt wird mit der Funktion

$$\begin{aligned} \tau_{\mathcal{O}_{\text{Bew}}, \mathcal{O}_{\text{Sta}}} : M_{\mathcal{C}} \times \mathbb{R}^3 &\rightarrow M_{\mathcal{C}}, \\ \tau_{\mathcal{O}_{\text{Bew}}, \mathcal{O}_{\text{Sta}}} (\mathcal{C}_i, \vec{d}_L) &\mapsto \mathcal{C}_{i+1}, \end{aligned}$$

durch eine translatorische Bewegung entlang einer Bewegungsrichtung  $\vec{d}_L$  gemäß Kapitel 3.4 eine neue Kontaktpose für ein bewegtes  $\mathcal{O}_{\text{Bew}}$  und statisches  $\mathcal{O}_{\text{Sta}}$  Objektmodell berechnet, mit der der erste Kontakt zwischen den beiden Modellen hergestellt wird.

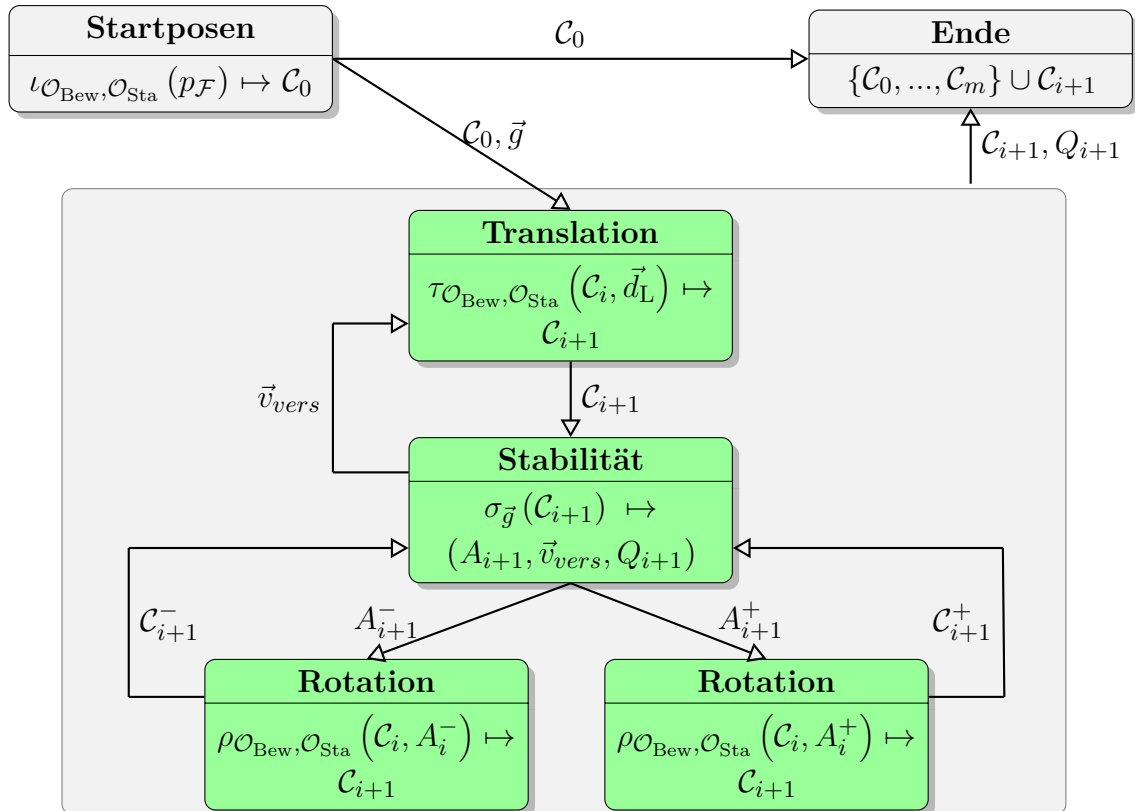


Abbildung 49: Ablauf der Ablageplanung

Im dritten Schritt wird der Kontaktzustand der Stabilitätsberechnung aus Kapitel 5.4.1 unterzogen. Zu diesem Zeitpunkt werden auch die neuen Kontaktpunkte zwischen  $\mathcal{O}_{\text{Bew}}$  und  $\mathcal{O}_{\text{Sta}}$  berechnet. Ebenso wie bei der Greifplanung werden nur Kontaktpunkte berücksichtigt, die auf sensorvalidierten Bereichen der Objektmodelle liegen (Kapitel 4.4.2). Ein Kontaktpunkt muss in beiden Modellen auf einer

sensorvalidierten Oberfläche liegen, ansonsten kann nicht davon ausgegangen werden, dass der Kontaktpunkt auch einem Kontakt beider realer Objekte entspricht. Die Stabilitätsberechnung sei in der Funktion

$$\begin{aligned}\sigma_{\vec{g}} : M_{\mathcal{C}} &\rightarrow M_A \times \mathbb{R}^3 \times \mathbb{R}, \\ \sigma_{\vec{g}}(\mathcal{C}_i) &\mapsto (A, \vec{v}_{vers}, Q_i)\end{aligned}$$

gekapselt. Sie berechnet die Achse  $A$ , um die das Objekt  $\mathcal{O}_{\text{Bew}}$  kippen kann, die Versagensrichtung  $\vec{v}_{vers}$ , in die  $\mathcal{O}_{\text{Bew}}$  rutschen kann, und mit  $Q_i$  das Qualitätsmaß. Falls  $Q_i < 0$  ist  $\mathcal{C}_i$  als instabil bewertet, andernfalls als stabil. Wenn der Kontaktzustand als stabil bewertet wird, terminiert die iterative Berechnung neuer Kontaktzustände.

Im Fall eines instabilen Kontaktzustands muss entschieden werden, ob dieser durch eine translatorische oder rotatorische Bewegung in einen neuen Kontaktzustand transformiert werden soll.

Wenn die Stabilitätsanalyse ergibt, dass der Kontaktzustand keinem Kräftegleichgewicht genügt, wird eine rotatorische Bewegung ausgeführt. Aus der Stabilitätsanalyse ist die dafür nötige Achse  $A \in M_A := \{p + \lambda \vec{d} \mid p, \vec{d} \in \mathbb{R}^3, \lambda \in \mathbb{R}^+\}$  gegeben. Die Funktion

$$\begin{aligned}\rho_{\mathcal{O}_{\text{Bew}}, \mathcal{O}_{\text{Sta}}} : M_{\mathcal{C}} \times M_A &\rightarrow M_{\mathcal{C}}, \\ \rho_{\mathcal{O}_{\text{Bew}}, \mathcal{O}_{\text{Sta}}}(\mathcal{C}_i, A) &\mapsto \mathcal{C}_{i+1}\end{aligned}$$

berechnet diese Bewegung gemäß Kapitel 3.5. Es wird jeweils eine Rotation in mathematisch positiver Richtung um die Achsen  $A^+ : p + \lambda \vec{d}$  und  $A^- : p + \lambda(-\vec{d})$  ausgeführt. Dadurch teilt sich die Sequenz der Kontaktzustände in zwei Zweige auf und es entsteht der Kontaktzustandsbaum  $T_{\mathcal{C}}(p_{\mathcal{F}})$ .

Wenn die Stabilitätsanalyse ergibt, dass der aktuelle Kontaktzustand auf Grund des Überschreitens der zulässigen Haftkraft an einem Kontaktpunkt versagt, folgt eine translatorische Bewegung in Richtung des Versagens  $\vec{v}_{vers}$ .

Diese iterative Erzeugung eines Kontaktzustandsbaumes wird solange fortgeführt bis

- a) ein Kontaktzustand als stabil bewertet wird,
- b) der Kontaktzustandsbaum  $T_{\mathcal{C}}(p_{\mathcal{F}})$  eine Tiefe von  $t_{T_{\mathcal{C}}}$  erreicht hat,

c) ein neuer Kontaktzustand sich nicht vom vorherigen unterscheidet, oder

Bezüglich der Rechenzeit des Verfahrens ist die Tiefe von  $T_{\mathcal{C}}(p_{\mathcal{F}})$  ein entscheidendes Kriterium. Es ist notwendig, da Abfolgen von Kontaktzuständen berechnet werden können, die sich unterscheiden aber von der Stabilitätsheuristik nicht als stabil bewertet werden. Es lässt sich basierend auf der ungünstigsten Abfolge möglicher Kontaktzustände eine untere Schranke bezüglich  $t_{T_{\mathcal{C}}}$  angeben. Diese Abfolge ist ein einzelner Punktkontakt, gefolgt von einem reinen Linienkontakt, gefolgt von einem ebenen Kontaktzustand, gefolgt von einem allgemeinen Kontakt. Daraus ergibt sich mit der initialen Bewegung in Richtung der Ablage die Baumtiefe  $t_{T_{\mathcal{C}},\min} = 5$ , die erforderlich ist, um auch bei der ungünstigsten Abfolge an Kontaktzuständen stabile Ablagen zu berechnen. Wenn diese unterschritten wird ist es für allgemeine Geometrien von  $\mathcal{O}_{\text{Sta}}$  nicht mehr möglich eine stabile Kontaktpose für  $\mathcal{O}_{\text{Bew}}$  zu finden. Alle Baumtiefen  $t_{T_{\mathcal{C}}} > t_{T_{\mathcal{C}},\min}$  ermöglichen dem Verfahren instabile Kontaktzustände weiter zu optimieren.

Alle Blätter von  $T_{\mathcal{C}}(p_{\mathcal{F}})$  besitzen eine Bewertung bezüglich der Stabilität. Der stabilste Kontaktzustand  $\mathcal{C}_i$  ist der, dessen Bewertung  $Q_i$  maximal ist (Kapitel 5.4.1). Ausgehend von diesem Kontaktzustand wird die Sequenz der Kontaktzustände, die zu dieser stabilen Kontaktpose für  $\mathcal{O}_{\text{Bew}}$  führt durch einfaches Backtracking im Baum bis zum Wurzelement synthetisiert.

Durch die explizite Modellierung der sensorisch nicht erfassten Bereiche der Objektgeometrie im Objektmodell (Kapitel 2) ist es möglich diese Unsicherheit nicht nur bei der Bewertung der Stabilität (Kapitel 5.4.1) zu berücksichtigen, sondern auch direkt bei der Erzeugung ausführbarer Roboterbewegungen mit einfließen zu lassen.

Nachdem die Berechnung der translatorischen oder rotatorischen Bewegung die Unterscheidung in sensorisch validierte und nicht validierte Oberflächenbereiche nicht berücksichtigt, wird kein neuer Kontaktzustand berechnet, der nicht realisierbar ist, da die Objekte sich in der Realität durchdringen würden.

Nachdem bekannt ist, ob ein Kontaktpunkt auf einem sensorisch validierten Bereich (Kapitel 2.3.4) ist, kann diese Information verwendet werden, um eine Roboterbewegung über die berechnete Kontaktpose hinaus weiter zu führen. Das setzt natürlich voraus, dass der Roboter über zusätzliche Sensorik verfügt, die erkennt, dass eine Bewegung abgebrochen werden muss, da diese sensorbasiert in einem neuen vom Plan abweichenden Kontaktzustand geendet hat.

In dieser Situation kommt eine weitere Stärke des Algorithmus zum Tragen. Selbst wenn der aktuelle reale Kontaktzustand vom geplanten abweicht, kann der Ablageplaner weiterhin verwendet werden. Nachdem basierend auf dem aktuellen Kontaktzustand ausschließlich mit geometrischen Informationen entschieden wird, welche anschließende Bewegung den folgenden Kontaktzustand erzeugt, kann der Planer ausgehend von einem beliebigen Kontaktzustand aus gestartet werden.

Durch diese Eigenschaft ist eine wechselseitige Planung und Ausführung möglich. Diese kommt besonders zum Tragen, wenn die geometrische Rekonstruktion viele sensorisch nicht validierte Oberflächenbereiche enthält.

## 5.5 EXPERIMENTE UND DISKUSSION

Eine Benchmark-Problemstellung ist das Aufräumen eines (Schreib-)Tisches. Dieses Problem ist etwas unpräzise formuliert, da das Kriterium, wann ein Tisch aufgeräumt ist, nicht eindeutig ist. Daher wird zuerst ein Kriterium anhand der Lagebeziehungen eingeführt, wann die Objekte auf einem Tisch aufgeräumt sind. Das ist genau dann der Fall, wenn kleine Objekte in einen Aufbewahrungsgegenstand gelegt sind und die anderen Objekte gestapelt, gepackt oder nebeneinander gemustert nach gewissen Eigenschaften platziert sind. Daraus lässt sich ableiten, dass alle Lagebeziehungen aus Kapitel 5.1 verwendet werden müssen, um diese Aufgabe zu lösen.

In diesem Unterkapitel wird das Verfahren aus Kapitel 5.4 zur Planung von stabilen Ablageposen und den dazugehörigen Kontaktzustandssequenzen untersucht, welche nach ihren Lagebeziehungen unterteilten Ablageaufgaben gelöst werden können. Des Weiteren werden die Rechenzeit des Algorithmus und die Leistungsfähigkeit des Planers diskutiert.

Für die Experimente werden die aus Kapitel 4.5 bekannten Objekte verwendet. Zusätzlich dazu sind weitere Ablagen, wie ein liegender Aktenordner, hinzugekommen, um die Leistungsfähigkeit des Planers beim Stapeln zu analysieren. Insgesamt werden 68 Objekt-Ablage-Kombinationen evaluiert. Die folgenden Abbildungen zeigen beispielhaft einzelne Kombinationen.

Die Anzahl der Startposen wird dem Algorithmus vorgegeben. Nachdem davon ausgegangen wird, dass die Ablageplanung auf einem räumlich begrenzten statischen Objekt durchgeführt wird und der euklidische Abstand zwischen den Startpo-



sen abhängig von dem bewegten Objekt ist, verwendet der Ablageplaner zwischen einer und vier Startposen für die hier vorgestellten Szenarien. Für die Szenarien wird angegeben, wie viele Startposen zur Berechnung verwendet werden. Wenn mit einer Startpose keine stabile Ablage berechnet werden kann, dann wird die nächste Startpose herangezogen. Diese Linearisierung der Berechnung dient dazu, die Anzahl der nötigen Startposen zu evaluieren. Für eine optimale Laufzeit sollte auf diese sequenzielle Ausführung verzichtet werden und gleich mehrere Startposen gleichzeitig berechnet werden.

Die geringe Anzahl an Startpose ist insbesondere durch den Optimierungsschritt instabiler Posen während des Planungsvorgangs möglich. Nur bei komplexen Objekten, bei denen diese Optimierung keine signifikanten Verbesserungen erzielen kann, muss das statische Objekt dichter abgetastet werden.

In Abbildung 50 ist ein Ergebnis für die Lagebeziehung *ineinander* abgebildet. Sie zeigt eine sehr schwierige Ausgangssituation: Die Startpose des Objektes über dem Ablagepunkt liegt so, dass es nicht direkt in das Innere der Schüssel translattiert werden kann, sondern Kontakt mit deren Rand herstellt. Des Weiteren besitzt die Schüssel im Inneren keine ebene Fläche. Somit ist es nahezu unmöglich Objekte abzulegen ohne sie an die Schüsselwand zu lehnen. Dieses Szenario markiert die Leistungsfähigkeit des Ablageplaners. Wie in der Abbildung zu sehen ist, werden auch in solchen ungünstigen Ausgangssituationen stabile Ablagen berechnet. Für Objekte, die einer einfachen geometrischen Form genügen, werden auch in diesen Situationen valide Resultate erzeugt. Bei komplexen Geometrien, die nicht in das Innere der Schüssel passen, werden dagegen auch Ablageposen erzeugt, die voraussichtlich nicht stabil sind, da beispielsweise das Objekt ausschließlich auf dem Rand der Schüssel liegt. Bei einer leicht fehlerhaften Berechnung des Schwerpunktes wird das Objekt in der Realität kippen. Die durchgeführten Experimente zeigen, dass bei komplexen Ablagegeometrien der Ablageplaner mehr Startposen benötigt (in diesem Szenario vier), da instabile Ablageposen nur schlecht verbessert werden können. Wenn das Objekt vollständig auf dem Rand der Schüssel abgelegt ist, reicht eine einzelne translatorische oder rotatorische Bewegung nicht aus, um eine neue signifikant veränderte Kontaktpose zu berechnen und das Objekt in das Innere der Schüssel zu bewegen.

Bei der Lagebeziehung *aufeinander* müssen zwei Situationen unterschieden werden. Die eine Situation ist, dass das Objekt nicht auf das unterstützende Objekt

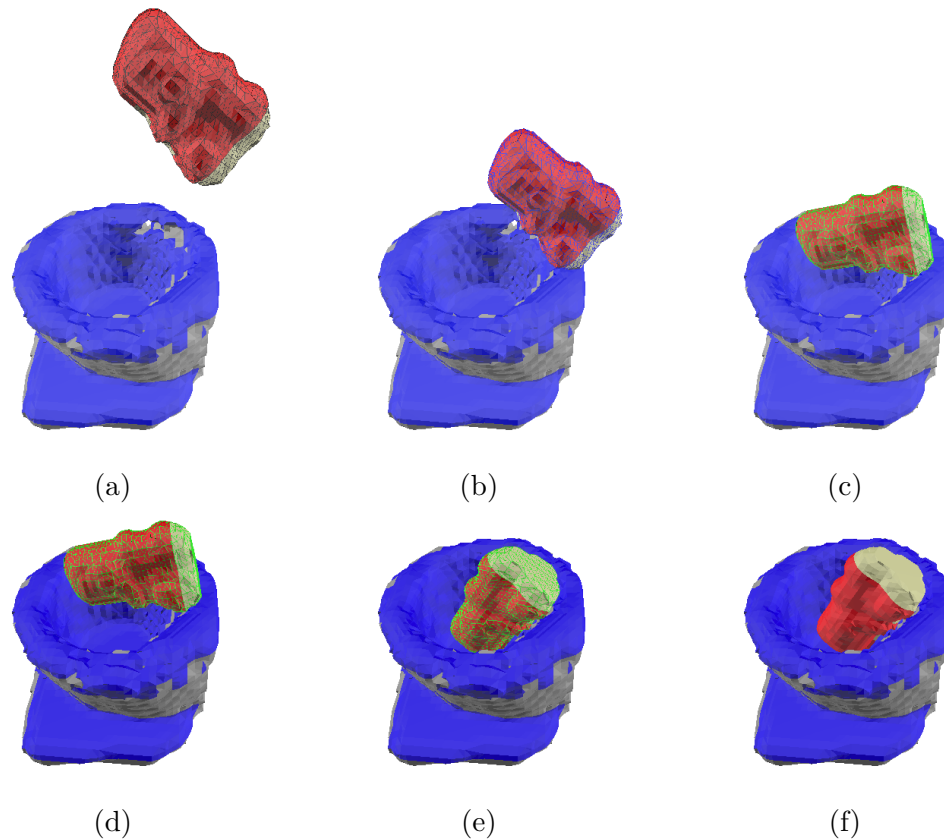


Abbildung 50: Ergebnissequenz der Kontaktzustände der Ablageplanung für die Lagebeziehung *ineinander*, angefangen von der Startpose (a) bis zur finalen Ablagepose (f). Die Farbe des Gittermodells verweist auf die zuvor ausgeführte Bewegung: Translation (blau), Rotation (grün) und Startpose (schwarz). Die sensorvalidierten Bereiche des Objektes (rot) und der Ablage (blau) sind farblich von den nicht validierten Bereichen (gelb-grau bzw. grau) unterschieden.

gestapelt werden kann. Die andere Situation ist, dass ein Stapeln prinzipiell möglich ist.

Die Abbildungen 51 und 52 zeigen zwei ausgewählte Resultate. Das unterstützende Objekt ist ein Aktenordner, der auf einem Tisch liegt. Das Objekt, eine Minigießkanne, soll auf diesem Ordner abgelegt werden. Wenn ausschließlich der Aktenordner rekonstruiert wird, würde in diesem Szenario keine Ablagepose berechnet werden, da auf dem Ordner selbst keine stabile Pose existiert. In den beiden

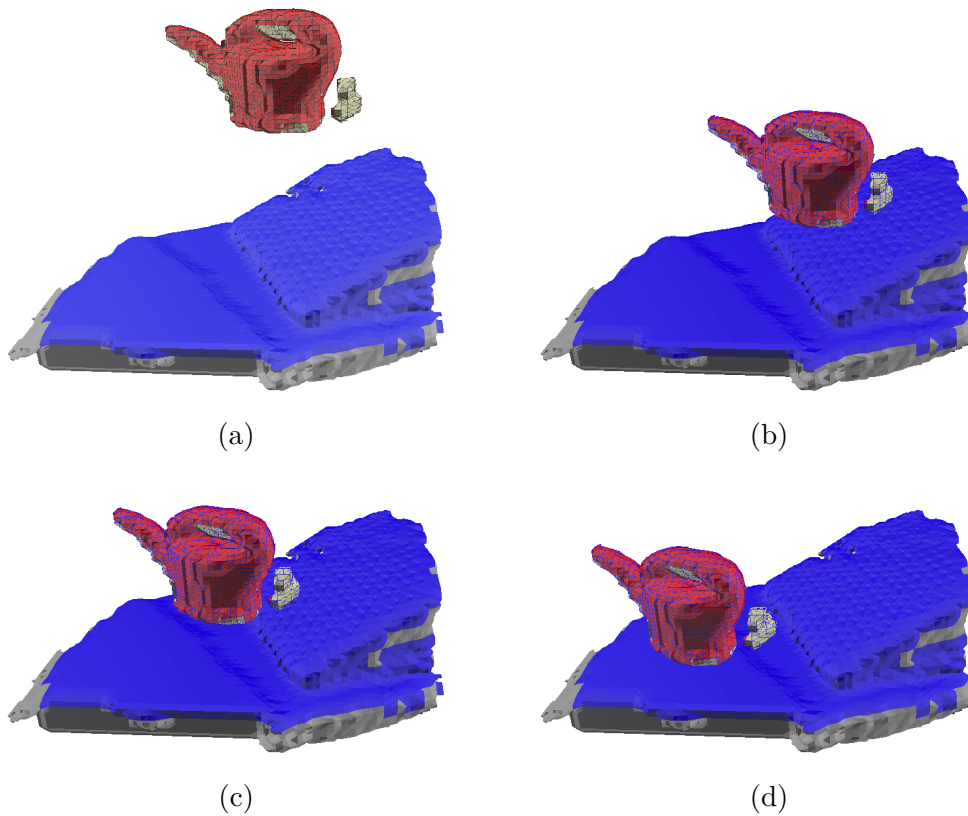


Abbildung 51: Ausgewählte Kontaktzustände der Ergebnissequenzen der Ablageplanung für die Lagebeziehung *aufeinander* bei einer Ablage auf die nicht gestapelt werden kann anfangen von der Startpose (a) bis zur finalen Ablagepose (d). Die Bedeutung der farblichen Markierung ist äquivalent zu Abbildung 50.

Abbildungen sind Ergebnisse für zwei unterschiedliche Startposen zu sehen. Bei deren Vergleich fällt auf, dass einmal die Kanne gekippt wird und einmal nicht. Ansonsten sind die Kontaktzustandssequenzen vergleichbar. Dieser Unterschied entsteht, wenn ein Scheinobjekt mit rekonstruiert wird, das zusätzlich noch einen kleinen sensorvalidierten Bereich aufweist, und durch numerische Ungenauigkeiten der Berechnung der Kontaktpunkte. Daraus folgen unterschiedliche Ergebnisse bei der Entscheidung, ob es sich um ein Versagen der Reibung oder des Kräftegleichgewichts handelt. Gerade beim initialen Kontakt zwischen Objekt und Ablage ist die Anzahl der Kontaktpunkte gering. Dadurch beeinflussen einzelne fehlerhafte Kon-

taktnormalen die Entscheidung des Versagenstyps der Stabilitätsbewertung. Im Rahmen der Experimente ist es jedoch nicht vorgekommen, dass dieser Umstand zu einer fehlerhaften Stabilitätsaussage geführt hat.

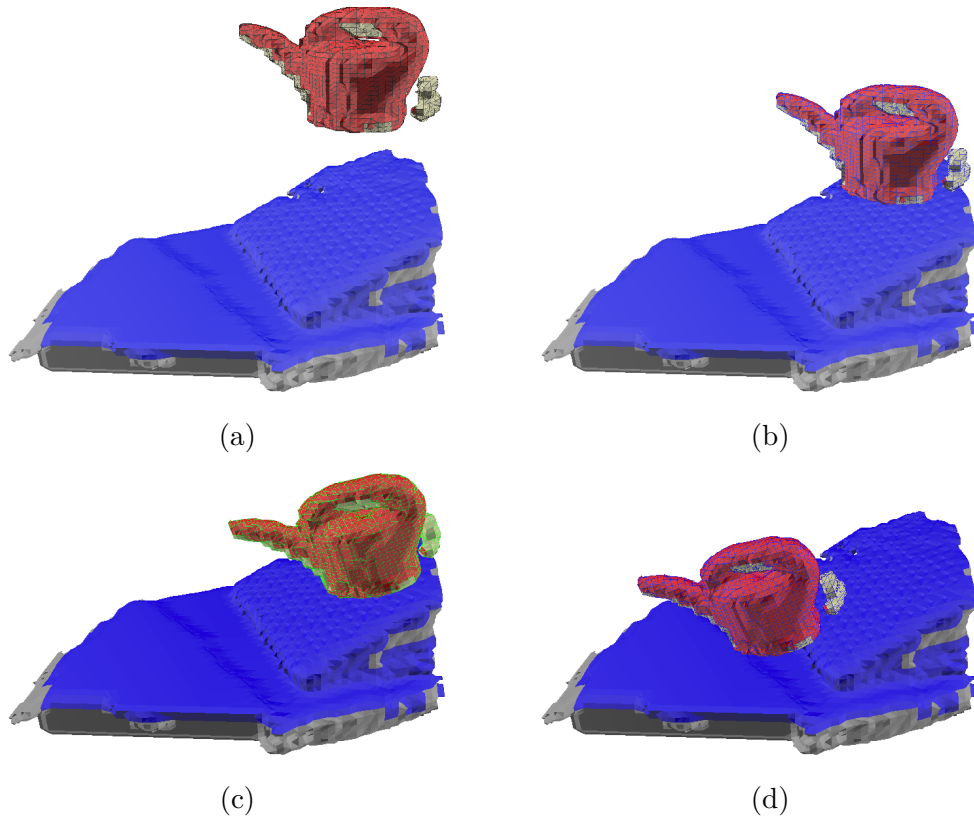


Abbildung 52: Alternatives Ergebnis für eine andere Startpose wie in Abbildung 51

In Abbildung 53 ist beispielhaft ein Szenario abgebildet, in dem es möglich ist Objekte zu stapeln. In diesem Fall wird das Objekt stabil abgelegt. Hier ist absichtlich das Objekt nicht gemäß des Algorithmus vororientiert, sondern in einer Startpose, die es ermöglicht das Objekt durch eine rotatorische Bewegung von dem unterstützenden Objekt herunter zu bewegen. Wie das Resultat zeigt, wird die stabile Ablagepose auf dem unterstützenden Objekt berechnet und kein falsch positives Resultat erzeugt.

Festzuhalten ist, dass wenn ein Objekt nicht auf das vorgesehene Objekt abgelegt werden kann, der Ablageplaner dennoch stabile Ablageposen erzeugt, die allerdings

nicht mehr der geforderten Lagebeziehung entsprechen. Gerade das Optimieren instabiler Posen hat zur Folge, dass die Anzahl der Startposen klein gehalten werden kann. Für die beiden Stapelszenarien werden nur zwei Startpose benötigt.

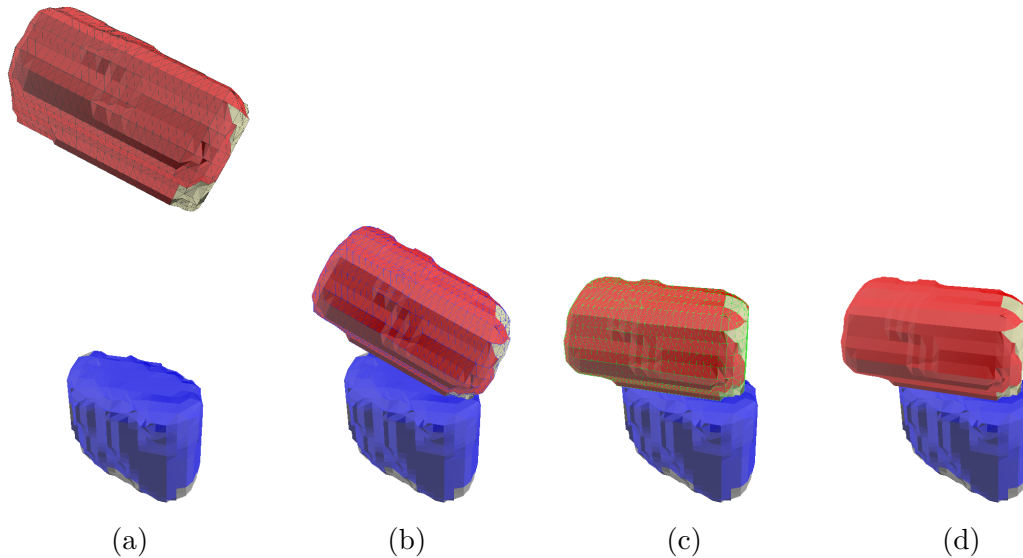


Abbildung 53: Ergebnissequenz der Kontaktzustände der Ablageplanung für die Lagebeziehung *aufeinander*, angefangen von der Startpose (a) bis zur finalen Ablagepose (d). Die Bedeutung der farblichen Markierung ist äquivalent zu Abbildung 50.

Das Anordnen auf ebenen horizontalen Ablagen ist die wohl einfachste Problemstellung bei der Planung stabiler Ablageposen. In Abbildung 54 ist ein Beispiel abgebildet. Die beiden Objekte sind anhand ihrer Hauptachsen an denen der Ablage ausgerichtet und die Schwerpunkte der Objekte liegen auf der x-Achse des Koordinatensystems, dass durch die Hauptachsen der Ablage definiert ist. Insbesondere ist festzustellen, dass der Planer bei allen Testläufen mit ebenen Ablagen unabhängig von der initialen Orientierung des Objektes immer stabile Ablageposen berechnet.

Bei der Aufgabe Objekte *anzuordnen* kann man ohne semantische Informationen über die Objekte oder weiterführende Heuristiken nur einfache Regeln verwenden. Nachdem der vorgestellte Ablageplaner zuverlässig stabile Posen für die unterschiedlichen Objekte berechnet, besteht an dieser Stelle keine Einschränkung der

Leistungsfähigkeit des Planers. Vielmehr ist es nötig zusätzlich, unter Umständen aufgabenspezifische, Informationen mit einzubeziehen.

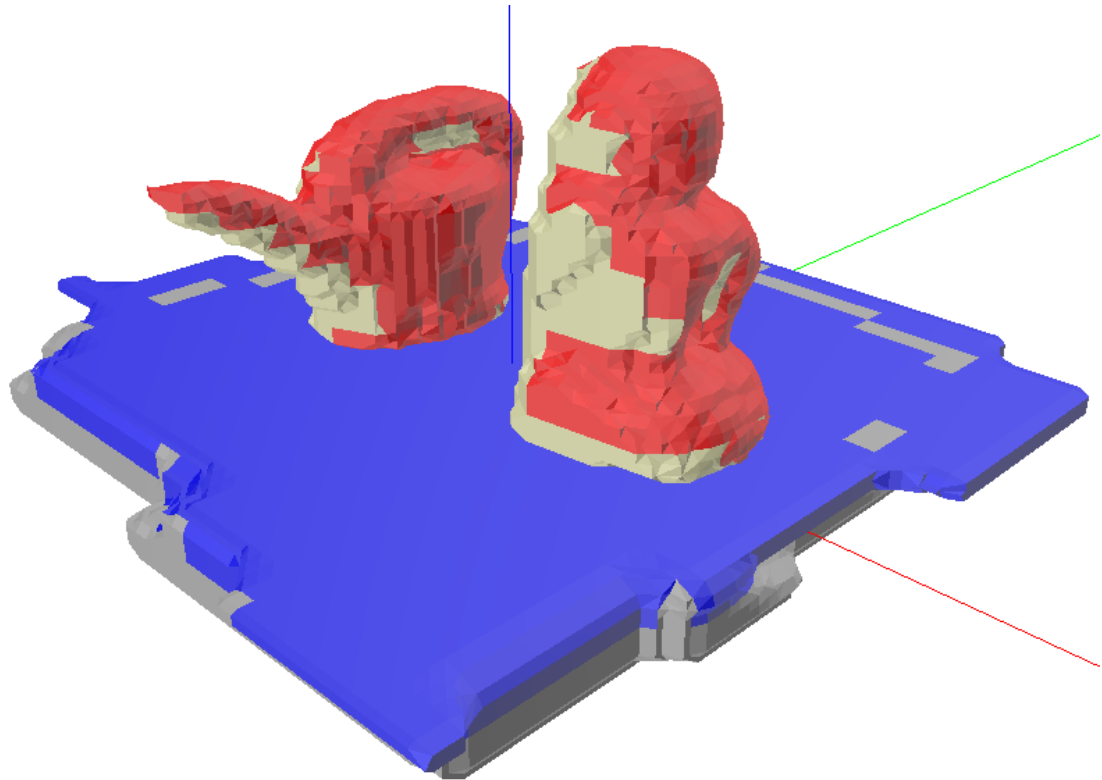


Abbildung 54: Die Szene zeigt die Platzierung zweier Objekte, die an den Hauptachsen der Ablage ausgerichtet sind. Die Bedeutung der farblichen Markierung ist äquivalent zu Abbildung 50.

Die Kombination der gezeigten Lagebeziehungen ermöglicht es grundsätzlich Objekte zu packen. Ebenso wie bei dem Anordnen von Objekten muss neben dem Maß für die Stabilität der Ablage auch noch ein zusätzliches Maß für das Online-packproblem verwendet werden. Insbesondere müssen weitere Kriterien gefunden werden, die die Ausführbarkeit der Kontaktsequenz mit dem Roboter beschreiben. Beispielsweise muss das Objekt unter Umständen geschoben werden, da es nicht möglich ist es gegriffen in die Zielpose zu bewegen.

Die Art und Weise wie die Startposen und wie die initiale Bewegungsrichtung während der Ablageplanung festgelegt sind schließt aus, dass Objekte beispielswei-

se an ein anderes Objekt gehängt werden. Demnach ist die Lagebeziehung *hängen* mit dem Ablageplaner in der Form nicht möglich.

Es lassen sich folgende Grenzen des Ablageplaners formulieren: Je komplexer die Geometrie der Ablage wird, desto mehr Startposen müssen bei der Planung berücksichtigt werden. Gerade in dem Szenario mit der Schüssel als Ablage ist es notwendig, dass eine Startpose existiert, die es ermöglicht das Objekt in das Innere der Schüssel zu translatieren. Nur in dem Fall werden zuverlässige Ablageposen berechnet. Wenn keine solche Startpose existiert, kommt es zu der Berechnung von augenscheinlich labilen Ablageposen. Bei diesen ist zu beobachten, dass sie in deren Kontaktzustandsübergängen viele kleine Rotationen beinhalten, der Konfigurationsbaum demnach sehr tief ist. Falls solche Konfigurationen nicht vorher abgebrochen werden, bleibt die Auswahl der Ablagepose anhand der Stabilitätsbewertung. In allen Experimenten ist nie der Fall aufgetreten, dass eine augenscheinlich stabilere Konfiguration schlechter bewertet wird. Beispielsweise ist die Ablage in Abbildung 51 besser bewertet als die aus Abbildung 52.

Falls die Objektmodelle große nicht sensorvalidierte Bereiche aufweisen, beispielsweise wenn diese nur durch eine ungünstige Kameraperspektive rekonstruiert werden, ist es dem Ablageplaner nicht möglich eine Ablagepose zu berechnen. Für das Modell der Ablage ist gleiches zu beobachten. Nachdem der Planer die Objekte in Richtung der Gravitation ablegt, ist es ausreichend die Ablage auf Basis einer einzigen Perspektive zu rekonstruieren. Bis auf Abbildung 50 werden die Ablagen auf Basis einer Kameraperspektive in Richtung der Gravitation zeigend und die Objekte aus zwei gegenüberliegenden Perspektiven rekonstruiert.

Die Festlegung des Reibungskoeffizienten mit 0,2 ist zusätzlich eine Fehlerquelle. Wenn nämlich die tatsächliche Reibung kleiner ist als angenommen, verharrt das abgelegte Objekt nicht in der berechneten Pose. Im Gegenzug ermöglicht die Berücksichtigung der Reibung es Ablageposen zu berechnen, die nur unter Berücksichtigung dieser stabil sind. Unter der ausschließlichen Verwendung der Kräfte in der Stabilitätsheuristik ist es in dem Szenario *ein Objekt in die Schüssel zu legen* nicht möglich stabile Ablageposen im Inneren der Schüssel zu berechnen.

In Abbildung 55 sind die mittleren Rechenzeiten<sup>2</sup> der Einzelkomponenten und der gesamten Planung dargestellt. Für jeden Planungsdurchlauf werden zwei Start-

<sup>2</sup> Die Experimente sind auf einem Computer mit einem 22 nm "Ivy Bridge" 2.9 GHz Intel "Core i7" Prozessor (3520M) und 8 GB RAM durchgeführt worden.

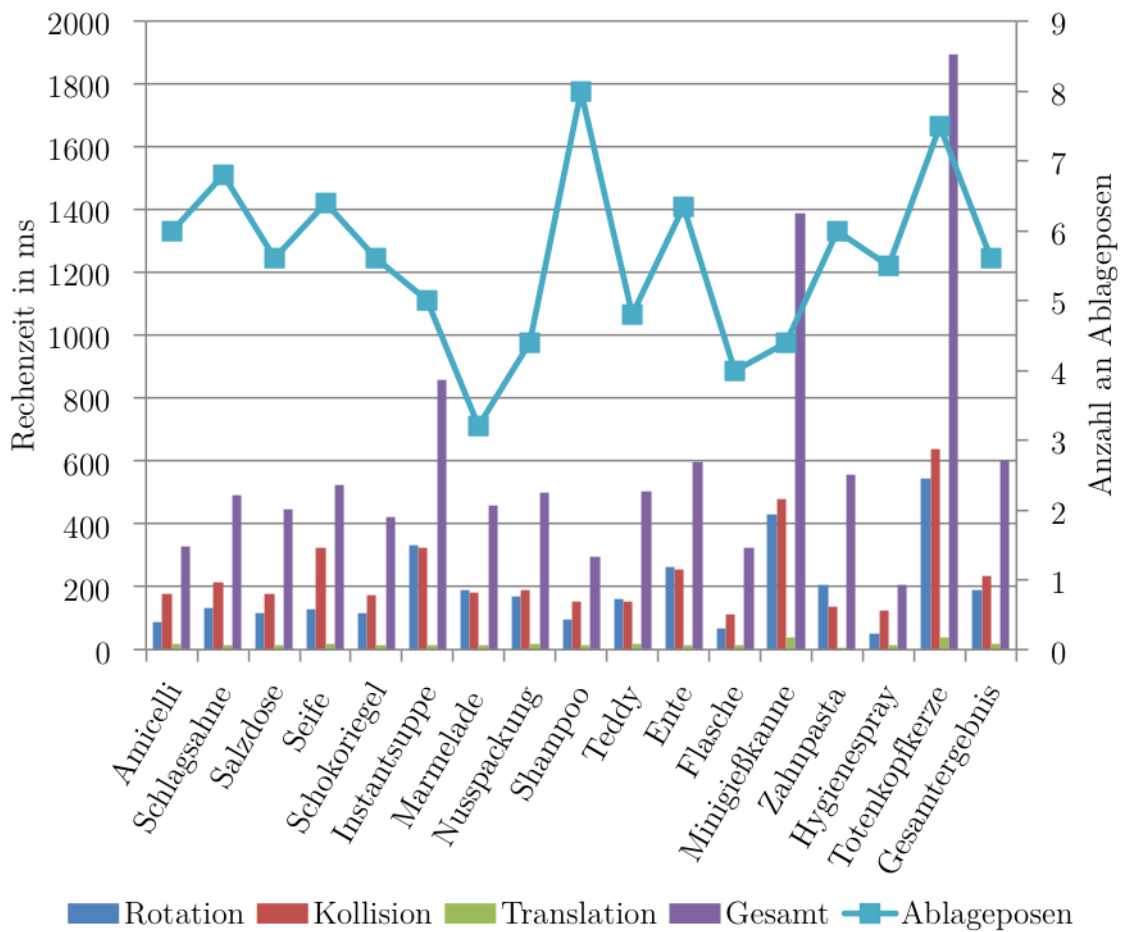


Abbildung 55: Mittelere Rechenzeiten und mittlere Anzahl an berechneten Ablageposen für die unterschiedlichen Testobjekte einschließlich dem Mittelwert über alle Testläufe

posen mit fünf zufälligen Orientierungen und vier verschiedene Ablageobjekte verwendet (vgl. Abbildungen 50, 51, 53 und 54). Die Objektmodelle und Modelle der Ablage sind mit den äquivalenten Rahmenbedingungen wie in Kapitel 4.5.2 mit dem Rekonstruktionsverfahren aus Kapitel 2 erstellt. Für jede Startpose werden alle möglichen Konfigurationen berechnet. Insbesondere wird die Berechnung nicht abgebrochen, sobald eine Ablagepose gefunden ist. Die maximal erlaubte Tiefe des Konfigurationsbaumes ist mit  $t_{T_c} = 12$  festgelegt. Damit sind theoretisch 4096 Ablageposen berechenbar. Aus der Abbildung ist abzulesen, dass die Anzahl berechneter Ablageposen zwischen drei und acht liegt. Dieser starke Unterschied



zu der möglichen Anzahl entsteht, wenn sich der neu berechnete Kontaktzustand nicht vom vorhergehenden unterscheidet und daher dieser Zweig des Konfigurationsbaumes nicht weiter verfolgt wird.

Die Rechenzeit des Ablageplaners pro Startpose ist dominiert von der Anzahl an Rotationsberechnungen (Kapitel 3) und der Berechnung der Kontaktpunkte. Die Rechenzeiten für die Bestimmung einer linearen Kontaktpose und die Stabilitätsbewertung sind vernachlässigbar.

Die Ergebnisse zeigen, dass Objekte mit einer primitiven Geometrie (Kapitel 4.5) annähernd gleiche Rechenzeit benötigen. Die komplexen Objekte benötigen eine deutlich höhere Rechenzeit, weil die Anzahl der Oberflächendreiecke fast doppelt so hoch ist wie bei den anderen Objekten (ca. 2000 zu 4000 Dreiecke).

Zusammenfassend ist festzustellen, dass der Ablageplaner in durchschnittlich 600ms 5,5 Ablageposen berechnet. Damit erfüllt er die Anforderungen bezüglich der Rechenzeit aus Kapitel 1.2. Dieser geometrische Planer ermöglicht es unbekannte Objekte auf unbekannte Ablagen abzulegen, ohne dass zusätzliche Informationen über die Objektdynamik verwendet werden.

## 5.6 SCHLUSSFOLGERUNG

In diesem Kapitel wird ein Ablageplaner vorgestellt, der ausschließlich auf rekonstruierten Oberflächenmodellen und Ablagen ohne Annahmen oder Informationen bezüglich der Objektdynamik und unter Berücksichtigung der Unsicherheiten bezüglich der rekonstruierten Objektmodelle arbeitet. Aus diesem Grund ist eine Heuristik entwickelt worden, die basierend auf dem ebenen Fall eine Objektpose ausschließlich basierend auf der Lage des Schwerpunktes, der Kontaktpunkte und -normalen und der Richtung der Gravitation bewertet. Zusätzlich zu einer stabilen Objektpose wird resultierend aus rotatorischen und translatorischen Bewegungen des Objektes eine Kontaktzustandssequenz berechnet. Diese Bewegungen können direkt vom Roboter ausgeführt werden. Damit ist die Notwendigkeit einer nachgelagerten Feinplanung nicht gegeben.

Der Ablageplaner ermöglicht es insbesondere Objekte auf nicht ebenen Ablagen zu platzieren, Objekte aufeinander zu stapeln und ineinander zu legen. Aus der Kombination dieser Ablageformen ist es möglich Objekte in eine Kiste zu packen beziehungsweise Objekte dicht aneinander auf eine horizontale Fläche zu stellen.

Mit dem entwickelten Ablageplaner ist es nicht möglich Objekte an anderen Objekten aufzuhängen.

Der Erfolg der Planung hängt maßgeblich von der initialen Orientierung des Objektes und der Auswahl der initialen Ablagepunkte ab. Diese werden mit einfachen Heuristiken unter anderem basierend auf den Hauptachsen des Objektes bestimmt. An dieser Stelle bietet sich die Möglichkeit die Objektfunktionalität mit einfließen zu lassen. Damit wäre es möglich Ablagen für unbekannte Objekte zu planen die nicht nur stabil sind, sondern auch der Objektfunktionalität genügen. Aus der Objektfunktionalität und der Ablage ließe sich eine initiale Orientierung und auch mögliche Lagebeziehungen ermitteln.

Im Bezug auf die Lagebeziehung lässt sich der Planer in Zukunft erweitern, so dass bei einer Bewertung der Stabilität einer möglichen Objektanlage auch bewertet wird inwieweit die geforderte Lagebeziehung erreicht ist. Eine heuristische Bewertung könnte regelbasiert auf Basis der verfügbaren geometrischen Informationen und Kontaktzustände erfolgen.

Mit der erzeugten Kontaktzustandssequenz zu einer Ablagepose besteht die Möglichkeit den Algorithmus zukünftig mit einer sensorüberwachten Ausführung zu kombinieren. Damit wäre es möglich Fehler bei der Ausführung zu erkennen und basierend auf dem ungeplanten Kontaktzustand erneut eine Ablageplanung durchzuführen, damit eine neue Zielpose geplant werden kann. Diese Erweiterung würde eine fehlertolerante Ausführung ermöglichen (vgl. Kapitel 1.1).

# 6

## PROTOTYP

In den vorangegangenen Kapiteln werden die entwickelten Komponenten zur Greifplanung (Kapitel 4) und Ablageplanung (Kapitel 5) unbekannter Objekte basierend auf einer Objektrekonstruktion aus mehreren Tiefenbildern (Kapitel 2) vorgestellt. In diesem Kapitel werden diese an einem prototypischen Personal Roboter getestet, der unbekannte Objekte aufgreifen soll und diese in unterschiedlichen Szenarien auf unbekanntem Ablagen platzieren soll.

Dazu stellt Kapitel 6.1 die verwendete Hardware vor. In Kapitel 6.2 wird die Reihenfolge der Berechnungen und ausgeführten Bewegungen diskutiert. Die durchgeführten Experimente werden in Kapitel 6.3 beschrieben und deren Ergebnisse interpretiert. In Kapitel 6.4 werden Schlussfolgerungen gezogen.

### 6.1 HARDWAREAUFBAU

Die Basis des verwendeten Systems ist der KUKA LWR 4+ Manipulator [KUK15]. Dieser verfügt über sieben Freiheitsgrade und besitzt durch seine Impedanzregelung eine gewisse Nachgiebigkeit. Weiter verfügt dieser Manipulator über Momentensensoren in jedem Gelenk, deren Informationen in diesem Aufbau nicht aktiv verwendet werden, da evaluiert werden soll, inwieweit geometrische Informationen ausreichen unbekannte Objekte zu manipulieren (Kapitel 1.2), ohne dass Kraft- oder Momenten-Informationen verwendet werden.

Als Mehrfingergreifer ist ein Robotiq Adaptive Gripper [Rob14] mit drei Fingern montiert. Dieser verfügt über ein Scheren-Nebengelenk (Kapitel 4.1) und drei Hauptgelenke pro Finger. Das System steuert die Aktuatoren des Greifers so an, dass die Finger möglichst synchron Kontakt mit dem Objekt herstellen. Um diese Funktionalität zu realisieren, ist experimentell bestimmt worden, wie sich die Aktuatorstellungen auf die Gelenkwinkel der passiven Gelenke auswirken. Dieser funktionelle Zusammenhang ermöglicht es die Geschwindigkeit und Zielstellung je-

des Fingermotors auf Basis der Gelenkwinkel zu berechnen, die vom Greifplaner bestimmt worden sind.

An dem Handgelenk des Manipulators ist zusätzlich eine Ensenso N10 Stereokamera [IDS] montiert, mit der die Objektrekonstruktion vorgenommen wird. Die Kamera ist auf das Tool-Center-Koordinatensystem des Roboters kalibriert.

Nachdem die Kamera direkt am Roboter befestigt ist, bleibt der Roboter transportabel und ist nicht auf einen zusätzlichen externen Aufbau angewiesen. Diese Kompaktheit des Aufbaus wiegt stärker als der Nachteil, dass sich der Roboter zum Rekonstruieren seines Arbeitsraumes aktiv bewegen muss.

## 6.2 SYSTEMARCHITEKTUR

Die verschiedenen Planungsalgorithmen und die Ausführung deren Resultate durch den Roboter besitzen gegenseitige Abhängigkeiten. Basierend auf den verwendeten Softwarekomponenten werden diese Abhängigkeiten im Folgenden diskutiert. Zusätzlich zu der Objektrekonstruktion, dem Greifplaner und Ablageplaner wird ein Bahnplaner [Bag98] verwendet, um kollisionsfreie Bahnen zu berechnen.

Die Annahmen bezüglich der Umwelt sind wie folgt: Es sind zwei disjunkte Raumvolumina gegeben, in denen der Rekonstruktionsalgorithmus operiert. Innerhalb des ersten Volumens werden die Objekte rekonstruiert, die gegriffen werden sollen. Dieses wird zukünftig *Aufgreifbereich* genannt. Innerhalb des zweiten Volumens, dem *Ablagebereich*, sollen die Objekte abgelegt werden. Zusätzlich wird angenommen, dass der Roboter auf einem Tisch montiert ist. Daher ist der Arbeitsraum so beschränkt, dass der Roboter nur oberhalb der Tischebene operiert. Es wird angenommen, dass außerhalb dieser beiden Volumina keine weiteren Hindernisse existieren und der Roboter sich außerhalb derer frei bewegen kann.

Um die Rekonstruktionsvolumen mit der Stereokamera aufzunehmen, muss der Roboter verschiedenen Kameraposen anfahren. Aufgrund der Annahmen über die Umwelt können diese Bewegungen immer ausgeführt werden, wenn die Pose innerhalb des Arbeitsraums liegt, ohne dass der Roboter in eines der Volumen eindringen muss.

Sobald die Tiefenbilder für den Aufgreifbereich verfügbar sind, können die Objekte rekonstruiert werden und die Greifplanung ausgeführt werden. Erst wenn zusätzlich die Tiefenbilder für den Ablagebereich und die Rekonstruktion der Ab-

lage zu den rekonstruierten Objekten vorliegen, kann die Ablageplanung beginnen. Sobald für ein Objekt die Greifplanung und auch die Ablageplanung abgeschlossen sind, kann eine kollisionsfreie Bahn geplant werden und diese auch ausgeführt werden.

Für die Abfolge dieser einzelnen Schritte existieren drei mögliche Varianten. Alle anderen Kombinationen schließen sich auf Grund der oben beschriebenen Abhängigkeiten zwischen Rekonstruktion und Planung aus. Die Möglichkeiten sind in Abbildung 56 illustriert. In der Variante aus Abbildung 56a kann die Reihenfolge der Rekonstruktionen getauscht werden. Daraus ergibt sich die zweite Variante (Abbildung 56b). Diese ist jedoch nicht in Betracht zu ziehen, da man nicht auf das Ergebnis der Ablagerekonstruktion warten muss, um mit der Greifplanung zu beginnen.

Wenn der Ablagebereich zuerst rekonstruiert wird, ist es nicht möglich einen der Planer auszuführen, solange der Roboter die Kameraposen anfährt, um den Aufgreifbereich zu rekonstruieren (Abbildung 56a). Die Greif- beziehungsweise Ablageplanung kann erst ausgeführt werden, wenn auch die Objektmodelle innerhalb des Aufgreifbereichs rekonstruiert sind. Die beiden Algorithmen müssen jetzt parallel ausgeführt werden, wobei der Roboter auf die Planungsergebnisse warten muss bis ihm ein Ziel zur Verfügung steht, das er anfahren soll. Nachdem die Rekonstruktion des Aufgreifbereichs abgeschlossen ist, muss wenn überhaupt nur eine kurze Bewegung ausgeführt werden, bevor die geplante Bahn zum Aufgreifen eines Objektes abgefahren wird. Diese Abfolge entspricht der klassischen *sense-plan-act* Architektur.

Wenn zuerst der Aufgreifbereich rekonstruiert wird (Abbildung 56c), findet die Greifplanung für das Objekt und die Rekonstruktion des Ablagebereichs mit den nötigen Roboterbewegungen zeitgleich statt. Diese Reihenfolge hat mehrere Vorteile. Zum einen wird der Greifplaner zum frühest möglichen Zeitpunkt gestartet. Daraus ergibt sich, dass die Roboterbewegungen für die Rekonstruktion des Ablagebereichs zeitgleich zur Greifplanung erfolgt. Weiter ergibt sich, dass die Bahnplanung zum Erreichen der Greifpose schon während der Ablageplanung begonnen werden kann, wobei nach dem das endgültige Ergebnis der Ablageplanung vorliegt, diese geplante Bewegung zum Greifen des Objektes angepasst werden muss. Zum Anderen kann die Ablageplanung schon berechnete Griffe als Nebenbedingungen berücksichtigen.

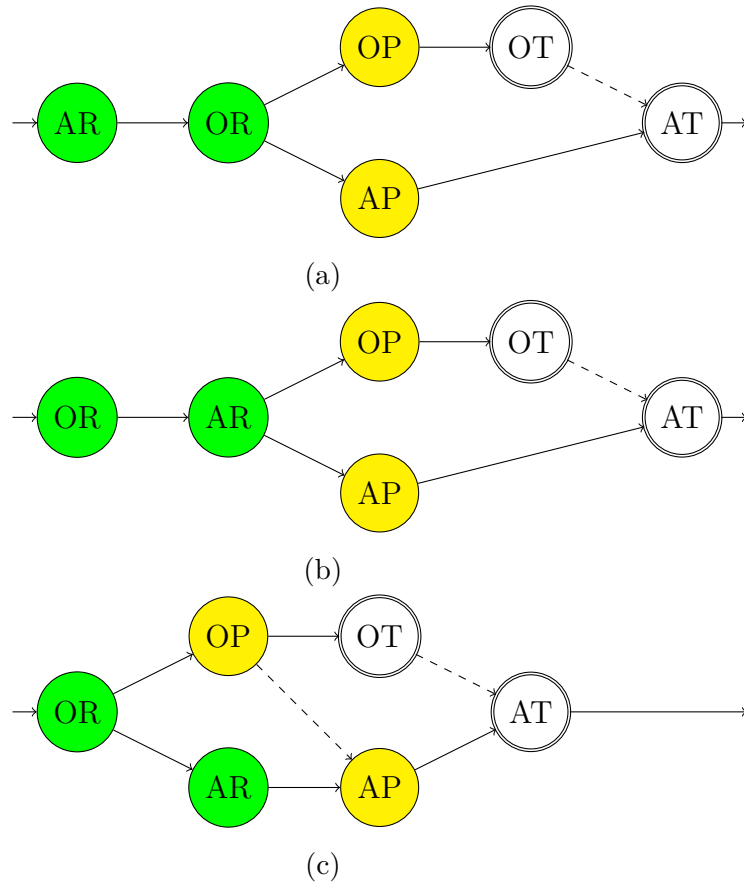


Abbildung 56: Illustration der möglichen Varianten an Reihenfolgen der Rekonstruktion des Aufgreifbereichs (OR) und Ablagebereichs (AR), der Greifplanung (OP) und Ablageplanung (AP), sowie der Bahnplanung zum Greifen des Objektes (OT) und dessen Ablage (AT). Die gestrichelten Pfeile bedeuten, dass das Ergebnis als Nebenbedingung berücksichtigt wird. Am Ende der doppelt umrandeten Berechnungen kann der Roboter beginnen die dynamisch berechneten Bewegungen zum Greifen beziehungsweise Ablagen des Objektes beginnen.

Nachdem in der dritten Variante (Abbildung 56c), bei der zuerst der Aufgreifbereich rekonstruiert wird, eine Parallelisierung der Roboterbewegungen und Ausführung der Planer ermöglicht, wird die Abfolge der einzelnen Komponenten gemäß dieser festgelegt.

### 6.3 EXPERIMENTE UND DISKUSSION

Mit dem beschriebenen Hardwareaufbau und der gewählten Systemarchitektur werden verschiedene Greif- und Ablageszenarien basierend auf der Taxonomie aus Kapitel 5.1 untersucht. Diese Szenarien werden in aufsteigender Schwierigkeit vorgestellt. Im Aufgreifbereich sind immer ein oder mehrere Objekte separiert platziert. Für den Ablagebereich werden zwei Startposen während der Ablageplanung verwendet. Diese werden wie in Kapitel 5.4.2 berechnet, wobei die initialen Punkte für die Berechnung der Startposen den Knoten des Voronoi-Graphs der Oberseite der Boundingbox bezüglich der z-Achse des Ablagebereichs entsprechen. Durch diese Einschränkung ist es möglich während den Experimenten verschiedene Extremsituationen zu erzeugen, um die Leistungsfähigkeit der Planer besser austesten zu können. Je nach Szenario unterscheiden sich die verwendeten Objekte, die im Ablagebereich platziert sind. Es wird vorausgesetzt, dass die Anordnung der Objekte im Ablagebereich mindestens so stabil ist, dass die übrigen Objekte nicht durch das Eigengewicht des abgelegten Objektes verrutschen.

Im Folgenden werden die einzelnen Szenarien vorgestellt und anschließend diskutiert, ob die Anforderungen aus Kapitel 1.2 erfüllt sind und welche Fehler aufgetreten sind.

Im leichtesten Szenario sollen die Objekte gegriffen werden und vereinzelt nebeneinander im Ablagebereich platziert werden. Die Ablage ist anfangs ein leerer Tisch. In Abbildung 57 ist beispielhaft eine Bildersequenz mit dem zugehörigen Planungsergebnis zu sehen, in der eine Schüssel wie geplant manipuliert wird. Nachdem diese nur am Rand gegriffen werden kann, muss der Griff während der Roboterbewegung starken externen Momenten standhalten. Die abgebildete Sequenz zeigt, dass dies gelingt. Wenn die Schüssel während der Bewegung verrutscht wäre, könnte die Ablage nicht mehr wie geplant durchgeführt werden.

Im nächst schwierigeren Szenario sollen alle Objekte innerhalb des Aufgreifbereichs aufeinander gestapelt werden. Die Objekte sind so gewählt, dass sie in jeder

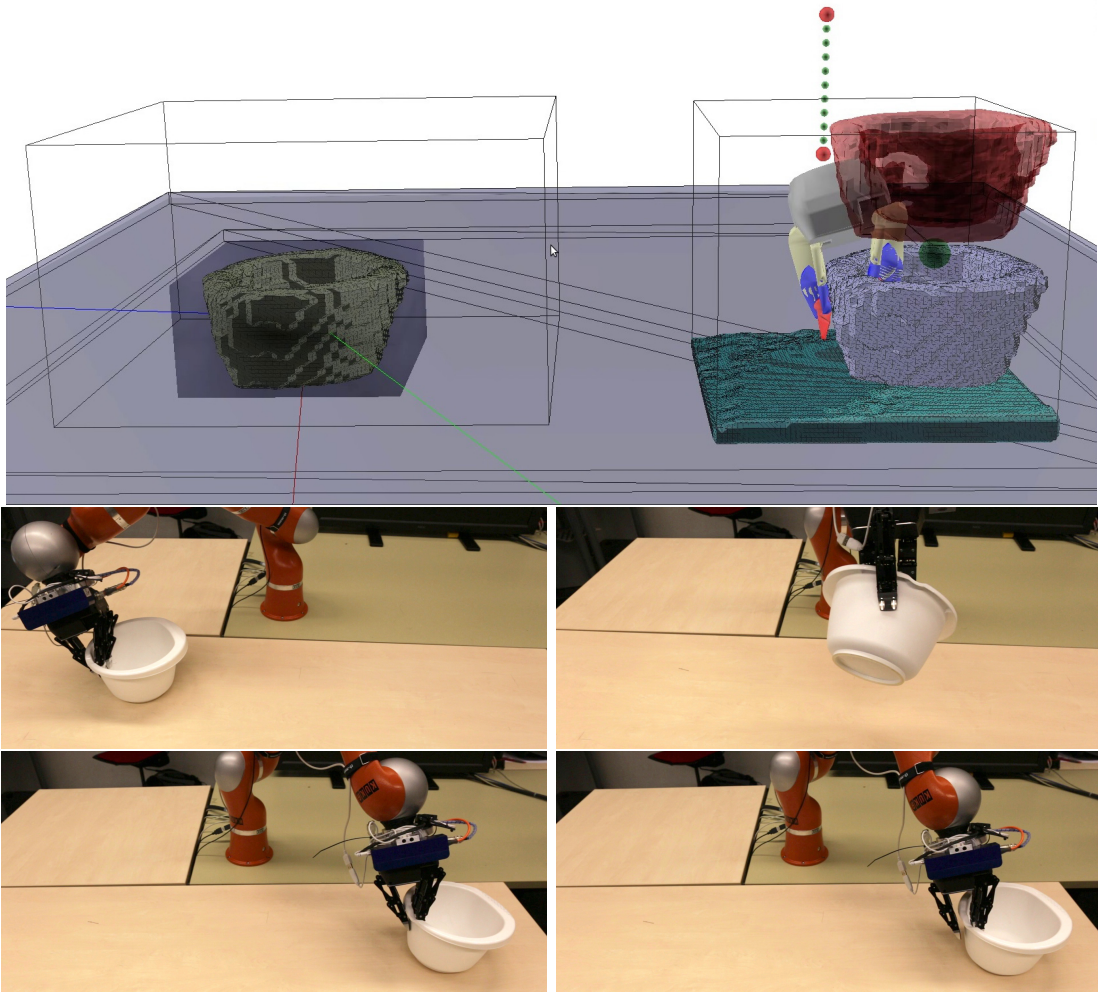


Abbildung 57: Erstes Szenario: Anordnen von Objekten auf einem Tisch. In der 3D-Ansicht sind die Rekonstruktionsvolumen inklusive der rekonstruierten Geometrien eingezeichnet. Die berechnete Ablagepose und alle Kontaktzustände angefangen bei der Startpose (rot) sowie der berechnete Griff sind zusätzlich dargestellt. Die berechnete Feinbewegung, um die Ablage durchzuführen, ist gepunktet dargestellt. Die roten Punkte markieren einen Kontaktzustand der vom Ablageplaner berechnet worden ist.

Kombination gestapelt werden können. Wie im vorherigen Szenario beinhaltet die Ablagefläche anfangs ausschließlich einen leeren Tisch. Abbildung 58 zeigt eine



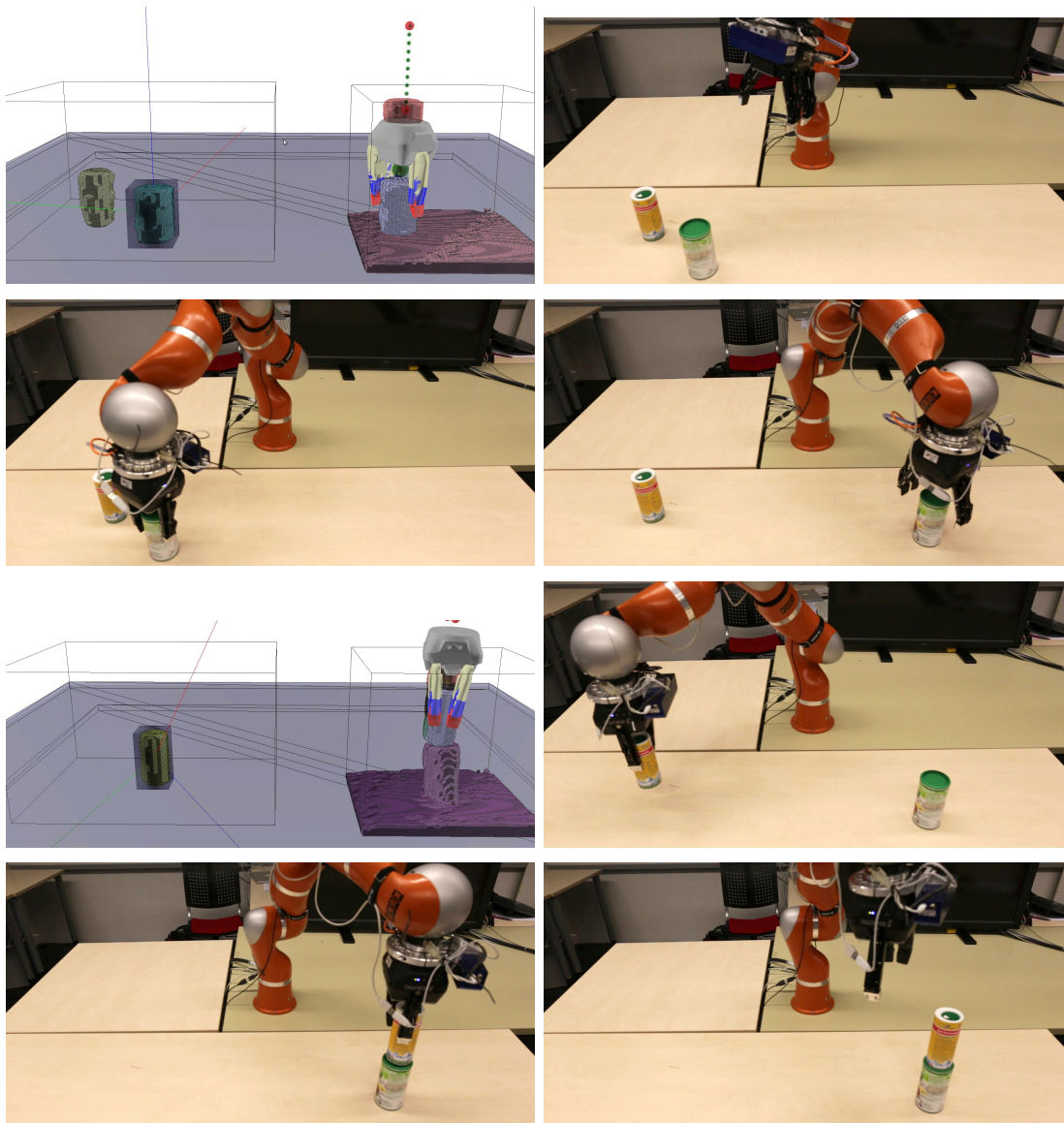


Abbildung 58: Zweites Szenario: Stapeln von Objekten auf einem Tisch. Die Erklärung der 3D-Ansicht ist Abbildung 57 zu entnehmen.

bildliche Abfolge eines Testlaufs dieses Szenarios. Im Gegensatz zu dem vorherigen Szenario ist es hier unbedingt notwendig, dass die Objekte präzise gegriffen werden, da sich jede Ungenauigkeit unmittelbar auf den Erfolg der Ablage auswirkt. Falls der Griff nicht exakt mit der Planung übereinstimmt, wird der Ablagevorgang unweigerlich scheitern.

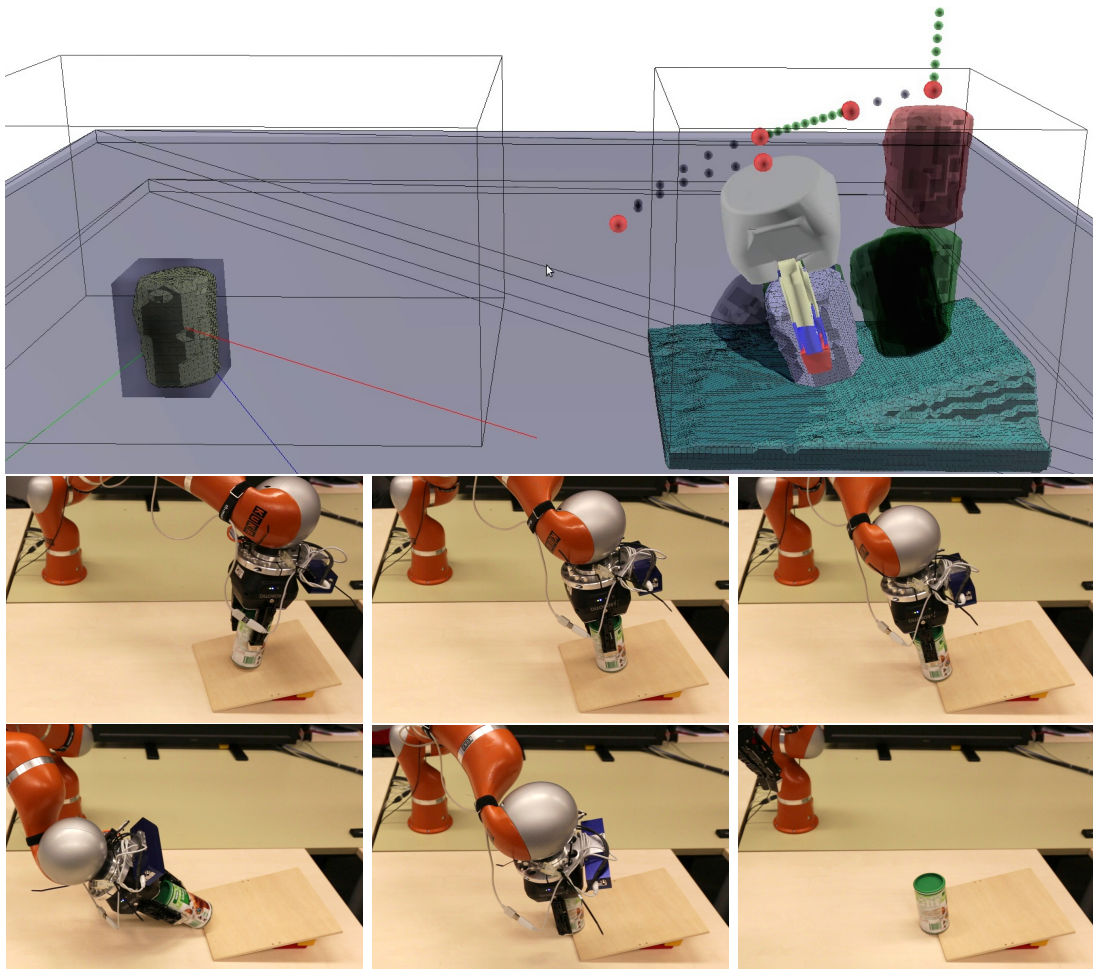


Abbildung 59: Drittes Szenario: Stapeln von nicht stapelbaren Objekten. Die Erklärung der 3D-Ansicht ist Abbildung 57 zu entnehmen.

Das nächste Szenario ist wieder eine Stapelaufgabe, die sich von der vorherigen dadurch unterscheidet, dass das Objekt auf der Ablage, auf welches das gegriffene Objekt gestapelt werden soll, keine stabile Ablagepose bietet. Abbildung 59 zeigt ein Beispiel, wie das Objekt aufgesetzt und dann, so gedreht wird, dass die Kontaktfläche zur Ablage vergrößert wird. Die anschließende Bewegung, die das Objekt Richtung Tisch führt, erfolgt mit minimalem Kontakt zur Ablage. Diese besteht aus einem leichten Holzbrett, das locker auf einigen glatten Bauklötzen liegt. Dieser instabile Aufbau zeigt, dass der Ablageplaner zusammen mit der Rekonstruktion so exakt zusammenarbeitet, dass es selbst bei Positionsfahrten durch

den Roboter auch ohne Kraftregelung zu keinen ungewollten Veränderungen der Umwelt kommt.

Das schwierigste Szenario deckt nicht mehr eindeutig eine einzige Lagebeziehung ab. Es stellt eine Kombination aller Lagebeziehungen dar, um Probleme wie das Packen einer Kiste zu simulieren, auch wenn keinerlei Beurteilung einer Ablagepose bezüglich der Kompaktheit der Objktanordnung erfolgt. Der Unterschied zu den vorherigen Szenarien ist, dass nicht für alle Objekte ebene horizontale Bereiche erreichbar sind, um sie dort abzulegen. Damit ist es unbedingt erforderlich, dass die abzulegenden Objekte abgestützt werden. Abbildung 60 zeigt beispielhaft einen der Testläufe. Durch dieses Szenario wird überprüft, ob die Stabilitätsanalyse aus Kapitel 5.4.1 falsch positive Resultate liefert.

Insgesamt sind 91 Testdurchläufe durchgeführt worden. Bei den manipulierten Objekte handelt es sich um alltägliche Gegenstände aus dem Haushalt, die bereits in den Experimenten der vorherigen Kapitel 2, 4 und 5 verwendet worden sind.

In Kapitel 1.2 sind wissenschaftliche Fragestellungen formuliert, die bisher nur einzeln in Simulationen überprüft worden sind. Durch die vorgestellten Szenarien werden diese Fragestellungen in Realexperimenten untersucht. In dieser Untersuchung ist inbegriffen, dass die Zusammenarbeit der entwickelten Planer und des Rekonstruktionsalgorithmus evaluiert wird. Falls diese Komponenten des Systems nicht miteinander kompatibel sind, addieren sich die Fehler und können nicht korrigiert werden, da die Ausführung durch den Roboter so gestaltet ist, dass keine Rückmeldung über Fehlerzustände oder Abweichungen erfolgt.

Bei der Rekonstruktion der Objektmodelle steht die Frage im Raum, wieviele Kameraperspektiven benötigt werden, um die Objekte so exakt zu rekonstruieren, dass die geforderte Manipulationsaufgabe erfüllt werden kann. Wie schon festgestellt, genügt diese Fragestellung dem next-best-view Problem (Kapitel 2). Deshalb werden die Perspektiven und deren Anzahl händisch gewählt. Unter der gegebenen Anordnung und Größe der Rekonstruktionsvolumen zusammen mit der verwendeten Kamera ergibt sich folgende Anzahl an Kameraperspektiven: Für den Aufgreifbereich sind drei Perspektiven nötig, damit auch mehrere Objekte voneinander segmentiert werden können. Bei dem Ablagebereich genügt ein Kamerabild aus der Vogelperspektive.

Nachdem die Rekonstruktion aus Kapitel 2 konservativ ist und dadurch die Objektmodelle selbst in den sensorvalidierten Bereichen (Kapitel 2.3.4) um wenige

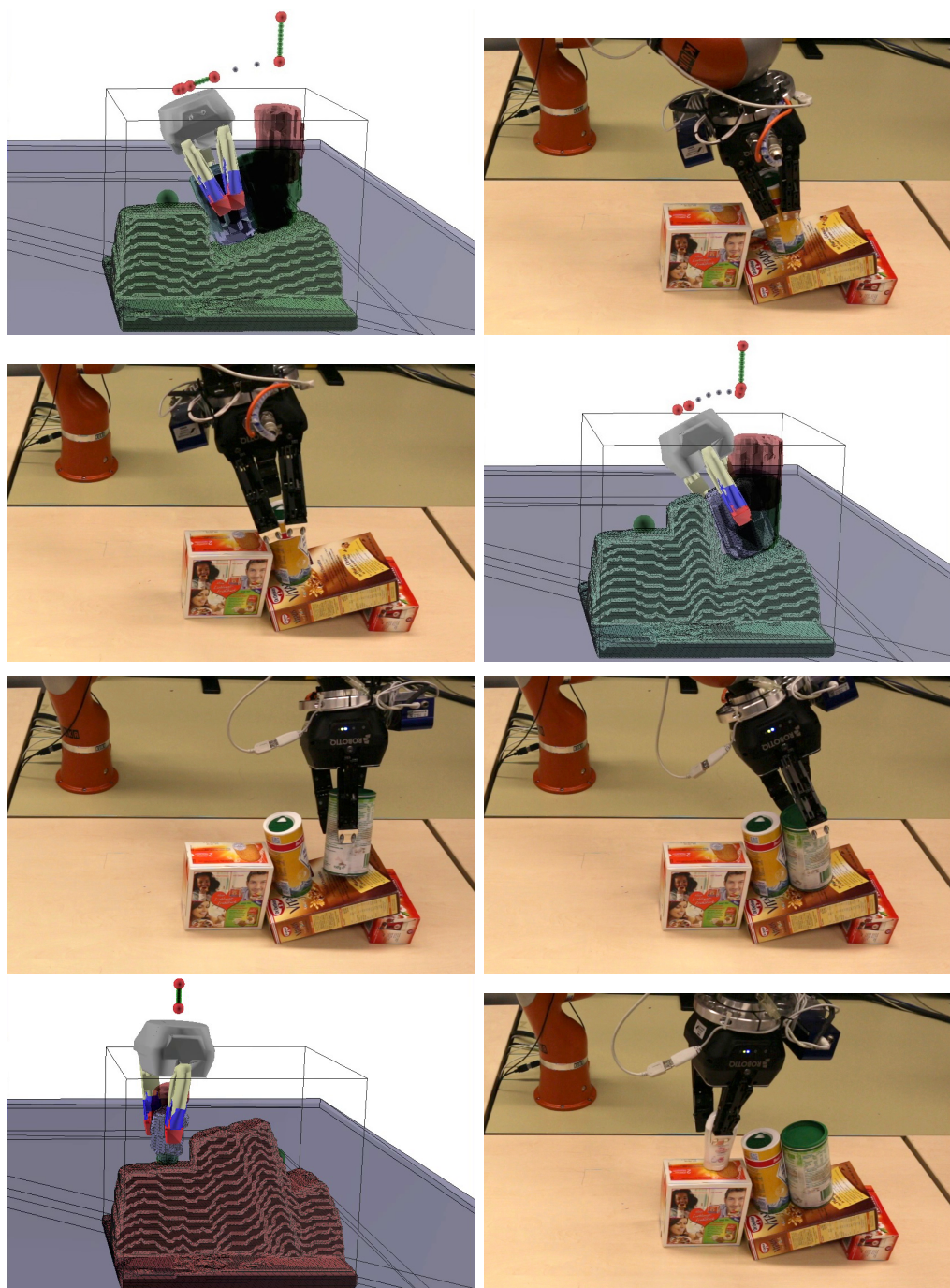


Abbildung 60: Viertes Szenario: Packen. Die Erklärung der 3D-Ansicht ist Abbildung 57 zu entnehmen.

Millimeter größer als das reale Objekt sind, kommt es wie in Abbildung 61 dazu, dass der Kontaktzustand aus der Planung zwischen den realen Objekten nicht existiert. Dieser Fehlerfall hat insgesamt dreimal (von 91 Testläufen) dazu geführt, dass die Ausführung der Planung fehlgeschlagen ist.

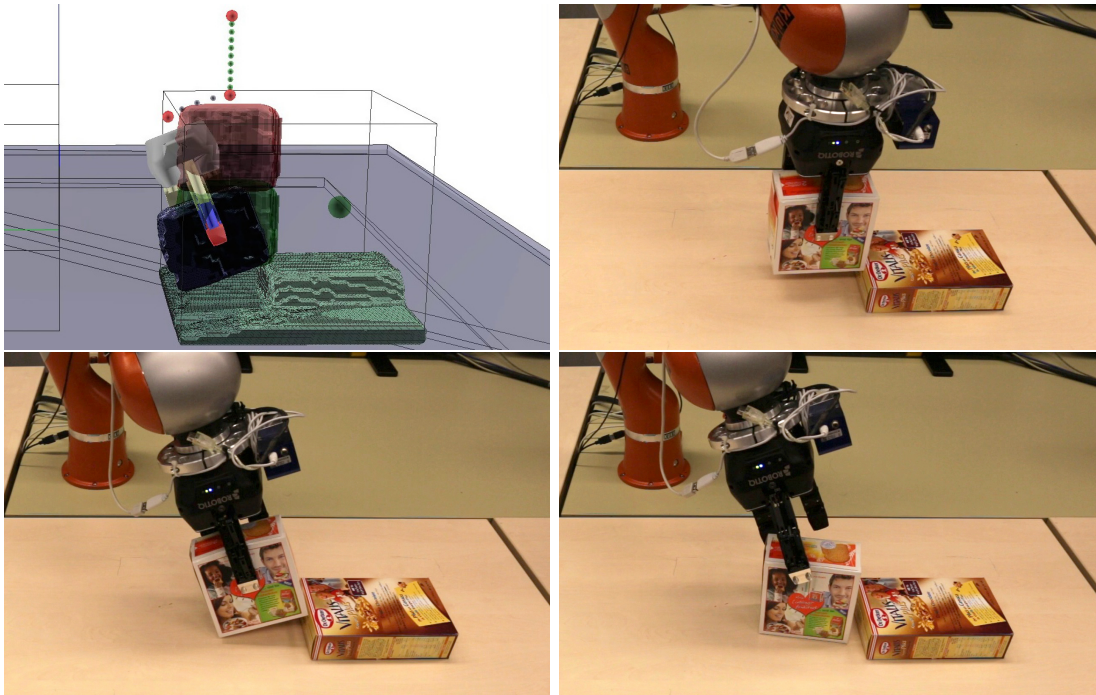


Abbildung 61: Ungenauigkeiten des Objektmodells führen zu fehlerhaft berechneten Kontaktzuständen. Die Erklärung der 3D-Ansicht ist Abbildung 57 zu entnehmen.

Der verwendete Sensor erzeugt auf spiegelnden Oberflächen nur lückenhaft Sensorwerte. Diese Detektionsfehler, die nicht von der Tiefenbildvervollständigung (Kapitel 2.3.3) korrigiert werden können, da zu große Teile des Tiefenbildes betroffen sind, und kleine Kalibrierungsfehler führen dazu, dass es zu Auslöschungen bei der Rekonstruktion kommt. Mit diesen fehlerhaften Modellen ist eine valide und ausführbare Planung nicht mehr möglich. Dieser Fehlerfall wird nicht weiter berücksichtigt, da die formulierten Annahmen (Kapitel 2) bezüglich der Rekonstruktion diese ausschließen.

Gerade ab dem zweiten Szenario, bei dem explizit die Genauigkeit des Zusammenspiels zwischen Planung und Ausführung überprüft wird, kommt es nicht ein-

mal vor, dass der Roboter mit dem gegriffenen Objekt die Ablage signifikant verrutscht hat. Insbesondere verrutscht die absichtlich labile schräge Ablage aus Abbildung 59 nicht. Nachdem die Objektmodelle konservativ sind, gleiten die Objekte nicht auf der schiefen Ebene, sondern bewegen sich frei mit einem etwa einen Millimeter großen Abstand über dieser. Diese Toleranz bedingt jedoch auch, dass die geplante Ablagepose nicht exakt mit der Realität übereinstimmt. Nachdem der Roboter den Griff löst, kann es dazu kommen, dass sich das Objekt leicht bewegt und erst dann in Ruhe verharrt. Diese leichten Ungenauigkeiten lassen sich mit zunehmender Komplexität der Szenarien beobachten.

Die Berechnung des Schwerpunktes des Objektmodells nimmt an, dass die Massenverteilung homogen ist (Kapitel 5.4.1). Abbildung 62 zeigt einen Fehlerfall, der durch diese Annahme bedingt ist. Die beiden abgebildeten Objekte sollen gestapelt werden. Die ausgeführte Ablage ist gerade so als stabil eingestuft, da der projizierte Objektschwerpunkt sehr dicht an dem Abschluss der konvexen Hülle der Kontaktpunkte (Kapitel 5.4.1) liegt. Durch die inhomogene Massenverteilung kippt das Objekt dennoch nachdem der Griff gelöst wird. Die Annahme einer homogenen Massenverteilung hat zu vier (von 91 Testläufen) fehlerhaften Ablagen durch den Roboter geführt.

Die Annahme bezüglich der Rauheit der Oberflächen und damit über den Reibungskoeffizienten führt dazu, dass bei glatten Oberflächen die Objekte abrutschen, obwohl sie als stabil bewertet werden. Der festgelegte Reibungskoeffizient von 0,2 genügt gängigen Materialkombinationen [Mül10]. Die Haftreibung versagt bei Objektkombinationen, die besonders glatte Oberflächen haben. Um diese Fehlerfälle zu minimieren, muss dieser Koeffizient weiter gesenkt werden. Dadurch schließt man allerdings stabile Ablageposen von anderen Objekten mit aus. Bei drei Ablagen während der 91 Testläufe sind die Objekte so abgerutscht, dass der Versuch als fehlerhaft zu werten ist.

Die Experimente des vierten Szenarios (Abbildung 60) zeigen, dass die entwickelte Stabilitätsheuristik Ablageposen korrekt als stabil einstuft, selbst wenn das Objekt in dieser Pose nicht ohne Unterstützung der Ablage in Ruhe verharren würde. Dabei ist sie so restriktiv ausgelegt, dass es zu nur in Kombination mit anderen Fehlern zu falsch positiven Resultaten kommt. Falsche Bewertungen der Stabilität haben in den Experimenten aus einer fehlerhaften Kontaktpunktberech-

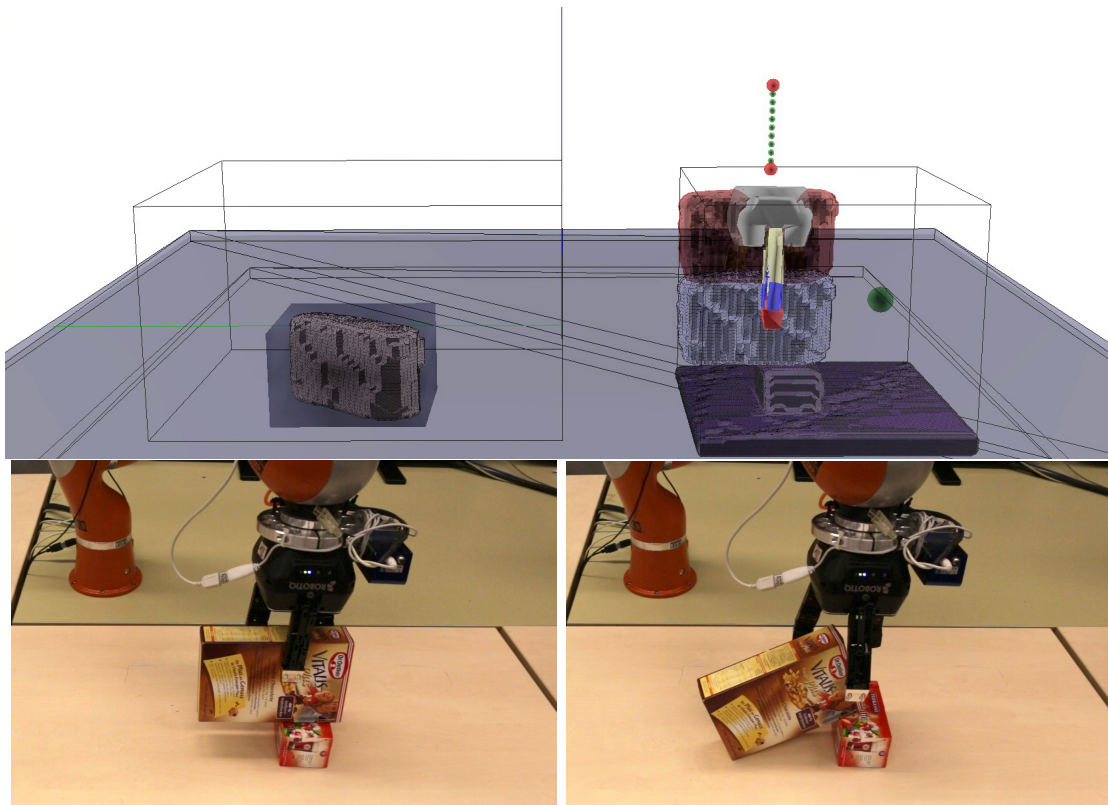


Abbildung 62: Fehler bei der Massenverteilung. Die Erklärung der 3D-Ansicht ist Abbildung 57 zu entnehmen.

nung ausgehend von einer fehlerhaften Rekonstruktion resultiert. Dieser Fall ist szenarioübergreifend insgesamt fünf mal (von 91 Testläufen) aufgetreten.

Große und komplexe Objekte, wie die abgebildete Rührschüssel, werden ebenso präzise gegriffen und im ersten Szenario auch wie geplant abgelegt. Nachdem allerdings die Anzahl der verfügbaren Greiferkonfigurationen deutlich kleiner als bei den primitiven Objekten ist (Kapitel 4.5), ist die Erreichbarkeit der berechneten Ablageposen durch den Roboter in den anderen Szenarien nicht immer gegeben. Gründe dafür sind, dass der Roboter während der Feinbewegung zur Ablage mit der Ablage selbst kollidieren würde oder die Kinematik des Roboters verhindert, dass die Feinbewegungen geplant und durchgeführt werden können.

Die Präzision der Griffe ist die Voraussetzung für eine erfolgreiche Ablage. Ein Griff ist nicht präzise, wenn das Objekt während des Schließens des Greifers verrutscht, so dass die tatsächliche Pose nicht mehr mit der geplanten übereinstimmt.

Primitive Objekte werden stets mit dem hier entwickelten Greifplaner präzise gegriffen. Dieser Fehler ist während der Experimente nur bei der Manipulation von komplexen Objekten aufgetreten. Während der Durchführung der Szenarien sind sechs Objekte so unpräzise gegriffen worden, dass sich nicht wie geplant abgelegt werden konnten. Insgesamt sind 29 Testläufe mit komplexen Objekten erfolgt.

Die Zeit, die der Roboter benötigt, um die Rekonstruktionsvolumen sensorisch zu erfassen, die Objektmodelle zu rekonstruieren, den Griff und die Ablage zu planen und auszuführen, beläuft sich durchschnittlich auf 1 min 35 sec. Die Rechenzeit der aller Softwarekomponenten zusammen beläuft sich auf durchschnittlich vier Sekunden. Durch die gewählte Reihenfolge zur Ausführung der einzelnen Systemkomponenten (Kapitel 6.2) kommt es quasi nicht dazu, dass der Roboter still steht und auf das Ergebnis einer Planung wartet. Die Fälle, dass der Roboter auf die Planungsergebnisse warten muss, entstehen, wenn die Erreichbarkeit der Greifpose beziehungsweise der Ablagepose nicht gegeben ist und daher viele Kombinationen getestet werden müssen und der Bahnplaner mehrmals neu planen muss.

Insgesamt werden über 90% der geplanten Griffe und über 74% der Ablagen erfolgreich ausgeführt. Fehlerhafte Griffe<sup>1</sup> lassen sich neben den beschriebenen Fehlerquellen hauptsächlich auf eine schlechte Konvergenz des Levenberg-Marquardt-Verfahrens (Kapitel 4.4) bei komplexen Teilgeometrien zurückführen.

Die fehlerhaften Ablagen<sup>2</sup> lassen sich zu etwa 50% auf die Nichterfüllung der Annahmen bezüglich des Reibungskoeffizienten und der Massenverteilung des Objektes oder auf Detektionsfehler des Sensors zurückführen. Die übrigen Fehler resultieren aus fehlerhaften Berechnungen der Kontaktpunkte und Abweichungen zwischen rekonstruierten Modell und realer Szene.

#### 6.4 SCHLUSSFOLGERUNGEN

In diesem Kapitel werden die entwickelte Rekonstruktion wie auch der Greifplaner und Ablageplaner anhand eines prototypischen Hardwareaufbaus experimentell evaluiert.

---

1 Ein Griff wird als fehlerhaft eingestuft, wenn das Objekt während des Schließens der Greiferfinger verrutscht oder nicht gegriffen werden kann.

2 Eine Ablage wird als fehlerhaft eingestuft, wenn das Objekt nachdem der Roboter den Griff löst nicht in Ruhe verharrt.



Die Algorithmen arbeiten so exakt, dass Greif- und Ablageaufgaben in unterschiedlichen Szenarien mit unüberwachten Roboterbewegungen durchgeführt werden können. Unter anderem durch die Berücksichtigung der Nebengelenke des Greifers und der zwischen sensorvalidierten und nicht validierten Oberflächenbereichen differenzierenden Rekonstruktion entstehen während des Aufgreifens der Objekte nur so geringe Abweichungen zwischen Objektpose und Planungsergebnis, dass die nachfolgenden Feinbewegungen zum Durchführen der Ablageplanung exakt ausgeführt werden können. Die durch den Ablageplaner entwickelten Kontaktzustände und Ablageposen sind so exakt, dass der Roboter beziehungsweise ein gegriffenes Objekt nicht mit der Ablage kollidiert. Die entwickelte Stabilitätsheuristik liefert nur dann falsch positive Bewertungen, wenn eine der Annahmen (Reibung oder Massenverteilung des Objektes) nicht zutrifft oder es zu numerischen Fehlern bei den Kontaktpunktberechnungen kommt. Die 91 Testläufe werden zu über 74% erfolgreich absolviert, wobei die Rechenzeit der Softwarekomponenten nur einen Anteil von 4% an der Gesamtzeit der Ausführung ausmacht. Für die Rekonstruktion der Objekte werden bei diesem experimentellen Aufbau drei Kameraperspektiven benötigt. Für die Rekonstruktion ist es ausreichend den Ablagebereich einmal aus der Vogelperspektive aufzunehmen.

Die aufgetretenen Fehlerfälle lassen sich zum großen Teil durch eine robustere Ausführung beispielsweise durch sensorgeregelte Bewegungen eliminieren. Die Planer stellen schon Kontaktzustände und die jeweiligen Roboterbewegungen bereit, um leicht die notwendigen zusätzlichen Bedingungen wie der Abbruchbedingung einer solchen sensorgeregelten Bewegungen zu ermitteln.

Die Anzahl der Kameraperspektiven zur Rekonstruktion und damit sowohl die Genauigkeit und Vollständigkeit der Rekonstruktion wie auch die Gesamtlaufzeit der Ausführung lassen sich durch die Verwendung eines next-best-view-Ansatzes weiter verringern und szenarioabhängig wählen.



# 7

## FAZIT

In diesem Kapitel wird die Arbeit zunächst zusammengefasst (Kapitel 7.1) und anschließend ein Ausblick (Kapitel 7.2) auf mögliche Erweiterungen aufgezeigt.

### 7.1 ZUSAMMENFASSUNG

Die übergeordnete Fragestellung dieser Arbeit ist es, inwieweit es möglich ist ausschließlich auf Basis der geometrischen Rekonstruktion eines unbekanntes Objektes dieses präzise zu greifen und stabil auf einer nicht notwendigerweise ebenen Ablage abzulegen. Dabei wird auf jede Sensorik verzichtet, die Informationen über Kräfte und Momente liefert. Es kommt ausschließlich eine Tiefenkamera zum Einsatz.

Um die Modelle der Objekte zu berechnen wird ein Volumenrekonstruktionsalgorithmus weiterentwickelt, der basierend auf Tiefenbildern ein konservatives geometrisches Oberflächenmodell erstellt. Das Verfahren vervollständigt Löcher in den Tiefenbildern, so dass mit diesen in einem subtrahierenden Vorgang eine Volumenrekonstruktion erstellt werden kann. Diese Volumenrekonstruktion wird mittels eines angepassten Marching-Cubes Algorithmus in ein Oberflächenmodell transformiert. Diese Modelle unterscheiden Oberflächenbereiche, die durch Sensordaten validiert sind und Bereiche, die konservativ abgeschätzt sind. Diese Unterscheidung der Oberflächenbereiche wird an Greif- und Ablageplaner weitergereicht, damit sie diese Informationen bezüglich des Planungsergebnisses berücksichtigen können.

Ebenso wird auf Basis dieser Oberflächenmodelle ein Greifplaner (eine Weiterentwicklung von [Bau+14a; Bau+14b]) eingeführt, der speziell für Mehrfingergreifer ausgelegt ist. Unter Verwendung einer parametrisierbaren geschlossenen geometrischen Form werden dynamisch die Greifervorkonfiguration von Mehrfingergreifern angepasst und Griffe geplant, die die Unsicherheit basierend auf der Unterscheidung zwischen durch Sensorwerte validierten und konservativ abgeschätzten Ober-

flächenbereichen der Objektmodelle berücksichtigt. Bei der Greifervorkonfiguration werden die Gelenkwinkel der Nebengelenke anhand der Parameter der geometrischen Form bestimmt, so dass die Greiferfinger flächigen Kontakt mit dem Objekt herstellen können. Weiter werden aus der Parametrisierung die möglichen Posen der Greiferbasis berechnet.

Des Weiteren wird ein Ablageplaner (eine Weiterentwicklung von [Bau+13] und [Bau+14c]) entwickelt. Er ermöglicht es unter Verwendung des obigen Objektmodells Ablageposen für unbekannte Objekte auf nicht-ebenen Ablagen zu planen. Um die Planung mit unbekanntem Objekt durchführen zu können, wird eine Stabilitätsheuristik und eine Kontaktposenberechnung eingeführt. Die Heuristik ermöglicht es eine Pose unter Verwendung der Kontaktpunkte und -normalen zusammen mit dem geometrischen Schwerpunkt auf Stabilität zu prüfen, ohne dass die Kontaktkräfte benötigt werden. Die Kontaktposenberechnung ermöglicht es kontinuierlich ausgehend von einer bestehenden Kontaktpose neue Kontaktposen zu berechnen. Dadurch entstehen für eine stabile Ablagepose zugehörige Sequenzen von Kontaktposen. Diese können direkt vom Roboter ausgeführt werden ohne zusätzlich eine gesonderte Feinplanung durchführen zu müssen.

In Experimenten mit einem realen Roboter werden die entwickelten Softwarekomponenten auf ihre Leistungsfähigkeit untersucht. Insbesondere wird evaluiert, ob es möglich ist unbekannte Objekte präzise zu manipulieren, ohne dass die Roboterbewegungen mit zusätzlichen Sensorinformationen überwacht beziehungsweise geregelt werden. Weiter wird experimentell untersucht, ob die Rechenzeit so gering gehalten werden kann, dass die Planung während der Transferbewegungen des Roboters abgeschlossen werden kann und wieviele Tiefenbilder notwendig sind, um ausreichend Umweltinformationen zu erfassen, so dass die Greif- und Ablageaufgaben erfüllt werden können. Diese Fragestellungen werden in vier unterschiedlich komplexen Ablageszenarien mit unterschiedlich komplexen Objekten untersucht.

Die Berücksichtigung aller Freiheitsgrade des Mehrfingergräfers ermöglicht es anfangen von einfachen Objekten, deren Geometrie einer primitiven geometrischen Form entspricht, bis hin zu Objekten, die eine komplexe Geometrie besitzen und nicht vom Greifer umschlossen werden können, zu greifen. Dabei ist die Planung so präzise, dass die Objekte während des Schließens der Greiferfinger im Allgemeinen nicht verschoben oder verdreht werden. Insbesondere ermöglicht die

vollständig variable Greifervorkonfiguration, dass die Anzahl an stabilen Griffen für ein Objekt erhöht wird und daher die möglichen Kombinationen aus Greif- und Ablagepose den vollständigen Greif- und Ablagevorgang nicht einschränken.

Die Ablageplanung ermöglicht es stabile Objektposen relativ zu den ebenso rekonstruierten Ablagegeometrien zu berechnen. Der Planer erzeugt stabile Ablagen zusammen mit der zugehörigen Feinplanung auf den unterschiedlichsten nicht-ebenen Ablagegeometrien. Die Experimente mit dem prototypischen Hardwareaufbau in Kombination mit den entwickelten Algorithmen zeigen, dass Greif- und Ablageaufgaben in verschiedenen Szenarien, angefangen bei der Anordnung von Objekten über das Stapeln von Objekten bis hin zu einfachen Packaufgaben durchgeführt werden können. Die entwickelten Planer arbeiten basierend auf der konservativen Rekonstruktion so genau, dass die Ausführung durch den Roboter ausschließlich positionsbasiert erfolgen kann. Insbesondere können alle Szenarien mit hoher Erfolgsrate ohne sensorisch überwachte Bewegungen erfüllt werden.

Die Rekonstruktion der involvierten Objekte benötigt zur Durchführung der Szenarien insgesamt nur vier unterschiedliche Kameraperspektiven. Zusammen mit der Planungszeit, die so kurz ist, dass sie während der Roboterbewegungen erfolgt, werden im Schnitt nur 1,5 Minuten benötigt, damit der Roboter ein Objektmodell und ein Modell der Ablage rekonstruiert und das Objekt wie geplant manipuliert. Die dabei benötigte Rechenzeit, um die Planer auszuführen und die Objekte zu rekonstruieren, bleibt unter zwei Sekunden. In den Experimenten ist es im Allgemeinen zu keinen Wartezeiten des Roboters gekommen, weil die Planer noch keine Ergebnisse vorliegen hatten.

## 7.2 AUSBLICK

Die Verfahren lassen sich in verschiedenen Bereichen erweitern. Diese Möglichkeiten werden im Folgenden dargelegt.

Durch die Unterscheidung zwischen sensorisch validierten und konservativ abgeschätzten Oberflächenbereichen der rekonstruierten Modelle könnte die nächst beste Kameraperspektive ermittelt werden, um die Anzahl der nötigen Kameraperspektiven aufgaben-unabhängig automatisch zu minimieren.

Mit einem Verfahren, das basierend auf der Objektfunktionalität eines unbekanntes Objektes Teilgeometrien identifiziert, wäre es mit dem entwickelten Ablageplaner möglich Objekte so zu greifen, dass sie gemäß ihrer Funktion aufgabenabhängig vom Roboter verwendet werden. Mit diesem zusätzlichen Verfahren werden das Übergeben von Objekten an den Menschen, das Ablegen von Objekten in die menschliche Hand und das Verwenden der gegriffenen Objekte als Werkzeug möglich. Beispielsweise kann dann ein Personal Roboter als Handlanger arbeiten, um einem Mechaniker das nötige Werkzeug zu reichen. In diesem Szenario greift der Mensch nicht aktiv nach dem Werkzeug, sondern streckt seine Hand blind aus und wartet darauf, dass das Werkzeug in seine Hand gelegt wird. Seine Hand stellt dabei eine unbekanntes Ablage dar, da man als Mensch in unterschiedlichen Situationen seine Hand unterschiedlich ausstreckt. Die Objektfunktionalität kann nicht nur die Möglichkeiten der Greifplanung erweitern, sondern auch die Ablageplanung. Mit der gegebenen Funktionalität ließe sich die Lagebeziehung und die Vorzugsorientierung der Objekte ermitteln.

Die Bewertung einer Ablagepose könnte in Zukunft nicht nur auf Basis ihrer Stabilität erfolgen, sondern auch berücksichtigen inwieweit die geforderte Lagebeziehung erreicht ist und inwieweit diese Pose für die Objektfunktionalität geeignet ist.

Die Ausführung der Kontaktzustände, die von dem Ablageplaner berechnet werden, ist nicht immer möglich solange das Objekt gegriffen ist. Wenn der Roboter beispielsweise eine Tasse in einen Schrank stellen soll und diese dicht an die anderen Tassen schieben soll, muss der Roboter die Tasse abstellen und anschließend diese schieben. Mit einem Verfahren, das basierend auf der Kontaktzustandssequenz Schiebebewegungen ein Umgreifen oder Zwischenablagen plant und durchführt, sind zusammen mit den Planern aus dieser Arbeit auch solche Szenarien durchführbar.

In dieser Arbeit werden die berechnete Roboterbewegungen ohne Rückmeldung an das System ausgeführt. Nachdem der Ablageplaner auch von einem bestehenden Kontaktzustand aus beginnen kann eine stabile Ablage zu planen, wäre es möglich mit sensorischer Rückmeldung während der Ausführung Fehler zu erkennen und basierend auf dem detektierten Kontaktzustand die geplante Ablagepose zu korrigieren beziehungsweise neu zu planen.

## LITERATUR

- [AIS13] euRobotics AISBL. *Robotics 2020: Strategic Research Agenda For Robotics in Europe*. [www.eu-robotics.net](http://www.eu-robotics.net). [Online; abgerufen 17 Mai 2016]. 2013.
- [AIS14] euRobotics AISBL. *Robotics 2020 Multi-Annual Roadmap*. [www.eu-robotics.net](http://www.eu-robotics.net). [Online; abgerufen 17 Mai 2016]. 2014.
- [Ama+87] John Amanatides und A Woo. “A fast voxel traversal algorithm for ray tracing”. In: *Eurographics*. Amsterdam, 1987, S. 1–10.
- [Ame+01] Nina Amenta, Sunghee Choi und Ravi Krishna Kolluri. “The power crust, unions of balls, and the medial axis transform”. In: *Computational Geometry* 19.2-3 (2001), S. 127–153.
- [Att+13] Marco Attene, Marcel Campen und Leif Kobbelt. “Polygon mesh repairing: An Application Perspective”. In: *ACM Computing Surveys* 45.2 (2013), S. 1–33.
- [Bag98] Boris Baginski. “Motion Planning for Manipulators with Many Degrees of Freedom – The BB-Method”. Diss. Technische Universität München, 1998.
- [Bar+92] David Baraff und Andrew Witkin. “Dynamic simulation of non-penetrating flexible bodies”. In: *ACM SIGGRAPH Computer Graphics* 26.2 (1992), S. 303–308.
- [Bar84] Alan H. Barr. “Global and local deformations of solid primitives”. In: *Conference on Computer graphics and interactive techniques*. 1984, S. 21–30.
- [Bar94] David Baraff. “Fast contact force computation for nonpenetrating rigid bodies”. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94*. New York, New York, USA: ACM Press, 1994, S. 23–34.

- [Bau+12] Johannes Baumgartl und Dominik Henrich. “Fast Vision-based Grasp and Delivery Planning for unknown Objects”. In: *7th German Conference on Robotics (ROBOTIK 2012)*. 2012, S. 1–5.
- [Bau+13] Johannes Baumgartl, Per Kaminsky und Dominik Henrich. “A geometrical placement planner for unknown sensor-modelled objects and placement areas”. In: *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2013, S. 364–371.
- [Bau+14a] Johannes Baumgartl und Dominik Henrich. “GPU-based Grasp And Placement Planners For Sensor-modelled Objects”. In: *ISR/Robotik 2014; 41st International Symposium on Robotics*. Munich, 2014, S. 1–6.
- [Bau+14b] Johannes Baumgartl, Konrad Ley und Dominik Henrich. “Multi-Finger Grasp Planning For Sensor-Modeled Objects Using Local Curvature Information”. In: *5th CATS 2014 - CIRP Conference on Assembly Systems and Technologies*. 2014.
- [Bau+14c] J. Baumgartl u. a. “A fast, GPU-based geometrical placement planner for unknown sensor-modelled objects and placement areas”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, S. 1552–1559.
- [Bay+08] John G. Baylog u. a. “Robust search for structured object placement using unmanned vehicles”. In: *OCEANS 2008*. IEEE, 2008, S. 1–10.
- [Ber+14] Matthew Berger u. a. “State of the Art in Surface Reconstruction from Point Clouds”. In: *Eurographics STAR (Proc. of EG’14)*. 2014, S. 161–185.
- [Ber04] Gino van den Bergen. *Ray casting against general convex objects with application to continuous collision detection*. [www.detecta.com](http://www.detecta.com). [Online; abgerufen 19 Juli 2015]. 2004.
- [Bic+00] Antonio Bicchi und V Kumar. “Robotic grasping and contact: a review”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation*. IEEE, 2000, S. 348–353.



- [Bie+07] Georg Biegelbauer und Markus Vincze. “Efficient 3D Object Detection by Fitting Superquadrics to Range Image Data for Robot’s Object Manipulation”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, S. 1086–1091.
- [Big+09] Ali Bigdelou, Alexander Ladikos und Nassir Navab. “Incremental Visual Hull Reconstruction”. In: *Proceedings of the British Machine Vision Conference 2009*. British Machine Vision Association, 2009, S. 35.1–35.11.
- [Boh+11] Jeannette Bohg u. a. “Mind the gap - robotic grasping under incomplete observation”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, S. 686–693.
- [Boh+14] Jeannette Bohg u. a. “Data-Driven Grasp Synthesis - A Survey”. In: *IEEE Transactions on Robotics* 30.2 (2014), S. 289–309.
- [Bon+08] Gary M Bone, Andrew Lambert und Mark Edwards. “Automated modeling and robotic grasping of unknown three-dimensional objects”. In: *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, S. 292–298.
- [Bra13] Dietrich Braess. *Finite Elemente*. Berlin: Springer-Verlag, 2013.
- [Bre+98] D.E. Breen, S. Mauch und R.T. Whitaker. “3D scan conversion of CSG models into distance volumes”. In: *IEEE Symposium on Volume Visualization*. IEEE, 1998, S. 7–14.
- [Bre65] J. E. Bresenham. *Algorithm for computer control of a digital plotter*. 1965.
- [Bro+10] E. Brown u. a. “Universal robotic gripper based on the jamming of granular material”. In: *Proceedings of the National Academy of Sciences* 107.44 (2010), S. 18809–18814.
- [Buc+13] Dirk Buchholz u. a. “Efficient bin-picking and grasp planning based on depth data”. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, S. 3245–3250.

- [Buc+14] Dirk Buchholz u. a. “Combining visual and inertial features for efficient grasping and bin-picking”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, S. 875–882.
- [Bul13] Bullet Physics Library. *Manual*. [bulletphysics.org/](http://bulletphysics.org/). [Online; abgerufen 19 Juli 2015]. 2013.
- [Cho+05] Howie Choset u. a. *Principles of Robot Motion*. Boston: MIT Press, 2005.
- [Cho+06] Y.-K. Choi u. a. “Continuous Collision Detection for Two Moving Elliptic Disks”. In: *IEEE Transactions on Robotics* 22.2 (2006), S. 213–224.
- [Chr+07] Vassilios N. Christopoulos und Paul Schrater. “Handling shape and contact location uncertainty in grasping two-dimensional planar objects”. In: *IEEE International Conference on Intelligent Robots and Systems*. IEEE. IEEE, 2007, S. 1557–1563.
- [Cio+09] M. T. Ciocarlie und P. K. Allen. “Hand Posture Subspaces for Dexterous Robotic Grasping”. In: *The International Journal of Robotics Research* 28.7 (2009), S. 851–867.
- [Cio+14] M. Ciocarlie u. a. *Experimental Robotics*. Hrsg. von Oussama Khatib, Vijay Kumar und Gaurav Sukhatme. Bd. 79. Springer Tracts in Advanced Robotics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 241–252.
- [Coc+12] Tiberiu T. Cocias, Sorin M. Grigorescu und Florin Moldoveanu. “Multiple-superquadrics based object surface estimation for grasping in service robotics”. In: *13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*. IEEE, 2012, S. 1471–1477.
- [Cos+11] Akansel Cosgun u. a. “Push planning for object placement on cluttered table surfaces”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, S. 4627–4632.

- [Cou13] Murilo G. Coutinho. *Guide to Dynamic Simulations of Rigid Bodies and Particle Systems*. Hrsg. von Louis G. Birta. Simulation Foundations, Methods and Applications. London: Springer London, 2013, S. 1–399.
- [Dan+01] Dan Ding, Yun-Hui Lee und Shuguo Wang. “Computation of 3-D form-closure grasps”. In: *IEEE Transactions on Robotics and Automation* 17.4 (2001), S. 515–522.
- [Det+12] Renaud Detry u. a. “Generalizing grasps across partly similar objects”. In: *Proceedings - IEEE International Conference on Robotics and Automation*. 2012, S. 3791–3797.
- [Dia+08] Rosen Diankov und James Kuffner. *OpenRAVE : A Planning Architecture for Autonomous Robotics*. Techn. Ber. Pittsburgh: Carnegie Mellon University, 2008.
- [Dia10] Rosen Diankov. “Automated Construction of Robotic Manipulation Programs”. Diss. Carnegie Mellon University, 2010.
- [Dog13] Mehmet R Dogar. “Physics-Based Manipulation Planning in Cluttered Human Environments”. Diss. Carnegie Mellon University, 2013.
- [Dun+08] C. Dune u. a. “Active rough shape estimation of unknown objects”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, S. 3622–3627.
- [Dun+13] Kester Duncan u. a. “Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects”. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, S. 4238–4243.
- [Eds+06] Aaron Edsinger und Charles Kemp. “Manipulation in Human Environments”. In: *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, S. 102–109.
- [Ege08] Jens Egeblad. “Heuristics for Multidimensional Packing Problems”. Diss. University of Copenhagen, 2008.

- [Epp+13] Clemens Eppner und Oliver Brock. “Grasping unknown objects by exploiting shape adaptability and environmental constraints”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, S. 4000–4006.
- [Epp+15] Clemens Eppner und Oliver Brock. “Planning Grasp Strategies That Exploit Environmental Constraints”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, 2015.
- [Erl05] Kenny Erleben. “Stable, Robust, and Versatile Multibody Dynamics Animation”. Diss. Denmark: University of Copenhagen, 2005.
- [Fel+09] Javier Felip und Antonio Morales. “Robust sensor-based grasp primitive for a three-finger robot hand”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, S. 1811–1816.
- [Fin+03] B. Finkemeyer, T. Kröger und F. M. Wahl. “Placing of objects in unknown environments”. In: *9th IEEE International Conference on Methods and Models in Automation and Robotics*. Międzyzdroje, 2003, S. 975–980.
- [Fis+11] Matthew Fisher, Manolis Savva und Pat Hanrahan. “Characterizing structural relationships in scenes using graph kernels”. In: *ACM Transactions on Graphics* 30.4 (2011), S. 1.
- [Fis+12] David Fischinger und Markus Vincze. “Empty the basket - a shape based learning approach for grasping piles of unknown objects”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, S. 2051–2057.
- [Fu+08] Hongbo Fu u. a. “Upright orientation of man-made objects”. In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), S. 42.
- [Gol+07] Corey Goldfeder u. a. “Grasp Planning via Decomposition Trees”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. April. Roma: IEEE, 2007, S. 4679–4684.
- [Gol+11] Corey Goldfeder und Peter K. Allen. “Data-driven grasping”. In: *Autonomous Robots* 31.1 (2011), S. 1–20.

- [Gor+09] Nicolas Gorges und Heinz Wörn. “Learning an Object-Grasp Relation for Silhouette-Based Grasp Planning”. In: *Advances in Robotics Research*. Hrsg. von Torsten Kröger und Friedrich M Wahl. Springer Berlin Heidelberg, 2009, S. 227–237.
- [Gug+12] Fabio Guggeri, Riccardo Scateni und Renato Pajarola. “Shape Reconstruction from Raw Point Clouds using Depth Carving.” In: *Eurographics 2012 - Short Papers*. The Eurographics Association, 2012.
- [Gup+11] Kalyan Moy Gupta u. a. “Representing and reasoning with functional knowledge for spatial language understanding”. In: *Workshop on Computational Models of Spatial Language Interpretation-2*. Boston, MA, 2011.
- [Han+14] Kaiyu Hang u. a. “Combinatorial optimization for hierarchical contact-level grasping”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, S. 381–388.
- [Har+12] Kensuke Harada u. a. “Object placement planner for robotic pick and place tasks”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, S. 980–985.
- [Har+14] Kensuke Harada u. a. “Validating an object placement planner for robotic pick-and-place tasks”. In: *Robotics and Autonomous Systems* 62.10 (2014), S. 1463–1477.
- [Hav+14] Giray Havur u. a. “Geometric rearrangement of multiple movable objects on cluttered surfaces: A hybrid reasoning approach”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, S. 445–452.
- [Her+12] Alexander Herzog u. a. “Template-based learning of grasp selection”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, S. 2379–2384.
- [Hol+13] Anne Holladay u. a. “Object placement as inverse motion planning”. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, S. 3715–3721.

- [Hop94] Hugues Hoppe. “Surface Rekonstruktion from Unorganized Points”. Diss. Washington: University of Washington, 1994.
- [Hsi+10] K Hsiao u. a. “Contact-reactive grasping of objects with partial shape information”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, S. 1228–1235.
- [Hue+09] Kai Huebner u. a. “Grasping Known Objects with Humanoid Robots: A Box-Based Approach”. In: *International Conference on Advanced Robotics*. Munich, 2009, S. 1–6.
- [IDS] IDSI Imaging Development Systems GmbH. *Ensenso Manual*. [www.ensenso.de/manual/](http://www.ensenso.de/manual/). [Online; abgerufen 17 Mai 2016].
- [ISO12] ISO. *Robots and robotic devices – Vocabulary*. ISO 8373:2012. 2012.
- [Iza+11] Shahram Izadi u. a. “Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera”. In: *In Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), S. 559–568.
- [Jai+10] Advait Jain und Charles C Kemp. “EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces”. In: *Autonomous Robots* 28.1 (2010), S. 45–64.
- [Jia+12] Yun Jiang u. a. “Learning to place new objects”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, S. 3088–3095.
- [Jia+13] Yun Jiang und Ashutosh Saxena. “Hallucinating Humans for Learning Robotic Placement of Objects”. In: *International Symposium on Experimental Robotics*. 2013, S. 921–937.
- [Kai+12] Bernhard Kainz u. a. “OmniKinect: Real-Time Dense Volumetric Data Acquisition and Applications”. In: *Proceedings of the 18th ACM symposium on Virtual reality software and technology*. 2012, S. 25–32.
- [Kaj+08] S. Kajita und B. Espiau. *Springer Handbook of Robotics*. Hrsg. von Bruno Siciliano und Oussama Khatib. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, S. 361–389.

- [Kas+12] A. Kasper, Z. Xue und R. Dillmann. “The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics”. In: *The International Journal of Robotics Research* 31.8 (2012), S. 927–934.
- [Kaz+06] Michael Kazhdan, Matthew Bolitho und Hugues Hoppe. “Poisson surface reconstruction”. In: *Eurographics Symposium on Geometry Processing*. 2006, S. 61–70.
- [Kaz+13] Michael Kazhdan und Hugues Hoppe. “Screened poisson surface reconstruction”. In: *ACM Transactions on Graphics* 32.3 (2013), S. 1–13.
- [Kim+03] Byungmoon Kim und Jarek Rossignac. “Collision prediction for polyhedra under screw motions”. In: *Proceedings of the eighth ACM symposium on Solid modeling and applications - SM '03*. New York, New York, USA: ACM Press, 2003, S. 4.
- [Kle+11] Oliver Kleine, Martin Haegele und Nikolaus Bluemlein. *Wirtschaftlichkeitsanalysen neuartiger Servicerobotik- Anwendungen und ihre Bedeutung für die Robotik-Entwicklung*. Techn. Ber. Fraunhofer-Institute IPA und ISI, 2011, 1–371 (but only 318 are in paper copy).
- [Kli+11] Ellen Klingbeil u. a. “Grasping with application to an autonomous checkout robot”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, S. 2837–2844.
- [Kli+14] Matthew Klingensmith, Martin Herrmann und Siddhartha S. Srinivasa. “Object Modeling and Recognition from Sparse, Noisy Data via Voxel Depth Carving”. In: *International Symposium on Experimental Robotics*. 2014.
- [Koh03] Leonard Kohen. *Arranging Things: A Rhetoric of Object Placement*. Stone Bridge Press, 2003.
- [Koo+12] G. Kootstra u. a. “Enabling grasping of unknown objects through a synergistic use of edge and surface information”. In: *The International Journal of Robotics Research* 31.10 (2012), S. 1190–1213.

- [Kro+14] Athanasios Krontiris u. a. “Rearranging similar objects with a manipulator using pebble graphs”. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, S. 1081–1087.
- [Kru+10] Sébastien Krut u. a. “Extension of the Form-Closure Property to Underactuated Hands”. In: *IEEE Transactions on Robotics* 26.5 (2010), S. 853–866.
- [Kru+14] Robert Krug u. a. “Velvet fingers: Grasp planning and execution for an underactuated gripper with active surfaces”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, S. 3669–3675.
- [Kru+15] Robert Krug u. a. “On Using Optimization-based Control instead of Path-Planning for Robot Grasp Motion Generation”. In: *IEEE International Conference on Robotics and Automation (ICRA) - Workshop on Robotic Hands, Grasping, and Manipulation*. 2015.
- [Kru14] Robert Krug. “Optimization-based Robot Grasp Synthesis and Motion Control”. Diss. Örebro University, 2014.
- [Kuh12] Stefan Kuhn. “Wissens- und sensorbasierte geometrische Rekonstruktion”. Diss. Bayreuth: Universität Bayreuth, 2012.
- [KUK15] KUKA Roboter GmbH. *KUKA LBR. Bedienerfreundlich, feinfühlig und flexibel*. [www.die-roboter.de/fileadmin/content/files/kuka/KUKA\\_LBR4plus\\_DEUTSCH.pdf](http://www.die-roboter.de/fileadmin/content/files/kuka/KUKA_LBR4plus_DEUTSCH.pdf). [Online; abgerufen 17 Mai 2016]. 2015.
- [Kun+11] Kun Zhou u. a. “Data-Parallel Octrees for Surface Reconstruction”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.5 (2011), S. 669–681.
- [Kwa+11] Sung Jo Kwak, Tsutomu Hasegawa und Seong Youb Chung. “Efficiency improvement in generation of a contact state graph by eliminating unnecessary elements”. In: *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*. IEEE, 2011, S. 1–6.



- [Lad+08] Alexander Ladikos, Selim Benhimane und Nassir Navab. “Real-time 3D reconstruction for collision avoidance in interventional environments”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008* 5242 (2008), S. 526–534.
- [Lau94] Aldo Laurentini. “The visual hull concept for silhouette-based image understanding”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.2 (1994), S. 150–162.
- [Lee+14] Adam Leeper u. a. “Using Near-Field Stereo Vision for Robotic Grasping in Cluttered Environments”. In: *The 12th International Symposium on Experimental Robotics*. 2014, S. 253–267.
- [Leo+14] Beatriz Leon u. a. “Characterization of grasp quality measures for evaluating robotic hands prehension”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong: IEEE, 2014, S. 3688–3693.
- [Li+11] Yangyan Li u. a. “GlobFit: consistently fitting primitives by discovering global relations”. In: *ACM Transactions on Graphics* 30.4 (2011), S. 1.
- [Lia+13] Jian Liang, Frederick Park und Hongkai Zhao. “Robust and Efficient Implicit Surface Reconstruction for Point Clouds Based on Convexified Image Segmentation”. In: *Journal of Scientific Computing* 54.2-3 (2013), S. 577–602.
- [Lip+13] Vincenzo Lippiello u. a. “Visual Grasp Planning for Unknown Objects Using a Multifingered Robotic Hand”. In: *IEEE/ASME Transactions on Mechatronics* 18.3 (2013), S. 1050–1059.
- [Lod+02] A Lodi, S Martello und M Monaci. “Two-dimensional packing problems: A survey”. In: *European Journal of Operational Research* 141.2 (2002), S. 241–252.
- [Lor+87] William E. Lorensen und Harvey E. Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. In: *SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. ACM, 1987, S. 163–169.

- [Loz+89] T. Lozano-Perez u. a. “Task-level planning of pick-and-place robot motions”. In: *Computer* 22.3 (1989), S. 21–29.
- [Mai+10] Jeremy Maitin-Shepard u. a. “Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding”. In: *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, S. 2308–2315.
- [Mal+10] Alexis Maldonado, Ulrich Klank und Michael Beetz. “Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: IEEE, 2010, S. 2586–2591.
- [Mam+09] Khaled Mamou und Faouzi Ghorbel. “A simple and efficient approach for 3D mesh approximate convex decomposition”. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, S. 3501–3504.
- [Mar+10] Z Marton u. a. “General 3D modelling of novel objects from a single view”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, S. 3700–3705.
- [Mea82] Donald Meagher. “Geometric modeling using octree encoding”. In: *Computer Graphics and Image Processing* 19.1 (1982), S. 85.
- [Mil+01] Victor J Milenkovic und Harald Schmidl. “Optimization-based animation”. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*. New York, New York, USA: ACM Press, 2001, S. 37–46.
- [Mil+03] A.T. Miller u. a. “Automatic grasp planning using shape primitives”. In: *2003 IEEE International Conference on Robotics and Automation*. IEEE, 2003, S. 1824–1829.
- [Mil+99] A.T. Miller und P.K. Allen. “Examples of 3D grasp quality computations”. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*. IEEE, 1999, S. 1240–1246.
- [Mil01] Andrew T. Miller. “Graspit!: A versatile simulator for robotic grasping”. Doktorarbeit. Columbia University, 2001.

- [Mol+13] Bogdan Moldovan, Plinio Moreno und Martijn van Otterlo. “On the use of probabilistic relational affordance models for sequential manipulation tasks in robotics”. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, S. 1290–1295.
- [Moo+88] Matthew Moore und Jane Wilhelms. “Collision detection and response for computer animation”. In: *Proceedings of the 15th annual conference on Computer graphics and interactive techniques - SIGGRAPH '88*. Bd. 22. 4. New York, New York, USA: ACM Press, 1988, S. 289–298.
- [Mor+06] Antonio Morales u. a. “Vision-based three-finger grasp synthesis constrained by hand geometry”. In: *Robotics and Autonomous Systems* 54.6 (2006), S. 496–512.
- [Mor78] Jorge J. Moré. “The Levenberg-Marquardt algorithm: Implementation and theory”. In: *Numerical Analysis*. Bd. 630. Springer Berlin Heidelberg, 1978, S. 105–116.
- [Mos+11] L. Mosenlechner und Michael Beetz. “Parameterizing actions to have the appropriate effects”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, S. 4141–4147.
- [Mul+10] Patrick Mullen u. a. “Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets”. In: *Computer Graphics Forum* 29.5 (2010), S. 1733–1741.
- [Mül10] Reiner Müller. *Klassische Mechanik: Vom Weitsprung zum Marsflug*. 2. Aufl. Berlin: de Gruyter, 2010, S. 488.
- [Nap56] J R Napier. “The prehensile movements of the human hand.” In: *The Journal of bone and joint surgery. British volume* 38-B.4 (1956), S. 902–13.
- [Noz+13] Shunichi Nozawa u. a. “Description and execution of humanoid’s object manipulation based on object-environment-robot contact states”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, S. 2608–2615.

- [Obe+14] Antje Ober-Gecks, Marius Zwicker und Dominik Henrich. “Efficient GPU Photo Hull Reconstruction for Surveillance”. In: *Proceedings of the International Conference on Distributed Smart Cameras - ICDCS '14*. New York, New York, USA: ACM Press, 2014, S. 1–8.
- [Ope14] Open Dynamics Engine. *Manual*. [www.ode.org](http://www.ode.org). [Online; abgerufen 19 Juli 2015]. 2014.
- [Ort10] F Ortmann. “Heuristics for Offline Rectangular Packing Problems”. Doktorarbeit. Department of Logistics, Stellenbosch University, 2010.
- [Pau+08] Mark Pauly u. a. “Discovering structural regularity in 3D geometry”. In: *ACM Transactions on Graphics* 27.3 (2008), S. 1.
- [Pop+11] M. Popovic u. a. “Grasping unknown objects using an Early Cognitive Vision system for general scene understanding”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, S. 987–994.
- [Prz+10] Markus Przybylski, Tamim Asfour und Rudiger Dillmann. “Unions of balls for shape approximation in robot grasping”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, S. 1592–1599.
- [Rao+10] D Rao u. a. “Grasping novel objects with depth segmentation”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, S. 2578–2585.
- [Red+00] S Redon, A Kheddar und S Coquillart. “An algebraic solution to the problem of collision detection for rigid polyhedral objects”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation*. IEEE, 2000, S. 3733–3738.
- [Red+02] Stéphane Redon, Abderrahmane Kheddar und Sabine Coquillart. “Fast Continuous Collision Detection between Rigid Bodies”. In: *Computer Graphics Forum* 21.3 (2002), S. 279–287.

- [Ric+08] Mario Richtsfeld und Markus Vincze. “Grasping of Unknown Objects from a Table Top”. In: *Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments*. 2008.
- [Roa+12] Maximo a. Roa u. a. “Power grasp planning for anthropomorphic robot hands”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, S. 563–569.
- [Rob14] Robotiq. *Adaptive Gripper 3-Finger*. [robotiq.com/products/industrial-robot-hand/](http://robotiq.com/products/industrial-robot-hand/). [Online; abgerufen 17 Mai 2016]. 2014.
- [Rom+12] Eric Rombokas u. a. “Biologically inspired grasp planning using only orthogonal approach angles”. In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2012, S. 1656–1661.
- [Sah+12] A Sahbani, S El-Khoury und P Bidaud. “An overview of 3D object grasp synthesis algorithms”. In: *Robotics and Autonomous Systems* 60.3 (2012), S. 326–336.
- [San+12] Artur Santos u. a. “Understanding the Efficiency of kD-tree Ray-Traversal Techniques over a GPGPU Architecture”. In: *International Journal of Parallel Programming* 40.3 (2012), S. 331–352.
- [Sau+12] Jean-Philippe Saut und Daniel Sidobre. “Efficient models for grasp planning with a multi-fingered hand”. In: *Robotics and Autonomous Systems* 60.3 (2012), S. 347–357.
- [Sax+08] Ashutosh Saxena, Justin Driemeyer und A. Y. Ng. “Robotic Grasping of Novel Objects using Vision”. In: *The International Journal of Robotics Research* 27.2 (2008), S. 157–173.
- [Sch+04] Fabian Schwarzer, Mitul Saha und Jean-Claude Latombe. “Exact Collision Checking of Robot Paths”. In: *Algorithmic foundations of robotics V*. Bd. 7. 2004, S. 25–41.

- [Sch+09] Ruwen Schnabel, Patrick Degener und Reinhard Klein. “Completion and Reconstruction with Primitive Shapes”. In: *Computer Graphics Forum* 28.2 (2009), S. 503–512.
- [Sch+10] Martin J Schuster u. a. “Perceiving clutter and surfaces for object placement in indoor environments”. In: *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2010, S. 152–159.
- [SCH15a] SCHUNK GmbH & Co. KG. *GSW-V*.  
[www.schunk.com/schunk\\_files/attachments/GSW-V\\_EN.pdf](http://www.schunk.com/schunk_files/attachments/GSW-V_EN.pdf).  
[Online; abgerufen 19 Juli 2015]. 2015.
- [SCH15b] SCHUNK GmbH & Co. KG. *Servoelektrische 3-Finger-Greifhand SDH*. [mobile.schunk-microsite.com/fileadmin/user\\_upload/pdf/SCHUNK\\_Technische\\_Daten\\_SDH.pdf](http://mobile.schunk-microsite.com/fileadmin/user_upload/pdf/SCHUNK_Technische_Daten_SDH.pdf). [Online; abgerufen 17 Mai 2016]. 2015.
- [Sha+10] Shy Shalom u. a. “Cone carving for surface reconstruction”. In: *ACM Transactions on Graphics* 29.6 (2010), S. 1.
- [Shi96] K.B. Shimoga. “Robot Grasp Synthesis Algorithms: A Survey”. In: *The International Journal of Robotics Research* 15.3 (1996), S. 230–266.
- [Sim+04] Thierry Siméon u. a. “A General Manipulation Task Planner”. In: *Algorithmic Foundations of Robotics V*. Bd. 7. 2004, S. 311–327.
- [Soa+07] Luciano Soares u. a. “Work stealing for time-constrained octree exploration: Application to real-time 3d modeling”. In: *Eurographics Symposium on Parallel Graphics and Visualization*. 2007.
- [Sol+90] Franc Solina und Ruzena Bajcsy. “Recovery of parametric models from range images: the case for superquadrics with global deformations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.2 (1990), S. 131–147.
- [Ste00] David E. Stewart. “Rigid-Body Dynamics with Friction and Impact”. In: *SIAM Review* 42.1 (2000), S. 3–39.

- [Suá+06] Raúl Suárez, Máximo Roa und Jordi Cornella. *Grasp quality measures*. Techn. Ber. Institut d'Organització i Control de Sistemes Industrials, Universitat Politècnica de Catalunya, 2006.
- [Sun+10] Sung Jo Kwak, Seong Youb Chung und Tsutomu Hasegawa. "Generating a contact state graph of polyhedral objects for robotic application". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, S. 4522–4527.
- [Tan+14] Min Tang, Dinesh Manocha und Young J. Kim. "Hierarchical and Controlled Advancement for Continuous Collision Detection of Rigid and Articulated Models". In: *IEEE Transactions on Visualization and Computer Graphics* 20.5 (2014), S. 755–766.
- [Täu+12] Holger Täubig und Udo Frese. "A New Library for Real-time Continuous Collision Detection". In: *7th German Conference on Robotics (ROBOTIK 2012)*. 2012, S. 108–112.
- [Tay94] G. Taylor. "Point in Polygon Test". In: *Survey Review* 32.254 (1994), S. 479–484.
- [Tho+03] U. Thomas u. a. "Error-tolerant execution of complex robot tasks based on skill primitives". In: *2003 IEEE International Conference on Robotics and Automation*. IEEE, 2003, S. 3069–3075.
- [Tri+10] Lluís Trilla und G. Alenyà. "Planning Stacking Operations with an Unknown Number of Objects". In: *7th International Conference on Informatics in Control, Automation and Robotics*. SciTePress - Science, 2010, S. 348–353.
- [Uck+12] Andre Uckermann, Robert Haschke und Helge Ritter. "Real-time 3D segmentation of cluttered scenes for robot grasping". In: *IEEE-RAS International Conference on Humanoid Robots (2012)*, S. 198–203.
- [Uto+13] Soichiro Uto u. a. "Grasp Planning using Quadric Surface Approximation for Parallel Grippers". In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Shenzhen, 2013, S. 1611–1616.

- [Vah+11] Nikolaus Vahrenkamp u. a. “Bimanual grasp planning”. In: *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2011, S. 493–499.
- [Wat99] Alan Watt. *3D Computer Graphics*. 3. Addison-Wesley, 1999.
- [Wel13] René Weller. “A Brief Overview of Collision Detection”. In: *New Geometric Data Structures for Collision Detection and Haptics*. 2013, S. 9–46.
- [Wer+14] Tobias Werner und Dominik Henrich. “Efficient and Precise Multi-Camera Reconstruction”. In: *Proceedings of the International Conference on Distributed Smart Cameras - ICDS '14*. 2014, S. 1–6.
- [Whi82] D. E. Whitney. “Quasi-Static Assembly of Compliantly Supported Rigid Parts”. In: *Journal of Dynamic Systems, Measurement, and Control* 104.1 (1982), S. 65.
- [Xue+07] Zhixing Xue, Marius und R Dillmann. “Grasp planning: Find the contact points”. In: *IEEE International Conference on Robotics and Biomimetics*. IEEE, 2007, S. 835–840.
- [Zha03] Yan Zhang. “Superquadric representation of scenes from multi-view range data”. Diss. Knoxville: University of Tennessee, 2003.



## EIDESSTATTLICHE VERSICHERUNG

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin erkläre ich, dass ich die Hilfe von gewerblichen Promotionsberatern bzw. –vermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe, noch künftig in Anspruch nehmen werde.

Zusätzlich erkläre ich hiermit, dass ich keinerlei frühere Promotionsversuche unternommen habe.

Mietingen, den 27.5.2015

Unterschrift