Topology-based Representations for Motion Planning and Grasping

Von der Fakultät 5 Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Dmitry Alexandrovich Zarubin

aus Rostov-on-Don

Hauptberichter: Prof. Dr. Marc Toussaint

Mitberichter: Prof. Dr. Oliver Brock

Tag der mündlichen Prüfung: 24.03.2015

Institut für Parallele und Verteilte Systeme der Universität Stuttgart

2015

"Человечество всегда мне представлялось в виде множества блуждающих в тумане огоньков, которые лишь смутно чувствуют сияние, рассеиваемое всеми другими, но связаны сетью ясных огненных нитей, каждый в одном, двух, трех... направлениях. И возникновение таких прорывов через туман к другому огоньку вполне разумно называть "ЧУДОМ".

А. Н. Колмогоров

"Restate my assumptions:

- 1. Mathematics is the language of nature.
- 2. Everything around us can be represented and understood through numbers.
- 3. If you graph these numbers, patterns emerge. Therefore: There are patterns everywhere in nature."

Maximillian Cohen, Pi.

"Die Ewigkeit ist bloß ein Augenblick, gerade lang genug für einen Spaß."

Hermann Hesse, Steppenwolf.

"Arc, amplitude, and curvature sustain a similar relation to each other as time, motion, and velocity, or as volume, mass, and density."

Carl Friedrich Gauss

Acknowledgments

I wish to thank my supervisor Marc Toussaint for being very supportive and open to new ideas. His guidance throughout my PhD studies was invaluable and gave me a great chance to learn. His friendly and helpful attitude created a very enjoyable and inspirational atmosphere in the MLR group.

I am grateful to Oliver Brock for agreeing to be my second thesis supervisor and for providing support for the MLR group during the moving process.

I was lucky to be a part of very exciting and productive TOMSY project. Florian Pokorny, Vladimir Ivan, Sethu Vijayakumar, Danica Kragic and other people involved helped to broaden my vision and inspired me with new ideas.

I owe a lot to my colleagues Peter, Andrea, Vien, Stefan, Stanio and others. We had good times together.

I would like to thank my father Alexander and mother Ekaterina for continuous support and motivation. My brothers Mikhail and Georgy did not let me to overwork. Furthermore, I want to thank all my friends in Berlin and Rostov.

Contents

Zι	Zusammenfassung xi				
Summary x					
1	1 Introduction		ion	1	
	1.1	Abstra	act Representations for Motion Planning	. 2	
	1.2	Grasp	ing and Caging of Complex Objects	. 3	
	1.3	Thesis	of Outline and Contributions	. 5	
		1.3.1	Previously Published Work	. 9	
2	Bac	kgrou	nd	12	
	2.1	Open	Robot Simulator - libORS	. 12	
	2.2	Introd	luction to Topology-based Representations	. 13	
		2.2.1	Linking Numbers And Writhe	. 15	
		2.2.2	Total Gaussian Curvature	. 16	
2.3 Approximate Inference Control (AICO)		oximate Inference Control (AICO)	. 18		
		2.3.1	Discrete Formulation of SOC Problem	. 18	
		2.3.2	LQG case	. 19	
		2.3.3	Method Formulation	. 19	
	2.4	Robot	tic Caging and Grasping	. 20	

CONTENTS

3	Rep	resentations	24			
	3.1	Derivation of Writhe Task Variable	25			
		3.1.1 Writhe Jacobian	27			
	3.2	Derivation of Winding Number Task Variable	28			
	3.3	Geodesic Balls	30			
	3.4	Discrete Gauss Curvature	31			
	3.5	Winding Angles	33			
4	Mo	ion Planning Using Topology-based Spaces	36			
	4.1	Related Work	38			
	4.2	Extended AICO With Tasks in Abstract Spaces	39			
		4.2.1 Expressing Motion Priors in Topology-based Spaces and Cou-				
		pling Spaces	39			
		4.2.2 Probabilistic Model for Final Posture Estimation	41			
		4.2.3 Estimation of the Trajectory Duration	43			
	4.3	Experiments	43			
		4.3.1 Experiments with Writhe Representation	44			
		4.3.2 Reaching Through a Loop Using Writhe Representation	44			
	4.4	Results	46			
5	Gra	Grasp Transfer 5				
	5.1	Related Work				
	5.2	Topological Synergies	54			
		5.2.1 Invariant Representation	55			
		5.2.2 Motion Planning	56			
	5.3	Experiments	57			

		5.3.1	The Generation of the Benchmarking Data	58
		5.3.2	Grasp Transfer Using Motion Planning	58
	5.4	Result	S	60
		5.4.1	Conclusions	62
6	Cag	ging Sy	nthesis	64
	6.1	Relate	ed Work	66
	6.2	Pre-pr	rocessing of Geometric Structures	67
	6.3	Manip	pulators	69
		6.3.1	Caging with a Net of Points	69
		6.3.2	Articulated Manipulators	70
		6.3.3	Positioning and Automatic Closing of the Hands	71
	6.4	Caging	g Heuristics	72
		6.4.1	Circle Caging, \mathbb{S}^1	73
		6.4.2	Sphere Caging, \mathbb{S}^2	73
	6.5	Evalua	ation Methodology	75
		6.5.1	Robustness With Respect to Noise	75
		6.5.2	Physical Simulation	75
		6.5.3	Computational Complexity	76
	6.6	Exam	ples of Caging Grasps	77
	6.7	Exper	iments	77
		6.7.1	Results	79
		6.7.2	Discussion	81
	6.8	Comp	arison of Curvature Integral and Charge Integral Heuristics	82

CONTENTS

7	Con	onclusions 8			
	7.1	Benefi	ts for Motion Planning	89	
	7.2	Benefi	ts for Grasping and Caging	89	
	7.3	Future	Work	90	
\mathbf{A}	Der	ivation	of the Final State Estimators	94	
	A.1	Probal	bilistic Model for 1-step Posture Estimation	94	
		A.1.1	Kinematic Case	95	
		A.1.2	Pseudo-dynamic Case	96	
		A.1.3	Dynamic Case	97	
	A.2	A.2 Posterior Posture & Temporal Optimization			
		A.2.1	Two Approaches to Estimate the Trajectory Duration	99	
		A.2.2	Estimation of the Final State q_T (E-step)	100	
		A.2.3	Gradient Optimization of the Trajectory Duration D	100	
	A.3	A.3 Trajectory Duration Estimation in the Dynamic Case			
		A.3.1	Estimation of the Final State q_T	101	
		A.3.2	Gradient Optimization	103	
Bi	bliog	raphy		105	

viii

List of Figures

1.1	Topology-based representations for character animation	3
1.2	Example grasps with robotic manipulators	4
1.3	Thesis outline	5
2.1	Manipulators	13
2.2	Interaction of two curves described by <i>writhe matrices</i>	16
2.3	Example of Operations on Point Clouds	17
2.4	Standard inference control using <i>Dynamic Bayesian Network</i>	19
2.5	Toy Examples of Grasping and Caging	21
3.1	An illustration of analytical writhe calculation for two segments \ldots .	26
3.2	Winding number of a curve around a point	29
3.3	An illustration of the <i>Geodesic Ball</i> representation	31
3.4	Geodesic balls of different radii	32
3.5	Illustration of <i>Winding Angle</i> representation	33
4.1	AICO in configuration and topology-based space	40
4.2	Box Folding experiment	43
4.3	Unwrapping experiment	45
4.4	Reaching through a loop experiment	46
4.5	Comparison to <i>RRTs</i>	47
5.1	Illustration of grasp transfer problem	51
5.2	Grasping objects using electric flux	54
5.3	Low-dimensional representation of a grasp posture	55
5.4	An illustration of <i>Winding Number</i> in three dimensions	56

5.5	Grasp Transfer experiments	59
5.6	Trajectories in topology-based space	61
6.1	An illustration of curvature-based Sphere Caging heuristic	65
6.2	Example of the preprocessing output	68
6.3	Illustration of caging with a net of points	70
6.4	Examples of manipulators used in our experiments	71
6.5	Illustration of circle caging	73
6.6	A geodesic ball (in red) and generated sphere cages $\ldots \ldots \ldots \ldots$	74
6.7	The effect of additional noise on the original mesh	76
6.8	Examples of generated caging grasps	78
6.9	Models used for the experiments	82
6.10	Stability histograms for curvature- and charge-based heuristics	84
6.11	Force closure histograms for curvature- and charge-based heuristics	85
7.1	An example of KUKA LWR 4 robotic arm reaching through a hollow box	88
7.2	Invariance with respect to slight changes in morphology of the objects .	91
A.1	An illustration of the 1-step inference approach.	95
A.2	An illustration of iterative temporal optimization algorithm	99

List of Tables

5.1	Stability evaluation of transferred grasps	60
6.1	Robustness of sphere caging heuristics w.r.t size of an object	80
6.2	Robustness of circle caging w.r.t size of an object	80

Zussamenfassung

Die Planung und Synthese von menschenähnlichen Bewegungen ist ein offenes Problem in der Robotik-Forschung. Diese Dissertation hat das Ziel Ideen und Methoden aus der Topologie zu verwenden um aktuelle Bewegungsplanungsalgorithmen zu verbessern und die Synthese von geschickten Manipulationen zu ermöglichen. Die erste Hälfte der vorliegenden Arbeit beschäftigt sich mit einem Repräsentationsproblem und wie es mit dem Verfahren von Topologie-basierten Invarianten gelöst werden kann. Dabei leiten wir die Abbildung von abstrakten Räumen in den Konfigurationsraum her. Repräsentationen werden auf der Grundlage ihrer Fähigkeit, mit der sie Informationen über die Interaktion zwischen Objekten extrahieren ausgewählt. Wir schlagen ein Regelungsverfahren basierend auf approximierter Inferenz vor, das mit einer zusätzlichen Schicht der Inferenz erweitert wird um neuartigen Repräsentationen zu bewältigen. Unser Ansatz nutzt eine natürliche Hierarchie von mehreren Repräsentationen und verbessert die Qualität und die Rechenzeit der Bewegungsplanung. Die Vorteile der Verwendung des Writhe Raumes und des Raumes der Winding Numbers werden in der Simulation gezeigt. Wir entwickeln auch ein gradientenbasiertes Verfahren zur optimalen zeitlichen Steuerung, dass in ein allgemeines Planungsframework eingebettet wird.

Die zweite Hälfte der Arbeit ist der Synthese von Greifbewegungen gewidmet. Zuerst betrachten wir das Problem der Übertragung von Griffen mit Topologie-basierten Repräsentationen um die Dimensionalität des Problems zu reduzieren. Die Bewegungplanungsmethoden, die in der ersten Hälfte der Arbeit hergeleitet wurden, werden dann dazu verwendet eine optimale Trajektorie zu erzeugen, die zu der gewünschten Griff-Konfiguration führt. Insbesondere betrachten wir welche Teile eines Zielobjekts zum Ergreifen besser geeignet sind. Wir entwickeln eine neuartige Hand-Repräsentation, die wir Geodesic Ball nennen. Diese Repräsentation ist invariant zu einer bestimmten kinematischen Struktur des Manipulators und ermöglicht eine systematische Analyse der Struktur eines Objekts. Wir entwickeln zwei neue Caging Heuristiken auf der Basis des diskreten Gaußschen Krümmungs Integrale und von Wickelwinkel. Wir zeigen die Wirksamkeit der neuartigen Repräsentation durch Auswertung der Stabilität der erzeugten Hand-Konfigurationen und einen Vergleich mit anderen Heuristiken.

Summary

The planning and synthesis of human-like motions remains an open area of robotics research. This thesis aims to employ ideas and methods from topology in order to improve current motion planning algorithms and to enable the synthesis of dexterous manipulations. The first half of the presented work is devoted to a *representational problem* and how it can be addressed using topology-based invariants. The mapping from abstract spaces to the configuration space is derived. Representations are selected based on their ability to extract essential information about the interaction between objects. We propose to extend an approximate inference control framework with an additional layer of inference in order to cope with novel representations. Our approach exploits a natural hierarchy of several representations and improves the quality and computation time of motion planning. The advantages of using the *writhe space* and the space of *winding numbers* are demonstrated in simulation. We also derive a gradient-based method for temporal optimal control within a common framework.

The second half of the thesis is dedicated to the synthesis of grasping and caging movements. Firstly, we address the problem of *grasp transfer* using topology-based representations in order to reduce the dimensionality of the demonstrated hand postures. The motion planning methods derived in the first half of the thesis will then be used to generate an optimal trajectory leading to the desired grasp configuration. We focus in particular on which parts of a target object are better suited for grasping. We develop a novel *hand-local* representation, which we call a *geodesic ball*. This representation is invariant to a particular kinematic structure of a manipulator and allows for a systematic analysis of an object's structure. We introduce two novel caging heuristics based on the *discrete Gaussian curvature integral* and *winding angles* respectively. We demonstrate the effectiveness of a novel representation by evaluating the stability of generated caging grasps and by comparing it to other heuristics.

Chapter 1 Introduction

Robots and intelligent systems are slowly coming into our everyday life. Machines for cleaning and for controlling home facilities, surgery robots and self-navigating cars are not in the field of science fiction anymore. Instead, there are currently many attempts to change the dominating image of robots being massive state-machines unable to adjust to rapidly changing environment. Those industrial robots are being in service for decades and indeed do not provide a desired level of adaptation. Humans are still capable of tying knots, wrapping an object in a tissue, dexterously manipulating an object, putting a key inside a keyhole or handing over a household tool in a much effective and seemingly effortless way.

Despite great advances in the area of robotics, these tasks are beyond current state-of-the-art systems. Part of the problem lies in the perception of subtle changes in the environment. Articulated and flexible objects are extremely difficult to track with vision systems and slight fluctuations in friction are usually not captured by haptic feedback devices. Yet another problem lies in actual planning and execution of such motion. Well established description of robotic manipulators in terms of joint angles and simple 3D end-effector positions as goals, is not that efficient for describing tasks requiring close interaction of objects. Given how many degrees of freedom a, for example, rope has, the task space would simply explode.

We state here, that part of the motion planning problem can be addressed using an appropriate *representation*. Abstract, topology-based representations capture essential connectivity between sub-parts of objects. They are to some degree invariant to local geometric deformations of described interactions. We shall show, that such coordinate systems are much better suited for several classes of tasks, which can not be solved using classical approaches. We propose methods that can combine the different representations and plan simultaneously to efficiently generate optimal, collision free motion that satisfy constraints.

1.1 Abstract Representations for Motion Planning

Classical representations such as configuration space and end-effector space are not suitable for describing complex interaction between robot and the environment. Certain types of such interaction problems, for example winding around an object or wrapping an object in a tissue , can suitably be described by so-called writhe and winding number representations, as shown in the following chapters.

Task of reaching a certain point with a finger is obviously very different from the task of tying shoes. It is clear, that representation of such problem will affect the performance. One could also see that there is a great demand in multiple layers of task description. These different levels are motivated by intrinsic geometric or topological structures of manipulators and objects involved.

Simultaneous motion planning in abstract and low-level configuration spaces requires more extensive theoretical and methodological investigation. Usually, control tasks of robotic systems are defined in world coordinates, the targets or obstacles can be observed in camera coordinates, and the joint limits of the manipulators are typically described in joint coordinates. Therefore, the main challenge is to derive motion synthesis methods that combine the benefits of reasoning in abstract coordinates while accounting for constraints in the world coordinates.

In work of (Edmond and Komura, 2009), topology-based representations were used for describing interaction of characters in context of computer animation. Embracing an agent with arms of another one (see Figure 1.1 for illustration) is difficult to formulate in configuration space. Considering instead an interaction of body segments separately gives much better generalization to agents of different sizes and positions. This is a good example of a successful fusion of different representations.



Figure 1.1: Topology-based Representations for character animation. Adapted from (Edmond and Komura, 2009)

1.2 Grasping and Caging of Complex Objects

Humans can easily grasp objects in order to perform desired manipulations. We can create a mental picture of the whole reaching and grasping process. Similarly, robots have to imagine, or represent, the sequence of motions leading to the interaction with its environment. There should be an internal representation capable of describing the most important features of possible interactions. In the following we shall briefly review several methods proposed for *grasp transfer* and *grasping and caging*.

Grasp Transfer

The grasp transfer problem can be roughly described as mapping of demonstrated human grasps to a robotic manipulator with different kinematic structure. This problem has been addressed in fields of teleoperation and learning from demonstration. One could distinguish three main approaches. First suggests a mapping of angles between phalanges of the fingers. This method is suitable for robotic hands with similar to human hand kinematics. In the work of Ciocarlie et al. (2007) dimensionality reduction methods have been used to effectively map human grasps onto different robotic hands.

The second approach considers the positions of fingertips as a grasp transfer representation. The relation between spaces of fingertip positions is studied in Peer et al. (2008). This method is suitable for precision grasps transfer, although it is not clear how well it could generalize for hands with dissimilar kinematics or objects with different shapes.

The third most popular approach suggests to map joint angles first to some intermediate space, find a correspondence between postures in that space and then map it

CHAPTER 1. INTRODUCTION



Figure 1.2: Example grasps with robotic manipulators. Schunk hand on the left and Universal Gripper in the middle and on the right (Brown et al., 2010).

back. The technique proposed by Pao and Speeter (1989) uses algebraic transformation from human hand pose into space of robotic manipulators.

Grasping and Caging

There are two main approaches to formalize robotic manipulations. First employs a notion of *force closure* as a model-based concept coming from physical interaction between the hand and an object. According to (Murray et al., 2006), a *force closed* grasping configuration is only achieved when rotations in all directions can be compensated by opposing fingers of the manipulator. In order to check this condition one has to estimate forces applied at contact points based on their normals. Several evaluation measures have been introduced by (Ferrari and Canny, 1992), (Suárez et al., 2006) and others. Given a precise object model one could also consider finding a good grasping posture as an optimization problem (El-Khoury et al., 2012).

Another approach addresses the problem by generating *caging grasps*. It was originally formulated for planar objects by (Kuperberg, 1990). The object is "caged" by a set of points when it can not be moved arbitrarily far away from these points. There was a extensive amount of work for planar objects and only few recent studies have made an attempt to extend the approach to 3D space. In the review of (Rodriguez et al., 2012), caging is considered to be an intermediate step towards a full solid grasp. Caging grasps have potentially wider real world application, since they are not that strictly constrained and could be used e.g. for manipulations of objects with holes (Pokorny et al., 2013) and (Stork et al., 2013b).



Figure 1.3: Thesis outline.

In this thesis, we propose a novel topology-based representation and scoring functions for generation of caging grasps. We claim that these functions are less sensitive to the local noise due to their integral nature. We describe in detail how physically stable caging grasps can be synthesized and evaluated.

1.3 Thesis Outline and Contributions

Figure 1.3 summarizes main topics of the thesis. The key component is **topology-based representations**, which were used for **motion planning**, **grasp transfer** and **caging synthesis**. The figure also shows conference and journal papers which were published during the course of doctoral studies and contributed to the manuscript.

Chapter 2: Background will provide us with a brief introduction to essential concepts needed for better understanding the following chapters. We will introduce a general motion planning problem formulation and discuss how the usage of alternate

representations may affect the resulting motion. We shall give basics of stochastic optimal control and introduce the approximate inference control framework. We will cover notions of topological invariants such as Gauss Linking Integral, which is used as a basis for derivation of our representations. Our caging and grasping heuristics require a basic knowledge of point cloud segmentation, surface reconstruction and related characteristics.

In chapter 3: Representations we will introduce four specific examples of topology-based representations: Winding numbers, Writhe coordinates, discrete Gauss curvature and Winding angles. We shall discuss that some robot interaction problems, e.g. placing an end-effector "inside" or "outside" of a box or wrapping a manipulator around an object, can be conveniently expressed using writhe or winding number representations. We investigate properties of these representations and derive corresponding Jacobians.

Contributions:

- Analytical derivation of Writhe Matrix Jacobian
- Analytical derivation of Winding Number Jacobian
- Development of a novel hand-local representation of Geodesic Ball
- Derivation of Discrete Gauss curvature scoring function over a geodesic ball
- Derivation of Winding Angle scoring function

In chapter 4: Motion Planning Using Topology-based Spaces we explain how a target motion can be defined in alternative *topology-based* representations. We state the difference from joint configuration or end-effector spaces in terms of implications on metric and Voronoi bias of the motion space. We develop novel methods, which allow us to merge different representations for movement generation. We propose a unified motion planning framework, based on an optimal control as an approximate inference problem. It can be formulated as a graphical model and allows for extensions for multiple representations (e.g., topology-based spaces, end-effector and joint spaces). In section 4.2 we elaborate on how a motion generalization can be performed within our framework. Forward and inverse projection of motions from topology-based to joint configuration space are discussed in detail. We demonstrate the advantages of our framework in section 4.4 on several problems, where direct motion planning in configuration space is extremely hard whereas local optimal control using topologybased representations can efficiently find optimal trajectories. *Contributions:*

- The introduction of **topology-based representations** tuned to the domain of robot **motion synthesis and manipulation**, with a strong focus on the interaction of manipulation and the environment.
- A method for **Final state** estimation.
- A method for trajectory duration estimation
- We develop an *extension* of stochastic optimal control framework **AICO** that integrates various representations for motion planning. We also provide a **graph***ical model* that couples motion priors at different layers of representations.
- We design experiments that prove the **benefit** of exploiting **topology-based rep**resentations in problems involving close interactions.

In chapter 5: Grasp Transfer we have engaged our efforts on particularly interesting field of grasping. First, we investigated how already demonstrated or synthesized physically stable grasps can be *transferred* between different hand kinematics. In section 5.2 we propose a novel topologically inspired coordinate representation which we call *topological synergies*. This approach is motivated by the fact that grasping is in many cases similar to wrapping the hand surface around an object. Thus we decided to adapt the representation of winding numbers, described in chapter 3. We define the transfer problem as a stochastic optimization task using the framework developed in chapter 4. This approach allows us to compute not only the final grasp itself, but also a trajectory in configuration space leading to it. We designed and carried out experiments on transfer between a simulated human hand and a 3-finger Schunk hand. For the first time, stability evaluation has been done using the simulation framework PhysX. The results 5.4 demonstrate that our approach makes it possible to transfer a large percentage of grasps. Contributions:

- **Development** of a novel low-dimensional topologically inspired **grasp representation** for the purpose of grasp transfer which describes how much a hand's surface is wrapped around an object.
- Integration of this representation into an AICO planning framework.
- Successful **transfer of grasps** between a simulated human hand and a 3-finger Schunk hand.
- A novel stability evaluation method for benchmarking transferred grasps using a realistic physics simulation carried out in the PhysX simulation software.
- Several extensions of libORS library development of a human hand kinematic model, improvement of an *interface* to the PhysX engine, development of *specific task variables* for grasping problems.

In chapter 6: Caging Synthesis we propose a novel approach for the synthesis of grasps of objects whose geometry can be observed only in the presence of noise. The motivation behind this idea was to increase robustness of grasp transfer method developed in chapter 5. We shifted our focus to the problem of generating caging grasps, which can be seen as a relaxed form of solid grasps. We introduce the idea of using geodesic balls on the object's surface in order to approximate the maximal contact surface between a robotic hand and an object. We derive two types of heuristics which extract local information from approximate geodesic balls in order to find areas that can potentially be used to generate a caging grasp. Our heuristics are based on two scoring functions defined in sections 6.4.1 and 6.4.2. The first one uses an extended version of winding numbers representation - winding angles - measuring how much a geodesic ball on the surface winds around a dominant axis. The second explores using the total discrete Gaussian curvature, introduced in section 2.2.2, of a geodesic ball to rank potential caging postures. We evaluate our approach with respect to variations in hand kinematics, for a selection of complex real-world objects and with respect to its robustness to noise. We show that our method can generate successful grasps even on complex objects.

Contributions:

- We introduce the idea of using **geodesic balls** on an object's surface to approximate the **contact surface** between a hand and an object.
- We develop a novel heuristic based on winding angles designed for generation of circle cages (S¹).
- We develop a novel heuristic based on **discrete curvature integrals**, capable of selecting suitable grasp centre points for **sphere caging** (S²).
- We evaluate our approach in simulation with respect to **noise**, for various **objects** and for several hand **kinematics**: a deformable hand simulation by a net of points, a simulated multi-joint 6-finger hand and a 2-finger hand, a 3-finger Schunk hand and a 5-finger anthropomorphic hand.

1.3.1 Previously Published Work

This thesis enhances our previously published work with extended benchmarking experiments and more detailed methods description. Parts of the work presented in the thesis have been used within EU project "TOMSY". Writhe representation with a motion planning framework was introduced in (Zarubin et al., 2012), the representation of winding numbers was appended in (Ivan et al., 2013). Initial grasp transfer work has been shown in (Zarubin et al., 2013a), caging grasps heuristics were presented in (Zarubin et al., 2013b). Two journal papers are currently in submission and include work on temporal optimization (Rawlik et al., 2014) and on a combination of grasping heuristics and motion planning (Sandilands et al., 2014). The content of this thesis is partially adapted from these papers and only represents work conducted by the author.

Publications

 Zarubin, D., Ivan, V., Toussaint, M., Komura, T., and Vijayakumar, S. (2012). Hierarchical motion planning in topological representations. In *Robotics: Science and Systems (RSS)*, Sydney, Australia

- Ivan, V., Zarubin, D., Toussaint, M., Komura, T., and Vijayakumar, S. (2013). Topology-based representations for motion planning and generalization in dynamic environments with interactions. *I. J. Robotic Res.*, 32(9-10):1151–1163
- Zarubin, D., Pokorny, F. T., Song, D., Toussaint, M., and Kragic, D. (2013a). Topological synergies for grasp transfer. In *Hand Synergies - how to tame the* complexity of grapsing, Workshop, IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany
- Zarubin, D., Pokorny, F. T., Toussaint, M., and Kragic, D. (2013b). Caging complex objects with geodesic balls. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan

Submissions

• Rawlik, K., Zarubin, D., Toussaint, M., and Vijayakumar, S. (2014). An approximate inference approach to temporal optimization for robotics

In preparation

Sandilands, P., Zarubin, D., Pokorny, F. T., Toussaint, M., Komura, T., Kragic, D., and Vijayakumar, S. (2014). Taming variability in robotic manipulation with electrostatics and geodesic balls

Chapter 2

Background

In this chapter we shall review definitions and technical terms needed for understanding the following chapters. Firstly, we discuss a role of representation in robotics problems and how this choice determines the system behaviour. We will introduce notions of topological invariants such as Gauss Linking Integral, Writhe Matrix and Winding Number. After that, we shall formulate basics of stochastic optimal control and introduce the approximate inference control framework. This method treats motion planning as a trajectory optimization problem. Such approach allows us to use Bayesian inference for motion generation under constraints of different nature. Our work on synthesis of caging grasps require also a basic knowledge of surface reconstruction from point clouds and related characteristics.

2.1 Open Robot Simulator - libORS

In simulations robots usually modelled as a set of rigid bodies with connecting joints. Such kinematic trees together with the angles between joints determine the current robot configuration. In order to perform e.g. a grasping motion, given positions of finger tips in world coordinates, one has to compute a sequence of transformations for a given robot kinematics.

We have decided to conduct all our simulations using openly available library libORS developed by (Toussaint, 2012). The simulator offers tools for modelling kinematic structures, includes collision detection libraries and allows for generating tra-



Figure 2.1: A simple actuator with 2 degrees of freedom is shown on the left. A simulation of a Schunk robot penetrating a target area (in red), is shown on the right.

jectories for robots. Figure 2.1 (right) shows an example of a simulated Schunk hand and Schunk arm. Kinematic mapping and Jacobians are computed using chaining of translations and rotations.

The main features of the Open Robot Simulator can be summarized as follows:

- Implementation of manipulators of arbitrary kinematic structure
- Pre-defined "task variables" (motion objectives) (e.g. position, alignment, q itself, collision potentials, orientations)
- Simulation of robot trajectories and estimation of their efficiency
- Straight-forward execution of planned trajectories on real hardware

2.2 Introduction to Topology-based Representations

Robots should be aware of geometric world around and should not, for example, collide with obstacles while executing a sequence of motions required for achievement an abstract goal, e.g. putting a key into a keyhole. Coupling abstract and geometric constraints is a difficult task. Topology-based representations could provide exactly a "missing link" - they describe the core interface between low-level geometric representations and high level, almost symbolic ones. The idea of using such representations comes from the analogy to human behaviour. We do not calculate distance between all possible points on object's surface (as it is often done in classical motion planning) and do not plan how this distance should be changed in order to e.g. touch the object. We rather extract the essential structure of the *interaction*.

Suppose we have a planar manipulator with 2 degrees of freedom (as depicted on Figure 2.5a). Let us define a *configuration space* (C-space) as a space of all components necessary for the description of the current robot configuration - $q \in \mathbb{R}^N$, where $q = (q_1, q_2, ..., q_N)$. In our toy example q is a set of joint angles (α, β) Varying these components will change configuration and the position of the *end-effector* - end of the second bar in our case. The actual *task* for this robot would be to reach a certain position XY with the end-effector. The XY coordinates form the *task space*. One has to define a correspondence between C-space and task space in order to allow a meaningful motion control.

We define a motion problem as an optimization problem with a cost function reflecting task constraints. A successful performance requires computing a sequence of intermediate states α_i , β_i - or *trajectory*. In a basic case shown above, a classical inverse kinematics (IK) method could give us a solution (Paul, 1982). Before applying this method we need to know several properties:

- Kinematic Mapping $\phi: q \mapsto y$ from configuration to the task space
- Jacobian J computed for every part of the robot
- Current and desired position in the task space
- Optimality criteria (in terms of metric in the task space).

In our example, *Kinematic mapping* determines the transformation of the endeffector according to changes in joint angles. *Jacobian* is a matrix of partial derivatives of a kinematic map ϕ . It tells how much a change in configuration space will affect the position XY of the end-effector in the task space. The *metric* of the task space could be a simple Euclidean distance. For more detailed introduction to kinematics in robotics context we shall refer to (Toussaint, 2013).

Essentially the problem of mapping leads to the question of *representation*. The literature (Edmond and Komura, 2009; Tamei et al., 2011) discusses how alternate formulation of a problem using a different representations allows to significantly benefit

in computational costs. The main advantages of using alternate representations can be summarized as follows:

- They can provide a better generalization,
- Naturally describe deformations of flexible and articulated objects,
- Avoid global search and allows of using local optimization,
- Reduce dimensionality of the problem.

In the context of randomized search, such representations alter the Voronoi bias and therefore, the efficiency of RRT or randomized road maps. (Lindemann and LaValle, 2004) demonstrate this effect in the case of RRTs. An alternate representation changes a *metric*, such that a trajectory that is a complex path in one space becomes a simple geodesic in another. Different representations may allow us to express different *motion priors*, for instance, a prior preferring "wrapping-type motions" can be expressed as a simple Brownian motion or Gaussian process prior in one space, whereas the same Brownian motion prior in configuration space renders wrapping motions extremely unlikely.

2.2.1 Linking Numbers And Writhe

We shall introduce several types of topology based spaces which have been investigated in context of animation and robotics. All of them exploit either Euler characteristic or Linking Integrals.

Writhe of two closed curves describes how one curve winds around another (see Figure 2.2 for illustration). The average amount of twists can be computed using Gauss Linking Integral (GLI):

$$GLI(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int_{\gamma_1} \int_{\gamma_2} \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_1 - \gamma_2)}{\|\gamma_1 - \gamma_2\|^3},$$
(2.1)

where γ_1 and γ_2 are interacting curves. The discrete version of GLI is derived in work of Klenin and Langowski (2000). The curves are approximated by discrete segments and writhe is computed for all pairs of segments of these two curves. The *writhe matrix*



Figure 2.2: Examples of close interaction of two curves (upper row) and corresponding *writhe matrices* (lower row). Adapted from (Edmond and Komura, 2009)

is composed of all pair-wise values and gives very intuitive yet mathematically correct representation of the interaction. The total writhe is then can be computed by sum of all matrix elements.

The writhe matrix W can be seen as a detailed description of the relative configuration of two chains. Figure 2.2 illustrates several configurations together with corresponding writhe matrices. The amplitude of the writhe (shading) along the diagonal illustrates which segments are wrapped around each other. Due to discrete nature, the writhe matrix has very attractive properties from robotics point of view. Most of robotic structures can be described as *kinematic chains* and thus can be directly used for computation of linking integrals.

2.2.2 Total Gaussian Curvature

There are many robotics problems which could be solved using algorithms from *computational geometry*. Most of the research in this field is focused on perception problems, e.g. finding two closest points in a point cloud or mesh reconstruction. Nevertheless, concepts and methods of computational geometry can also be applied to control problems. In simple case these abstractions provide an additional source of information for motion planners. In this section , we shall give several informal definitions of concepts from computational geometry, which are used in the following chapters.

A *point cloud* is a set of data points, which can come from laser scanners, stereo or RGBD cameras. It is typical information about environment which is available for robot perception module. The main properties of point cloud data include a set of



Figure 2.3: Noisy point cloud (a), reconstructed mesh (b), and Gaussian curvature (c)

data points $\{P\}$ in world coordinates and (optionally) point normals $\{N\}$. Given a set of data points in 3D one could estimate many interesting geometric and topological properties. The *center of mass* of a set is used for estimation of weight distribution. *Point normals* are usually computed using plane fitting algorithms. Another important operation is *surface reconstruction* - the process of connecting points in order to make triangles (triangulation). For collision detections it is sometimes necessary to cover the whole cloud with a *convex hull*.

The *surface* obtained after triangulation consist of polygons and vertices. It is called *watershed* when there are no holes in it.

Curvature at a certain point measures how much the surface twists in different directions.

Gaussian curvature tells how salient particular part of the surface is. It is 0 for flat surface, positive for spherical and negative for hyperbolic parts.

Total Gaussian curvature is a sum of Gauss curvatures at all points which belong to a certain part of the surface. Some examples of introduced operations are illustrated on Figure 2.3.

The *Euler characteristic* $\chi(S)$ is a topological quantity, which describes shape's structure no matter how much it is bended.

Probably most known and important topological invariant is genus. For orientable surface S, its genus can be computed via Euler characteristic as $g = \frac{\chi(S)-2}{2}$. In simple words genus tells how many holes the object has.

The total Gaussian curvature is connected with Euler characteristic via Gauss-Bonnet theorem (Polthier and Schmies, 2006).

2.3 Approximate Inference Control (AICO)

Unlike their models, real robotic systems have to deal with all kind of noise. Thus the exact solutions can not be implemented directly. Typical approach to this problem is to assume variation in controls and develop motion planning algorithms in probabilistic framework. Stochastic optimal control (SOC) aims to find a control law, which leads to a desired motion under given constraints and accounting for uncertainty in some part of the robotic system. If a cost function is defined then one could formulate an *optimal* control problem w.r.t to this functional.

This alternative problem formulation leads to derivation of approximation methods which would be non-obvious in the classical formulation. In the following we will adopt the approximate inference perspective to propose a specific approximation method to solve the SOC problem.

2.3.1 Discrete Formulation of SOC Problem

Let x_t be the state of the system at certain time point t and let u_t be the control signal. The discrete time stochastic controlled process can then be defined as:

$$x_{t+1} = f_t(x_t, u_t) + \xi_t , \ \langle \xi_t, {\xi_t}^T \rangle = Q_t(x_t, u_t)$$
(2.2)

Where dynamics f_t is non-linear and ξ_t is Gaussian noise with covariance Q_t . For a sequence of controls and states $u_{0:T}, x_{0:T}$ we define a cost function:

$$C(x_{0:T}, u_{0:T}) = \sum_{t=0}^{T} c_t(x_t, u_t).$$
(2.3)

The closed loop SOC problem is to find the policy $\pi_t^* : x_t \to u_t$, that has minimal expected costs:

$$\pi^* = \underset{u}{\operatorname{argmin}} \langle C(x_{0:T}, u_{0:T}) \rangle.$$
(2.4)


Figure 2.4: The graphical model for standard inference control using dynamic Bayesian network. Adapted from (Rawlik et al., 2010)

2.3.2 LQG case

In case in of linear dynamic system with quadratic costs and Gaussian noise (the LQG case) we can integrate controls out according to (Toussaint, 2009):

$$P(q_{t+1} \mid q_t) = \int_u \mathcal{N}(q_{t+1} \mid Aq_t + a + Bu_t, Q) \mathcal{N}[u_t \mid 0, H] = \mathcal{N}(q_{t+1} \mid Aq_t + a, Q + BH^{-1}B'),$$
(2.5)

2.3.3 Method Formulation

Approximate Inference Control (AICO) addresses the problem of optimal control as a problem of probabilistic inference in a dynamic Bayesian network (see Figure 2.4 for illustration and (Toussaint, 2009) for extended definition). Let us consider the dynamic case of the SOC problem, where the state $x_t = (q_t, \dot{q}_t)$. We define the problem of minimizing (the expectation of) the cost

$$C(x_{0:T}, u_{0:T}) = \sum_{t=0}^{T} [c_x(x_t) + c_u(u_t)]$$
(2.6)

where c_u describes costs for the control and c_x describes state dependent task costs.

The dynamics of the robot is described by the transition probabilities $P(x_{t+1}|u_t, x_t)$. The AICO framework translates the corresponding problem into the graphical model (depicted on Figure 2.4)

$$p(x_{0:T}, u_{0:T}) \propto P(x_0) \prod_{t=0}^{T} P(u_t) \prod_{t=1}^{T} P(x_t | u_{t-1}, x_{t-1}) \cdot \prod_{t=0}^{T} \exp\{-c_x(x_t)\}$$

The control prior $P(u_t) = \exp\{-c_u(u_t)\}$ reflects the control costs. The last term $\exp\{-c_x(x_t)\}$ reflects the task costs. The interpretation of this term is that if we had introduced an auxiliary random variable z_t (representing task fulfilment) with

$$P(z_t = 1|x_t) \propto \exp\{-c_x(x_t)\},$$
 (2.7)

then z = 1 if the task costs $c_x(x_t)$ are low in time slice t. The above defined distribution is then the posterior

$$p(x_{0:T}, u_{0:T}) = P(x_{0:T}, u_{0:T} | z_{0:T} = 1).$$
(2.8)

Generally speaking, AICO tries to estimate p, in particular the posterior trajectory and controls. In original work (Toussaint, 2009), this is done using Gaussian message passing (similar to Kalman smoothing) based on local Gaussian approximations around the current belief model.

In work (Rawlik et al., 2012) it has been demonstrated that the general SOC problem can be reformulated in the framework of approximate inference, or more precisely, as a problem of minimizing a Kullback-Leibler divergence.

The AICO approach is very similar to differential dynamic programming (Murray and Yakowitz, 1984) or iLQG (Li and Todorov, 2006) methods with the difference that not only backward messages or cost-to-go functions are propagated but also forward messages. This modification allows AICO to compute a local Gaussian belief estimate $b(x_t) \propto \alpha(x_t)\beta(x_t)$ as the product of forward and backward message and exploit it to iterate message optimization within each time slice.

2.4 Robotic Caging and Grasping

One of the most challenging and interesting problem in robotics is grasping and manipulation of objects. It is a crucial component of many state-of-the-art robotic systems,



Figure 2.5: Toy examples for concepts introduced in the chapter. a) Grasping and b) caging of planar objects

where interacting with the environment is being addressed.

We have already mentioned in Chapter 1, there exist two main concepts in formal description of robotic manipulation. According to (Murray et al., 2006) force-closure grasp can resist any external wrench in arbitrary direction. The idea behind is that forces applied at *contact points* induce internal forces which compensate each other. The most common approach for checking whether a grasp is in force closure is to analyze a *convex hull* made of all contact points with the object. Popular simulators, such as GraspIT (Miller and Allen, 2004) or OpenGRASP (León et al., 2010), test the *force-closure* property by random sampling and applying some quality measure to the synthesized posture. This procedure is obviously prone to the reconstruction errors and perception noise.

Unlike solid grasps, caging is somewhat relaxed form of an interaction with an object. Contacts and consequently forces are not required to be established in such case. As defined by (Kuperberg, 1990) an object is caged when it can not be moved arbitrary far away from the manipulator. There is not necessarily a contact with the object, but rather a restriction of some degrees of freedom. Caging grasps have mainly been studied in connection with simple 2-dimensional polygonal objects where analytic methods provide a solution. An example of grasping and caging of planar objects is given on Figure 2.5.

In 3D space, caging has not been studied that extensive. In work (Diankov et al., 2008), the authors used caging grasps for the manipulation of articulated objects with

handles. The method achieved greater success rate compared to a local contact based approach. Recent work of (Rodriguez et al., 2012) considered cages as an intermediate state towards synthesis of solid grasps.

Chapter 3

Representations

In this chapter we will introduce the topology-based spaces, which can significantly alter the metric and topology of the search space. As we already discussed in chapter 2, points that are near in the topological space can be separated by a significant distance in a configuration space. This insight gives a key to understanding the motivation of selecting particular types of spaces for the robotics needs. A simple linear interpolation in a topological space could lead to complex non-linear movement in configuration space.

Our main objective for exploiting topology-based representations was to find invariants that enable us to represent interactions between robot and its environment more effectively.

Here we will introduce four specific examples of topology-based representations: Winding numbers, Writhe coordinates, discrete Gauss curvature and Winding angles. They are used later in our experiments. Writhe representation has previously been used in the context of computer animation (Edmond and Komura, 2009), attempting to capture the "windedness" of one object around another. Basic introduction to winding numbers and Gauss curvature is given in sections 2.2.1 and 2.2.2. We have extended the previous work and developed methods for forward and backward mapping between the configuration space and the topology-based space.

Writhe, Winding number and Winding angle representations can be formalized by a mapping $\phi : q \mapsto y$ from configuration space to the topology-based space, where $q \in \mathbb{R}^n$ is the configuration state with n degrees of freedom. Our motion synthesis framework, described in chapter 4 require computation of ϕ and its Jacobian J for any q. We call a space with Jacobian J and mapping ϕ , in which goals and constraints can be defined, a *Task Variable*.

The representation of a geodesic ball, although is not topology-based *per se*, serves as an important link connecting object and hand geometries. It allows us to define Winding angle and Discrete curvature scoring functions, which are in turn become topological invariants in the continuous limit. Each representation has its own strength and weaknesses and coupling them can help solve a wider range of problems.

Contributions of this chapter can summarized as follows:

- Analytical derivation of Writhe Matrix Jacobian
- Analytical derivation of Winding Number Jacobian
- Development of a novel hand-local representation of Geodesic Ball
- Derivation of **Discrete Gauss curvature** scoring function over geodesic ball
- Derivation of Winding Angle scoring function

3.1 Derivation of Writhe Task Variable

According to (Klenin and Langowski, 2000) and (Edmond and Komura, 2009), the writhe is a property of the interaction of two kinematic chains. Intuitively, the writhe tells to what degree the two chains are wrapped around each other. Let us describe two kinematic chains by positions $p_{1:K}^{1,2}$ of their joints, where $p_k^i \in \mathbb{R}^3$ is the *k*th point of the *i*th chain. Using standard kinematics, we know how these points depend on the configuration $q \in \mathbb{R}^n$, that is, we have the Jacobian $J_k^i := \frac{\partial p_k^i}{\partial q}$ for each point. The writhe matrix is a function of the link positions $p_{1:K}^{1,2}$.

More precisely, the **writhe matrix** W_{ij} describes the relative configuration of two points (p_i^1, p_{i+1}^1) on the first chain and two points (p_j^2, p_{j+1}^2) on the second where i, jare indexes of points along the first and the second chain respectively. For brevity, let us denote these points by $(a, b) = (p_i^1, p_{i+1}^1)$ and $(c, d) = (p_j^2, p_{j+1}^2)$, respectively (see Figure 3.1 for illustration).



Figure 3.1: An illustration of analytical writhe calculation for two segments. One - r_{ab} - belongs to the manipulator and another - r_{cd} - is a fixed part of the rope object. We assume that only points a and b change their positions according to control signals - in other words, jacobians of these two points are known and we would need to calculate Jacobian of the whole writhe matrix w.r.t these points. Reproduced from (Zarubin et al., 2012).

According to (Klenin and Langowski, 2000) an analytical solution for Gauss linking integral between two segments r_{ab} and r_{cd} can be expressed as a sum of inverse trigonometric functions of normalized cross products of vectors connecting points of two segments. So that:

$$W_{i,j} = \left[\arcsin(n_a n_b) + \arcsin(n_b n_c) + \arcsin(n_c n_d) + \arcsin(n_d n_a) \right] \operatorname{sign} \left[\overline{ab}^T (\overline{ac} \times \overline{cd}) \right]$$
(3.1)

Where i, j are indices of segments of the respective chains:

$$n_a = \frac{r_{ac} \times r_{ad}}{\|r_{ac} \times r_{ad}\|}, n_b = \frac{r_{ad} \times r_{bd}}{\|r_{ad} \times r_{bd}\|}, n_c = \frac{r_{bd} \times r_{bc}}{\|r_{bd} \times r_{bc}\|}, n_d = \frac{r_{bc} \times r_{ac}}{\|r_{bc} \times r_{ac}\|},$$
(3.2)

where n_a, n_b, n_c, n_d are normalized normals at the points a, b, c, d with respect to the opposing segment (see Figure 3.1). The solution of this integral is based on an analogy with the solid angle formed by all view directions in which segments (a, b) and (c, d) intersect (Klenin and Langowski, 2000) multiplied by an appropriate sign.

3.1.1Writhe Jacobian

We are going to use writhe matrix representation for our motion synthesis experiments, which require the computation of Jacobian w.r.t to all segments in a chain. Given the analytical expression of the interaction of link positions $p_{1:K}^{1,2}$ above, we can compute the Jacobian using the chain rule.

Consider the example described on Figure 3.1, we assume that change of joint angle at point a is proportional to the change of position of point b. Thus we will calculate the Jacobian of the writhe matrix with respect to the Jacobian of these points - coordinates of the tip of a body. Let us calculate explicitly the derivative of, e.g. $\arcsin(n_a n_b)$.

$$\operatorname{arcsin}\left(\frac{r_{ac} \times r_{ad}}{\|r_{ac} \times r_{ad}\|} \frac{r_{ad} \times r_{bd}}{\|r_{ad} \times r_{bd}\|}\right) = \operatorname{arcsin}\left(\frac{A}{B}\frac{C}{D}\right)$$
(3.3)

where A, C are vectors and B, D are scalars.

_

$$\left(\arcsin\left(\frac{A}{B}\frac{C}{D}\right)\right)' = \frac{1}{\sqrt{1 - \left(\frac{A}{B}\frac{C}{D}\right)^2}} \left(\frac{A}{B}\frac{C}{D}\right)' =$$
(3.4)

$$\frac{1}{\sqrt{1 - \left(\frac{A}{B}\frac{C}{D}\right)^2}} \left(\left(\frac{A}{B}\right)'\frac{C}{D} + \frac{A}{B}\left(\frac{C}{D}\right)' \right) =$$
(3.5)

$$=\frac{1}{\sqrt{1-(\frac{A}{B}\frac{C}{D})^2}}\left(\left(\frac{A'B-AB'}{B^2}\right)\frac{C}{D}+\frac{A}{B}\left(\frac{C'D-CD'}{D^2}\right)\right)$$
(3.6)

So now it is required to have only derivatives of cross products for calculation A'and C'. Thus:

$$(r_{ac} \times r_{ad})' = (r_{ac})' \times r_{ad} + r_{ac} \times (r_{ad})'$$
(3.7)

$$(r_{ad} \times r_{bd})' = (r_{ad})' \times r_{bd} + r_{ad} \times (r_{bd})'$$

$$(3.8)$$

$$(r_{bd} \times r_{bc})' = (r_{bd})' \times r_{bc} + r_{bd} \times (r_{bc})'$$

$$(3.9)$$

$$(r_{bc} \times r_{ac})' = (r_{bc})' \times r_{ac} + r_{bc} \times (r_{ac})'$$
(3.10)

derivative of the Euclidean norm

$$\frac{\delta \|f(x)\|}{\delta x} = \frac{f(x)^T}{\|f(x)\|} f(x)'$$
(3.11)

Consequently all B', D' values can be replaced by respective $B' = \frac{A}{B}A'$ and $D' = \frac{C}{D}C'$

Once we have an analytical Jacobian expression, we can derive simpler metrics from the full writhe matrix. For example, the Gauss linking integral, which counts the mean number of intersections of two chains when projecting from all directions, can be computed as a sum of all elements of the writhe matrix. In our experiments, we have also used the vector $w_j = \sum_i W_{ij}$ as a representation of the current configuration. In general, writhe, however, does not provide a unique mapping to joint angles which is why we require additional constraints and cost terms especially in scenarios where wrapping motion is not dominant.

3.2 Derivation of Winding Number Task Variable

In case when a problem can be described as a wrapping motion in 2D, we can use a special case of the writhe representation. Recall that the winding number of a closed curve $\gamma : [0,1] \to \mathbb{R}^2$ not containing the origin in \mathbb{R}^2 is an integer determining how many times γ 'wraps around the origin' with the sign being determined by the orientation of γ (see Figure 3.2).

Definition 3.2.1 Let $\gamma : [0,1] \to \mathbb{R}^2$ be closed, smooth and suppose γ does not traverse the origin. Then the winding number $w(\gamma) \in \mathbb{Z}$ around the origin is given by

$$w(\gamma) = \frac{1}{2\pi} (\theta(1) - \theta(0)).$$
 (3.12)

where $\gamma(t) = (r(t), \theta(t))$ in polar coordinates, and where r denotes the radial and θ the angular coordinate.

The quantity $w(\gamma)$ for a closed curve is a topological invariant and does not change under continuous deformations of γ not intersecting the origin. This formula allows to compute how γ wraps around the origin, also in the case of piece-wise linear curves. In latter case, the quantity is not a topological invariant anymore.



Figure 3.2: Winding number of a point p_c surrounded by the doubly wound curve: $\hat{w} = 2$.

We compute the Winding number using the approximate algorithm derived in (O'Rourke, 1998) which is based on calculating inverse trigonometric functions of the scalar product of two normalized vectors, formed by consequent points p_i and p_{i+1} on a curve and a central point p_c :

$$\hat{w} = \frac{1}{2\pi} \sum_{i=1}^{n-1} \arccos\left(\frac{(p_i - p_c)^\top (p_{i+1} - p_c)}{|p_i - p_c||p_{i+1} - p_c|}\right)$$
(3.13)

where n is the number of points along the curve. This scalar continuous function can be thus viewed as a simplification of a writhe representation defined in 3.1. In our experiments, outlined in section 4.3, we assume that all joints and points lie within one plane. We use the standard kinematics to define the points p_i depending on the configuration $q \in \mathbb{R}^n$. We can therefore use the chain rule to compute the Jacobian of the winding number. For convenience let us denote $A = (p_i - p_c), C = (p_{i+1} - p_c)$ and $B = |p_i - p_c|, D = |p_{i+1} - p_c|$, then the derivative

$$\hat{w}' = \frac{1}{2\pi} \sum_{i=1}^{n-1} \left(\arccos(\frac{A}{B} \frac{C}{D}) \right)' =$$
 (3.14)

$$= \frac{1}{2\pi} \sum_{i=1}^{n-1} \frac{-1}{\sqrt{1 - (\frac{A}{B}\frac{C}{D})^2}} \left(\left(\frac{A'B - AB'}{B^2} \right) \frac{C}{D} + \frac{A}{B} \left(\frac{C'D - CD'}{D^2} \right) \right).$$
(3.15)

The latter equation is very similar to the one we derived for the writhe matrix Jacobian.

3.3 Geodesic Balls

The notion of a geodesic ball is a key concept of our approach to caging and grasping. The motivation behind is to find a representation which would describe both – an object and a hand. Standard approaches towards the synthesis of both caging grasps and force-closed grasps simplify the geometry of the robot's hand by working only with a discrete number of contact points or caging points respectively (Asfour et al., 2006; Vahedi and van der Stappen, 2009).

In this thesis, we propose to instead think about an approximation of the 2D contact surface that a robot hand can make with an object. A geodesic ball can be seen as a rough approximation of a hand surface projected on an object. Technically, we do not project a hand, but rather approximate it by a patch of surface of a certain radius, which usually equals to a finger length. The geodesic attribute comes from the analogy to a shortest path between two points on a curved surface. In our case we include all vertices along all shortest edge-paths, going from center to the border. One could imagine a geodesic ball representation as if the hand surface was projected for all possible rotations of the wrist.

More formally, we model the maximal contact surface between the robot's hand and a given closed surface S as a geodesic ball $B_r(p)$ of radius r and centred at a point $p \in S$. Such a geodesic ball then represents the 'hand-local' geometry of M for a robot hand whose fingers are approximately of length r. We think of $B_r(p)$ as a first order approximation of the projection of the robot's hand onto the object as visualized in Figure 3.3.

Since the calculation of an exact geodesic ball $B_r(p)$ is computationally expensive, we use the mesh M with vertex set V and consider a rough approximation of $B_r(p)$ by a sub-mesh $G_r(p) \subseteq M$ instead. Since all triangles in M are approximately equilateral, according to section 6.2, we use Dijkstra-like algorithm to determine $G_r(p)$ as follows: we approximate the distance between two vertices $p, q \in V$ by calculating the number



Figure 3.3: We approximate the projection of the robot hand's surface onto the turtle model by a geodesic ball whose radius r is proportional to the robot hand's finger length. Adapted from (Zarubin et al., 2013b).

of vertices in the shortest edge-path between them and scaling the result by the mean edge length. $G_r(p)$ then denotes the sub-mesh containing all triangles whose vertices are of distance at most r from p. Figure 3.4 displays $G_r(p)$ for a fixed vertex p and for four different radii of increasing size. Since r varies with the size of the robotic hand, different geometric information is captured by $G_r(p)$ for differing hand size.

3.4 Discrete Gauss Curvature

We have already introduced a topological invariant of total Gauss curvature in section 2.2.2. Here we shall formulate a novel representation based on discrete Gauss curvature.

Let us consider a mesh M with vertex set V(M) and where F(v) defines the set of faces containing vertex $v \in V(M)$. In this case, the total Gauss curvature K(v) for a vertex $v \in V(M)$ can be calculated by $K(v) = 2\pi - \sum_{f \in F(v)} \theta(v, f)$ (Polthier and Schmies, 2006), where $\theta(v, f)$ denotes the angle of the face f at vertex v. Recall that point v is called Euclidean, spherical or hyperbolic if K(v) = 0, K(v) > 0 or K(v) < 0respectively (Polthier and Schmies, 2006).

According to the Gauss-Bonnet theorem for a smooth closed oriented surface $S \subset \mathbb{R}^3$ the integral of the standard Gaussian curvature should satisfy $\int_S K \, \mathrm{dVol} = 2\pi \chi(S)$, where $\chi(S)$ denotes the Euler characteristic of S, introduced in 2.2.2. It is invariant under certain continuous deformations of S, so that we have $\chi(S) = 2 - 2g$, where g is the genus of the surface.



Figure 3.4: Geodesic ball approximations $G_r(p)$ (coloured white) for a fixed vertex p and for radii r of increasing size on an isotropically remeshed bunny object. The size of the radius r corresponds to the finger-length of different robotic hands. Adapted from (Zarubin et al., 2013b).

As discussed in (Polthier and Schmies, 2006; Reshetnyak, 1993), the Gauss-Bonnet theorem has a discrete analogue for a closed polyhedral surface $M \subset \mathbb{R}^3$:

$$\sum_{v \in V(M)} K(v) = 2\pi \chi(M),$$
(3.16)

where the sum is over vertices in M and K now denotes the *discrete total Gauss* curvature.

While K(v) shares some properties with the standard Gaussian curvature, another more closely related version of K which we shall not use in this work can be provided by normalizing K(v) by the 'mixed area' around the vertex v (see (Meyer et al., 2002)



Figure 3.5: Illustration of winding angle representation: The blue line highlights vertices close to the plane $\Pi_r(p)$ for a fixed vertex p and orthogonal to the main axis $A_r(p)$ of the vertices lying on the depicted geodesic ball $G_r(p)$ with red border. The wrapping of $G_r(p)$ around the horse is approximated using the angles between the end-points of the blue curve around the central axis of $G_r(p)$. The upper right sketch illustrates the solid angle $Angle_{Z_r(p)}(a, b)$ (in green) between the centre of mass $Z_r(p)$ and the projections of the end-points of $P_r(p)$ onto $\Pi_r(p)$. Reproduced from (Zarubin et al., 2013b)

for details).

3.5 Winding Angles

Writhe and Winding number representations have shown an outstanding improvement of the motion planning algorithms within our experiments (as described in the section 4.4). Yet for the problem of synthesis of caging grasps they were not suitable to apply directly. We developed another version of winding numbers for non-closed curves in order to describe the wrapping of a hand around an object or its sub-part.

Recall that the winding number of a closed curve $\gamma : [0,1] \to \mathbb{R}^2$ not containing the origin in \mathbb{R}^2 is an integer determining how many times γ 'wraps around the origin' with the sign being determined by the orientation of γ .

CHAPTER 3. REPRESENTATIONS

According to section 3.2, the winding number defines how many times a curve is wound around a point on a 2D plane (3.2). We extend this notion to a 3D euclidean space and make use of a geodesic balls representation, introduced in section 3.3.

We suggest to measure the winding of a closed curve $\gamma : [0,1] \to \mathbb{R}^3$ around the object as follows: for each vertex p of our model mesh M, we consider a set of vertices of $G_r(p)$ as a set of 3D points. We can thus compute a major axis $A_r(p)$, centre of mass $Z_r(p)$ and a plane $\Pi_r(p)$ going through $Z_r(p)$ which is orthogonal to $A_r(p)$. This plane should separate the geodesic ball into two approximately equal sub-parts along some "cutting curve". We shall compute an edge-path $P_r(p)$ in the model mesh approximating $\Pi_r(p) \cap G_r(p)$. On Figure 3.5 the approximation of such "cutting curve" is shown in blue.

To measure the amount of winding of $P_r(p)$ around the object we finally project the path $P_r(p)$ onto the plane $\Pi_r(p)$ and compute the winding angle

$$W_r(p) = 2\pi - Angle_{Z_r(p)}(a, b),$$
 (3.17)

where $Angle_{Z_r(p)}(a, b)$ denotes the solid angle between the centre of mass $Z_r(p)$ and the projections of the end-points of $P_r(p)$ onto $\Pi_r(p)$ (as illustrated on Figure 3.5).

Winding angle $W_r(p)$ takes values in $[\pi, 2\pi]$. This quantity can be computed without having to represent the projection $\gamma : [0, 1] \to \Pi_r(p)$ of the curve $P_r(p)$ in polar coordinates centred at $Z_r(p)$ as $\gamma(t) = (r(t), \theta(t))$. Furthermore, since for our purposes γ satisfies $\pi \leq |\theta(1) - \theta(0)| \leq 2\pi$ in all but degenerate cases, we opted to use the winding angle $W_r(p)$ to measure the amount of wrapping of γ around $Z_r(p)$.

Chapter 4

Motion Planning Using Topology-based Spaces

Many robotic problems concern close interactions of the robot and its environment, consisting of complex objects. While standard motion planning methods describe motion in configuration space, such tasks can often more appropriately be described in spaces that reflect the interaction more directly.

In chapter 1 we have discussed a problem of wrapping of arms around an object, e.g. embracing a human. Defined in joint space, such a motion is complex and varies greatly depending on the target object. When describing the motion more directly in terms of the interaction of arm segments with object parts (e.g. using the writhe matrix representation that we introduced in chapter 3) we gain better generalization to other objects.

Topology-based spaces described in previous chapter may provide better metrics or topology for motion generation. In ideal case, they should enable local optimization methods, operating in such spaces, to solve problems that would otherwise require more expensive global search in configuration space. Target scenarios can be, e.g. multi-link articulated robots reaching through small openings and complex structures, surfaces wrapping around objects and fingers grasping and manipulating objects. In such cases, the abstract representations greatly simplify the problem of motion generalization as well as planning.

In this chapter we introduce our method for exploiting topology-based representa-

tions for motion synthesis in an optimal control context. We decided to formulate the approach within the framework of Approximate Inference Control (Rawlik et al., 2012), which is closely related to differential dynamic programming (Murray and Yakowitz, 1984) or iLQG (Li and Todorov, 2006) (more details are given in the section 2.3). The framework allows us to use a graphical model to describe the coupling of geometric and topology-based representations.

The main contributions of this chapter are:

- The introduction of **topology-based representations** tuned to the domain of robot **motion synthesis and manipulation**, with a strong focus on the interaction with the environment.
- Final state estimation algorithm.
- Trajectory **duration** estimation algorithm.
- We develop an **extension** of stochastic optimal control framework **AICO** that integrates various representations for motion planning. We also provide a **graph**ical model that couples motion priors at different layers of representations.
- We design experiments that prove the **benefit** of exploiting **topology-based representations** in problems involving close interactions.

These extensions contributed to several publications :

- Zarubin, D., Ivan, V., Toussaint, M., Komura, T., and Vijayakumar, S. (2012). Hierarchical motion planning in topological representations. In *Robotics: Science* and Systems (RSS), Sydney, Australia
- Ivan, V., Zarubin, D., Toussaint, M., Komura, T., and Vijayakumar, S. (2013). Topology-based representations for motion planning and generalization in dynamic environments with interactions. *I. J. Robotic Res.*, 32(9-10):1151–1163
- Rawlik, K., Zarubin, D., Toussaint, M., and Vijayakumar, S. (2014). An approximate inference approach to temporal optimization for robotics

This chapter is structured as follows, we will first review related work on the use of topology-based representations for character animation and motion planning in section 4.1. Section 4.2 presents our approach to coupling topological and configuration space representations in an optimal control setting through the Approximate Inference Control (AICO) (Toussaint, 2009; Rawlik et al., 2012, 2014)) framework. This modification naturally leads to an extension that includes random variables for both the topological and configuration space representations, with their specific motion priors coupled via the graphical model. In Section 4.3 we describe experiments on using the proposed methods to solve motion synthesis problems like folding a box, unwrapping and reaching through a loop. These problems are extremely difficult or even infeasible without exploiting abstract representations.

4.1 Related Work

Dimensionality reduction of the state space of robots has been of interest of robotics research for decades. From the machine learning perspective, dimensionality reduction techniques are presented by feature extraction and projections to lower dimensions. For instance, in (Bitzer and Vijayakumar, 2009), a latent manifold in joint space was computed using Gaussian process from sample configurations produced by an expert. This manifold was, however, defined by samples from a valid trajectory in joint space and it did not capture state of the environment directly.

In order to deal with problems involving close interactions, it is necessary to introduce an abstract space based on the spatial relations between the interacting robots and objects. Several researchers have developed knotting robots that generalize the status of the strands and plan the motions using probabilistic road maps (Takamatsu and et.al., 2006; Wakamatsu et al., 2006). These works represent the rope state based on how it is overlapping with itself when viewing it from a certain direction (Dowker and Morwen, 1983). The transition between states was achieved by moving the end points toward a specific direction. Such a representation is not very useful in practice due to the view-dependence and the difficulty of moving the rope. Abstract topology-based representations that describe the interactions between 1D curves using their original configurations were proposed for motion synthesis (Edmond and Komura, 2009; Tamei et al., 2011) and for classifying paths into homotopy groups (Bhattacharya et al., 2011). In (Edmond and Komura, 2009), a representation based on the Gauss Linking Integral was suggested to generate winding movements. Positions of characters were mapped from new representation to the joint angles using the least squares method.

In (Tamei et al., 2011), the same representation was applied for synthesis of motion of a robot that puts a shirt on a human. The coupling between the new representation to the low level representation was learned through demonstrations by humans. The approach required the corresponding sample points to be fixed and defined a priori.

Another interesting idea, described in (Bhattacharya et al., 2011), was to abstract the paths connecting a start point and the end point. The paths were only classified into homotopy classes and there was no discussion about the mapping from the topological representation to the low level control coordinates. Besides, this representation was only applicable for simple one dimensional curves and was not useful for describing the relationship between 2D surfaces, which is often the case in robotics applications.

4.2 Extended AICO With Tasks in Abstract Spaces

4.2.1 Expressing Motion Priors in Topology-based Spaces and Coupling Spaces

In this section we shall extend the Approximate Inference Control framework (introduced in 2.3) in order to cope with abstract representations. We start with a discussion on motion priors and how they can be used in conjunction with topological spaces. In order to estimate the posterior, the controls u_t can be integrated out, implying the following motion prior:

$$P(x_{t+1}|x_t) = \int_u P(x_t|u_{t-1}, x_{t-1}) P(u_t) du .$$
(4.1)

This motion prior emerges as the combination of the system dynamics and our choice of control costs $c_u(u_t)$ in configuration space. For Linear-Quadratic systems discussed



Figure 4.1: AICO in configuration and topology-based space. The blue arcs represent the approximation used in the final-state posterior estimation. Adapted from (Zarubin et al., 2012).

in section 2.3.2 it is a linear Gaussian prior.

As we have already mentioned in chapter 2 the choice of the representation affects the Voronoi bias, the metric, or the topology. In some sense, successful trajectories are likely to be "simpler" (easier to find, shorter) in an appropriate space. More formally, in Machine Learning terms, this is expressed as a prior. From this perspective topologybased spaces are essentially a means to express priors about potentially successful trajectories. In our case we employ the linear Gaussian prior in a topology-based space to express the belief that trajectories may appear "simpler" in a suitable abstract space.

One has to be aware that using Gaussian motion prior only in abstract topologybased space is not sufficient to solve general motion synthesis problems. AICO algorithm will return a posterior, which does not directly specify an actual state trajectory or control law on the *joint level*. Besides, we would ignore tasks originally defined on the joint level. In order to address these issues, we need mechanisms to merge inference in topology-based and state space. We do so by coupling topology-based and joint state representations in AICO's graphical model framework as depicted on Figure 4.1. The bottom layer corresponds to the classical AICO setup, with the motion prior

$$P(x_{t+1}|x_t) = \int_u P(x_t|u_{t-1}, x_{t-1}) P(u_t) du$$
(4.2)

implied by the system dynamics and control costs. In addition, it includes the task costs represented by

$$P(z_t = 1|x_t) = \exp\{-c_x(x_t)\}.$$
(4.3)

The top layer represents tasks in topology-based space with a given linear Gaussian motion prior $P(y_{t+1}|y_t)$. Two types of tasks are coupled by introducing additional factors

$$f(x_t, y_t) = \exp\{-\frac{1}{2}\rho \|\phi(q_t) - y_t\|^2\},\tag{4.4}$$

which essentially aim to minimize the squared distance between the topology-based state y_t and the one computed from the joint configuration $\phi(q_t)$, weighted by a precision constant ρ .

Note that a local linearization of ϕ and Jacobians of the topology-based spaces, derived in chapter 3, are enough for efficient Gaussian message passing between layers of our model. According to the definition of the factors 4.4, the topology-based state y_t can be seen as an additional task variable for the lower level inference, similar to other potential task variables like end-effector position or orientation.

4.2.2 Probabilistic Model for Final Posture Estimation

Instead of message passing over configuration and topology-based variables in the full factor graph given in 4.1, which could lead to loops, we approximate probabilistic inference in several stages.

- 1. We first approximate directly a final-state posterior $b(x_T)$ for the end-state x_T which fulfils the task defined in *configuration space*. The details are given below and in the Appendix A.
- 2. We compute a trajectory posterior in topology-based space, accounting for the coupling of this final-state posterior to the topology-based representation.
- 3. We then project this posterior down to the configuration space, using AICO coupled to the topology-based space via factors introduced above.

As you might have noticed from the inference model 4.1, the initial inference in topology-based space only accounts for the task at the final time step. In order to overcome this limitation, we decided to iterate inference between levels. For the problems investigated in our experiments, the approximation scheme above is sufficient. Final state posterior estimation computes an approximate belief

$$\hat{b}(x_T) \approx P(x_T \mid x_0, z_{0:T} = 1)$$
(4.5)

about the end-state given the initial state and conditioned on the task. This approximation neglects all intermediate task costs and assumes linear Gaussian system dynamics of the form

$$P(x_t \mid x_{t-1}) = \mathcal{N}(x_t \mid A_t x_{t-1} + a_t, W_t) .$$
(4.6)

We can integrate the system dynamics,

$$P(x_T \mid x_0) = \sum_{x_{1:T-1}} \prod_{t=1}^T P(x_t \mid x_{t-1}) , \qquad (4.7)$$

which corresponds to the blue arc in 4.1. For stationary linear Gaussian dynamics, we have

$$P(x_T \mid x_0) = \mathcal{N}(x_T \mid A^T x_0 + \sum_{i=0}^{T-1} A^i a_i, \sum_{i=0}^{T-1} A^i W A^{'i}) , \qquad (4.8)$$

where superscript on A stands for a power of matrix, defined iteratively $A^i = A * A^{i-1}$. To estimate $\hat{b}(x_T)$, we condition on the task,

$$P(x_T \mid x_0, z = 1) = \frac{P(z_T = 1 \mid x_T) \ P(x_T \mid x_0)}{P(z_T = 1 \mid x_0)} \ . \tag{4.9}$$

Since we assume the distribution $P(x_T | x_0)$ to be Gaussian, using a local Gaussian approximation of the task $P(z_T = 1 | x_T)$ around the current mode of $\hat{b}(x_T)$, $P(x_T | x_0, z = 1)$ can be approximated with a Gaussian as well. We iterate this by alternating between updating the final state estimate $\hat{b}(x_T)$ and re-computing the local Gaussian approximation of the task variable. The exact derivation of the final state estimators for different cases of system dynamics is given in the Appendix A.



Figure 4.2: Initial (left) and final (right) state of simulation of box folding motion using winding numbers representation. Two skeletons that are being wound around the centre of the box are overlaid over the sides of the box. Reproduced from (Ivan et al., 2013).

4.2.3 Estimation of the Trajectory Duration

The temporal optimization is based on the idea of one-step inference described in the previous section. We estimate the final state of the robot configuration under simple kinematic constraints and try to apply the likelihood maximization with respect to the duration of the trajectory. It is somewhat similar to the notion of Macro-states in (He et al., 2010), but differs in terms of splitting trajectory into keyframes. Presented algorithm is much simpler than EM used for temporal optimization in (Rawlik et al., 2010), since we first estimate the exact final configuration and have an intuition about the possible time consumption. More details are given in the Appendix A.

4.3 Experiments

Having a motion planning framework, accounting for constraints in topology-based spaces, we agreed to conduct first experiments on a toy example. Being the most intuitive, we chose the representation of winding numbers and designed a simulated chain of joints suitable for wind-like motions. We decided to demonstrate the elegance and simplicity of this abstraction on a folding a box from layout experiment (See Figure 4.2).

The goal was to generate a folding motion, but instead of controlling all joint angles directly (although trivial in this case), we set only two scalar goals for our optimization problem - degrees of twist for the two chains or winding numbers as described in section 3.2. Close up or folding motion is then generated by the approximate inference in dynamic Bayesian network. Winding number, despite of its simplicity, has good generalization properties (e.g. it is invariant w.r.t. number of elements in chain) and can be used in arbitrary combination with other topological or geometrical goals.

4.3.1 Experiments with Writhe Representation

Writhe representation, introduced in section 3.1, is particularly interesting to use for describing an interaction of two chains of joints. As an example of a possible application of the this abstraction, we simulated a manipulator, consisting of 20 cylindrical segments and a hand with three fingers, making in total 29 degrees of freedom. Initial configuration of this rope-like manipulator was set to be twisted two and a half times around a striped pole, giving us approximately 900° of writhe density (See 4.3(a)).

The task was to generate a trajectory which should lead to grasping of the black cylinder without colliding with the striped stick (Figures 4.3(a),4.3(b)). Obviously, a local feedback approach using Inverse Kinematics will experience failure in this task. Naive AICO, with only end-effector task variable and collision avoidance task variable activated, is unable to converge to a solution due to a deep local minima in this space. Solution of this planning problem in configuration space would require to use exploratory motion planning methods such as Rapidly-exploring Random Trees (RRT).

On the other hand, a successful trajectory can be well captured as a linear interpolation in writhe space and projected back to the configuration space using the coupling method described in 3.1. Figure 4.3(e) illustrates an example of a unwrapping trajectory in topology-based space when all rows of writhe matrix are summed up into one column, representing the current state.

4.3.2 Reaching Through a Loop Using Writhe Representation

As described above, Writhe space is a suitable representation for tasks that involve interactions with chains—or loops—of obstacles. In order to demonstrate that, we



(c) initial writhe matrix (d) final writhe matrix

Figure 4.3: The experimental task is to grasp the object without collisions. Corresponding writhe matrices (c, d) are depicted below the configurations (a, b) - the darkness represents the amplitude of the writhe value. Each row in writhe space evolves over time as shown in (e). Reproduced from (Ivan et al., 2013).

have designed another task - reaching through a loop. The rim of an object forms the first chain (a loop of segments) and several joints of a robotic Schunk form another chain as shown on Figure 4.4. The interaction between these two chains can thus be described by the writhe representation.

Classically this problem would be addressed by conditioning on the end-effector position and collision avoidance. The advantage of using Writhe as a description of the interaction is in defining the task as a *relative* configuration of the robot and the loop—this relative description remains useful also when the box is moved dynamically.

Writhe matrices corresponding to the initial (left) and the final (right) configura-



Figure 4.4: Result of successful reaching motion. The task was defined in topologybased space. Two lower plots show the writhe matrix for chains defined on the Schunk hand and on the circle. The final Writhe matrix contains a peak around the link which passes through the loop, indicating it is fully wrapped by the circle segments.

tions are shown on Figure 4.4. The target in the Writhe space does not strictly define the task for all arbitrary positions of the circle, unlike the unwrapping task in the previous section 4.3.1. It allows us to define sub-goals such as precisely controlling the end-effector position via another task variable ("remain inside of the box"). This became possible due to the extension of AICO and appending an additional type of tasks to the graphical model. We can therefore achieve accurate manipulation within a spatially constrained dynamic environment.

4.4 Results

Extended AICO was conditioned on the end-state in writhe space y_T , estimated using our method from section 4.2.2. It was able to generate locally optimal trajectories, consisting of 50 time steps, in only few iterations, requiring a relatively small number of usually expensive collision checks (less than 1000). Initial comparison with RRTs planning for this reaching task revealed a dependence of the performance on distance between end-effector and object position. Moreover, full costs of obtained trajectories were on average 100 times higher than those generated with the local optimizer. Here, the end-state in configuration space q_T was given as a target for RRTs.



Figure 4.5: Performance of planning algorithms for unwrapping task. Computational time is proportional to the number of collision detection calls. Reproduced from (Zarubin et al., 2012).

For more systematic benchmarking of our planning platform, we have designed a set of final configurations q_T , gradually increasing the relative angle to be unwrapped. This sequence of final states was given as goals to uni- and bi-directional RRT planners. The results show that for simple trajectories (e.g. in case of nearby lying objects) all methods have no difficulties, whereas starting with one and a half of full twist, unidirectional search fails and bi-directional significantly slows down (See figure 4.5).

In this comparison, the RRTs solved a somewhat simpler problem than our system: For the RRTs we assumed to know the final state q_T in configuration space – estimated final pose using method from section 4.2.2 as the target q_T for RRTs. This is in contrast to our planning framework, where we use the end pose estimate only to approximate a final topology-based state y_T and then use the extended AICO to compute an optimal trajectory (including an optimal q_T) conditioned on this final topology-based state.

Therefore, the RRT's problem is reduced to growing to a specific end state. We applied the standard method of biasing RRT search towards q_T by growing the tree 10% of the time towards q_T instead of a random sample of the configuration space.

Knowing q_T also allowed us to test bi-directional RRTs, each with 10% bias to grow towards a random node of the other tree. Even under such simplified constraints, the RRT-based planners used significantly more computations for interpolating the states when complex winding is required. Furthermore, RRTs output non-smooth paths whereas AICO produces (locally) optimal dynamic trajectories since it minimizes dynamic control costs.

In summary, a combination of approximate inference methods and task-specific alternate representation gives an outstanding improvement of both quality and computational costs of the resulting motion. Particular types of interactions are much more natural to describe in Writhe or Winding coordinates. "Brute force" approach using randomized search in the C-space is not optimal and fails for such interactions.

Chapter 5

Grasp Transfer

In this chapter we show how grasp transfer problem can be addressed using our motion planning framework developed in chapter 4. We start with a discussion on how invariants defined in chapter 3 can be employed for this task. Then we move to combination of the motion planning in topology-based spaces, described in chapter 4, and grasping representations. We claim such a fusion to be advantageous for generating hand postures, similar to the demonstrated by a human. We conclude with experiments involving motion planning and winding numbers representation implemented in one system. This chapter is an extension of previous author's work (Zarubin et al., 2013a).

Contributions of this chapter according to Zarubin, D., Pokorny, F. T., Song, D., Toussaint, M., and Kragic, D. (2013a). Topological synergies for grasp transfer. In Hand Synergies - how to tame the complexity of grapsing, Workshop, IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany:

- **Development** of a novel low-dimensional topologically inspired **grasp representation** for the purpose of grasp transfer which describes how much a hand's surface is wrapped around an object.
- Integration of the above mentioned representation into an AICO planning framework.
- Successful **transfer of grasps** between a simulated human hand and a 3-finger Schunk hand.



Figure 5.1: Illustration of grasp transfer problem. The initial recording of a human grasp posture (left) is done using the Polhemus magnetic tracking sensors. The reconstructed grasp with a simulated human hand is in the middle and a result of motion planning for a robotic arm and a Schunk hand is on the right.

- A novel **stability evaluation** method for benchmarking transferred grasps using a **realistic physics** simulation carried out in the PhysX simulation software.
- Several extensions of libORS library development of a **human hand** kinematic model, improvement of an **interface** to the PhysX engine, development of **specific task variables** for grasping problems.

Despite of huge progress in grasping research, human demonstrations still remain the "ground-truth" data for motion synthesis algorithms. People do not only perform significantly better in manipulation tasks but also capable of doing so even if one or two fingers are damaged. Some fields of robotics research focus explicitly on transferring such demonstrations to robotic manipulators. It might be much cheaper to "copy" or transfer the grasp, instead of generating a desired posture "from scratch". The question of representation of a stable grasp and how to transfer such grasps between different hand kinematics remains an open area of research (Bicchi and Kumar, 2000).

Currently, the most popular approach towards describing good grasps is to analyse contact points and normals between the hand's surface and the object. This approach is known as *force closure analysis* and estimates how forces applied, e.g. on tips of opposite fingers, would correlate and increase the stability of the resulting grasp. In practice though, it might be difficult to achieve theoretically predicted positions of

CHAPTER 5. GRASP TRANSFER

finger tips due to noise in robotic sensors.

Even if a perfect model is provided there is no unique solution, which could generate a stable grasp. Popular simulators, e.g. such as GraspIT (Miller and Allen, 2004), offer a computationally expensive brute force search. The algorithm samples hand postures all over the object and computes the force closure values until a good candidate is found. The number of samples is usually of order of thousands, which makes it difficult to carry out on a real robot. Besides, the resulting grasp is only "optimal" w.r.t a particular hand kinematics and particular mesh.

In this chapter, we claim that the choice of *representation* of the state space can significantly simplify the transfer problem and overcome the disadvantages of the force closure analysis. We have already discussed before in chapters 3 and 4, that operating in the joint space coordinates directly is not always the best idea. In grasping context, there exists no obvious way of transferring a grasp to a new kinematic hand structure with different geometry and number of joints.

We suggest to exploit topology-based representations for describing hand postures. These spaces have shown good generalization properties (see section 4.4) and improvement of the motion synthesis involving close interactions. We show here how *winding number* representation can be employed for transferring grasps between different hand kinematics.

Grasping motion can be divided in several phases such as forming a pre-shape posture, approaching the object, closing of the fingers. When described directly in configuration space, such a motion can vary greatly depending on the target object. When describing the motion more directly in terms of the interaction of hand segments with object parts, we gain, as shown below, better generalization to manipulators of different kinematic structures.

5.1 Related Work

There are many robotic manipulators available on the market today. A human hand has a state space with more than 20 degrees of freedom (Buchholz and Armstrong, 1992; Dragulescu et al., 2007; Santello et al., 1998), while, for example, a Schunk robot hand has only 7 degrees of freedom. This inequality in degrees of freedom makes the question of transferring a human hand posture to a robot hand highly non-trivial. The problem of course is not about making a full copy of a human hand, but rather a general problem of correspondence between different hand kinematics.

Evidence from neuroscience and electrophysiological experiments on human subjects (Arbib et al., 1985; Santello et al., 1998) suggested that humans use some sort of *low-dimensional representation*, to which authors refer as *postural synergies*, in order to perform a grasping motion. In later work (Santello and Soechting, 2000), researchers introduced *force synergies* in attempt to define a linear subspace, which could represent a subset of grasping forces. A similar concept of *postural synergies* is derived in (Romero et al., 2012).

In the context of programming by demonstration, the exact transfer of a demonstration from a human subject to the robot is required for teaching new skills by the robot (Friedrich et al., 1998; Kang et al., 1997). Similarly, in the context of teleoperation (Rohling and Hollerbach, 1993) it is highly desirable to be able to transfer grasps from a human to a robotic hand (e.g. in case of manipulating household objects (Hu et al., 2005)).

Several approaches to the transfer problem have been investigated. The early work (Rohling et al., 1993) discussed three broad methods for transfer given by a) linear joint mapping - which is applicable if the robot's hand kinematics are very similar to those of the human hand, b) pose mapping - using least squares fitting and c) fingertip mapping. In more recent work of (Ekvall and Kragic, 2004) and (Kang et al., 1997), approaches related to the notion of virtual fingers have been explored. In this case, a subset of the fingers of the human hand was manually mapped to one or more fingers of a robot hand.

Dimensionality reduction is the key feature that the virtual finger, fingertip and synergy approaches share. They all attempt first to reduce the number of dimensions needed to describe a hand pose. Similarly, in the case of postural synergies (Santello and Soechting, 2000; Bicchi et al., 2011), a lower dimensional linear subspace of the full joint space is extracted using principal component analysis.

Another work of (Sandilands et al., 2013) exploits electrostatic coordinates for grasp transfer. Although, this approach is rather computationally expensive, it leads to good results for certain types of objects. The experimental setup is very similar to



Figure 5.2: Grasping objects using electric flux. Adapted from (Sandilands et al., 2013).

the one used in our approach (see Figure 5.2 for illustration).

Recent work (Romero et al., 2012) investigates cases where a linear dimensionality reduction might be suboptimal and explores the use of the nonlinear GP-LVM dimensionality reduction framework. The representation which we develop here falls into this non-linear class of state space representations, but while (Romero et al., 2012) attempt to *find such a representation by data-analysis*, in this work, we consider *designing* such a representation by finding analogies to methods and representations in topology.

5.2 Topological Synergies

In this section we propose a novel approach to transfer grasps between different hand kinematics. We develop a low dimensional topologically inspired coordinate representation of winding numbers. We call a set of such numbers *topological synergies*. In section 4.3 we have shown the efficiency of writhe matrix and winding number representations in robotics context. The multi-layered extension allowed to speed up the motion planning and to simplify definition of related tasks. Proposed here topological synergies were developed as an application of winding number representation to more specific field of motion planning - grasp transfer. The recent work of (Pokorny et al., 2013) suggested to use winding numbers to generate caging grasps on objects with holes. Here we extend a standard definition of winding numbers to a \mathbb{R}^3 and propose a novel grasp descriptor for synthesis of grasping motions.


Figure 5.3: We define curves going through the thumb and other fingers, resulting in 4 curves in total for the human hand. The winding of these curves around the centre of the object is then considered to be a low-dimensional representation of a grasp. Adapted from (Zarubin et al., 2013a).

5.2.1 Invariant Representation

We assume that for any robotic or human hand with $n \ge 2$ fingers f_1, \ldots, f_n , one of the fingers can be labelled as a *thumb*. In practice, the thumb for a robotic hand such as the Barrett, Schunk and Shadow hand, can be easily identified.

We can consequently define piecewise linear curves $\gamma_1, \ldots \gamma_{n-1}$, such that $\gamma_i(0)$ starts at the tip of the thumb f_n and ends at the tip of f_i by traversing the joints of f_n , then going through the center of the base of the hand and continuing through the joints of f_i . On Figure 5.3 such curves $\gamma_1(0)$ are depicted in white. A similar definition of winding curve is also used in (Pokorny et al., 2013).

Given that the classical winding number $w(\gamma) \in \mathbb{Z}$, for a closed curve γ in \mathbb{R}^2 computes the winding around a point $p \in \mathbb{R}^2$ by calculating the total change in angular coordinates, we now define a similar quantity for piecewise linear curves in \mathbb{R}^3 .

Let us define $\gamma : [0, 1] \to \mathbb{R}^3$ to be the piecewise linear curve connecting the points $X_0, \ldots, X_n \in \mathbb{R}^3$ by linear line segments from X_i to X_{i+1} , for $i = 0, \ldots, n-1$ such that a fixed central point $p \in \mathbb{R}^3$ is not contained in the image of γ (see Figure 5.4 for



Figure 5.4: An illustration of winding number topological invariant in three dimensions. The actual winding value (sum of $\alpha_i = angle_p(X_i, X_{i+1})$) for the non-closed curve (solid line) is computed with respect to a reference point p, which is not necessary lying on the same plane with γ .

illustration). For such a curve we can define

$$\hat{w}(\gamma) = \frac{1}{2\pi} \sum_{i=0}^{n} \text{angle}_p(X_i, X_{i+1}),$$
(5.1)

where $\operatorname{angle}_p(X_i, X_{i+1})$ denotes the angle between the vectors $X_i - p$ and $X_{i+1} - p$.

In case when γ is a closed curve that lies completely in a plane containing p, the above defined quantity is just the usual winding number. We will show in this chapter that $\hat{w}(\gamma)$ can be reliably used in grasping applications to quantify how much a curve, representing a robotic manipulator, is 'wrapped' around an object.

5.2.2 Motion Planning

In line with our previous research, we also incorporate motion planning using the extended AICO framework (see chapter 4 for details). The method allows us to combine our topological tasks with traditional constraints such as collision avoidance. One of the motivations for the use of alternate representations is that these spaces can better describe "wrapping-type motions", which are typical for grasping. Given the framework developed above, it is straight forward to incorporate winding numbers as a simple Gaussian process prior.

In the context of grasp transfer, we shall use quantities $\hat{w}(\gamma_i)$, as a compact reduced representation of the "wrappedness" of the hand surface. As we already defined above, the posture of a human hand grasp p can be described by four numbers $\hat{w}(\gamma_1), \ldots, \hat{w}(\gamma_4)$. We reduce it further and shall consider the quantity y = $(\hat{w}(\gamma_a), \hat{w}(\gamma_b)) \in \mathbb{R}^2$, where γ_a, γ_b are the two curves with the highest winding \hat{w} around the centre of mass of the target object. The value of y represents a description of a grasp in a topology-based space. We shall compute y for a human demonstration grasp and shall use this quantity as a topological task. We setup goals for the Schunk robotic hand to attain the same winding quantity with respect to the two curves γ'_1, γ'_2 running through the Schunk robotic hand.

Recall that in the extended AICO framework (see section 4.2), a motion prior in the topology-based space is coupled with the configuration space by introducing additional factors

$$f(x_t, y_t) = \exp\{-\frac{1}{2}\rho \|\phi(q_t) - y_t\|^2\}.$$
(5.2)

This factor essentially tries to minimize the squared distance between the state in topological space, in our case - space of winding numbers $y = (\hat{w}(\gamma_a), \hat{w}(\gamma_b)) \in \mathbb{R}^2$, and the one computed using forward mapping from the configuration space $\phi(q_t)$. This extension allows us to treat the winding representation of a grasp as an additional task variable.

5.3 Experiments

All experiments have been carried out in libORS, described in the section 2.1. In addition to extended AICO motion planning, we have implemented an interface to NVidia PhysX engine. We have also created a kinematic model of a human hand based on a 3d scan with 20 degrees of freedom (DOF) (see Figure 5.3, left), which served as a "demonstrator". As a target manipulator we chose a model of the Schunk robot hand (Figure 5.3, right). The robotic hand was attached to the Schunk arm and motions were generated for the full 14 DOF robot (arm + hand).

5.3.1 The Generation of the Benchmarking Data

Instead of recording training data set from human subjects, we decided to automatically generate a large set of grasps in simulation. This way we avoided solving a problem of fitting the data from the magnetic sensors from different subjects to one kinematic model of the hand, but still obtained a statistically significant amount of test data.

We have opted to use grasps generated with a part-based grasp planning system – BADGr (Huebner, 2012). The benchmarking set consisted of stable grasps with a human hand model for two objects. These models - a bottle and a hammer - are shown on Figure 5.5.

Models were first approximated through a set of oriented bounding boxes and then decomposed according to a convexity index. For each reachable side of the boxes, 4 grasp hypotheses were generated by aligning the approach vector to its normal and the 4 orientations to its 4 edge vectors. The best grasp was then chosen according to quality function. More details of the grasp planning process can be found in (Huebner, 2012).

We tested the initial set of 251 grasps for physical stability using the simulation engine PhysX. We copied the whole configuration - the hand and the object into the physical environment with gravity and friction forces. After that, we performed a sequence of random rotational motions of the hand with the object for a 1000 of simulation steps. A grasp was considered to be stable if an object did not fall on the "floor" during this procedure.

After testing stability in PhysX, we selected 67 stable grasps for the hammer and 32 stable grasps for the bottle model. We have used this set as our benchmarking data set in order to evaluate the transfer to the Schunk robot hand.

5.3.2 Grasp Transfer Using Motion Planning

We addressed the transfer problem in two phases:

- 1) motion planning using AICO framework and
- 2) an automatic closing of the fingers.



Figure 5.5: Left half: examples of bottle grasps generated with human hand and transferred Schunk hand grasps. Right half: examples for the hammer model. Adapted from (Zarubin et al., 2013a).

This is one of the main differences of our approach from other similar methods. They focus mainly on the second phase or try to find the contact points. We instead let the motion planning algorithm decide what an *optimal configuration* (position and orientation) of the hand should be.

For the first phase we used the winding numbers extracted from human hand example grasps from the benchmarking data set. We measured winding around the centre of mass of the object with respect to all four curves $\gamma_1, \ldots, \gamma_4$, running along the human hand. After that, we selected the winding values with with the largest winding values (e.g. γ_1, γ_2) and defined these values as goals for winding task variables, derived in section 3.2.

In addition, we have also included a collision task variable, which had very high costs for states penetrating an object. Since the robot platform we had was not mobile and not all orientations can be reached by the robot arm, we imitate the approach direction of the simulated human grasp by rotating the object. This heuristic ensures that the grasp approach direction is feasible for the robot arm. For the second phase, we

Objects	Hammer	Bottle
Total number of grasps	101	151
Stable grasps	67	32
Autoclose based on initial human hand configuration	13%	25%
Grasp transfer using topological synergies	53%	63%

Table 5.1: Stability evaluation of transferred grasps

performed an automatic finger closing - we moved each segment of the finger gradually until a contact or a certain small distance was achieved.

5.4 Results

The benchmarking of the transferred grasps has been done using a realistic physical simulation using PhysX engine as described in the section 5.3.1. In this case we again copied the hand (without an arm) to the physical environment and performed a sequence of rotational motions. This procedure determined a *successful* classification of the grasp.

Recall, that our approach to transfer problem consists of two phases. Firstly, the trajectory is generated with AICO conditioned on winding number and collision task variables. The efficiency of the planner is discussed in the chapter 4. In this work, we focused on benchmarking the second phase - the actual grasp posture after an automatic finger closing.

In order to show that the motion planning is advantageous compare to direct "transplantation" of the hand, we tested this hypothesis on our dataset. We placed the Schunk hand into the same position and set the same orientation as it was for the human hand. After that, we closed the fingers and checked the stability as described above.

Results of the experiments can be seen in the table 5.1. The direct "transplantation" method has a very low success rate with respect to the total number of stable grasps. Whereas our approach, exploiting topological synergies, produces much higher



Figure 5.6: Trajectories in topology-based space consisting of $y_t = (\hat{w}(\gamma_1(t)), \hat{w}(\gamma_2(t)))$. The graph displays steps t in the simulation (horizontal axis) against the winding values of $\hat{w}(\gamma_1(t))$ and $\hat{w}(\gamma_2(t))$. These curves were defined as going through thumb-index and thumb-middle fingers respectively. Adapted from (Zarubin et al., 2013a)

percentage of successful grasps. At least half of the transferred grasps were stable under realistic physical conditions. These results prove that a novel grasp representation based on the notion of *winding numbers* is capable of transferring from 20 DOF human hand to a 7 DOF Schunk hand.

It is interesting that the amount of information necessary for our method is significantly less than required by other approaches - only two winding values and the orientation of the object is sufficient for transferring grasps of simple objects. The trajectory in topology-based space is also comparatively simple (see Figure 5.6). However, due to non-linear properties of the mapping from the topology-based space used in our planner, we obtain a complex grasping behaviour in the configuration space.

5.4.1 Conclusions

In this chapter we have continued to explore topology-based representations in context of motion planning. We applied our framework to the problem of *grasp transfer*. We have used *winding numbers*, which measured how much a hand is wrapped around a target object. The results demonstrated that "topological synergies" can be used to successfully transfer grasps between a human hand and a Schunk hand. This is partly due to the fact that we were able to use topology-based task goals in conjunction with more traditional task variables, e.g. collision potentials, in order to synthesize complex motions.

Note that we chose the centre of mass as a reference point for the winding measure. This is a strong assumption and reduces the applicability of this particular approach. The shape of object is being completely ignored. The question of how the structure of the shape can affect grasping is addressed in the next chapter 6. The results and methodology developed here have served as a good basis for our further research and led to the development of other interesting topology-based representations.

Chapter 6

Caging Synthesis

A popular approach for grasp synthesis in robotics is based on *local contact-level tech*niques. Given an accurate mesh-representation of the object and friction coefficients, these methods use a certain grasp quality scoring function Q to generate hand configuration. The function Q can be defined in terms of contact points and surface normals on the object (*e.g.* as in (Ferrari and Canny, 1992)).

The most popular scoring function is based on computation of concatenation of possible forces applied at finger tips in normal directions - thus assuring *force-closure grasps* (Murray et al., 2006). The main drawback of these methods is that one needs to know friction coefficients and normals. They typically are not particularly robust in the presence of noise since even a small variation in the vertex positions on a mesh can result in large deviations of the estimated normal vectors. Besides, the actual calculation of Q function requires exhaustive collision checks within some simulator (e.g. GraspIT (Miller and Allen, 2004)) and have to be done for every hand separately.

Caging grasps are deprived of these drawbacks. They do not necessarily form contact with the object, but rather restrict its mobility. So that the object remains within a small distance from the hand surface. Caging grasps have mainly been studied in connection with simple 2-dimensional polygonal objects where analytic methods provide a solution.

Encouraged by success of using topology-based representation for general motion planning and grasp transfer, described in previous chapters, we have developed a novel heuristic synthesis algorithm, which exploits topology of the object. For the first time,



Figure 6.1: An illustration of our curvature based sphere caging heuristic. The red area in the left picture highlights an approximate geodesic ball on the surface with a large value of our discrete curvature integral. The two other plots demonstrate an application this heuristic - generated grasps with a net of points (middle) and a 3-finger Schunk hand (right). Adapted from (Zarubin et al., 2013b).

our method is capable of caging complex 3D objects. This chapter enhances previous author's work (Zarubin et al., 2013b).

The contributions of this chapter can be summarized as follows:

- We introduce the idea of using **geodesic balls** on an object's surface to approximate the **contact surface** between a hand and an object.
- We develop a novel heuristic based on **winding angles** designed for generation of **circle cages** (S¹).
- We develop a novel heuristic based on **discrete curvature integrals**, capable of selecting suitable grasp centre points for **sphere caging** (S²).
- We evaluate our approach in simulation with respect to **noise**, for various **objects** and for several hand **kinematics**: a deformable hand simulation by a net of points, a simulated multi-joint 6-finger hand and a 2-finger hand, a 3-finger Schunk hand and a 5-finger anthropomorphic hand.

Most research on caging consider mainly planar scenarios with fingers represented by points or discs such as in (Vahedi and van der Stappen, 2009). Unlike this approach, we are interested in the synthesis of caging grasps in 3D, for a complex object and using a real robotic manipulator. To our best knowledge, there exist no analytic solution to the general caging grasp synthesis problem. That is why we decided to explore a heuristic approach which enables us to:

- a) synthesize likely caging grasp configurations using information about the 'handlocal' geometry of an object
- b) evaluate different robot hand kinematics for the purpose of generating caging grasps.

Our approach consists of three main parts: a suitable object mesh representation described in section 6.2, a representation of the object/robot hand interaction introduced in section 3.3 and a quality scoring function for two types of caging heuristics which we call circle and sphere caging (defined in section 6.4).

For *circle caging*, which is motivated by caging an object by a curve which is almost closed, we develop a method for choosing appropriate grasping points using *winding angles*. For *sphere caging*, which is motivated by the idea of enclosing an object as much as possible by a geodesic ball, our approach is based on integrated discrete Gaussian curvature.

6.1 Related Work

We have already discussed two main concepts in grasping research in chapter 2. One approach uses force-closure measure, which tells if the object can resist external wrenches in arbitrary directions. This measure provides a ranking of potential grasp hypothesis. This concept is embedded in popular simulators such as GraspIT (Miller and Allen, 2004) and OpenGRASP (León et al., 2010). They use random sampling or heuristics in order to determine stable grasp configurations. An example of a sampling based approach is described in the work of (Borst et al., 2003). The more recent work (Saut and Sidobre, 2012) integrates grasp ranking procedure and sampling with a suitable object representation. The force closure measure can be computed once the local contact geometry of the object is known.

The second concept of a *caging grasp* relies on analysis of the global geometry of the object. The early work of (Rimon and Burdick, 1994) considered only plannar objects. Caging was defined there as a situation when the object could not be moved

arbitrarily far away from a fixed set of points in the plane. An interesting work on the relation between grasping and caging is presented in (Rodriguez et al., 2012). There, a caging grasp is considered to provide a useful waypoint towards a stable force-closure grasp. In work (Diankov et al., 2008), the authors investigated caging grasps for the manipulation of articulated objects with handles such as doors and windows. They generated a set of caging grasps on such handles and achieved greater success rate compared to a local contact based approach. In (Stork et al., 2013a), a caging approach based on topological features of objects with holes was investigated.

A few papers have studied curvature of object's model in order to identify grasping configurations. In work (Calli et al., 2011), it is assumed that concave points of a 2D elliptic Fourier descriptor of an object are most suitable for grasping. Good candidates are preselected by means of curvature extrema and then evaluated further using the concept of force-closure.

Another related work (El-Khoury and Sahbani, 2010) proposes a grasping algorithm for unknown 3D objects. There, Gaussian curvature is employed during the segmentation of the object. The labelling of sub-parts of the object is then done with respect to neighbourhoods of extrema of Gaussian curvature. This method depends on the use of point-wise approximations of Gaussian curvature and is hence rather unstable under noise. While the robustness with respect to noise can be increased by an additional smoothing step, we shall take a different approach in our work since we will work with a discrete version of Gaussian curvature defined for any mesh.

6.2 Pre-processing of Geometric Structures

We assume that we have a potentially noisy mesh representation M of an object of interest available. Such object meshes are used *e.g.* in the work of (Asfour et al., 2006) and can be obtained using laser range or Kinect sensor data. Since we will be interested in efficiently computing approximate geodesic balls, defined in the following section, on M, and in order for our discrete curvature K to be distributed evenly across the mesh, we would like to work with a mesh with approximately equal triangle size. For this purpose, we chose isotropic remeshing as a pre-processing step. Isotropic remeshing preserves the shape of the surface and provides an almost equilateral triangulation.



Figure 6.2: Example of the preprocessing output. a) Original pony model b) Remeshed model.

We start with an input mesh M and perform modifications using the tool (Fuhrmann et al., 2010) in three steps:

- oversampling and a sub-division is applied to M,
- vertices are uniformly resampled,
- the positions of the vertices are optimized using area equalization and Lloyd's relaxation method.

The resulting triangle mesh then has almost equal edge lengths (e.g. the one shown on Figure 6.2). We chose parameters so that the maximal deviation between edge-lengths was about 15% of the mean edge length, but this deviation can be significantly reduced further by increasing the number of triangles per mesh.

Another important part of the preprocessing is a convex decomposition of the object. The *PhysX Engine* (NVIDIA, 2012) which we are using for realistic physical simulations requires all shapes to be convex in order to efficiently compute collisions. For this purpose, we used the convex decomposition library (Ratcliff, 2009) and generated a decomposition of each object into approximately 100 convex meshes.

6.3 Manipulators

We shall now relate the representation of geodesic balls and winding angles (defined in sections 3.3 and 3.5) to the kinematics of a robot hand. We shall compute caging scoring functions, introduced in 6.4, on an object mesh using geodesic balls. The exact radius of a ball depends exclusively on kinematics of a particular manipulator and usually equals to a single finger length.

In our experiments, which we shall describe later in section 6.7, we would like to show that our heuristics are also robust with respect to different hand kinematics. For this purpose, we need to be able to define a correspondence between a pose of a robotic hand and an approximate geodesic ball $G_r(p)$ for \mathbb{S}^2 case or a correspondence between a pose and the projection of the path $P_r(p)$ onto the plane $\Pi_r(p)$ for \mathbb{S}^1 case.

In this section we describe several types of manipulators: an artificial net of small spheres distributed over an approximate geodesic ball, a simulated multi-joint 6 finger hand (for sphere caging), a multi-joint 2 finger hand (for circle caging), a realistic simulation of a 3 finger Schunk hand and a 5 finger anthropomorphic hand (see Figure 6.4 for illustration).

6.3.1 Caging with a Net of Points

This is the closest approximation of a geodesic ball. A net of points is a net of small disconnected meshes enveloping an object. As we already mentioned in previous sections, the key element of our method is a projection and approximation of a hand's contact surface by a geodesic ball. It is logical then to study the robustness of our approach if we had a robotic manipulator whose contact surface was very similar to such a ball. The recently proposed Universal Gripper (Brown et al., 2010) is an example of such deformable manipulator coming close to this idealization.

In our simulation environment (see section 2.1 for details), we produced a artificial structure designed to imitate a soft robotic manipulator wrapping around a geodesic ball on the object. For every vertex inside $G_r(p)$, we created a small sphere and moved it along the normal direction by a fixed offset away from the object. After that, we gradually shifted all spheres back towards the object along the normal direction until a small threshold distance was achieved (see Figure 6.3 for illustration).



Figure 6.3: Illustration of caging with a net of points. a) $G_r(p)$ ball on a dumbbell object, b) an artificial net, created for every vertex c) a "contracted" net, used in our experiments.

For circle caging, we chose only those vertices on the mesh close to the edge-path $P_r(p)$ and applied the same procedure to obtain spheres lying near $P_r(p)$ but outside the object. This procedure then simulates a contact surface obtained using a deformable robot hand and which is approximated by a net of points. Examples for both \mathbb{S}^1 and \mathbb{S}^2 cases are given in the section 6.6 (second column).

6.3.2 Articulated Manipulators

The 'Hexapus' and The 2-finger Hand

The next approximation of a geodesic ball we thought about was a 6-finger, 'hexapus', gripper with multiple joints within each finger (Figure 6.4b). It was also created within the libORS simulator (see section 2.1). The structure is aimed to be similar to biologically inspired soft robotic actuators described in (Deimel and Brock, 2013) and (Ilievski et al., 2011). Every finger consists of 6 square segments coupled via revolute joints between each other and with the hand base. We have also defined a 2-finger hand (displayed in Figure 6.4e) by using only 2 opposing fingers of this 'hexapus' hand. The latter manipulator is designed for winding around circular subparts of the object and is better suited for testing the winding angle representation. The 'hexapus' hand is used only for \mathbb{S}^2 caging and the 2-finger hand only for \mathbb{S}^1 caging.



Figure 6.4: Examples of manipulators used in our experiments. a, d) A Schunk hand b, e) a multi-joint artificial hand c) a KCL metamorphic hand f) a human hand model.

The Schunk Hand

We have implemented a simulation of the 7 degrees of freedom 3-finger Schunk hand commonly used in grasping applications (depicted in Figure 6.4a,d).

The KCL Anthropomorphic Hand

We have further tested a 'human-like' hand in our experiments. Every finger of this metamorphic anthropomorphic hand developed in the laboratory of Prof. Jian S. Dai at KCL (Wei et al., 2011) consists of 3 segments. The reconfigurable palm allows for additional freedom of the thumb (as shown in Figure 6.4c). We decided to have the same preshape for both \mathbb{S}^1 and \mathbb{S}^2 caging before applying an automatic finger closing procedure, described in the next section.

6.3.3 Positioning and Automatic Closing of the Hands

Our experiments are devoted to verification of the usefulness of our caging heuristics and not of the motion planning system itself. It has been already evaluated in chapter 4. That is why we decided to skip the trajectory planning phase. Instead, we place our manipulators directly on the object's surface and then perform an automatic finger closing operation. The caging areas were selected according to the S^1 and S^2 heuristics.

Recall that the radius of the approximate geodesic ball $G_r(p)$ is chosen to match the finger length of our robotic manipulators, so as to provide a possibility for the hand to fully cover such balls with an open palm posture (see Figures 3.3 and 6.4).

Our grasp synthesis approach consists of three phases and is implemented as follows: for a geodesic ball $G_r(p)$ that has highest value with respect to our scoring functions $W_r(p)$ or $S_r(p)$ (defined in section 6.4), the hand base is first placed at a certain distance from the center of $G_r(p)$ along the surface normal direction. A suitable preshape, depending on current heuristic, as in the Figure 6.4 is then adopted and the hand base orientation is set to be parallel to the surface. Additionally, for the circle caging case the main axis along the stretched fingers is aligned orthogonal to the main axis of $G_r(p)$. After that, we gradually moved the hand towards the surface until the distance to the surface reached a small threshold, varying with the object size.

In the final phase, we performed a standard autoclose operation: we moved the finger segments (starting from those closest to the palm) towards the object until the distance between each segment and the object was smaller than a predefined threshold value. This is done preserving the opposing finger configuration for the circle (6.4d,e), and the equal angular spread between the fingers for the sphere caging case (6.4a,b). For the "net of points" manipulator, we simply place the net over the geodesic ball $G_r(p)$ as described in the section 6.3.1.

6.4 Caging Heuristics

We develop two types of caging grasp heuristics which we call circle (\mathbb{S}^1) and sphere (\mathbb{S}^2) caging respectively. They are based on approximate geodesic balls $G_r(p)$ representation (see Section 3.3 for details). Using these heuristics we are capable of ranking the likelihood of finding a caging grasp for every object.



Figure 6.5: Illustration of circle caging on a dumbbell object (left) and a geodesic ball and winding path in blue (right).

6.4.1 Circle Caging, \mathbb{S}^1

Inspired by our grasp transfer experiments, we decided to exploit winding for synthesis of caging grasp. We have already introduced the representation of *winding angles* in section 3.5. The hypothesis here is that geodesic balls with higher winding values are better suited for the circle caging.

In other words, we define a heuristic for caging an object with a curve $\gamma : [0, 1] \to \mathbb{R}^3$ which wraps around an elongated part of an object such as the red curve in Figure 6.5 left. We call this type of grasp an \mathbb{S}^1 , or circle cage, since the curve γ is a closed circle in cases when full winding is possible. Of course, this measure is also applicable to other cases (not necessarily fully winding) as shown on the right part of the Figure 6.5.

We now propose to rank *all vertices* of our object model by the resulting winding angles $W_r(p)$, with large winding angles indicating a higher likelihood that the curve $P_r(p)$ yields an \mathbb{S}^1 caging grasp for the object. Let us now discuss a related concept of \mathbb{S}^2 caging before connecting the above with the geometry of a robot hand and the actual synthesis of such cages.

6.4.2 Sphere Caging, \mathbb{S}^2

Consider the case when an object or some part of it is fully enveloped, covered by a geodesic ball. For example, the upper part of the dumbbell object in Figure 6.6 is

CHAPTER 6. CAGING SYNTHESIS



Figure 6.6: A geodesic ball (in red) and generated sphere cages

almost completely enclosed by the red contact surface resembling a part of a sphere. This part is clearly caged - it has lost some degrees of freedom - although it is still possible to imagine a rotational motion around the main axis.

Recall, that Gaussian curvature indicates how salient a particular vertex is (see section 2.2.2). There is also a discrete formulation of the the Gauss-Bonnet theorem (explained in section 3.4). We can now define a heuristic for measuring how $G_r(p) \subseteq M$ wraps around the M using the integrated discrete total Gaussian curvature on $G_r(p)$. If a geodesic ball $G_r(p)$ covers the model M completely then this quantity equals $2\pi\chi(M)$. Therefore our hypothesis is that geodesic balls with a *large total curvature* are good candidates for applying a caging grasp.

More formally, we consider the scoring function S_r defined by

$$S_r(p) = \sum_{v \in V(G_r(p))} K(v),$$
 (6.1)

where K denotes the total Gaussian curvature introduced in Section 3.4 and where $V(G_r(p))$ denotes the vertex set of an approximate geodesic ball $G_r(p)$ on M.

CHAPTER 6. CAGING SYNTHESIS

Note that, the total Gaussian curvature does not distinguish convex and concave parts of the model. Since we would prefer to grasp locally convex parts of the object, we furthermore exclude geodesic balls with centre vertices v which have a concave 1ring neighbourhood in the mesh from further analysis. This procedure ensures caging of locally convex parts of the object.

6.5 Evaluation Methodology

6.5.1 Robustness With Respect to Noise

In practical robotic applications, the perception is usually presented by visual and depth information. This raw data is obtained from noisy sensors, affected by artifacts and distortions. A triangulated 3D model of an object can then be reconstructed out of point clouds. We claim here that our heuristics are to some degree invariant to the additional noise. To prove that, we designed a sequence of experiments with deformed meshes.

In order to simulate perceptual noise on a synthetic object mesh, we shifted every vertex of the original mesh by a uniformly distributed offset in $[-\sigma, \sigma]$. After that, we reconstructed a new mesh from the "deformed point cloud" using the Poisson tool from the Meshlab package (Cignoni et al., 2008). The value of the maximal offset σ used in our simulations was chosen empirically to be equal to 1% of the bounding sphere radius. The preprocessing was then done as described in section 6.2. As a result of this operation, the original mesh differed significantly from the resulting noisy reconstructed mesh as shown in Figure 6.7.

6.5.2 Physical Simulation

We have continued to exploit a realistic physics simulation using the PhysX software package to test if an object was successfully caged (see chapter 5 for reference). After placing the hand and closing the fingers, the object and the manipulator were, as before, copied to the physical environment with gravity and standard friction forces (the friction coefficient was set to 1). We simulated 10 consecutive random rotational motions of the manipulator 'in air' for 100 simulation steps each and deemed a grasp



Figure 6.7: The effect of additional noise on the original meshes (left column) is shown. For illustration purposes, we display five probable caging areas (in dark) with highest scoring function values, W_r for upper row and S_r for lower row. Noisy meshes (right column) are generated from the original meshes (left column) by adding Gaussian noise. Adapted from (Zarubin et al., 2013b).

stable if the object was still within a small distance of the hand and had not fallen onto a simulated floor under the influence of gravity.

6.5.3 Computational Complexity

All experiments were done on a computer with an Intel i5, 4 Core, 2.40GHz processor with 4 GB memory. The computation time of our scoring functions scales in the worst case (small object size compared to r in $G_r(p)$) quadratically in the number of vertices. For small objects, the average computation time was about 20 seconds (1000 vertices per object), while the computation for medium and large objects took less than a second.

6.6 Examples of Caging Grasps

Figure 6.8 displays geodesic balls of highest scoring function value together with corresponding grasps for several examples. The first cup object has smallest scale and can thus be fully covered by all manipulators. Since all $G_r(p)$ in this case have the same integrated curvature (*e.g.* zero for the original cup mesh), the first vertex in the list of vertices for this mesh is chosen to be the central vertex p of $G_r(p)$. The synthesized grasps for the medium size dumbbells object (integrated value of $3\pi \pm 0.16\pi$) also appeared to be stable. The last two examples demonstrate generated cages for large and medium size objects in the case of \mathbb{S}^1 caging. They were robust according to our evaluation procedure.

6.7 Experiments

In this section we describe a series of experiments which have been designed to test several aspects of our circle and sphere caging heuristics:

- the general success rates of our approach and its robustness to noise,
- dependence on object's shape and size,
- applicability for various hand kinematics,
- stability threshold values.

The methodological details of the experimental setup are outlined in Algorithm 1. In total, we generated 10 synthetic noisy meshes of 4 types and of 3 different sizes. Success evaluation of our caging heuristics for these meshes is given in section 6.7.1. We discuss results for several example cages and suggest a threshold value for which the successful caging is most probable in section 6.7.2. We also conducted a comparative study of a curvature-based grasping heuristic versus a charge-based heuristic in section 6.8.



Figure 6.8: We display examples of grasps generated by our method. The three top rows represent attempted S^2 caging grasps and the lower two attempted S^1 cages respectively. All displayed grasps apart from the grasps on the leg of the pony in the case of the net of points, the Schunk hand and the anthropomorphic hand (second, fourth and fifth column in the third row) pass our PhysX caging test. The first column shows reconstructed noisy meshes with the selected approximate geodesic ball $G_r(p)$ in red and centred at the vertex with highest $S_r(p)$ (first three objects) and highest $W_r(p)$ (lower two meshes). The edge-path $P_r(p)$ is highlighted in blue for circle cages. The other four columns demonstrate generated grasps for four types of manipulators. The topmost cup object has smallest size (scale factor of 1). The dumbbells are of medium size and the pony in the middle and the second cup are large. Finally, the last row shows generated S^1 grasps for a medium sized pony object.

Algorithm 1 Evaluation algorithm
Require: original 3D model M , scale s , noise level σ
Ensure: stability analysis of cages for K noisy meshes
for 1 to K do
Remove all faces.
Add uniform noise from $[-\sigma, \sigma]$ to every vertex.
Reconstruct a new mesh using MeshLab package (Cignoni et al., 2008).
Perform isotropic remeshing.
Compute scoring functions according to section 6.4.
Select best grasp centre point on the <i>noisy</i> mesh.
Position and autoclose the hand on the <i>original</i> mesh.
Test if the grasp is a caging grasp using $PhysX$.
end for
return $\%$ of successful caging grasps (robustness)

6.7.1 Results

For our first set of experiments we produced 10 synthetic noisy meshes with 1000 vertices according to the method described in section 6.2 for the pony, cup, bunny and dumbbells models. Some of them are depicted in Figures 6.7 and 6.6.

We have tested three different object sizes with a scaling factor between 1 and 8. The scaling factor was computed as a ratio $\pi(R_b/F_l)$, where R_b is a bounding sphere radius and F_l the Schunk hand's finger length. So that a value of 1 for the scale of an object represents an object size such that a full caging inside the robot hand is likely to be possible. Other scales were chosen empirically in order to represent medium (3-4) and large (8) scale objects.

We have followed Algorithm 1 in order to evaluate our approach. The stability rates were computed as percentage of stable out of total number of previously generated K = 10 noisy meshes. Results are presented in the Table 6.1 for sphere caging and in Table 6.2 for circle caging respectively. The maximal integrated curvature and winding angles are given as multiples of π .

(1		-			
Objects	Scale	Curvature	Net of	Hexapus	Schunk	Anthropo-
		integral	points	hand	hand	morphic
		value, π				hand
Dumbbells	1	4 ± 0	100%	100%	50%	60%
Dumbbells	4	3 ± 0.16	100%	100%	100%	80%
Dumbbells	8	0.89 ± 0.02	0%	0%	0%	0%
Pony	1	-7.2 ± 1.6	100%	100%	80%	80%
Pony	4	1.56 ± 0.12	90%	80%	20%	30%
Pony	8	1.77 ± 0.04	0%	40%	0%	0%
Cup	1	-2.36 ± 2.65	100%	100%	80%	60%
Cup	4	1.27 ± 0.53	50%	10%	10%	40%
Cup	8	0.70 ± 0.15	20%	60%	40%	70%
Bunny	1	4 ± 0	100%	100%	80%	60%
Bunny	3	2.58 ± 0.11	100%	100%	60%	40%
Bunny	8	1.97 ± 0.04	20%	20%	30%	30%

Table 6.1: Robustness of sphere caging heuristics w.r.t size of an object

Table 6.2: Robustness of circle caging w.r.t size of an object

			-	-		
Objects	scale	Winding	Net of	2-finger	Schunk	Anthropo-
		value, π	points	hand	hand	morphic
						hand
Dumbbells	1	2 ± 0	100%	40%	90%	80%
Dumbbells	4	2 ± 0	100%	90%	80%	100%
Dumbbells	8	1.92 ± 0.04	90%	90%	40%	90%
Pony	1	2 ± 0	90%	20%	80%	70%
Pony	4	2 ± 0	100%	90%	20%	70%
Pony	8	1.98 ± 0.03	70%	40%	0%	30%
Cup	1	1.99 ± 0.01	90%	30%	30%	40%
Cup	4	2 ± 0	90%	30%	10%	30%
Cup	8	1.97 ± 0.04	60%	50%	10%	30%
Bunny	1	2 ± 0	100%	80%	90%	80%
Bunny	3	2 ± 0	100%	40%	0%	10%
Bunny	8	1.90 ± 0.05	20%	20%	0%	10%

6.7.2 Discussion

In this section we review the results in terms of manipulators, scales and two types of heuristics.

The net of points demonstrated the best average success rate over all experiments and was the best on average for S^1 caging (followed by the anthropomorphic hand) and second best (after the 'hexapus' hand) for S^2 caging. The 3-finger Schunk hand showed a high amount of variations in stability rates depending on the size and type of an object. For this hand, the object might simply slip out between fingers when small enough, whereas additional fingers and flexibility prevent this in case of the 'hexapus' and anthropomorphic hand which performed slightly better.

The smallest scale was selected, as stated above, to enable a full caging by a robot hand. Thus, the stability rate for the net of points or the 'hexapus' hand in this case is close to 100% with a few exceptions in the S^1 case. For the medium size objects, the results are also good and demonstrate the ability of our method to select the most suitable subparts of an object. Large scale objects are generally unstable and success in this case depends fully on the object's structure.

The success of \mathbb{S}^2 caging heuristic can be explained using the relation to the genus of the object. The total Gaussian curvature integral for a closed polyhedral surface $M \subset \mathbb{R}^3$ equals $2\pi\chi(M)$ (see section 2.2.2). Furthermore, recall that $\chi(M) = 2 - 2g$ where g denotes the genus of M which is non-zero for the cup and pony objects. In the case of genus zero objects such as the bunny and the dumbbells, a good caging success rate seemed to occur if the integrated curvature over $G_r(p)$ was larger than 50% of the total - *i.e.* larger than 2π (we call it a 2π hypothesis). There was a substantial amount of variation between the robot hands.

In case of S^1 caging, the winding value was always close to 2π (full circle). Interestingly, this does not always guarantee a good performance since a cylindrical subpart might be too small or not reachable at all, preventing a good caging grasp. Nevertheless, the net of points, the anthropomorphic and the 2-finger hand have shown promising stability rates. The third finger in the case of the Schunk hand might be both beneficial (*e.g.* when the object is very small) and disadvantageous (*e.g.* due to collisions).

In summary, the high percentage of stable grasps for the 'net of points' indicates



Figure 6.9: Models used for the experiments. Areas of potentially stable cages are highlighted in red.

that our heuristics are capable of successful generation of caging grasps. However, different robotic hands are not equally well suited for \mathbb{S}^1 and \mathbb{S}^2 cages as can be seen from Table 6.1.

6.8 Comparison of Curvature Integral and Charge Integral Heuristics

In this section we shall evaluate our S^2 caging heuristic in a more systematic way. For this purpose we compare our approach to an adapted charge-based heuristic. This method is described in details in work of (Sandilands et al., 2014). Briefly, it uses electrostatics based parametrization which is obtained by computing the electric field after electrically charging the object in simulation. Note, that charge density values also could be used for determining areas of interest in grasping context. Salient subparts receive higher charge density than others. They can also be integrated over all faces within a geodesic ball and then be used exactly the same way as in the case of curvature-based heuristic. As before, we chose the highest value (of charge density integral) and tested it in physical environment. The charge density values were generously provided by Peter Sandilands from the group of Sethu Vijayakumar.

In order to compare two methods, we decided to use object models from the database of (Chen et al., 2009). We selected 10 different models (see Figure 6.9), representing all kinds of shapes. The meshes were resampled and isotropically remeshed according to our methodology described above. Also, the convex decompositions were made for a physical evaluation. The actual experiments were done as follows: we computed discrete curvature integrals for every vertex of a mesh, we chose the highest value, we evaluated the cage using a "net-of-points" manipulator in PhysX. This procedure returns stability value (1 for stable, 0 otherwise), a curvature integral and a force closure value (an approximation if pushed to the contact).

We changed the radius of a geodesic ball from 2 to 16 for every model, imitating varying object sizes. Thus, the resulting set consists of 150 datapoints, which we analyzed using histogram approach. We split the whole range of curvature integrals $[0.5; 3.5]\pi$, into 10 bins and computed the ratio of stable cages over total number of datapoints within each bin. The resulting histogram is shown in Figure 6.10. We have also evaluated force closure data, although there was no direct contact of the manipulator to the model. The artificial net manipulator (see section 6.3.1) was placed in sub-millimeter distance to the object and then used for grasp quality evaluation. We performed similar histogram analysis and calculated average force closure for every bin. The distribution can be seen on Figure 6.11.

The resulting cages were stable in 54 cases for charge-based heuristic and in 80 cases for curvature-based heuristic. One possible explanation for the worse performance of the charge-based heuristic can be that, unlike Gaussian curvature approach, hyperbolic parts were not penalized. That potentially could lead to selection of unstable areas between e.g. two salient parts.

The stability histogram 6.10 confirms our 2π hypothesis. There is a rapid increase of caging success ratio after the value of curvature integral crosses the 6.26 or 2π threshold. Force closure also supports this conclusion, although the average value grows more gradually in this case. The charge-based heuristic, on average, behaves worse both in terms of force closure analysis and physical stability.



Histogram of stable cages

Histogram of stable cages

for charge-based heuristic



Figure 6.10: Stability histograms for curvature- and charge-based heuristics.



Histogram of force-closure values

Histogram of force-closure values



Figure 6.11: Force closure histograms for curvature- and charge-based heuristics.

Chapter 7 Conclusions

The key idea and novelty of the thesis lies in using abstract topological representations for motion planning. This modification is advantageous with respect to classical planning algorithms in several points. First of all, topological abstraction is invariant to particular geometric constraints of the object and preserves only essential structure (via notions of connectivity and neighborhood). This gives rise to a significant generalization of controllable object classes. For example, one of the challenging task in modern Robotics is manipulation with articulated and deformable objects. We have addressed this problem by combining planning in a configuration space and by exchanging information within the alternate space. Second advantage is a natural hierarchy, which emerges from a particular topological mapping. Being a relatively simple deterministic transformations of a configuration space, topological representations may be combined in any arbitrary order as well as incorporated into stochastic control models. Third difference is based on the fact, that alternate representations are not restricted to a certain predefined situation – they have a potential to adapt to novel scenarios and thus improve planning in dynamic environments.

The choice of representations for motion planning is highly dependent on the particular problem. Spaces developed in this thesis have certain advantages and disadvantages, providing us with a set of instruments, but not with a universal solution. For some interaction problems there exist a suitable topology-based representation in which an interaction can be described in a way that local optimization methods can find solutions that would otherwise require expensive global search (as with the writhe

CHAPTER 7. CONCLUSIONS



Figure 7.1: KUKA LWR 4 robotic arm reaching through a hollow box with. The main task was defined in topological representation. Adapted from (Zarubin et al., 2012)

representations). Other representations can generalize better to novel or dynamic situations (as with the grasp transfer using winding number representation). These tools, however, should always be used in conjunction with classical tasks such as collision potentials in order to be compatible with real world applications (see Figure 7.1 for illustration).

Unlike previous work with such representations (e.g. described in (Dowker and Morwen, 1983; Edmond and Komura, 2009; Tamei et al., 2011)), where only basic approaches for inverse mapping from topological to configuration spaces were tested, we presented a framework that combines the different representations at the abstract and lower level for motion synthesis.

Consider for example, the reaching task (for an artificial "snake" manipulator from section 4.3.1) only in an end-effector space. Local optimization method would be trapped in a "deep local minima". We have demonstrated in section 4.4 that the solution of this problem is practically infeasible with global search approaches, e.g using RRTs. On the other hand, considering such a problem only in writhe space would not address the actual reaching task. The coupling of two spaces, however, allows a local optimization method to generate an unwrapping-and-reaching motion.

We decided to formulate our approach in the framework of optimal control as an approximate inference problem. This framework allows for a direct extension of the graphical model to incorporate multiple representations. Alternative formulations are possible, for instance as a structured constraint optimization problem. What we coined as a motion prior in topological spaces would here correspond to pseudo control costs for transitions in topological space. Which formulation will eventually lead to computationally most efficient algorithms is a matter of future research.

7.1 Benefits for Motion Planning

To our best knowledge, for the first time topology-based representations have been actually used for motion planning. We have shown that such spaces are particularly beneficial for describing close interaction of objects. One could imagine many applications where articulated and flexible objects are involved. Alternate representations could provide a semi-symbolic description of a task and can thus be used in combination with high level planners.

- We have proposed and evaluated a general framework for combination of tasks in configuration and alternate spaces.
- We addressed a temporal optimization problem and provided a solution for different kinds of linearization of the system dynamics.
- Task spaces developed in our work can simplify motion synthesis and can be efficiently used for motion planning in dynamic environments.
- We have shown the efficiency of our framework in comparison to randomized search algorithms.

7.2 Benefits for Grasping and Caging

In relation to grasping and caging synthesis our work has contributed in terms of theoretical framework and implemented software.

• We have proposed and evaluated two novel caging synthesis methods which we call circle and sphere caging.

- We have developed heuristic scoring functions for both caging types which allow us to find areas for robust caging grasps on complex 3D objects.
- Our approach is based on a novel representation of a geodesic ball, aimed to reflect the hand-object interaction.
- We evaluated the caging synthesis methods using a realistic physics simulation. We tested the robustness of the method under a simulated sensor noise for various object sizes and types of manipulators.

7.3 Future Work

Motion Planning and Abstract Representations

We plan to apply the proposed methods for dexterous robot manipulation of more complex, articulated or flexible objects. We would like to merge grasp synthesis and motion planning. We believe that multiple parallel representations will enable more robust and generalizing motion synthesis strategies.

We have applied our motion planning framework in context of grasp transfer, but not for generating grasps of complex objects. It is logical to ask, whether one could use motion planning for synthesis of physically stable grasps and whether usage of abstract representations is advantageous in such case. Initial work and preliminary results show that we can merge topology-based representations, caging heuristics and inference-based planning in one single system. Moreover such combination allows to generate an optimal trajectory leading to a stable grasp. Preliminary experiments in simulation and on real hardware with the KUKA arm and Schunk hand have proved the usefulness of our framework (Sandilands et al., 2014).

Caging and Grasping

The representation of a geodesic ball was developed in order to exploit the 'handlocal' geometry. It is, to some degree, independent of the exact kinematic structure of the robotic hand. One possible direction of the future research is to consider more elaborate approximations of either manipulator or object. We also hope that our caging synthesis methods will be tested and implemented in combination with soft


Figure 7.2: Invariance with respect to slight changes in morphology of the objects. 4 most probable S^2 caging areas on different teddy bear models.

robotic actuators. There are several existing robot hands suitable for applying such caging grasps (e.g. created in laboratories (Deimel and Brock, 2013), (Ilievski et al., 2011), and (Brown et al., 2010)).

Another potential direction could be to study curvature of point clouds. This extension could decrease the computation time of our scoring functions. It can be further utilized in research on exploration with mobile platforms and service robotics. One could think of using curvature based methods in order to identify potentially manipulable and movable objects. Such features, based on a geodesic ball representation, could be advantageous at least in terms of robustness due to the integration property.

Segmenting objects using geodesic balls representation can lead to extracting features for mapping between objects. This is of particular interest in the context of grasp transfer - such segmentation could simplify remapping of hand postures between similar objects. Possible applications range from precise surgery robotics and teleoperation to manipulating household objects (Hu et al., 2005).

Classification of objects is another field which could benefit from our heuristics. Integral of curvature values makes our scoring functions invariant to small deformations of object surface. Moreover, the combination of areas with highest scoring values may be a unique feature for a class of objects. For example, all teddy bears from our dataset had highest values on tips of their legs (see Figure 7.2 for illustration). Including such information as an additional feature for classification problem could improve the success ratio. In recent years, many academic and industrial institutions started to develop service robots for everyday usage. Such robots are especially trained to operate in typical home environments. Many tasks include interaction with objects and thus require a suitable internal representation. As we have shown here, alternate spaces can be very efficient in describing complex interactions. We hope that our work can become one of the steps towards really dexterous robotic manipulations.

Appendix A

Derivation of the Final State Estimators

A.1 Probabilistic Model for 1-step Posture Estimation

Part of our motion planning framework includes an estimation of the final configuration of the robot given a certain set of constraints. In this appendix we derive estimators for different cases of system dynamics and also obtain an optimal trajectory duration. We address two problems of robot trajectory optimization:

1) Backward messages for the last state of the trajectory usually defined a priori by constant values. Thus the estimation of the final configuration could help to tune backward message in a direction of faster convergence.

2) Amount of time needed for performing task is also used to be fixed and coupled with the number of time steps in the Markov chain. Optimizing likelihood of performing task with respect to the duration could decrease total trajectory costs. Related to the ideas presented in (Rawlik et al., 2010).

Derivations below use the same notations and terminology as in sections 2.3 and 4.2

Note, that joint distribution of graphical model on Figure A.1b can be written as:

$$P(q_0, q_T, z = 1) = P(q_0)P(q_T \mid q_0)P(z = 1 \mid q_T)$$
(A.1.1)



Figure A.1: An illustration of the 1-step inference approach. The red arrow shows a possible information flow.

1-step inference allows to estimate the final configuration of the system given an initial state q_0 and transition probabilities $P(q_{t+1} | q_t)$. We shall use equation 2.5 from section 2.3.2.

$$P(q_{t+1} \mid q_t) = \mathcal{N}(q_{t+1} \mid Aq_t + a, Q + BH^{-1}B'), \tag{A.1.2}$$

In the following sections we will derive equations for the final state estimation separately for three cases of system linearization: kinematic, pseudo-dynamic and dynamic case.

A.1.1 Kinematic Case

In the kinematic case we have A = B = I, a = 0, thus

$$P(q_{t+1} \mid q_t) = \mathcal{N}(q_{t+1} \mid q_t, W),$$

with $W = Q + H^{-1}$. Using properties of conditional distributions we obtain:

$$P(q_T \mid q_0) = \frac{P(q_0, q_T)}{P(q_0)} = \frac{\sum_{q_1, \dots, q_{T-1}} P(q_0) P(q_1 \mid q_0) \dots, P(q_T \mid q_{T-1})}{P(q_0)}$$

Now let us take the first two elements of the product and apply Gaussian multiplication rule:

$$P(q_1 \mid q_0) P(q_2 \mid q_1) = \mathcal{N}(q_1 \mid q_0, W) \mathcal{N}(q_1 \mid q_2, W) =$$

$$\mathcal{N}[q_1 \mid W^{-1}(q_0 + q_2), 2W^{-1}] \mathcal{N}(q_0 \mid q_2, 2W)$$
(A.1.3)

Taking into account the marginalization over all variables $q_1, .., q_{T-1}$,

$$P(q_1 \mid q_0)P(q_2 \mid q_1) \propto \mathcal{N}(q_0 \mid q_2, 2W) = \mathcal{N}(q_2 \mid q_0, 2W).$$

Iterating this procedure T - 1 times will give us probability of being in state q_T after T time steps

$$P(q_T \mid q_0) \propto \mathcal{N}(q_T \mid q_0, TW). \tag{A.1.4}$$

A.1.2 Pseudo-dynamic Case

In the pseudo-dynamic case we have a = 0 and thus

$$P(q_{t+1} \mid q_t) = \mathcal{N}(q_{t+1} \mid Aq_t, Q + BH^{-1}B'),$$

It is important that A, B and Q, H^{-1} have a very specific form:

$$A = \begin{pmatrix} I & \tau I \\ 0 & I \end{pmatrix}, B = \begin{pmatrix} \tau^2 I \\ \tau I \end{pmatrix}, Q = \begin{pmatrix} \tau Q_1 & 0 \\ 0 & \tau Q_2 \end{pmatrix}, H^{-1} = \begin{pmatrix} (\tau H_1)^{-1} & 0 \\ 0 & (\tau H_2)^{-1} \end{pmatrix}$$

Where τ is an integration step defined by the ratio $\tau = \frac{Time}{Steps}$ and $W = Q + BH^{-1}B'$. Similarly to the kinematic case we shall use properties of conditional distributions:

$$P(q_T \mid q_0) = \frac{P(q_0, q_T)}{P(q_0)} = \frac{\sum_{q_1, \dots, q_{T-1}} P(q_0) P(q_1 \mid q_0) \dots, P(q_T \mid q_{T-1})}{P(q_0)}$$

Now let us again take the first two elements of the product and apply Gaussian multiplication rule:

$$P(q_1 \mid q_0) P(q_2 \mid q_1) = \frac{1}{\det A} \mathcal{N}(q_1 \mid Aq_0, W) \mathcal{N}(q_1 \mid A^{-1}q_2, A^{-1}WA^{-T}) = (A.1.5)$$
$$\mathcal{N}[q_1 \mid W^{-1}Aq_0 + (A^{-1}WA^{-T})^{-1}A^{-1}q_2), W^{-1} + (A^{-1}WA^{-T})^{-1}] *$$
$$\mathcal{N}(Aq_0 \mid A^{-1}q_2, W + A^{-1}WA^{-T})$$

Taking into account the marginalization over all variables $q_1, .., q_{T-1}$,

$$P(q_2 \mid q_0) = \frac{1}{\det A} \frac{1}{\det A^{-1}} \mathcal{N}(q_2 \mid A^2 q_0, A(W + A^{-1}WA^{-T})A') = \mathcal{N}(q_2 \mid A^2 q_0, AWA' + W)$$

Iterating this procedure T - 1 times will give us probability of being in state q_T after T time steps in pseudo-dynamic case.

$$P(q_T \mid q_0) = \mathcal{N}(q_T \mid A^T q_0, \sum_{i=0}^{T-1} A^i W A^{'i}).$$
 (A.1.6)

A.1.3 Dynamic Case

In the full dynamic case we again start we the transition probability equation 2.5

$$P(q_{t+1} \mid q_t) = \mathcal{N}(q_{t+1} \mid Aq_t + a, Q + BH^{-1}B').$$
(A.1.7)

We define $W = Q + BH^{-1}B'$ and integrate the system dynamics,

$$P(q_T \mid q_0) = \sum_{q_{1:T-1}} \prod_{t=1}^T P(q_t \mid q_{t+1}).$$
(A.1.8)

For stationary linear Gaussian dynamics, we obtain the following:

$$P(q_T \mid q_0) = \mathcal{N}(q_T \mid A^T q_0 + \sum_{i=0}^{T-1} A^i a_i, \sum_{i=0}^{T-1} A^i W A'^i) , \qquad (A.1.9)$$

where a power of matrix, A with a superscript on it, is defined iteratively as $A^i = A * A^{i-1}$.

A.2 Posterior Posture & Temporal Optimization

The temporal optimization is based on the idea of one-step inference, where we estimate the final state of the robot configuration under simple kinematic constraints and try to apply the likelihood maximization with respect to the duration of the trajectory. It is somewhat similar to the notion of Macro-states in (He et al., 2010), but differs in terms of splitting trajectory into keyframes. Presented algorithm is much simpler than EM used for temporal optimization in (Rawlik et al., 2010), since we first estimate the exact final configuration and have an intuition about the possible time consumption.

The aim of including a task variable is to make sure that we not only have reached the final state but also performed a given task. In other words we estimate

$$P(q_T \mid q_0, z = 1).$$

Since we are interested only in the final configuration, we assume that all intermediate costs are equal to zero. We *approximate locally* the cost as exp-squared:

$$P(q_T \mid z = 1) = \mathcal{N}[q_T \mid r_T, R_T]$$

Note that this local approximation depends on the point q_T of localization. In practice, computation of the posterior $P(q_T \mid z = 1, q_0)$ is therefore an iterative Gauss-Newton-like process, that alternates between estimating q_T and re-computing the local approximate costs r_T , R_T .

Coupling this equation with (A.1.9) results in:

$$P(q_T \mid q_0, z = 1) = \mathcal{N}[q_T \mid (\sum_{i=0}^{T-1} A^i W A^{'^i})^{-1} A^T q_0 + r_T, (\sum_{i=0}^{T-1} A^i W A^{'^i})^{-1} + R_T] * (A.2.1)$$
$$\mathcal{N}(A^T q_0 \mid R_T^{-1} r_T, \sum_{i=0}^{T-1} A^i W A^{'^i} + R_T^{-1})$$

The first Gaussian represents the distribution over the final states. The second one is simply a likelihood of the success for a given number of time steps T. Equation (A.2.1) is an important result and is used in the algorithms, described in the following sections.



Figure A.2: An illustration of iterative temporal optimization algorithm

A.2.1 Two Approaches to Estimate the Trajectory Duration

We shall distinguish 2 approaches to optimize the duration (free parameter T) of the motion based on the 1-step model:

1. We can use *Expectation Maximization* approach, where the E-step computes $P(q_T \mid q_0, z = 1; T)$ given T. This is in itself an iterative process, since the cost r_T, R_T depends on the point of linearization, which we typically take as the mode of the posterior $P(q_T \mid q_0, z = 1; T)$.

In the M-step we then maximize the expected complete data log-likelihood

$$T_{\text{new}} = \underset{T}{\operatorname{argmax}} \int P(q_T \mid q_0, z = 1; T_{old}) \log P(q_T, z = 1 \mid q_0; T) \ dq_T$$

This approach is most similar to (Rawlik et al., 2010).

2. In the 1-step model the form of the likelihood is simple enough to be able to derive an analytic gradient for the likelihood. However, as for the E-step above, this gradient relies on choosing a specific cost approximation r_T , R_T – that is, on iteratively estimating the posterior $P(q_T \mid q_0, z = 1; T)$ to decide on the point of cost approximation (the posterior's mode).

Given a specific r_T , R_T the gradient of the likelihood is analytic and allows for gradient ascent to maximize it. This is derived in the following.

A.2.2 Estimation of the Final State q_T (E-step)

Input: initial duration D = T and a posterior probability of being in the state q_T and fulfill all tasks z = 1.

$$P(q_T \mid q_0, z = 1) \propto \mathcal{N}[q_T \mid (\sum_{i=0}^{T-1} A^i W A'^i)^{-1} A^T q_0 + r_T, (\sum_{i=0}^{T-1} A^i W A'^i)^{-1} + R_T]$$

Output: Belief about the final state: b, B; Mean and variance of related costs: r, RConsequently calculate variance and mean of the belief about the final state using:

$$B^{-1} = \left(\sum_{i=0}^{T-1} A^{i}WA^{'^{i}}\right)^{-1} + R_{T}; \ b = B\left(\left(\left(\sum_{i=0}^{T-1} A^{i}WA^{'^{i}}\right)^{-1} + R_{T}\right)q_{0} + r\right)$$

substitute the current state \hat{q} by

$$\hat{q}^{new} = \hat{q}^{old} + \alpha(b - \hat{q}^{old})$$

with convergence rate α .

A.2.3 Gradient Optimization of the Trajectory Duration D

Input: r, R and second part of the equation:

$$L(z=1;D) = \mathcal{N}(A^{T}q_{0} \mid R_{T}^{-1}r_{T}, \sum_{i=0}^{T-1} A^{i}WA^{'i} + R_{T}^{-1})$$
(A.2.2)

Output: Duration of the movement: D_{opt}

Equation (A.2.2) represents the likelihood of the success. It is a Gaussian with the mean $\mu = R_T^{-1}r_T$ and the variance $\sigma^2 = \sum_{i=0}^{T-1} A^i W A^{\prime i} + R_T^{-1}$

$$\frac{dL(z=1;D)}{dD} = \mathcal{N}(A^T q_0 \mid \mu, \sigma^2) [-h'(\delta_D A^T q_0) + \frac{1}{2}h^T(\delta_D \sigma^2)h - \frac{1}{2}\operatorname{tr}(\sigma^{-2}(\delta_D \sigma^2)\sigma^2)]$$

where $h = \sigma^{-2} (A^T q_0 - \mu)$.

Equation (A.2.3) can now be used in any gradient-ascent method in order to max-

imize the likelihood with respect to the duration of the trajectory.

A.3 Trajectory Duration Estimation in the Dynamic Case

Similarly to the kinematic case, "1-step-dynamic" inference allows to estimate the final configuration of the system given an initial state q_0 and transition probabilities $P(q_{t+1} \mid q_t)$. We will again use related equations from (Toussaint, 2009).

$$P(q_{t+1} \mid q_t) = \mathcal{N}(q_{t+1} \mid Aq_t + a, Q + BH^{-1}B').$$

In the pseudo-dynamic case we assume that a = 0 and thus

$$P(q_{t+1} \mid q_t) = \mathcal{N}(q_{t+1} \mid Aq_t, Q + BH^{-1}B').$$

It is important that A, B, Q and H^{-1} have a very specific form:

$$A = \begin{pmatrix} I & \tau I \\ 0 & I \end{pmatrix}, B = \begin{pmatrix} \tau^2 I \\ \tau I \end{pmatrix}, Q = \begin{pmatrix} \tau Q_1 & 0 \\ 0 & \tau Q_2 \end{pmatrix}, H^{-1} = \begin{pmatrix} (\tau H_1)^{-1} & 0 \\ 0 & (\tau H_2)^{-1} \end{pmatrix}$$

Where τ is an integration step defined by the ratio $\tau = \frac{Time}{Steps}$.

A.3.1 Estimation of the Final State q_T

Following the similar steps as for the kinematic case, we first obtain a posterior :

$$P(q_T \mid q_0) = \mathcal{N}(q_T \mid A^T q_0, \sum_{i=0}^{T-1} A^i W A'^i).$$

where $W = Q + BH^{-1}B'$. After that step, we couple the posterior with task related terms:

$$P(q_T \mid q_0, z = 1) = \mathcal{N}[q_T \mid (\sum_{i=0}^{T-1} A^i W A'^i)^{-1} q_0 + r_T, (\sum_{i=0}^{T-1} A^i W A'^i)^{-1} + R_T] *$$
$$\mathcal{N}(q_0 \mid R_T^{-1} r_T, \sum_{i=0}^{T-1} A^i W A'^i + R_T^{-1})$$

The main difference from the kinematic case is appearance of a sum in the posterior variance. Taking into account the specific form of A we can show that

$$A^i = \left(\begin{array}{cc} I & i\tau I \\ 0 & I \end{array}\right).$$

Consequently,

$$A^{i}WA^{\prime i} = \begin{pmatrix} I & i\tau I \\ 0 & I \end{pmatrix} \begin{pmatrix} W_{1} & W_{2} \\ W_{3} & W_{4} \end{pmatrix} \begin{pmatrix} I & 0 \\ i\tau I & I \end{pmatrix} = \begin{pmatrix} W_{1} + i\tau W_{2} + i\tau W_{3} + (i\tau)^{2}W_{4} & W_{2} + i\tau W_{4} \\ W_{3} + i\tau W_{4} & W_{4} \end{pmatrix}.$$

So now the summation of matrices has transformed into the summation of coefficients of the respective matrix blocks.

$$S_0 = \sum_{i=0}^{T-1} 1 = T, \ S_1 = \sum_{i=1}^{T-1} i = \frac{T(T-1)}{2}, \ S_2 = \sum_{i=1}^{T-1} i^2 = \frac{T(T-1)(2T-1)}{6}$$

Using these coefficients and the fact that W is a symmetric matrix $(W_2 = W_3)$ we get:

$$\Sigma = \sum_{i=0}^{T-1} A^{i} W A^{\prime i} = \begin{pmatrix} S_0 W_1 + 2\tau S_1 W_2 + \tau^2 S_2 W_4 & S_0 W_2 + \tau S_1 W_4 \\ S_0 W_2 + \tau S_1 W_4 & S_0 W_4 \end{pmatrix}$$

Which is further utilized for the posterior belief estimation algorithm.

A.3.2 Gradient Optimization

For solving the temporal optimization problem it is important to notice that time is included implicitly in $\tau = \frac{Time}{Steps}$. Thus, for the beginning we have to rewrite all likelihood terms as functions of τ . Particularly,

$$W = \begin{pmatrix} \tau Q_1 & 0 \\ 0 & \tau Q_2 \end{pmatrix} + \begin{pmatrix} \tau^2 I \\ \tau I \end{pmatrix} \begin{pmatrix} (\tau H_1)^{-1} & 0 \\ 0 & (\tau H_2)^{-1} \end{pmatrix} \begin{pmatrix} \tau^2 I & \tau I \end{pmatrix} = \begin{pmatrix} \tau^3 H_1^{-1} + \tau Q_1 & \tau^2 H_1^{-1} \\ \tau^2 H_1^{-1} & \tau H_1^{-1} + \tau Q_2 \end{pmatrix}$$

Combining it with the result from the previous section: $\sum_{i=0}^{T-1} A^i W A^{'^i} =$

$$\begin{pmatrix} \tau^3 H_1^{-1}(S_0 + 2S_1 + S_2) + \tau S_0 Q_1 + \tau^3 S_2 Q_2 & \tau^2 H_1^{-1}(S_0 + S_1) + \tau^2 S_1 Q_2 \\ \tau^2 H_1^{-1}(S_0 + S_1) + \tau^2 S_1 Q_2 & S_0 \tau (H_1^{-1} + Q_2) \end{pmatrix}$$

Now we substitute $\tau = \frac{D}{T}$ and take a derivative of the sum w.r.t D: $\frac{d\Sigma}{dD} =$

$$\left(\begin{array}{c} \frac{3D^2H_1^{-1}(S_0+2S_1+S_2)}{T^3} + \frac{S_0Q_1}{T} + \frac{3D^2S_2Q_2}{T^3} & \frac{2DH_1^{-1}(S_0+S_1)}{T^2} + \frac{2DS_1Q_2}{T^2} \\ \frac{2DH_1^{-1}(S_0+S_1)}{T^2} + \frac{2DS_1Q_2}{T^2} & \frac{S_0(H_1^{-1}+Q_2)}{T} \end{array}\right)$$

The latter equation is then used in gradient ascent algorithm in order to find the optimal time = D value.

Bibliography

- Arbib, M. A., Iberall, T., and Lyons, D. (1985). Coordinated control programs for movements of the hand. Hand Function and the Neocortex, A. W. Goodwin and I. Darian-Smith (Eds), pages 111–129. 53
- Asfour, T., Regenstein, K., Azad, P., Schröder, J., Bierbaum, A., Vahrenkamp, N., and Dillmann, R. (2006). Armar-iii: An integrated humanoid platform for sensorymotor control. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 169–175. 30, 67
- Bhattacharya, S., Likhachev, M., and Kumar, V. (2011). Identification and representation of homotopy classes of trajectories for search-based path planning in 3D. *Proc. of RSS*. 39
- Bicchi, A., Gabiccini, M., and Santello, M. (2011). Modelling natural and artificial hands with synergies. *Phil. Trans. R. Soc.* 53
- Bicchi, A. and Kumar, V. (2000). Robotic grasping and contact: A review. In *ICRA*. 51
- Bitzer, S. and Vijayakumar, S. (2009). Latent spaces for dynamic movement primitives. In Proc. of 9th IEEE-RAS International Conference on Humanoid Robots, pages 574–581. IEEE. 38
- Borst, C., Fischer, M., and Hirzinger, G. (2003). Grasping the dice by dicing the grasp. In *IEEE/RSJ IROS*, pages 3692–3697. 66
- Brown, E., Rodenberg, N., Amend, J., Mozeika, A., Steltz, E., Zakin, M. R., Lipson, H., and Jaeger, H. M. (2010). Universal robotic gripper based on the jamming of

granular material. *Proceedings of the National Academy of Sciences*, 107(44). 4, 69, 91

- Buchholz, B. and Armstrong, T. J. (1992). A kinematic model of the human hand to evaluate its prehensile capabilities. *Journal of Biomechanics*, 25(2):149 162. 52
- Calli, B., Wisse, M., and Jonker, P. (2011). Grasping of unknown objects via curvature maximization using active vision. In *IEEE/RSJ IROS*, pages 995–1001. 67
- Chen, X., Golovinskiy, A., and Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3). 83
- Cignoni, P., Corsini, M., and Ranzuglia, G. (2008). Meshlab: an open-source 3d mesh processing system. *ERCIM News*, (73):45–46. 75, 79
- Ciocarlie, M., Goldfeder, C., and Allen, P. (2007). Dimensionality reduction for handindependent dexterous robotic grasping. In *Proc. of IROS*, pages 3270–3275. 3
- Deimel, R. and Brock, O. (2013). A compliant hand based on a novel pneumatic actuator. In *IEEE ICRA*. 70, 91
- Diankov, R., Srinivasa, S., Ferguson, D., and Kuffner, J. (2008). Manipulation planning with caging grasps. In 2008 IEEE International Conference on Humanoid Robots. 21, 67
- Dowker, C. H. and Morwen, B. T. (1983). Classification of knot projections. *Topology* and its Applications, 16(1):19–31. 38, 88
- Dragulescu, D., Perdereau, V., Drouin, M., Ungureanu, L., and Menyhardt, K. (2007).
 3D active workspace of human hand anatomical model. *BioMedical Engineering* OnLine, 6(1):15. 52
- Edmond, S. L. H. and Komura, T. (2009). Character motion synthesis by topology coordinates. In *Computer Graphics Forum (Proc. Eurographics 2009)*, volume 28, Munich, Germany. 2, 3, 14, 16, 24, 25, 39, 88

- Ekvall, S. and Kragic, D. (2004). Interactive grasp learning based on human demonstration. International Conference on Robotics and Automation, 2004. Proceedings, 4:3519–3524 Vol.4. 53
- El-Khoury, S., Li, M., and Billard, A. (2012). Bridging the gap: One shot grasp synthesis approach. In *IEEE/RSJ IROS*, pages 2027–2034. 4
- El-Khoury, S. and Sahbani, A. (2010). A new strategy combining empirical and analytical approaches for grasping unknown 3d objects. *Robotics and Autonomous* Systems, 58(5):497 – 507. 67
- Ferrari, C. and Canny, J. (1992). Planning optimal grasps. *IEEE ICRA*, pages 2290– 2295. 4, 64
- Friedrich, H., Dillmann, R., and Rogalla, O. (1998). Interactive robot programming based on human demonstration and advice. In Sensor Based Intelligent Robots'98, pages 96–119. 53
- Fuhrmann, S., Ackermann, J., Kalbe, T., and Goesele, M. (2010). Direct Resampling for Isotropic Surface Remeshing. In *Proceedings of Vision, Modeling and Visualiza*tion (VMV), pages 9–16. 68
- He, R., Brunskill, E., and Roy, N. (2010). Puma: Planning under uncertainty with macro-actions. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010), pages 1089–1095. 43, 98
- Hu, H., Li, J., Xie, Z., Wang, B., Liu, H., and Hirzinger, G. (2005). A robot arm/hand teleoperation system with telepresence and shared control. In *Proc. of AIM*, pages 1312 –1317. 53, 91
- Huebner, K. (2012). A toolbox for box-based approximation, decomposition and GRasping. *Robotics and Autonomous Systems*, 60(3):367–376. 58
- Ilievski, F., Mazzeo, A. D., Shepherd, R. F., Chen, X., and Whitesides, G. M. (2011). Soft robotics for chemists. Angew Chem Int Ed Engl, 50. 70, 91

- Ivan, V., Zarubin, D., Toussaint, M., Komura, T., and Vijayakumar, S. (2013). Topology-based representations for motion planning and generalization in dynamic environments with interactions. *I. J. Robotic Res.*, 32(9-10):1151–1163. 9, 43, 45
- Kang, S. B., Ikeuchi, K., and Member, S. (1997). Toward automatic robot instruction from perception - mapping human grasps to manipulator grasps. *IEEE Transactions* on Robotics and Automation, 11:432–443. 53
- Klenin, K. and Langowski, J. (2000). Computation of writhe in modeling of supercoiled DNA. *Biopolymers*, 54(5):307–317. 15, 25, 26
- Kuperberg, W. (1990). Problems on polytopes and convex sets. In Workshop on Polytopes. 4, 21
- León, B., Ulbrich, S., Diankov, R., Puche, G., Przybylski, M., Morales, A., Asfour, T., Moisio, S., Bohg, J., Kuffner, J., and Dillmann, R. (2010). Opengrasp: a toolkit for robot grasping simulation. In *Proc. of the second international conference on simulation, modeling, and programming for autonomous robots*, SIMPAR'10, pages 109–120. 21, 66
- Li, W. and Todorov, E. (2006). An iterative optimal control and estimation design for nonlinear stochastic system. In Proc. of the 45th IEEE Conference on Decision and Control, pages 3242–3247, San Diego, CA, USA. IEEE. 20, 37
- Lindemann, S. R. and LaValle, S. M. (2004). Incrementally reducing dispersion by increasing voronoi bias in RRTs. In Proc. of International Conference on Robotics and Automation. 15
- Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. (2002). Discrete differentialgeometry operators for triangulated 2-manifolds. 32
- Miller, A. T. and Allen, P. K. (2004). GraspIt! A Versatile Simulator for Robotic Grasping. *IEEE Robotics and Automation Magazine*. 21, 52, 64, 66
- Murray, D. M. and Yakowitz, S. J. (1984). Differential dynamic programming and Newton's method for discrete optimal control problems. *Journal of Optimization Theory and Applications*, 43:395–414. 20, 37

- Murray, R. M., Li, Z., and Sastry, S. S. (2006). A Mathematical Introduction to Robotic Manipulation. CRC Press. 4, 21, 64
- NVIDIA (2012). Physx engine, http://www.geforce.com. 68
- O'Rourke, J. (1998). Computational Geometry in C. Cambridge University Press, New York, NY, USA. 29
- Pao, L. and Speeter, T. H. (1989). Transformation of human hand positions for robotic hand control. In Proc. of RA, pages 1758 –1763 vol.3. 4
- Paul, R. P. (1982). Robot Manipulators: Mathematics, Programming, and Control.MIT Press, Cambridge, MA, USA, 1st edition. 14
- Peer, A., Einenkel, S., and Buss, M. (2008). Multi-fingered telemanipulation mapping of a human hand to a three finger gripper. In *Proc. of RO-MAN*, pages 465 –470. 3
- Pokorny, F. T., Stork, J. A., and Kragic, D. (2013). Grasping objects with holes: A topological approach. In *IEEE ICRA*. 4, 54, 55
- Polthier, K. and Schmies, M. (2006). Straightest geodesics on polyhedral surfaces. In ACM SIGGRAPH 2006 Courses, SIGGRAPH '06, pages 30–38, New York, NY, USA. ACM. 17, 31, 32
- Ratcliff, J. W. (2009). Convex decomposition library. http://code.google.com/p/ convexdecomposition/. 68
- Rawlik, K., Toussaint, M., and Vijayakumar, S. (2010). An approximate inference approach to temporal optimization in optimal control. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *NIPS*, pages 2011–2019. Curran Associates, Inc. 19, 43, 94, 98, 99
- Rawlik, K., Toussaint, M., and Vijayakumar, S. (2012). On stochastic optimal control and reinforcement learning by approximate inference. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia. 20, 37, 38
- Rawlik, K., Zarubin, D., Toussaint, M., and Vijayakumar, S. (2014). An approximate inference approach to temporal optimization for robotics. 9, 38

- Reshetnyak, Y. G. (1993). *Geometry IV: Non-regular Riemannian Geometry*. Encyclopaedia of Mathematical Sciences. Springer. 32
- Rimon, E. and Burdick, J. W. (1994). Mobility of Bodies in Contact I: A New 2nd Order Mobility Index for Multiple-Finger Grasps. 66
- Rodriguez, A., Mason, M. T., and Ferry, S. (2012). From caging to grasping. I. J. Robotic Res., 31(7):886–900. 4, 22, 67
- Rohling, R. N. and Hollerbach, J. M. (1993). Optimized fingertip mapping for teleoperation of dextrous robot hands. *IEEE International Conference on Robotics and* Automation, pages 769–775. 53
- Rohling, R. N., Hollerbach, J. M., and Jacobsen, S. C. (1993). Optimized fingertip mapping: a general algorithm for robotic hand teleoperation. *Presence - Teleopera*tors and Virtual Environments, 2:203–230. 53
- Romero, J., Feix, T., Ek, C. H., Kjellström, H., and Kragic, D. (2012). Extracting postural synergies for grasping. *RSS*. 53, 54
- Sandilands, P., Ivan, V., Komura, T., and Vijayakumar, S. (2013). Dexterous reaching, grasp transfer and planning using electrostatic representations. *IEEE-RAS Interna*tional Conference on Humanoid Robots (Humanoids). 53, 54
- Sandilands, P., Zarubin, D., Pokorny, F. T., Toussaint, M., Komura, T., Kragic, D., and Vijayakumar, S. (2014). Taming variability in robotic manipulation with electrostatics and geodesic balls. 9, 82, 90
- Santello, M., Fl, M., and Soechting, J. F. (1998). Postural hand synergies for tool use. The Journal of Neuroscience. 52, 53
- Santello, M. and Soechting, J. F. (2000). Force synergies for multifingered grasping. Exp Brain Res. 53
- Saut, J.-P. and Sidobre, D. (2012). Efficient models for grasp planning with a multifingered hand. *Robotics and Autonomous Systems*, 60(3):347–357. 66

- Stork, J. A., Pokorny, F. T., and Kragic, D. (2013a). Integrated motion and clasp planning with virtual linking. In *IEEE/RSJ IROS*. 67
- Stork, J. A., Pokorny, F. T., and Kragic, D. (2013b). A topology-based object representation for clasping, latching and hooking. In *IEEE-RAS International Conference* on Humanoid Robots (HUMANOIDS). 4
- Suárez, R., Roa, M., and Cornella, J. (2006). Grasp quality measures. Technical report, Universitat Polytècnica de Catalunya. 4
- Takamatsu, J. and et.al. (2006). Representation for knot-tying tasks. *IEEE Transac*tions on Robotics, 22(1):65–78. 38
- Tamei, T., Matsubara, T., Rai, A., and Shibata, T. (2011). Reinforcement learning of clothing assistance with a dual-arm robot. *Proceedings of Humanoids*, pages 733– 738. 14, 39, 88
- Toussaint, M. (2009). Robot trajectory optimization using approximate inference. In Proc. of the 26th Int. Conf. on Machine Learning, pages 1049–1056. ACM. 19, 20, 38, 101
- Toussaint, M. (2012). Open robot simulator software. URL http://ipvs. informatik.uni-stuttgart.de/mlr/marc/source-code/index.html. 12
- Toussaint, M. (2013). Robotics, university lecture. URL http://ipvs.informatik. uni-stuttgart.de/mlr/marc/teaching/13-Robotics/. 14
- Vahedi, M. and van der Stappen, A. F. (2009). On the complexity of the set of threefinger caging grasps of convex polygons. In *Robotics: Science and Systems.* 30, 65
- Wakamatsu, H., Arai, E., and Hirai, S. (2006). Knotting/unknotting manipulation of deformable linear objects. International Journal of Robotics Research, 25(4):371– 395. 38
- Wei, G., Dai, J., Wang, S., and Luo, H. (2011). Kinematic analysis and prototype of a metamorphic anthropomorphic hand with a reconfigurable palm. *International Journal Of Humanoid Robotics*, 8(3):459 – 479. 71

- Zarubin, D., Ivan, V., Toussaint, M., Komura, T., and Vijayakumar, S. (2012). Hierarchical motion planning in topological representations. In *Robotics: Science and Systems (RSS)*, Sydney, Australia. 9, 26, 40, 47, 88
- Zarubin, D., Pokorny, F. T., Song, D., Toussaint, M., and Kragic, D. (2013a). Topological synergies for grasp transfer. In Hand Synergies how to tame the complexity of grapsing, Workshop, IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany. 9, 50, 55, 59, 61
- Zarubin, D., Pokorny, F. T., Toussaint, M., and Kragic, D. (2013b). Caging complex objects with geodesic balls. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan. 9, 31, 32, 33, 65, 76

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbstständig auf der Grundlage der in der Arbeit angegebenen Hilfsmittel und Hilfen verfasst habe.

Dmitry Alexandrovich Zarubin Stuttgart, den 29. Oktober 2014

Dmitry Zarubin

Curriculum Vitae

Mainzer Str 5 10247, Berlin, Germany ☎ +(49) 711 685 88229

Education

Since 2011	PhD student in Robotics, Machine Learning and Robotics Lab, University of Stuttgar	rt,
	Germany. http://ipvs.informatik.uni-stuttgart.de/mlr/	

- thesis title Topology-based Representations for Motion Planning and Grasping
- supervisor Prof. Dr. Marc Toussaint
- 2008-2011 Master in Computational Neuroscience, Bernstein Center for Computational Neuroscience, Technical University Berlin, Germany. http://www.bccn-berlin.de
- thesis title Stochastic Simulations of Single Cell Dynamics: Models of Coupled Ion Channels
- supervisor Prof. Dr. Susanne Schreiber
- 2004-2008 Bachelor in Applied Mathematics and Computer Science, The Faculty of Mathematics, Mechanics and Computer Science, Southern Federal University, former Rostov State University, Russia, http://sfedu.ru/international/
- thesis title Use of Genetic Algorithms for Time-series Forecasting with Neural Networks
- 1994-2004 Secondary school №92, Rostov-on-Don, Russia.

Work Experience

- Since 2013 **Research Associate**, Institute for Parallel and Distributed Systems, University of Stuttgart, Germany.
- 2011-2013 **Research Associate**, Machine Learning and Robotics Lab, Free University Berlin, Germany.
- 2010-2011 **Student Assistant**, Institute for Theoretical Biology, Humboldt University Berlin, Germany. http://www.neuron-science.de/

Projects

since 2014 EU project THIRD HAND http://www.3rdhandrobot.eu

task Development and implementation of sequential motion planning framework

- 2011-2014 EU project TOMSY http://www.tomsy.eu
 - task Development and implementation of novel planning algorithms using topology-based representations, development and benchmarking of grasping heuristics
 - 2010 Coherence of the feet imaginary movements. Berlin Brain-Computer Interface, TU Berlin, Germany. Supervisor Dr. Benjamin Blankertz
 - 2009 Analysis and classification of the MEG data. Donders Institute for Brain, Cognition and Behaviour, The Netherlands. Supervisor Dr. Ole Jensen

Research Interests

Alternate Representations for Motion Planning in Robotics, Grasp Transfer, Trajectory Optimization using Inference in Bayesian Networks, Synthesis of Caging Grasps, Surface Reconstruction and Point Clouds Segmentation, Grasp Heuristics, Machine Learning

Languages

Russian (native speaker), English (spoken and written proficiency), German (intermediate)

Awards and Honors

2008-2010 DAAD-Scholarship for Master studies in Germany. http://www.daad.de

- 2007 The winner of the 5th All-Russia conference "Youth of XXI century the future of the Russian science" in section "Information Technologies", Rostov-on-Don.
- 2004 Laureate of regional Mathematical Olympiad, Rostov-on-Don.
- 2004 Silver Medal for excellent scholar achievements.

Publications

- Dmitry Zarubin, Florian T. Pokorny, Marc Toussaint, and Danica Kragic. Caging complex objects with geodesic balls. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.
- [2] Dmitry Zarubin, Florian T. Pokorny, Dan Song, Marc Toussaint, and Danica Kragic. Topological synergies for grasp transfer. In Hand Synergies - how to tame the complexity of grapsing, Workshop, IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013.
- [3] Vladimir Ivan, Dmitry Zarubin, Marc Toussaint, Taku Komura, and Sethu Vijayakumar. Topology-based representations for motion planning and generalization in dynamic environments with interactions. *I. J. Robotic Res. (IJRR)*, 32(9-10), 2013.
- [4] Dmitry Zarubin, Vladimir Ivan, Marc Toussaint, Taku Komura, and Sethu Vijayakumar. Hierarchical motion planning in topological representations. In *Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.
- [5] Peter Sandilands, Dmitry Zarubin, Florian T. Pokorny, Marc Toussaint, Taku Komura, Danica Kragic, and Sethu Vijayakumar. Taming variability in robotic manipulation with electrostatics and geodesic balls. in submission, 2014.
- [6] Konrad Rawlik, Dmitry Zarubin, Marc Toussaint, and Sethu Vijayakumar. An approximate inference approach to temporal optimization for robotics. submitted, 2014.
- [7] Dmitry Zarubin, Ekaterina Zhuchkova, and Susanne Schreiber. Effects of cooperative ionchannel interactions on the dynamics of excitable membranes. *Physical Review E*, Jun 2012.
- [8] Dmitry Zarubin and Natalia N. Yachmeneva. Use of genetic algorithms for time-series forecasting with neural netwroks. In *Proceedings of the 5th All-Russia conference "Youth of XXI century the future of the Russian science"*, Rostov-on-Don, Russia, 2007.