Leveraging Problem Structure in Interactive Perception for Robot Manipulation of Constrained Mechanisms

vorgelegt von

Roberto Martín-Martín
geboren in Madrid

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –
genehmigte Dissertation

Promotionsausschuss
Vorsitzender: Prof. Olaf Hellwich
Gutachter: Prof. Dr. Oliver Brock
Gutachter: Prof. Dr. Dieter Fox
Gutachter: Prof. Dr. Tamim Asfour

Tag der wissenschaftlichen Aussprache: 26. Februar 2018

Berlin 2018

Thesis advisor: Professor Oliver Brock Roberto Martín-Martín

Leveraging Problem Structure in Interactive Perception for Robot Manipulation of Constrained Mechanisms

Abstract

In this thesis we study robot perception to support a specific type of manipulation task in unstructured environments, the *mechanical manipulation of kinematic degrees of freedom*. In these tasks the goal of the robot is to create controlled motion, i.e. to change configuration of the kinematic degrees of freedom (DoF) of the objects in the environment. Often, the environment contains articulated objects. Their manipulation is specially complex because the knowledge about their properties that would facilitate the task (e.g. their motion constraints, the geometry of their parts, their dynamic and frictional properties) are first revealed when the robot interacts with the object. Therefore, the perception of these objects should exploit interactions to create information-rich sensor signals. This type of problem and the perceptual methods that incorporate actions are called *interactive perception*. In this thesis we propose a general approach for interactive perception and instantiations of this approach into perceptual systems to build kinematic, geometric and dynamic models of articulated objects.

Perceptual problems in the domain of robot mechanical manipulation of DoF possess special challenges. While unstructured environments are usually continuously changing, robot mechanical manipulation exacerbates this characteristic. But in fact, these changes in the environment contain crucial information for a robot that aims to change purposely the state of the world. Perception for robot manipulation has to *extract information from changing sensor signals* and their relationship to changes in the environment and to actions. The perceptual process has to deliver information quickly and in an *online* manner, based only on past and current sensor signals, so that the information can be applied to ongoing interactions. And the perceptual solutions must be *versatile* enough to cope with a broad range of environmental and task conditions in which the robot should be able to manipulate DoF.

To address these challenges, we propose an approach for interactive perception that leverages four structural regularities of perceptual problems in the domain of robot mechanical manipulation of DoF. First, our approach leverages the *dependency between robot actions and changes in the sensor stream* using ideas from interactive perception. Second, our approach exploits the *temporal structure* of the physical processes involved in the mechanical manipulation of DoF using temporal recursion. Third, our approach makes use of task-specific priors that encode physical regularities of the world. These *physical priors* relate to the manipulation of DoF in unstructured environments and the sensor signal formation: physics laws that govern the motion of objects (e.g. kinematics), mathematical models for the signal formation (e.g. projective geometry), and assumptions about the physical properties of the environment (e.g. that the environment is composed of rigid solid parts). And fourth, our approach leverages *dependencies between multiple perceptual subtasks* that extract different information patterns about the same articulated object.

The approach we propose leverages the aforementioned problem structure with an interconnected network of recursive estimation processes encoding physical priors and exploiting robot interactions. We instantiated this approach in several robot perceptual systems, presented in consecutive chapters, to extract information about articulated objects –kinematic, geometric and dynamic properties– using only RGB-D information, or a combination of RGB-D and proprioceptive signals (e.g. applied wrenches, configuration of robot's joints). We study our proposed approach through these interactive perception systems. We evaluate if the systems can extract task-relevant information for the mechanical manipulation of DoF of articulated mechanisms for different objects and in varying and challenging environmental and task conditions. To truly demonstrate that the perceived information is useful for robot manipulation, we complement the perceptual systems with methods to monitor, control and steer the robot interaction based on the online perceived information. We also propose and evaluate a novel method to generate and select informative actions for interactive perception based on the information acquired so far.

Thesis advisor: Professor Oliver Brock Roberto Martín-Martín

Leveraging Problem Structure in Interactive Perception for Robot Manipulation of Constrained Mechanisms

Zusammenfassung

In dieser Dissertation untersuchen wir künstliche Wahrnehmungs-Methoden die ermöglichen, dass Robotern bestimmter Manipulationsaufgaben – die *mechanische Manipulation kinematischer Freiheitsgrade* – in unstruktierten Umgebungen lösen. Der Roboter soll dabei in die Lage versetzt werden die kinematischen Freiheitsgrade von Objekten in seiner Umgebung durch zielgerichtete Bewegungen zu verändern. Menschliche Umgebungen sind voll von artikulierten Objekten, die nur bestimmte kinematische Freiheitsgrade zulassen. Diese Objekte zu manipulieren ist besonders schwierig, da die Konsequenzen der Handlungen des Roboters von der kinematischen Struktur des Objekts, seiner Geometrie und den dynamischen Eigenschaften (z.B. Gelenkreibung) abhängt. Hinzu kommt, dass sich diese Eigenschaften nur erkennen lassen wenn der Roboter mit dem Objekt interagiert. Deshalb sollte die Wahrnehmung solcher Objekte Interaktionen ausnutzen, um Sensorsignale mit hohem Informationsgehalt zu generieren. Diese Art von Problemen und die Wahrnehmungsmethoden die Handlungen berücksichtigen nennt man *interaktive Wahrnehmung*. In dieser Dissertation schlagen wir einen allgemeinen Ansatz für interaktive Wahrnehmung vor um kinematische, geometrische und dynamische Modelle artikulierter Objekte zu erstellen.

Wahrnehmungsprobleme im Bereich der mechanischen Manipulation von Freiheitsgraden sind durch besondere Herausforderungen gekennzeichnet. Während sich unstrukturierte Umgebungen ohnehin permanent verändern, wird dieser Umstand durch manipulierende Roboter noch zusätzlich verschärft. Tatsächlich enthalten die Veränderungen der Umgebung jedoch wichtige Informationen, die ein Roboter ausnutzen kann, um den Zustand der Welt zielgerichtet zu verändern. Wahrnehmung für Manipulation muss *Informationen von sich verändernden Sensorsignalen*, deren Zusammenhang mit Änderungen in der Umgebung und deren verursachenden Handlungen extrahieren. Der Wahrnehmungsprozess muss Informationen unmittelbar zur Verfügung stellen, so dass diese in laufenden Interaktionen verwendet werden können. Zusätzlich müssen die Wahrnehmungslösungen *vielseitig* genug sein um mit einer breiten Palette an Umgebungen und Aufgaben zurechtzukommen, in denen der Roboter Freiheitsgrade manipulieren soll.

Zur Bewältigung dieser Herausforderungen stellen wir einen Ansatz für interaktive Wahrnehmung vor, der vier strukturelle Regularitäten von Wahrnehmungsproblemen im Bereich der Manipulation von Freiheitsgraden ausnutzt. Erstens nutzen wir die *Korrelation zwischen den Handlungen des Roboters und Änderungen im Sensorsignalfluss* aus. Zweitens machen wir uns die *zeitliche Struktur* der physikalischen Prozesse zu Nutze, indem unser Ansatz auf zeitlicher Rekursion basiert. Drittens benutzen wir aufgabenspezifisches Vorwissen, das physikalische Regelmäßigkeiten der Welt abbildet. Dieses *physikalische Vorwissen* bezieht sich auf die Manipulation von Freiheitsgraden in unstrukturierten Umgebungen und die Entstehung von

Sensorsignalen: Physikalische Gesetze, die die Bewegung von Objekten beschreiben (z.B. Kinematik), mathematische Modelle für die Signalentstehung (z.B. projektive Geometrie) und Annahmen über die physikalischen Eigenschaften der Umwelt (z.B. dass diese aus Festkörpern zusammengesetzt ist). Viertens nutzen wir die *Korrelation zwischen mehreren informationsverarbeitenden Prozessen* (Unteraufgaben der Wahrnehmung) bezüglich eines einzelnen artikulierten Objekts aus, indem Informationen zwischen Teilprozessen ausgetauscht werden.

Unser Ansatz macht sich die o.g. Problemstruktur mittels eines ineinandergreifenden Netzwerks aus rekursiven Schätzprozessen zu Nutze. Wir haben diesen Ansatz in Form mehrerer künstlicher Wahrnehmungssysteme implementiert. Diese werden in aufeinanderfolgenden Kapiteln vorgestellt und beziehen auf die Art der Information, die dabei über artikulierte Objekte gewonnen wird: kinematische, geometrische und dynamische Eigenschaften. Unser Ansatz benötigt lediglich RGB-D Daten oder eine Kombination aus RGB-D und propriozeptiven Signalen (z.B. angewendete Dyname oder Konfiguration der Robotergelenke). Wir analysieren unseren Ansatz mit Hilfe dieser interaktiven Wahrnehmungssysteme. Wir evaluieren ob die Systeme aufgabenrelevante Informationen für die mechanische Manipulation von Freiheitsgraden für unterschiedliche Objekte und unter wechselnden Umgebungs- und Aufgabenumständen extrahieren können. Um zu zeigen, dass die wahrgenommene Information für einen manipulierenden Roboter hilfreich ist, ergänzen wir das Wahrnehmungssystem mit Methoden zum Überwachen, Regeln und Steuern der Interaktion durch den Roboter. Zusätzlich stellen wir eine neue Methode vor die informative Handlungen für interaktive Wahrnehmung erzeugt.

Acknowledgments

Writing this thesis would not have been possible without the support and help of many marvellous people.

First, the incredible team at the *Robotics and Biology Lab*. I've learned from you, discussed with you, laughed and even cried. My experience and what I take with me from these years is in essence a melting pot of you, cooked at many conversations, projects and moments together. Thank you Alexander, Andreas, Angela, Arne, Armin, Can, Clemens, Dennis, Dov, Elöd, Emily, Eveline, Florian, Freek, Gabriel L., Gabriel Z., Georg B., George, Henrietta, Ingmar, Ines, Janika, Jessica, Johannes, José, Kolja, Mahmoud, Malte, Manuel, Marc, Marianne, Melinda, Michael, Nicolas, Oliver, Philipp, Raphael, Rico, Robert, Roman, Sebastian H., Sebastian K., Serena, Stanimir, Steffen, Tim, Thomas, Vincent, Wolf.

Thank you Dubi for being my first working colleague at RBO. Thank you Clemens for your Übung Unterricht where I learned to read your lips, an important asset for the many years of laughing and science with you that were to come. Thanks Sebastian, working with you was a motivation to come everyday. Thanks Arne, Raphael, Rico, Vincent, you have been my hardest critics and my best supporters. Thank you Manolo, behind the slow-moving man is a fast-moving mind. Thank you Janika, for helping me with the fierce monster of the German bureaucracy and the research life.

I feel so lucky I had the honour and pleasure of working with you and having you in my life.

I thank also to the institutions that made this thesis possible, the Technische Universität Berlin, the Deutsche Forschungs Gemeinschaft, the European Commission and the Alexander von Humboldt Foundation. And a very important institution that supported me from the beginning, my family. Thank you Dad, Mum, Pifo. Thank you Thomas, you have been so understanding and supporting these many years. Your positive energy helped me go further.

And last, but not least, thank you Oliver, my Thesis father. You infected me with the addiction to robotics and science, to question everything and everyone, even myself. You taught me how to be a researcher, a critical thinker and an expert in music videos from the 80s.

Prepublication and Statement of Contribution

Parts of this thesis have been previously published in the following peer-reviewed articles:

A. Roberto Martín-Martín and Oliver Brock, Online Interactive Perception of Articulated Objects with Multi-Level Recursive Estimation Based on Task-Specific Priors. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2494–2501. Chicago, USA. 2014

B. Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An Integrated Approach to Visual Perception of Articulated Objects. *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5091–5097. Stockholm, Sweden. 2016

C. Roberto Martín-Martín, Arne Sieverling and Oliver Brock, Estimating the relation of perception and action during interaction. *In International Workshop on Robotics in the 21st century: Challenges and Promises*. Göttingen, Germany. 2016

D. Roberto Martín-Martín and Oliver Brock, Building kinematic and dynamic models of articulated objects with multi-modal interactive perception. *In Proceedings of the AAAI Symposium on Interactive Multi-Sensory Object Perception for Embodied Agents*, pages 473–476. Palo Alto, USA. 2017

E. Roberto Martín-Martín and Oliver Brock, Cross-modal interpretation of multi-modal sensor streams in interactive perception based on coupled recursion. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vancouver, Canada. 2017

F. Clemens Eppner*, Roberto Martín-Martín* and Oliver Brock, Physics-based selection of actions that maximize motion for interactive perception. *In RSS workshop: Revisiting Contact – Turning a problem into a solution*. Boston, USA. 2017 (* contributed equally)

Own contributions to [A] *(© 2014 IEEE):* I (RMM) was the sole first author of this paper. I conceived, designed, implemented and evaluated the algorithm and experiments presented in the paper and made the main contribution to paper writing. The last author (OB) conceived the project idea, gave scientific advice and contributed to paper writing.

Own contributions to [B] *(© 2016 IEEE):* I (RMM) was the first author, conceived the algorithmic idea and contributed the major part of the implementation. The second author (SH) contributed to the implementation. SH and RMM equally contributed to the experimental evaluation and to paper writing. OB gave scientific advice and contributed to paper writing.

Own contributions to [C]: I (RMM) was first author, integrated the implementation of AS into the perceptual system I developed, and adapted the implementation for the real world experiments. The second author (AS) provided the conceptual idea and implemented the uncalibrated Jacobian estimation. RMM and AS contributed equally to the experimental evaluation and to paper writing. The last author (OB) gave scientific advice and contributed to paper writing.

Own contributions to [D] *(© 2017 AAAI):* I (RMM) was the sole first author of this paper. I conceived, designed, implemented and evaluated the algorithm and experiments presented

in the paper and made the main contribution to paper writing. The last author (OB) gave scientific advice and contributed to paper writing.

Own contributions to [E] *(© 2017 IEEE):* I (RMM) was the sole first author of this paper. I conceived, designed, implemented and evaluated the algorithm and experiments presented in the paper and made the main contribution to paper writing. The last author (OB) gave scientific advice and contributed to paper writing.

Own contributions to [F]: I (RMM) share first authorship with (CE). We both contributed equally to the conception, design and implementation of the algorithm and the experiments presented in this extended abstract. OB gave scientific advice and contributed to paper writing.

Appearance of Previous Publications in the Thesis

Chapter 1 , 2, and 3 are original to this thesis.

Chapter 4: The content of this chapter is based on the publication [A]. The perceptual system and parts of the related work has been previously published in [A]. This chapter contains a novel introduction to the context of the thesis and a discussion of the implications of the algorithm and future work. It extends and complements the explanation of the system. It also presents additional experimental results and analyzes the methods sensitivity to parameters and a new qualitative analysis of the results.

Chapter 5: The content of this chapter is based on the publication [B]. The contextualization of the perceptual system in the unified approach of the thesis is novel to this chapter.

Chapter 6: The content of this chapter is based on the publication [C,D,E]. The perceptual system and parts of the related work has been previously partially published in [C,D,E]. In this chapter we extend and provide complete explanations of the methods, specially the estimation of dynamic properties. This chapter contains a novel introduction to the context of the thesis and a discussion of the implications of the algorithm and future work. It also presents additional experimental results. The chapter links the work presented in these publications into a unified view.

Chapter 7: The approach has been significantly extended from the presented method of [F]. This chapter contains a novel introduction, related work, and a discussion of the implications of the algorithm and future work. It also presents additional experimental results.

Chapter 8: This chapter is original to this thesis

Table of Contents

List of Figures

List of Tables

1 Introduction

Robots have been a successful working force in factories for decades[1]. In these environments, most of the robot's tasks are considered *mechanical manipulation tasks*, e.g. picking up objects, placing them and assembling them. In these tasks, the robot interacts with the environment by exerting forces in order to move objects (Mason, 2001). When the robot moves an object, the robot is effectively changing the configuration of the kinematic *degrees of freedom* (DoF) of the environment. We consider the purposeful change of the DoF of the environment to be the goal of any mechanical manipulation task.

Robots in other types of human environment –homes, airports, streets, schools, hospitals,...– are not as successful as robots in factories. In these other environments robots are still not capable of manipulating kinematic DoF successfully and reliably. Until now, the only successful robot in this type of human environments is a rolling vacuum cleaner that suctions dust (with all my respect); very different to mechanical manipulation tasks robots execute in factories. The differences between factories and other types of human environments have been so far insurmountable obstacles to extend the success of mechanical industrial manipulators to these other environments.

Factories are *structured environments*: they are carefully designed and controlled to facilitate robotic manipulation. On the other hand, other human environments are *unstructured*: dynamic, uncontrolled, uncertain and very different from one another. These differences have crucial implications on the development of robots that aim to physically manipulate these environments (Kemp et al., 2007).

In structured environments the information that is relevant for the robot's manipulation task can be given a priori (e.g. where to find the parts to assemble the car, how to move without collisions from A to B, or how much force to apply on a tool) or assumed implicitly (e.g. "when the robot motion is finished, the picked part will be at the desired configuration"). On the contrary, in unstructured environments it is very difficult to give this information a priori because of the large variety of environments, tasks and conditions the robot could confront, and the ever-changing nature of these environments. In fact, the robot has to continuously monitor the progression of the task to adapt online towards the goal and detect the task termination. Thus, robots in unstructured environments are advocated to continuously

[1]The International Federation of Robotics estimated 1.632 million industrial robots on operation in 2015. They predict 2.589 million industrial robots active in 2019 (International Federation of Robotics, 2016).

acquire the information that is relevant for the task from their input sensor signals. In other words, robots have to **perceive** to compensate the absence of prior knowledge about the environment (Ersen et al., 2017). This process of continuously acquiring the information that is relevant for the manipulation task is what we call *perception for robot manipulation* and is the topic of this thesis.

Even in unstructured environments the robot's perceptual system needs to make some assumptions to be able to extract task-relevant information from the sensor signals. The input signals correlate to different physical processes or types of energy (e.g. vision to electromagnetic fields, audio to air pressure waves, haptics to mechanical forces). These input signals are usually high-dimensional and full of noise. Moreover, the input signals arrive continuously and change quickly, leading to a complex sensor stream. Searching for information in this input space is a process that the robot cannot approach naïvely, as an uninformed search. To put some numbers on this, a binary image of size 20×20 pixels spans a space of $2^{20\times20} \approx 10^{120}$ elements, impossible to be rendered, searched and analyzed element by element. Thus, the perceptual system of the robot needs to make assumptions about the underlying structure of the problem. These assumptions are prior information about the problem that the system leverages to solve it.

Artificial perception is plenty of solutions that leverage the right problem structure. We can analyze the structure leveraged by solutions of a largely studied perceptual task: object classification. Computer vision solutions for object classification have traditionally been based on feature descriptors like SIFT (Lowe, 2004) or SURF (Bay et al., 2008). These solutions assume that the classification of an image should not be sensitive to photometric effects like scale or orientation. More recent solutions apply artificial neural networks (ANN) both to generate an intermediate feature representation and to learn the mapping between features and classes (Krizhevsky et al., 2012a). These networks leverage the spatial structure of the image using convolutions (Rumelhart et al., 1985, LeCun et al., 1998) and the structure of the visual classification problem using hierarchical architectures (e.g. groups of pixels define elementary image structures that, when combined, define an object).

Perceptual solutions to object classification from robotics have exploited additional assumptions that are available to embodied agents. These solutions use the robot's capabilities to interact with the objects, revealing and generating a sensory response that makes the classification easier (Sinapov et al., 2014, Willimon et al., 2011, Venture et al., 2009, Atanasov et al., 2014). Known or learned correlations[2] between actions and changes in the sensor signals are leveraged as prior knowledge to extract information from the sensor stream and to solve the classification task. The common pattern in the aforementioned solutions for object classification, both in computer vision and in robotics, is to identify and leverage the right structure of the problem.

So far, we have seen two contradicting lines of argument with respect to the amount of prior information we should encode in the robot's perceptual system. On one hand, we have argued that perception in unstructured environment needs to reduce its dependency on a-priori given information. In this process, the robot will gain in generality at the cost of making robot perception more challenging, since the robot will be able to manipulate in a larger variety of environments based on the perceived task-relevant information.

[2]Some authors use the term *correlation* only to indicate a linear relationship between variables. In this thesis we adopt the less restricted definition from the Cambridge Dictionary of Statistics:

> **Correlation:** A general term for interdependence between pairs of variables (Everitt & Skrondal, 2002, p. 107).

On the other hand, we argued that perception needs prior assumptions about the structure of the problem and the environment. Only based on these assumptions the perceptual solution can extract information from the complex input sensor stream. The more information can be assumed a-priori, the less challenges for perception. But excessive assumptions could restrict the generality of the solution. This trade-off is depicted in Figure 1.1. The goal when developing a perceptual solution for robot manipulation in unstructured environments is to find a balance between assumptions about the problem and generality of the perceptual approach. Finding this balance involves 1) identifying generic prior assumptions that apply to a large variety of environments, problem conditions, and, possibly, perceptual tasks, and 2) devising methods that leverage these assumptions and are capable of extracting the task-relevant information from the robot's sensor stream.

In this thesis, our goal is to leverage the inherent structure of problems in the domain of perception for robot manipulation in unstructured environments. We will focus on the specific type of manipulation tasks we mentioned before, *mechanical manipulation tasks*, where the goal of the robot is to change the kinematic DoF of the environment. Often, the environment contains articulated objects: objects composed of rigid parts and connections between them (e.g. doors, drawers, laptops, scissors). Robots that aim to manipulate unstructured environments must be able to actuate this type of objects, i.e. to change the configuration of their internal DoF. This manipulation is specially complex because the knowledge about their properties that would facilitate the task (e.g. their motion constraints, the geometry of their parts, their dynamic and frictional properties) are first revealed when the robot interacts with the object. Perception that supports the mechanical manipulation of DoF of environments including articulated objects should be able to perceive motion constraints and other properties of these objects. These perceptual tasks will be the goal of the systems presented in this thesis, as we will see in Chapters 4, 5 and 6.

In robot mechanical manipulation the goal of the robot is to change the state of the environment (its DoF) purposefully through interactions. Therefore, we consider that the goal of perception for robot mechanical manipulation is to extract the information that is relevant for this task. Shifting the goal of perception from building a complete model (Marr, 1982) towards extracting task-relevant information is a journey that began with the concepts of *Purposive Active Vision* by Aloimonos (1990) and *Animate Vision* by Ballard (1991). This goal-shift brings the behavior of the interacting agent to the central role (Brooks, 1986).

There are multiple ways to define that something is *relevant* to a task. In this thesis, we adopt the definition by Hjørland & Christensen (2002):

> Something is relevant to a task if it increases the likelihood of accomplishing the goal which is implied by the task (Hjørland & Christensen, 2002, p. 964).

To evaluate if the perceptual systems presented in this thesis achieve the aforementioned goal of perception for robot mechanical manipulation, we will evaluate experimentally the improvement in the manipulation task when the robot uses the perceived information (e.g. Chapter 4.3.4, and Chapter 6.6.2). We will complement our proposed perceptual systems with manipulation methods to control and steer the robot actions based on the information. Additionally, in Chapter 7 we will present a method that uses the perceived information to generate, plan, select and execute information-rich robot actions.

In the rest of this chapter we will analyze further the structure of perceptual problems to support robot mechanical manipulation of DoF. We will see that the manipulation task imposes additional challenges to the already demanding problem of robot perception in unstructured environments. But we will also see that the structure of the problems presents some opportunities in the form of generally valid prior assumptions that we can leverage to

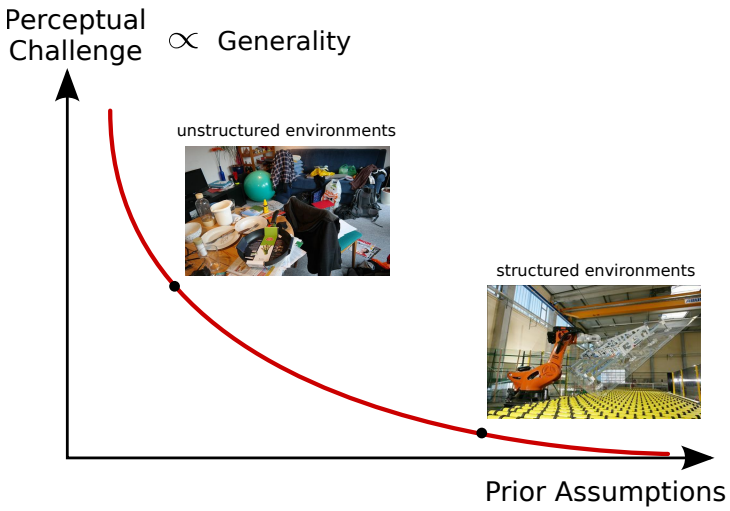


Figure 1.1: Pictorial representation of the trade-off in perceptual problems between prior assumptions and challenges (and generality); a point on the curve represents a perceptual task; perception in structured environments (e.g. a industrial robot factory, right picture) is less challenging because more information can be assumed a priori; perception in unstructured environments (e.g. an untidy room, left picture) is more challenging because less information can be assumed a priori

solve them. In the following we will identify the most important challenges and opportunities, and propose based on them an approach to leverage the opportunities and address the challenges.

A Note about Priors: In Bayesian probability theory the term "prior" has a clear and restricted definition: a prior is a probability distribution that, combined with an observation through the likelihood function in the Bayes Rule, generates the posterior distribution (see 2). In this thesis, we will use the term prior in a broader sense: priors are information about the problem that is known before the actual sensor data of the problem has been observed. In this sense, the term prior could be a probability distribution (Bayesian definition), but also a model, an algorithm to apply to a problem, the structure of a neural network or any other type of information about the task, the environment or the problem. This information represents assumptions about the structure of the problem that we leverage to solve it. Training data can be also considered prior information, since it is observed before the actual perceptual process (applying the trained model to new data).

1.1 Challenges in Perception for Robot Manipulation of DoF

Given that the goal of a mechanical manipulation task is to purposefully change the kinematic state of the environment, perception for robot mechanical manipulation must focus on these changes: detect them, track them, and understand their relationship to the robot's actions. In contrast, other fields of artificial perception (in the context of manipulation) are focused on extracting static, geometric models of the environment. These models cannot represent the dynamic nature of environment and task and are unrelated to actions. Changes in the environment and their relationship to actions, however, provide the most appropriate perceptual signal to support robot mechanical manipulation – more appropriate than static, geometric models–, since they can be used to guide the robot towards the manipulation goal. The **first challenge** in perception for robot manipulation (CH1) then consists of devising perceptual methods to extract information from changing sensor signals and their relationship to changes in the environment and to actions. Only if acquired quickly enough, this information can be used to monitor, steer and control robot's interaction and achieve the desired change in the environment. The **second challenge** (CH2) is to develop perceptual algorithms that can perceive this information online in unstructured environments.

As commented before, structured environments are challenging because they are uncontrolled and very different from one another. The **third challenge** in perception for robot mechanical manipulation (CH3) is to generate versatile algorithms that can cope with the variability of conditions of the environment and of the task in which the robot needs to successfully perceive and manipulate the DoF of the environment.

Other researchers studying the challenges of perception and manipulation in unstructured environments have suggested similar challenges. For example, Kemp et al. (2007) list the following properties of the environment as challenges for robots: dynamic variation, real time constraints, and variation in object placement, type, appearance, structure, and sensory signals. We think the three challenges we discussed (CH1-CH3) is a condensed version of the ones of their analysis, while we did not include the challenges due to human presence in the environment.

1.2 Opportunities in Perception for Robot Manipulation of DoF

Perceptual problems in the context of the robotic manipulation of DoF present structural regularities that we can leverage to address them. These regularities represent opportunities for a perceptual system to extract task-relevant information and overcome the aforementioned challenges. In the following we will identify these regularities in perceptual problems for robotic mechanical manipulation.

The goal in robotic mechanical manipulation is to modify the kinematic state of the environment. The manipulation causes changes in the sensor signals and exacerbates the dynamic behavior of the unstructured environments. In the previous section we identified the changing nature of the unstructured environments as one of the challenges in perception for robotic mechanical manipulation. However, what is a challenge can be also an opportunity. By changing the environment, robot interactions generate information-rich changing signals and reveal information that could not be perceived passively, e.g. motion constraints and dynamic properties of articulated mechanism. Moreover, knowledge of the interaction can be used as prior to restrict the space of possible perceptual solutions and simplify perceptual tasks. The **first opportunity** (OP1) is thus to exploit the additional knowledge provided by robot interactions to generate and interpret changes in the sensor signal and extract task-relevant information. Methods that integrate interactions as part of the perceptual solution are called *Interactive Perception (IP)* methods.

Our goal is that the robot perceives and manipulates DoF in unstructured environments based on perception. These environments vary strongly from one another. However, robots are embodied agents and therefore, their perceptual tasks are always grounded into the same physical world. Physical priors (e.g. physics laws, knowledge about the sensor signal formation) are universal constraints that help to understand the sensor data generated by physical events within it. A **second opportunity** (OP2) is to exploit the physical priors prevalent to all unstructured environments for the interpretation of sensor signals.

The sensor signals the robot acquires at a point in time are intimately related to the signals acquired before (and later). This relationship is stronger the shorter the time interval between sensor signals. This is a consequence of the smooth nature of many physical processes in the environment (e.g. motion): the environment does not change drastically from one time step to another. For perception that implies that the information acquired before is a strong prior to interpret current signals. In other words, we can exploit recursively what has been perceived so far to acquire information now, and to focus on the changes. A **third opportunity** (OP3) is to leverage the temporal structure of the perceptual task evidenced in the continuous sensor stream.

Robot mechanical manipulation does not rely on a single type of information or property of the environment. For a robot that aims to manipulate the DoF of an articulated object, multiple properties are relevant, e.g. the kinematic constraints of the object, its geometry or its dynamic and friction properties. These information patterns are the result of different information extraction processes, or perceptual subtasks. These subtasks are not completely independent because the properties they perceive depend on each other (e.g. perceiving friction on the joints of an object depends on assumed object's kinematic structure), and the sensor signals they use originate in the same physical interaction and/or the same object. Changes in the sensor signals are best explained combining information from these subprocesses (e.g. the change in visual appearance of an articulated object depends on its kinematic structure and the geometry of its parts). Therefore, information from one subprocess can be used to help the others. The **fourth opportunity** (OP4) in perception for robotics is to exploit the interrelation between perceptual subtasks so that information from one can be used as prior to interpret sensor signals in the other in a self-bootstrapping manner, and their information can be combined to better support the robot manipulation.

The following table summarizes the challenges and the opportunities we identify in perception for robot manipulation of kinematic DoF. The abbreviations OP1 to OP4, and CH1 to CH3 along the thesis document contain hyper-references that point to the definitions of this table:

Challenges in Interactive Perception	
Challenge 1 (CH1)	To extract information from changing sensor signals and their relationship to actions
Challenge 2 (CH2)	Online Perception: quickly enough to support ongoing interaction, based only on past and current sensor signals
Challenge 3 (CH3)	To be versatile to cope with different environmental and task conditions
Opportunities in Interactive Perception	
Opportunity 1 (OP1)	To exploit the additional knowledge provided by robot interactions to generate and understand changes in the sensor signal
Opportunity 2 (OP2)	To interpret sensor signals as manifestations of underlying physical processes and known properties
Opportunity 3 (OP3)	To use the information perceived before to help you interpret the sensor signals now
Opportunity 4 (OP4)	To use the information from one perceptual subtasks to help solving other subtasks

Table 1.1: Summary of the challenges and the opportunities in interactive perception for robot manipulation

1.3 Our Approach

In this thesis, we propose an approach to leverage the aforementioned opportunities and address the challenges of perception for robot mechanical manipulation of kinematic DoF. Our approach is in essence a structure of interconnected recursive estimation processes (Figure 1.2).

This algorithmic architecture allows us to exploit the structure of the perceptual problem, as we discuss in the following. Clearly, recursive estimation is a well suited computational solution to exploit the temporal structure in the sensor signals and the physical environment (OP3) and to extract information from changing sensor signals online (CH1, CH2). Recursive estimation processes require models to predict and update the perceived information online. In our approach these models are based on task-related physical priors (e.g. kinematics, rigid body assumption, projective geometry) encoding general physical knowledge (OP2). The physical priors "enrich" the sensor signals and allows the recursive process to interpret them as evidence of the modelled physical processes, while being general enough to apply to many unstructured environments (CH3).

As methodological approach, we propose to factorize the original perceptual task into simpler subtasks that we can address using recursive processes based on physical priors. To compose these factors, our approach intercommunicates the recursive processes in a way that reuses priors and results from other processes to help on each subtask (OP4). Finally, our approach for perception integrates robot interactions to create information-rich signals, and uses knowledge about these interaction as additional prior knowledge for the interpretation of these signals (OP1).

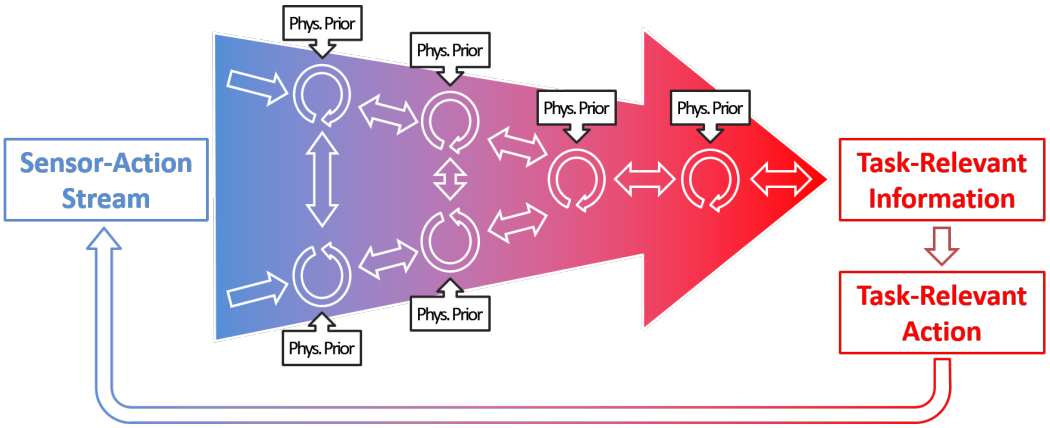


Figure 1.2: Our view of interactive perception for manipulation: a structure of highly connected subprocesses that interpret sensor-action signals based on task-specific physical priors and information from other subprocesses; generic sensor-action signals are enriched with the physics priors and transformed into a stream of task-relevant information; interactive perception generates information to monitor, steer, generate and select task-relevant actions; actions reveal information contained in the sensor-action stream

Figure 1.2 illustrates the proposed approach for interactive perception for robotic mechanical manipulation. We depict the recursive processes that tackle perceptual subtasks as loops. The recursive processes integrate sensor signals and information about the interaction (sensor-action stream), physical priors, previous estimates (recursively), and information from other processes to continuously extract patterns of information. The output can act again as an input to another recursive estimation process. The outputs can therefore be seen as signals of a virtual sensor. The combination, integration, and sequencing of these estimation processes leads to a general flow of information from sensor-action data (blue, on the left) to task-relevant information (red, on the right). The idea is to **incrementally process and interpret the sensor streams through cascades of interconnected estimation processes until the necessary perceptual information has been extracted** robustly and

efficiently. The interconnectivity of these processes reflects that different but correlated information can be leveraged to further increase the robustness and efficiency. Based on the perceived information robots can monitor, control, and plan actions, thus closing the loop and affecting the input to the processes.

We will instantiate this perceptual approach into several interactive perception systems in the following chapters (chapters 4, 5 and 6). These systems will perceive different properties of articulated objects that are relevant for robots manipulating the object's DoF. The evaluation of these systems will allow us to assess if our proposed approach for interactive perception acquires task-relevant information overcoming the challenges of the perceptual problem.

1.4 Contributions and Thesis Structure

The **first intellectual contribution** of this thesis is the approach for interactive perception we presented above. Our hypothesis is that, based on this general approach, we can build interactive perceptual systems that support robot mechanical manipulation of DoF in the environment. In the rest of the thesis we will propose and study perceptual systems based on this generic approach to validate and evaluate this hypothesis.

The **second contribution** of this thesis is an interactive perceptual system based on vision (an RGB-D stream) that builds kinematic models of articulated objects in an online manner. The system will be presented and evaluated in Chapter 4. Studying this first system we will evaluate if our general approach is applicable to a single perceptual task using a single sensor modality. This work led to one publication (Martín-Martín & Brock, 2014) and an open-source perceptual system (Online Interactive Perception, Martín-Martín (2014))[3].

The **second contribution** of this thesis is a system that integrates the segmentation of images and reconstruction of the shape of the parts of the articulated object with the perception of kinematic models. With this system we will study how to further exploit interdependencies between perceptual subtasks (OP4). We evaluate the benefits of the integration comparing the results of the integrated and non-integrated systems. This perceptual system will be presented and evaluated in Chapter 5. This work led to another publication (Martín-Martín et al., 2016a) and we included it as part of the previous open-source perceptual system.

The **third contribution** of this thesis is an interactive perceptual system based on multiple sensor modalities including signals about the specific robot action (proprioception). We will propose a method to leverage interdependencies between perceptual subtasks (OP4) using signals from different sensor modalities. To exploit their interdependencies we will make use of the concept of cross-modality: using information from one modality as prior to interpret another. Our goal will be to increase the versatility of the system to cope with a broader range of environmental and task conditions, and also to perceive new properties, e.g. the dynamic properties (friction) of articulated objects. In our path to further exploit the interdependencies between interactions (proprioceptive signals) and changes in the sensor signals we will need to hardcode models of these interdependencies in our perceptual system. These models will assume certain robot morphology and are therefore not general. We will explore simple online learning methods to obtain such interactive models from experiences and reduce the dependency on predefined interaction models. The description and evaluation of this third system is the content of Chapter 6. This work led to three publications (Martín-Martín & Brock, 2017a), (Martín-Martín et al., 2016b) and (Martín-Martín & Brock, 2017b), and an

[3] https://github.com/tu-rbo/omip

open-source perceptual system (Online Multi-Modal Interactive Perception, Martín-Martín (2016))[4].

Action is a crucial component of any interactive perception system. The **fourth contribution** of this thesis is the development of methods to generate robot motion with two objectives: 1) to safely explore and generate informative sensor signals for perception, and 2) to exploit the perceived information to support the mechanical manipulation task. As part of the perceptual system of Chapter 6 we will present a compliant controller for the safe actuation of articulated mechanisms, and a method to exploit the information obtained with our perceptual systems to generate new robot trajectories online. Chapter 7 will present a novel action selection algorithm for interactive perception based on the systems presented in Chapters 4, 5, and 6. The algorithm will allow the robot to build incrementally richer models by generating and choosing the most informative actions. This work was part of two publications (Martín-Martín et al., 2016b) and (Martín-Martín & Brock, 2017b), and led to a new publication (Eppner et al., 2017).

[4] https://github.com/tu-rbo/omip/tree/omip2

2
Related Work

In this thesis, we present a method to overcome the challenges of problems in interactive perception for robot manipulation leveraging structural regularities of these problems. In the introduction we identified four structural regularities shared by many perceptual tasks in robot manipulation. They represent opportunities to overcome the perceptual challenges exploiting the correlation between interactions and (changes in) sensor signals (OP1), the physical structure of the environment and the sensor signal formation (OP2), the temporal structure in the manipulation processes and its influence in the sensor stream (OP3), and the interdependencies between information extraction subprocesses (OP4). In this section, we will review how these four problem regularities have been exploited in previous perceptual solutions in the literature. We will see that many of the advances in artificial perception can be explained by a better exploitation of these regularities, which motivates a historically growing use of them.

Since the work presented here belongs to the family of interactive perception approaches, we will also analyze previous methods in that field. We will evaluate how they use the four aforementioned opportunities (problem regularities) to overcome the challenges of perception for robotics in unstructured environments. At the end of this chapter, we will summarize and classify the interactive perception methods included in our review in Table 2.1. We classify them by their application and their exploitation of the four opportunities.

Our ultimate goal is to provide robots with perceptual skills that enable manipulation in unstructured human environments. Obviously, a successful example of a perceptual system in this domain is the human perceptual system. We will review work in the fields of psychology, cognitive science, and philosophy that provide evidence of the crucial role of the four aforementioned problem regularities for the robustness and versatility of the human perceptual system.

2.1 Leveraging Interaction as Prior for Perception (OP1)

2.1.1 From Passive to Active to Interactive Perception

Since the early days of robotics until our days, most research in robot manipulation relies on perception that extracts geometric 3D models from sensor data. Obtaining such a complete and detailed 3D model has been the main goal in visual perception since the seminal work of Marr (1982). The hope was that any task in robotics could be easily solved given an

accurate geometric 3D model of the environment. The advent of RGB-D sensors has made the acquisition of such models particularly easy since they "solved" the hard problem of finding the inverse transformation from 2D to 3D. A variety of methods, mostly stemming from the SLAM community, integrated these 3D images into complete shape models of the static environment (Gonzalez-Aguirre et al., 2011, Kerl et al., 2013, Endres et al., 2014).

A break-through in the field was the work by Newcombe et al. (2011a), known as *KinectFusion*. In this work, the authors incrementally built a 3D reconstruction of the environment from depth images, represented as a truncated signed distance function (Curless & Levoy, 1996). For each new depth image the authors estimate the pose of the generating depth sensor within the map, and an extension to the map, in a SLAM-like approach.

KinectFusion made the generation of complete geometric 3D models of static scenes and the estimation of the pose of a depth sensor with respect to the model a "solved" problem. One of the reasons for the success of KinectFusion is that it leveraged correctly the structure of the perceptual problem: the synergies generated from the combination of the scene reconstruction and pose estimation processes, the exploitation of the known physics behind the formation of depth images from a known geometry, elegantly leveraged using a signed distance function, and the boost in performance from the recursive initialization of the pose. However, by not exploiting the correlations between interactions and changes in the sensor signals the method restricts itself to static environments. Its "detachment" from interactions limits the applicability of KinectFusion (and other methods that generate geometric 3D models) for robot manipulation tasks.

The resulting 3D models from the aforementioned methods serve as input to a variety of grasping and manipulation planning algorithms (Miller & Allen, 2004, Rusu et al., 2009, Papazov et al., 2012, Nieuwenhuisen et al., 2012, Jentzsch et al., 2015). However, this geometry-based perception cannot extract time-varying signals and therefore does not explicitly consider the robot's interactions. As a result, time-varying aspects of the robot's action must be planned *prior* to execution and therefore without access to up-to-date sensor feedback. The resulting limitations in reactivity necessitates complex planning under uncertainty (Smallwood & Sondik, 1973, Kaelbling et al., 1998, Hsiao et al., 2007). The reliance on static, geometric models also makes it impossible to extract certain object properties, including kinematic articulations and dynamic properties, although these properties are essential for robust and versatile manipulation.

Researchers have attempted to overcome the limitations of static scenes and handle the complexity of a changing sensor signal by building 3D deformable models (Schulman et al., 2013b, Furch & Eisert, 2012, Channoufi et al., 2016). Newcombe et al. (2015) extended KinectFusion with a warp-field that encode the deformation of the reconstructed surface from the nominal pose. These methods provide impressive reconstruction results, but the deformation field is not easily correlated to interactions and it cannot be applied to robot manipulation.

The reduced reactivity, the thereby necessitated complex reasoning, and the limitations on the type of properties that can be perceived—all consequences of the static, non-interactive view of perception—must be viewed as significant obstacles on the path towards perception tailored to manipulation.

Realizing the limitations of static perception, researchers began to consider time-varying sensor signals correlated to actions. The insight that time-varying signals together with knowledge of the actions that caused the changes contain important additional information led to a novel paradigm in computer vision: to *active vision* (Aloimonos et al., 1988, Bajcsy, 1988, Ballard, 1991). The active vision paradigm exploits correlations between changes in sensor's parameters (e.g. pose or focal length) and changes in sensor signal. This enables new

approaches to computer vision problems, such as image segmentation or structure from motion (Aloimonos, 1990, Salganicoff et al., 1992, Aloimonos, 1993, Aloimonos & Fermüller, 1995, Whitehead & Ballard, 1990, Blake & Yuille, 1993, Pahlavan et al., 1993, Chaumette et al., 1996, Hayman, 2000). It also led to the novel challenge of appropriately directing sensing resources to satisfy the requirements of a perceptual task, i.e. "where to look" to perceive effectively (Rizzi et al., 1996, Kragic et al., 2005). The field has been recently reviewed by some of its founders (Bajcsy et al., 2016). However, in spite of the appropriateness of active vision for perception and manipulation, this paradigm has not yet found widespread use in that context.

A reason for the limited dissemination of the active vision paradigm is the type of actions involved and thus, the type of information about the environment it can reveal. Active vision only considers changing the sensor's parameters and exploiting the correlation of these actions to changes in sensor signals. Properties related to the way the environment reacts to interactions that are crucial for manipulation (e.g. kinematic and dynamic properties) are not contained in the signals that active vision is able to generate and perceive.

Researchers in the intersection of artificial perception and robotics proposed to exploit the interactive capabilities of embodied agents to overcome the limitations of passive and active perception. The information revealed through physical interactions relating actions and environmental reactions is crucial for manipulation because it allows the agent to plan and predict the outcome of the manipulation.

The first to realize the potential of physical interaction as part of the perceptual process were Tsikos & Bajcsy (1991) (published even earlier as a technical report (Tsikos & Bajcsy, 1988)). In their approach they generate an initial set of hypotheses about objects on a table using range sensing and build a relational graph were the nodes are object hypotheses and the edges connecting nodes indicate a *on-top* relationship. Through interactions (shaking, pushing, picking) the robot was capable of refining the relational representation, identifying the topmost object at each step, and using this information to plan and execute actions to clear the table.

Later on Fitzpatrick & Metta (2002) (Fitzpatrick, 2003, Fitzpatrick et al., 2003) integrated interactions as part of a perceptual system for object segmentation. Their robot identified its own arm and correctly segmented objects in the visual stream using poking actions to create motion cues that are sufficient for the task. Extending this idea Katz & Brock (2007) used robot interactions to reveal and perceive kinematic properties of articulated objects. They coined the term *interactive perception* to design methods that integrate physical interaction with the environment as part of the perceptual process. Since then, interactions has been leveraged to simplify and solve multiple perceptual tasks including object recognition (Li & Kleeman, 2011, Sinapov et al., 2011), object singulation (Chang et al., 2012), image segmentation (van Hoof et al., 2014, Schiebener et al., 2012, Kenney et al., 2009), and the estimation of kinematic and dynamic properties of articulated objects (Atkeson et al., 1986, Katz et al., 2013a, Endres et al., 2013). A recent survey (Bohg et al., 2017) has summarized the most important existing approaches in interactive perception.

2.1.2 Interactions in Interactive Perception

Obviously, interactions are an intrinsic component of any interactive perception approach. They can play two main roles: as generators of information-rich sensor signals, and as prior to interpret these signals. Which role interactions play depends on how much information about the interaction and its consequences on the environment is available to the perceptual

method (i.e. self-interaction or observation of another agent, forward and other predictive models, . . .).

Initial methods in interactive perception used interactions just as generators of informative signals (Tsikos & Bajcsy, 1991, Fitzpatrick, 2003, Katz & Brock, 2007). These methods do not require to know the exact interaction; they only assume that the interaction will reveal the desired information contained in the sensor signals. Because of their low dependency on detailed information about the interaction, these methods can be applied to perceive from self-interactions and from interactions from other agents. However, these methods fail to perceive unequivocally in cases where the same sensor signals could result from multiple pairs of interaction - environment property, cases that could be easily disambiguated exploiting further information about the interaction as prior.

Trying to overcome these limitations, a second group of interactive perception methods make use of more detailed information about the action and its correlation to changes in the environment to interpret the sensor signal. Zhang & Trinkle (2012) use knowledge of the robot action and tactile sensing to solve a dynamic equation and track the motion of an interacted object. Similarly, Koval et al. (2013) also use tactile sensing and knowledge about the robot interaction to localize an object inside the robot's end-effector. They realized that the object's pose lies on a submanifold within all possible poses defined by the contact configurations. Hausman et al. (2015) heuristically define the possible outcomes of an action in terms of changes of the environment and sensory signals, and use the measured sensor signals to update the robot's internal belief. In general sensor modalities like haptics and tactile sensing that can only register spatially close events greatly benefit from this variant of the interactive perception paradigm because the information contained in the sensor signals extends spatially when interpreted with detailed information of the interaction that caused them (Schneider et al., 2009, Ilonen et al., 2014, Martinez-Hernandez et al., 2017, Michel et al., 2014). While methods exploiting deeper knowledge of the interaction can extract more information more accurately from the same sensor signals, they require complex models (forward models relating actions to changes in the state of the environment, measurement models relating changes in the state of the environment to changes in the sensor signals (Corke, 2017)) that are not available for many tasks.

Some recent methods try to learn these models directly from interactions (Agrawal et al., 2015, 2016). Learning the models from pairs of action-sensor signals avoids having to define them analytically. Moreover, the learned models can even outperform hard-coded models because they replace the intermediate represented states by representations that are more tailored to the specific action-sensor space. However, these methods require vast amounts of training data that is generally costly to obtain for interactions and robot manipulation. Also, the task-tailored representations cannot be easily adapted and shared between domains and tasks, which limits their generalization. A promising recent research line tries to alleviate these problems by imposing soft constraints in the resulting intermediate representations (Byravan & Fox, 2017).

There is a trade-off between the information that can be extracted from the sensor signals using knowledge about the interaction, and the complexity of the required prior knowledge about the correlation between actions and changes in the sensor signals. Simpler methods just assume that the actions generate information-rich signals, but cannot fully extract the information contained in the signals. More elaborate approaches obtain additional information at the cost of complex forward and measurement models. An additional advantage of having these complex models is that they allow to plan for the most informative actions to guide exploration for interactive perception (Krüger et al., 2011, van Hoof et al., 2014, Hausman et al., 2015, Otte et al., 2014, Kulick et al., 2015, Baum et al., 2017, Barragán et al., 2014).

There is an increasing interest in the robotics community for interactive perception methods that can obtain their own interaction models and apply them to interpret sensor signals and to plan informative actions.

2.1.3 Interactions in Human Perception

Traditionally, the dominating idea in the scientific community was to consider perception as a *passive* process[1]: signals acquired by our sensing organs are transmitted to the brain, where they are processed to generate a percept and possibly a response action. At the beginning of the 70s psychologists and cognitive scientists developed a new theory of the perceptual process that departed from this passive model.

In his seminal work the psychologist Gibson (Gibson, 1966) proposed a new view of perception that integrates actions in an *active* process of the agent within its environment. In his theory, perceiving agents are not waiting for information-rich signals to arrive, but they move, explore, interact with their environment to generate and find these signals (Gibson, 1979).

Experiments with human subjects support Gibson's active view of the perceptual process. In one of these experiments, subjects are asked to recognize pebbles of varying forms. The subjects are allowed to have different levels of interaction with the pebbles: no interaction, observing the pebbles rotating but without control on their motion, and full control of the pebble motion (and therefore, the obtained visual signals). While the first group could only recognize 49% of the pebbles, having changing sensor signals increased it to 72%. The group that could actively interact achieved 99% accuracy. The combination of known interactions and corresponding changes in sensor signals contains much richer information to solve the perceptual task.

Philosophers like Alva Noë (Noë, 2006) and cognitive scientists like Varela (Varela et al., 1993), O'Regan (O'Regan, 2011) and Gallagher (Gallagher, 2006) further developed the idea that action is part of perception into what they called the *enactive* perception paradigm. Within this paradigm, a property of the environment is just a link between a set of actions and the corresponding changes in the sensory responses.

For example, we say that a plate presents a circular-form not because its shape projects onto our retina as a circle, but rather because it projects as ellipses of varying eccentricity as we move our eyes around it and the variation pattern linking motion and ellipse eccentricity matches the one we associate to "perceiving a circle". Actions (in this case moving our head around) generates changes in the sensor signals (in this case images) defined the sensory property. Properties of the environment are then subsets in the combined space of actions and changing sensor signals, $A \times S \times t$. Elements of this space contain richer information than static sensor signals or changing signals alone. The changing signals acquire more meaning when combined with the actions that caused them.

The experiments by Held & Hein (1963), while not in humans, support the idea that perception in biological systems is intrinsically linked to actions. In their study, the authors placed two kittens in a carousel: both could visually observe their surroundings but only one kitten was controlling the motion with its walking movements, while the other was passively moved based on the motion of the first kitten (see Figure2.1). As a result, only the kitten that controlled its own motion learned to understand the correlation between walking and changing visual signals that is necessary to navigate. The second kitten, even though it had acquired the same visual signals, was unable to avoid obstacles or to follow a path.

[1]Some authors have argued against this historical narrative and suggest that the passive-view was not the only paradigm when studying perception (Wagner, 2016). However, these alternative paradigms were marginal compared to the predominant passive perception model.

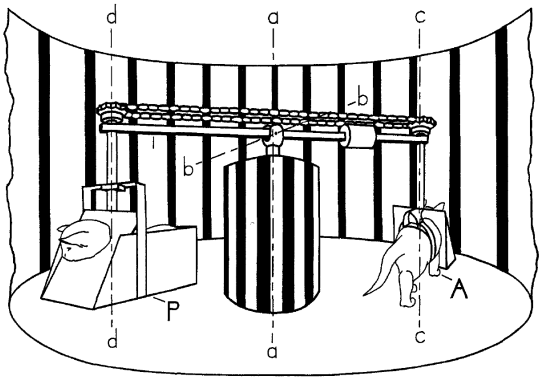


Figure 2.1: Experimental setup of (Held & Hein, 1963); one kitten (A) controls the motion while the other (P) moves passively; both kittens observe the same environment; only the active kitten (A) learns the correlation between actions and changes in the visual field and can use perception to support navigation (© 1963 APA)

Given the compelling indications from psychology and cognitive science about the crucial role of interactions in the perceptual process, our goal in this thesis is to exploit interactions in order to reveal and create information-rich sensor signals for robot perception.

2.2 Leveraging Physical Priors for Perception (OP2)

2.2.1 Physical Priors in Signal Processing and Artificial Perception

Subsumed under the term *physical priors*, we consider two types of regularities that a perceptual solution can exploit: 1) assumptions about the physical properties and structure of the environment and signal (e.g. the *rigid body prior*, the assumption that the environment is composed of rigid bodies), and 2) known models of the physical processes related to the signal generation (e.g. the *kinematics prior*, that motion in the environment is governed by known kinematic equations). Both types of priors have played a crucial role since the early days of artificial perception, when it was still called *signal processing*. Signal processing is the analysis, manipulation and transformation of sensor signals. Processing a signal requires to leverage physical priors about the signal itself and its generation. For example, processing an image to detect intensity discontinuities between areas (Canny detector (Canny, 1986)), compute intensity gradients and spatial-frequency properties (Fast Fourier Transform (Cooley & Tukey, 1965)), or find salient points (Harris & Stephens, 1988) requires to assume a certain image formation procedure (i.e. the *projective geometry prior*) and a fixed image spatial representation.

Artificial perception goes beyond the analysis of the sensor signal itself and interprets the signal as evidence of relevant properties of the world. For this interpretation, artificial perception methods use physical priors, not only about the signal, but also about the environment. The transition from image processing to computer vision is a clear example of this increasing role of physical priors (Rosenfeld & Pfaltz, 1966, Rosenfeld et al., 1976, Barrow & Tenenbaum, 1978).

Computer vision researchers used increasingly complex physical priors and models to develop the family of methods of *shape-from-X* (Ramachandran, 1988, Aloimonos, 1988, Nayar & Nakagawa, 1994). These methods obtain three dimensional information based on detailed models of illuminance and reflection (Phong, 1975, Oren & Nayar, 1994). Advances in geometric and projective physical models derived into multi-view geometry, and generated the first sparse three-dimensional reconstructions based on sets of images (Shashua, 1995, Ullman, 1979, Hartley & Zisserman, 2003). Assumptions about the *continuity* of the object

surfaces (i.e. the *surface continuity prior* in color, in curvature, in depth, ...) are at the core of most image segmentation algorithms Shi & Malik (2000), Kato & Pong (2001), Papon et al. (2013b). Recent artificial perception approaches have imported more complex physical priors from other fields like computer graphics (Seitz & Szeliski, 1999, Benno Heigl, 2000, Sigal & Black, 2006, Bogo et al., 2015), and analytic physics (Zhou et al., 2016, Battaglia et al., 2013, Pauwels & Kragic, 2015, Schenck & Fox, 2017).

As a reaction to this dependency on assumptions and complex physical priors, artificial perception researchers have proposed to extract these regularities directly from sensor data using machine learning techniques (Krizhevsky et al., 2012b, Deng et al., 2009). Unfortunately, this process requires large amounts of data to find the right general patterns in the signals. The most recent and promising trend to decrease this data-hunger is to combine simple physical priors with machine learning techniques (Jonschkowski & Brock, 2015, Schenck & Fox, 2016, Byravan & Fox, 2017). Nevertheless, researchers in artificial perception are still looking for the right combination of physical priors and sensor data.

2.2.2 Physical Priors in Interactive Perception

Interactive perception methods solving the same task usually leverage the same physical priors. In the following, we will take a look on the most important of these priors per interactive perception application. Most interactive segmentation algorithms assume that the environment is composed of rigid bodies (*rigid body prior*). Some of them create an initial segmentation hypothesis leveraging physical priors from computer vision, e.g. the *surface continuity prior* in (Bergström et al., 2011, Beale et al., 2011), object surface convexity (Tsikos & Bajcsy, 1991, Chaudhary et al., 2016), geometric primitives (Schiebener et al., 2012, 2014, Chang et al., 2012), smooth normal orientation, or combinations of the previous (van Hoof et al., 2013, 2014, Katz et al., 2013c), and refine incrementally this initial segmentation through interactions. For the refinement, some of them exploit additional knowledge of the kinematics of rigid bodies, i.e. the *kinematic prior* (Schiebener et al., 2012, 2014). The kinematic prior is also necessary to build kinematic models of articulated objects (Sturm et al., 2009, Katz & Brock, 2008), estimate the pose of an object (Koval et al., 2015, Zhang & Trinkle, 2012), or perceive dynamic properties from interactions (Endres et al., 2013, Atkeson et al., 1986).

Some interactive perception methods that reconstruct the shape of an object or recognize it from images rely on models of the projection of light from the environment onto the camera, i.e. the *projective geometry prior* (Ude et al., 2008, Katz & Brock, 2011b). These models are useful to plan the best next action (Krainin et al., 2011). Other interactive shape reconstruction approaches from sparser data (e.g. tactile information) resort to *continuity priors* to interpolate between data points (Ilonen et al., 2014, Martinez-Hernandez et al., 2017, Michel et al., 2014).

Essentially, interactive perception uses the same physical priors as other artificial perception fields like computer vision. However, in interactive perception the *kinematic prior* has a more prominent role, since one of the most common reasons to integrate interactions into the perceptual process is to generate motion and its association information-rich sensor signals.

2.2.3 Physical Priors in Human Perception

We observe two types of physical priors involved in human perception: physical priors "hard-coded" in the anatomical system, and "soft" priors about physics learned and used to make inference. The first type of physical priors exploit the known underlying process of signal formation. Being co-evolved for millions of years, sensor organs and brain have developed mech-

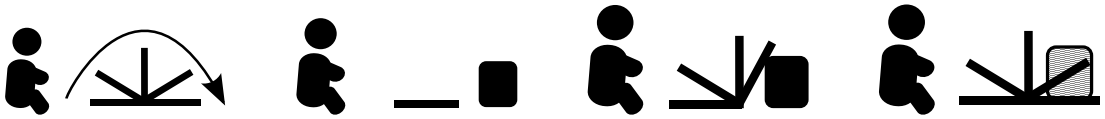


Figure 2.2: Experiments on intuitive physics by Baillargeon et al. (1985); (a) the baby is habituated to the rotation of a screen around a revolute axis; (b) the experimenter places an obstacle in the trajectory and repeat the motion; the baby predicts a collision based on its intuitive physical priors (c), and demonstrates surprise if the collisions does not take place because the experimenter removes the obstacle (without the baby noticing)

anisms that assume a known type of physical process generating the sensor signals (Corke, 2011). The result of extracting information with "hard-coded" priors is similar to the results signal processing: the transformation of the sensor signal into a more favourable representation.

For example, while the number of photoreceptor cells in a human eye is approx. 100 millions, the connections to the brain through the optic nerve contains less than 2 millions (Jonas et al., 1992). There is a first reduction of the raw visual signal of two orders of magnitude performed at the retina. This reduction is based on the geometric distribution of the cells: signals from adjacent cells are aggregated following an innate mechanism. This physiological process is tailored to the anatomy of the eye that dictates the formation of the images in the retina and constitutes an example of physical prior in human perception.

The second type of physical prior exploitation in human perception derives from an innate knowledge of some of laws of physics, called *naïve* or *intuitive physics*. Experiments with infants (Spelke et al., 1995, Hespos & vanMarle, 2012) support the hypothesis that we are born with a certain basic knowledge of the physical processes that govern our environment. The experiments show that infants are surprised when they observe illusions breaking physical concepts like *solidity*, *occlusions*, *object permanence*, and *containment*. For example, Baillargeon et al. (1985) exposed three and half months old babies to a screen rotating around a revolute joint (see Figure 2.2). They place an obstacle in the trajectory of the blade and actuate the mechanism, removing in some trials the obstacle without the babies noticing. In these trials, babies were surprise because of the absence of a collision (starring longer to the event). The experiment indicates a intuitive concept of solidity and rigid body physics. These concepts clearly relate to the physical priors we have seen for interactive perception, e.g. *rigid body* or *surface continuity*, and encode the same problem regularities.

Despite the initial studies highlighting human misjudgements and wrong predictions of physical phenomena (McCloskey et al., 1980), the majority and more recent experiments support the idea that humans can foreseen accurately the outcome of physical processes (Proffitt et al., 1990, Gilden & Proffitt, 1989, Nusseck et al., 2007) especially in the context of motion. This prior knowledge is encoded in our interactive perception systems as *kinematic priors*.

Physical priors are thus a crucial element of the human perceptual process to encode regularities of the problem that allow to interpret and predict the sensor signals. The approach presented in this thesis aims to similarly exploit simple physical priors about rigid bodies and image formation to interpret the sensor signals from interactions.

2.3 Leveraging Temporal Consistency as Prior for Perception (OP3)

2.3.1 From Snapshots to Batches to Continuous Stream Interpretation

In some subfields of artificial perception like audio analysis or speech recognition, the relevant information is contained in a time series of signals. In contrast, in other subfields like computer vision the sensor signal at a single time step (an image) contains rich information by itself. Initial methods of computer vision focussed on the extraction of patterns of information from single images (Rosenfeld & Pfaltz, 1966, Rosenfeld et al., 1976).

Based on the increasing understanding of the structure of a single signal and the improvement in computing and sensing technologies, computer vision researchers turned to the analysis of temporal sequences of signals, i.e. signal streams or video sequences. Adding the temporal dimension researchers could focus on new perceptual problems like tracking (Kass et al., 1988, Lucas & Kanade, 1981) or optical flow (Nagel & Enkelmann, 1986). Many initial methods exploit the information of the entire signal sequence to interpret each individual snapshot (Faugeras, 1992, Poelman & Kanade, 1997). These methods are batch processing: they assumed that the entire signal sequence is available at processing time. However, this processing approach is not suited for online applications, like perception for robot manipulation.

Applications with online constraints require solutions that interpret sensor signals as they arrive. Researchers found out that this is possible leveraging further the underlying temporal structure of the problem, e.g. by turning perceptual problems into recursive estimation problems. Using recursion the solution at the previous step acts as constraint to restrict the space of possible solutions at the next step, making the search of the most likely state easier (Thrun et al., 2005).

The first and most successful examples of this idea appeared in robot localization (Smith et al., 1990, Leonard & Durrant-Whyte, 1991, Fox et al., 1999) and simultaneous localization and mapping (SLAM) (Se et al., 2002, Thrun et al., 1998, 2005). Following this success, many other robot perceptual problems have been posed as recursive state estimation like object tracking (Weng et al., 2006, Choi & Christensen, 2013), manipulator state tracking (Garcia Cifuentes et al., 2017, Hebert et al., 2012), and semantic segmentation (Miksik et al., 2013).

Recursive state estimation became a crucial technique to overcome the challenges of perception for robotics (specially the online requirements, CH2) and remains today one of the most useful algorithmic approaches for robots to perceive the continuously changing state of a dynamic system from noisy observations. In fact, many artificial perception researchers consider Bayesian inference the most crucial mechanism in perception (Knill & Richards, 1996), and recursive state estimation as the application of Bayesian inference along the temporal dimension.

2.3.2 Temporal Priors in Interactive Perception

Initial methods in interactive perception did not extract information from the continuous sensor stream. They applied instead signal differencing between snapshots of the sensor stream before and after the interaction and perceived the changes in the environment caused by the robot (Tsikos & Bajcsy, 1991). This algorithmic methodology disregards all the information revealed in intermediate steps of the interaction that can be crucial to understand the outcome.

Later approaches moved towards an analysis of entire sensor sequences of interactions in a batch manner (Pillai et al., 2015, Katz & Brock, 2011b, 2008, Schiebener et al., 2012, 2014). These methods compare each signal to the first one and evaluate the changes during the entire

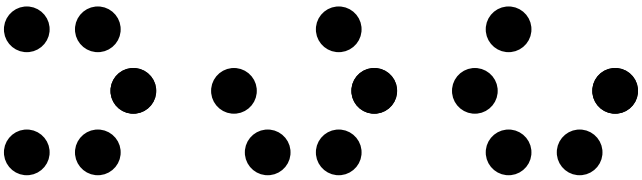


Figure 2.3: Example of the beta effect; the missing element in the sequence of images gives the illusion of motion, despite the large the distance between elements

manipulation. These approaches can thus extract more information from the sensor stream, but they cannot still be applied to extract task-relevant information online and are therefore not suited to address the challenges of perception for robotics.

As in other fields of artificial perception, some methods turned interactive perception problems into recursive state estimation problems. Hausman et al. (2013) proposed an online tracking framework to segment objects in a table top scenario using robot interactions. Combining color and depth visual salient points, their method can be applied to textured and texture-less objects. Other authors have proposed recursive state estimation solutions to perceive and track the pose of a pushed object from tactile measurements (Koval et al., 2013, 2015, Zhang & Trinkle, 2012). However, recursive state estimation is still not the prevalent algorithmic solution in interactive perception due to the complexity of the required models of the interaction (forward, measurement) and the additional challenge of only using the past and current measurements to perceive. However, only such an online approach can be applied to create interactive perception methods that can support and steer ongoing robot manipulation.

2.3.3 Temporal Priors in Human Perception

The human perceptual system is prone to apply temporal priors (consistency, smoothness) to interpret the continuously arriving sensor signals. This becomes evident from illusions like the *phi* and the *beta phenomenon* Wertheimer (1912). In this illusions (see Figure 2.3), a sequence of static images is shown, triggering immediately the sensation of motion into human subjects. These effects are called *long-range apparent motion*. A related effect, the *short-range apparent motion* effect, is responsible for creating the illusion of motion between consecutive images with small differences, e.g. in cinematographic movies Grossberg & Rudd (1992). Long and short-range apparent motion effects evidence the predisposition of the human perceptual system to apply temporal consistency priors for the interpretation of sensor streams. This predisposition shows how the human perceptual system has adapted to exploit the temporal consistency of the physical processes.

Numerous studies in cognitive science shed a light on the consequences of this adaptation: the human perceptual system performs better in tasks that present temporal consistency and correlation (Kristjánsson et al., 2010, Maljkovic & Nakayama, 1994, Maljkovic & Martini, 2005, Niemi & Näätänen, 1981). Given that many perceptual tasks present strong temporal structure, it seems reasonable to exploit it. This is one of the goals of the perceptual approach we present in this thesis: to increase the perceptual capabilities of robots leveraging the temporal structure of the task.

2.4 Leveraging Information from Other Processes as Prior for Perception (OP4)

2.4.1 From Independent to Collaborative Perceptual Subtasks

The easier way to build perceptual systems of arbitrary complexity is to structure them into modules. Modularity is a way of decomposing complexity by breaking down a problem into smaller subproblems that can be solved and tested individually. Researchers can focus on specific parts of the perceptual problem (e.g. image segmentation, object recognition or pose estimation) and implement their solutions assuming they are independent processes (Marr, 1982).

While this *divide and conquer* approach has enabled the study of each subproblem in isolation and the development of successful solutions, it commonly neglects the interdependencies between subproblems. However, exploiting these interdependencies could be necessary to solve them. To ensure maximum performance of a entire perceptual system, and to avoid making wrong commitments or addressing subproblems that are unnecessarily difficult, all components of the system should be chosen to maximally exploit potential synergies between components (Katz & Brock, 2011a).

Important advances in artificial perception research were achieved by overcoming existing modularizations and exploiting the interdependencies between predefined perceptual subtasks. The best known example is SLAM (Se et al., 2002, Thrun et al., 1998, 2005), a problem that couples localization and mapping and that has been addressed with recursive solutions that tackle effectively the joint problem. Other methods have shown the benefits of integrating segmentation, reconstruction and pose estimation to improve each subtask by leveraging their interdependencies (Stückler & Behnke, 2012, Ma & Sibley, 2014).

Perceptual systems can also exploit interdependencies between subtasks within hierarchical architectures. The usual procedure in a hierarchy is to propagate information bottom-up. However, a way to further exploit interdependencies between subtasks is to propagate information from higher levels into lower levels. This information helps to interpret lower level signals. Rao & Ballard (1999) present an elegant hierarchical architecture for image recognition based on bottom-up and top-down communication between interconnected neural networks. In their approach each network implement a recursive estimator that generates predictions and corrections for increasingly complex parts of the image. From the bottom-up communication, the higher levels recognize more complex structures composing information from smaller parts. But interestingly, from the top-down communication, the information from the higher levels helps the lower levels in the recognition of small patches. We deem the exploitation of interdependencies between perceptual subtasks at different abstraction levels an opportunity that we aim to exploits in perception for robot mechanical manipulation of DoF.

2.4.2 Interdependencies Between Subtasks as Prior in Interactive Perception

Initial methods in interactive perception focussed on a single perceptual task and neglected the synergies of a more holistic approach (Tsikos & Bajcsy, 1991, Fitzpatrick, 2003). However, subsequent methods achieved important advances by integrating multiple perceptual subtasks. For example, many interactive segmentation methods combine clustering and motion estimation to find the rigid bodies that move in from interactions (Schiebener et al., 2012, 2014, Chang et al., 2012). Katz et al. (2014) combined the problems of segmentation, 3D pose estimation and kinematic analysis to perceive the kinematic models of articulated objects.

These methods found a right factorization of the original task that allows to inject simpler priors at each level (e.g. simple physical priors), simplifies the analysis and evaluation of each subcomponent and facilitates possible extensions.

In the aforementioned methods the subproblems are combined following a sequential bottom-up pipeline structure: the outcome of one subtask is the input for the next one. This pipeline architecture to build perceptual systems was already proposed by Marr (1982). Information in the opposite direction (from higher levels of abstraction to lower levels, top-down) or between subprocesses pertaining to other information extraction processes (e.g. between subprocesses, or from different modalities) is thus not exploited. This one-way sequential processing structure cannot fully leverage the interdependencies between subproblems as priors to interpreting sensor signals.

In addition to the limitations of the one-way information flow mechanism, most interactive perception algorithms proposed so far are based on a single sensor modality, mainly vision (Bergström et al., 2011, Schiebener et al., 2012, Katz et al., 2014). Integrating information from other modalities, however, is crucial to improve robustness, versatility and accuracy of perceptual systems. Some few methods have explored the use of several modalities, but in a one-modality-per-task fashion. This is the case of Hausman et al. (2015). The authors apply vision to generate hypotheses of kinematic models, and proprioception (force-torque signals) to reject wrong hypotheses from robot interactions. The output of these processes is combined, but the processes do not help each other. This is a multi-modal version of the serial pipeline processing, and thus, it neglects the interdependencies between subtasks and top-down information flow. There is a clear opportunity to improve interactive perception with a tighter integration of perceptual subprocesses from different sensor modalities. This has been explored by Krainin et al. (2011) combining tracking and reconstruction, or more recently by Byravan & Fox (2017) combining segmentation and pose estimation.

A recent and elegant example of the benefits of a tighter integration is the work of Garcia Cifuentes et al. (2017). In their approach based on coupled recursive processes, the authors fuse proprioceptive signals (robot's arm noisy encoder values) and visual information (RGB-D stream) to perceive, in an online manner, the configuration of the robot arm. Perceptual subprocesses defined in one sensor modality pass information to the other subprocesses, which is used as prior to help each other and achieve accurate tracking performance. While this method is not applied to perceive the environment from interactions, we think (and propose in this thesis) that such an interconnection between online processes can be applied to exploit interdependencies between subtasks in interactive perception.

2.4.3 Interdependencies Between Subtasks as Prior in Human Perception

There is evidence from neuroscience, cognitive science and psychology supporting the hypothesis that the human brain processes sensor signals in parallel subprocessing units, and that these units share information at multiple areas to help each other. For example, Livingstone et al. (1988) discovered that the human perceptual system contains two parallel functional and anatomical subprocesses that share information at various levels. The first subprocess, called the *magno system*, focusses on the perception of motion and three-dimensional scene arrangement. The second subsystem, the *parvo system*, perceives color, shape and other surface properties.

What are possible reasons for their structural separation? As the authors say:

> Segregating the processing of different types of information into separate pathways might facilitate the interactions between cells carrying the same type of

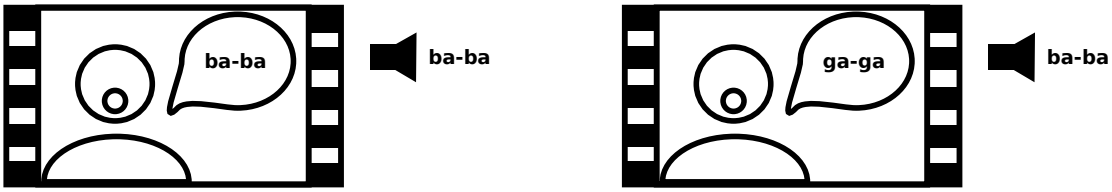


Figure 2.4: Illustration of the experiments on the McGurk Effect (McGurk & MacDonald, 1976); left: sound and video are congruent (no effect); right: the altered video leads to many subjects hearing a different syllable even though the audio signal is the same (McGurk effect)

> information. It might also allow each system to develop functions particularly suited to its specialization.(Livingstone et al., 1988, p. 748)

The segregation thus enables the adaptation of the processes to the type of information that need to be extracted, while the interconnection between processes permits the injection of additional information into the other subprocesses.

When it comes to the integration of sensor signals obtained from different sensing organs, the human perceptual system seamlessly generates a coherent unified experience. Ernst & Banks (2002) demonstrated that for tactile and visual integration, the information is merged in a statistical optimal fashion. The results of this integration can be modelled and well predicted using a Kalman-filter-like approach. Information from each information extraction process is merged into a combined percept based on its uncertainty. Uncertainty will also play a crucial role in the integration of information from different concurrent subprocesses of the perceptual approach proposed in this thesis.

Psychologist have also found evidence supporting that the integration of modalities does not follow a simple bottom-up approach –from sensor data to a higher level percept– but that there is information transfer between interpretation processes in different modalities. An example of this intercommunication is the McGurk effect. The McGurk effect (McGurk & MacDonald, 1976) is a perceptual illusion where a subject watches a video of a person pronouncing syllables. The subject is convinced to hear different utterances, such as da-da or ga-ga, when in fact the sounds are identical (see Figure 2.4). The subjects misjudges the sound because the video in fact shows the person saying different syllables but the sound has been altered to play the identical syllable. The illusion occurs because the visual cue produced by the facial motions influences the perception of the sound. As a result, identical sounds are perceived as being different. This illusion demonstrates that visual cues affect hearing. If the integration of information in the human perceptual system were purely centralized and bottom-up, the subject would notice the contradiction in the visual and audio signal. This cross-modal interpretation of multiple modalities is necessary for humans to robustly perceptive speech (Rosenblum et al., 2007). In this thesis, we will propose mechanisms to exploit the interdependencies between concurrent processes of the perceptual system (within the same modality or across different modalities) to help each other.

2.5 Conclusion

In this chapter, we have seen that the four opportunities for perception are crucial in human perception. This supports the hypothesis that leveraging these problem regularities could enable robust and versatile perception in unstructured environments.

We have also seen in this chapter that the four opportunities play a crucial role in many successful artificial perception and interactive perception methods. In Table 2.1 we summarize the interactive perception methods we reviewed in this chapter and classify them by application and the way they leverage the four opportunities for perception. The most important applications are defined in Table 2.2. We will conclude this chapter deriving conclusions about the contribution of the four opportunities in the IP methods we have reviewed, and linking them to the approach we presented in Chapter 1 and the perceptual systems we propose in following chapters.

Existing IP methods can use actions to 1) create information-rich sensor signals (CS), and 2) to interpret these signals (IS). Theoretically, IP methods of the IS group extract more information from the interaction because they use it to disambiguate between unclear events, restricting effectively the space of possibilities. However, IS methods require complex interaction models that are difficult to formulate. The approach we present in this thesis can be used to instantiate perceptual systems of both groups. If equipped with the necessary models, the recursive processes that compose our approach use information about the interaction to predict state and measurement changes, as we will see in Chapter 6. Nevertheless, the recursive processes can also interpret changes in sensor signals without explicit information about the generating action, only from the signals the interaction creates, the information perceived so far, and physics models, as we will see in Chapter 4 and 5.

The reviewed IP methods exploit different physical priors to extract information. Physical priors allow the IP methods to understand sensor signals as evidence of the underlying physical structure, to complete missing information and to reject noise in the sensor signals. Most IP methods presented so far assume that the environment is composed of rigid bodies (RB). Some of the methods use kinematic models to interpret the sensor signals (K). IP Methods tackling computer vision tasks (e.g. image segmentation) employ also projective geometry models (PG) and assume some degree of continuity (in color, depth, curvature, ...) on the surface of the environment (SC). Interestingly, as a reaction to the increasingly complex physical priors and models, some IP researchers are trying to reduce this dependency on hard-coded physical models by learning statistical regularities from the sensor data. However, even these approaches try to leverage physical priors to reduce the necessary amount of data. In this thesis, we will encode simple physical priors (projective geometry, rigid body assumption, kinematics of rigid bodies) in our perceptual solutions to help in the interpretation of the sensor signals.

While almost all IP methods assume some temporal structure in the problem (e.g. that after an interaction the environment will change) only a few IP methods try to exploit the mutual information between consecutive sensor signals to interpret the changes in the environment as they occur (*online processing*, OP). Most existing IP methods use either signal differencing (D) or batch processing (BP). These methods cannot be applied to continuously interpret changes in sensor signals and use the information to monitor and steer ongoing interactions, one of the challenges in perception for robot manipulation of DoF (CH2). We consider this challenge an important obstacle for robot perception. The IP approach and the perceptual systems that we present in this thesis are online, delivering information to support ongoing manipulation.

And finally, very few IP methods have explore how to maximally exploit the interdependencies between perceptual subtasks. Most of the reviewed methods are focussed on a single task (ST) or they compose multiple subtasks in serial manner (*serial pipeline*, SP). Very few IP approaches fully exploit the multi-modal sensor signal available to most robots nowadays. We believe that this design decisions are a consequence of an excessive modularization in robotics and artificial perception, and that neglecting the interdependencies between perceptual sub-

tasks renders them more difficult. In this thesis, we propose an approach that intercommunicates solutions to perceptual subtasks so that information from one helps to solve others, within one modality or across modalities.

Note: This chapter reviewed IP methods (and other artificial and human perception studies) in the context of the four problem regularities we propose to leverage for robot perception. In the following chapters we will present additional related work sections that discuss previous IP methods in the context of the perceptual applications addressed in the chapters: the perception of kinematics, geometry, and dynamics of articulated objects, from a single or multiple modalities. We believe this way of dividing and presenting prior work in the field helps to understand better the contribution of this thesis.

Taxonomy of IP Solutions by Application and Exploitation of Opportunities					
Applications	*IP Methods*	*OP1*	*OP2*	*OP3*	*OP4*
IS	Fitzpatrick & Metta (2002), Fitzpatrick (2003), Fitzpatrick et al. (2003)	CS	RB	D	ST*
	Kenney et al. (2009)	CS	RB	BP	ST
	Bergström et al. (2011)	CS	RB,K	D	SP
	Chaudhary et al. (2016)	CS	RB, K	D	SP
	Beale et al. (2011)	CS	RB, PG	BP	SP
	van Hoof et al. (2014, 2013, 2012)	CS	RB, SC	D	SP
IS+OR	Schiebener et al. (2012, 2014)	CS	RB, K	BP	SP
OR	Schneider et al. (2009)	CS	PG	BP	ST
	Li & Kleeman (2011)	CS	RB, K	BP	ST
	Sinapov et al. (2011)	IS	–	D	ST*
OS	Tsikos & Bajcsy (1991, 1988)	CS*	RB	D	ST
	Chang et al. (2012)	CS	RB	D	SP
	Katz et al. (2013c)	CS	RB	BP	SP
SR	Krainin et al. (2011)	IS*	RB, K, PG	OP	IT*
	Michel et al. (2014)	IS	RB, K, SC	BP	ST
	Ilonen et al. (2014)	IS	RB, K	OP	ST*
	Martinez-Hernandez et al. (2017)	IS	RB, K	OP	ST
PE	Zhang & Trinkle (2012)	IS	RB, K	OP	SP
	Hausman et al. (2013)	CS	RB, K, PG	OP	SP
	Koval et al. (2013, 2015)	IS	RB, K	OP	SP

Table 2.1: Taxonomy of the interactive perception (IP) methods discussed in this section, their application and how they leverage the four opportunities for perception for robotics presented in Chapter 1.2)
A glossary of applications and their initials is depicted in Table 2.2
OP1: CS= "to create signals", IS="to interpret signals", *="with action selection"
OP2: RB="rigid body", K="kinematics", PG="projective geometry", SC="surface continuity"
OP3: D="differencing", B="batch processing", OP="online perception"
OP4: ST ="single task", SP="serial pipeline", IT="interconnected tasks", *="multimodal"

Taxonomy of IP Solutions (Continued)					
KM	Katz & Brock (2007), Katz et al. (2013a), Katz & Brock (2011b, 2008), Katz et al. (2014)	CS	RB, PG, K	BP	SP
	Sturm et al. (2009)	CS	RB, K	BP	ST
	Pillai et al. (2015)	CS	RB, K	BP	SP
	Otte et al. (2014)	CS*	RB, K	BP	SP
	Hausman et al. (2015)	IS*	RB, K	OP	SP*
	Barragán et al. (2014)	IS*	RB, K	D	SP
	Baum et al. (2017)	CS*	RB, K	BP	SP*
DM	Atkeson et al. (1986)	IS	RB, K	BP	SP
	Endres et al. (2013)	IS	RB, K	BP	ST*
IML	Agrawal et al. (2015, 2016)	IS	PG	BP	ST*
	Byravan & Fox (2017)	IS	RB, K	BP	IT*

Table 2.1: (Continued) Taxonomy of the interactive perception (IP) methods discussed in this section, their application and how they leverage the four opportunities for perception for robotics presented in Chapter 1.2

Applications of Interactive Perception		
Image Segmentation	**IS**	Divide images into connected regions corresponding to the same object
Object Recognition	**OR**	Estimate the identity of an object within a set of known possibilities
Object Singulation	**OS**	Separate individual objects from an unordered group/structure
Kinematic Model Estimation	**KM**	Build a model of an articulated object defining the motion constraints between its movable parts
Dynamic Model Estimation	**DM**	Build a model of the dynamic properties of an object
Object Pose Estimation	**PE**	Estimate the pose in 6D space of a known object
Shape Reconstruction	**SR**	Generate a model of the geometric surface of an object
Interaction Model Learning	**IML**	Build a model that predicts the outcome of robot interactions with the environment (as changes of environment's state and/or sensor signals)

Table 2.2: Glossary of applications of interactive perception methods

3

Background

In this thesis, we propose an approach to overcome the challenges of perception for robot manipulation (CH1-CH3, Section 1.1) exploiting favourable structural properties of interactive perception problems (OP1-OP4, Section 1.2). This chapter will review the theories and algorithms that our approach is based on. These theoretical foundations are not part of the contribution of the thesis although they are crucial to understand its contributions, and thus, make the text self-complete.

One of the opportunities we aim to leverage is the temporal structure of the problem (OP3). Signals in the sensor stream evidence the temporal structure in the changing state of the environment. Information about the state that has been extracted from previous signals can be used as prior to interpret current sensor data. Exploiting algorithmically the temporal structure in this manner is called *recursive estimation*. Recursive estimation will allow us to build online systems to perceive environmental changes associated to interactions (CH1 and CH2). Supported by the online information, the robot will more likely accomplish its task, the manipulation of mechanical DoF in the environment. Therefore, we will begin this chapter by summarizing Bayesian filters, the most important family of recursive estimation algorithms in robotics that we will apply in this thesis.

As we explained in Chapter 1, in this thesis we focus on a specific type of robot manipulation: the mechanical manipulation of kinematic DoF in the environment, and the special case of articulated mechanisms. We presented a general approach for interactive perception (see Section 1.3) that we will use in the next chapters to instantiate perceptual systems to acquire information about articulated objects (kinematic, geometric, dynamic models). These perceptual systems extract task-relevant information leveraging physical priors (OP2), and known or learned correlations between interactions and changes in sensor signals (OP1). The task-related priors encode knowledge about **spatial descriptions and transformations**, **kinematics of rigid bodies**, and **articulated objects**. In this chapter, we will revise these fields to know how to exploit this regularities for perception.

Finally, this chapter serves also to define the mathematical notation used all along the text, summarized in Table 3.1.

3.1 Recursive Estimation

Estimation is the process of producing a reasonable statement about a latent (i.e. non-directly observable) variable based on input data. When the input data comes from a sensor and the latent variable is some task-relevant property of the world, the estimation problem is a perceptual task.

As we argued in Chapter 1, in the specific type of robot tasks we are focussed on, the mechanical manipulation of kinematic DoF of the environment, the goal is to *change* the kinematic state of the world, e.g. the pose of the objects and parts of articulated mechanisms. Perception to support mechanical manipulation tasks needs to estimate and monitor these changes continuously. The estimation of a dynamically changing latent (not directly perceivable) state is called state estimation (Thrun et al., 2005, Bar-Shalom et al., 2001, Barfoot, 2017).

The physical processes involved in mechanical manipulation of DoF present a strong temporal structure: the current state of the process (e.g. the pose of an object or the configuration of a joint) is strongly related to the previous states[1]. This temporal structure can be leveraged to help in the estimation of the current state (OP3in 1.2). The main idea is simple but powerful: use what has been perceived before as prior for the interpretation of current sensor signals, assuming a certain temporal evolution. Applying this idea, the estimation of the current state is guided by the previously estimated information, improving convergence (Young, 2012). Recursive state estimation is a family of algorithms that leverage the temporal structure in a perceptual problem to solve it.

To correctly exploit the information perceived before, recursive state estimation uses a model that correlates previous and current states. This model is called *forward model*, or also *dynamic or transition model* because describes to the underlying dynamic process and the transitions between its states.

So far we have not assumed any interactive capabilities for the perceiving agent: the agent could be a *passive observer* that has no influence on the state to estimate. However, a robot is an *active agent*: it performs actions that can change the state world. Recursive state estimation also provides mechanisms to leverage action for perception by encoding the relationship between interactions and changes in the state in the *interaction forward model*.

All the aforementioned properties make recursive estimation algorithms well suited to overcome the challenges of perception for robot manipulation.

In recursive state estimation the state can be represented either deterministically, or probabilistically. With a deterministic representation, the estimated state is a single element of the space of possible states. On the other hand, with a probabilistic representation the state is a random variable and what we estimate is its probability distribution over the space of possible states. By maintaining multiple hypothesis (i.e. a distribution over a space of possibilities) with a probabilistic representation, a recursive solution increases its robustness because it does not commit prematurely to a wrong estimate. Probabilistic representations also account for the uncertain nature of the unstructured environment and the noisy behavior of the sensor signals, and provide a mathematical framework to express the degree of certainty on the perceived information. The robot can use this degree of certainty to act not only based on the current estimate, but also on the uncertainty about it, e.g. to guide exploration (Settles, 2012) or to operate safely (Liu & Tomizuka, 2015). This makes probabilistic representations best suited for recursive state estimators for robot perception in unstructured environments.

[1] Kinematics provide mathematical models of the temporal evolution of these processes. We will cover them in the next sections of this chapter.

The downside is that representing and operating on distributions over entire state spaces is computationally expensive and in some cases intractable. Using a probabilistic representation we commit ourselves to use calculus and algebra of probability theory within our recursive state estimation algorithm. We will see that, to render recursive estimation problems with probabilistic representations solvable, we will have to make some assumptions (e.g. Gaussian distributions, linear or linearizable dynamics, ...) about the nature and properties of the problem and the environment that will restrict their applicability. The perceptual approach propose in this thesis aims to alleviate these limitations by factorizing perceptual problems into subproblems that are simpler to linearize.

In probability theory the way to exploit prior information (which in our recursive solution are the previously estimated states) is to apply the Bayes rule[2] so that we can integrate observations to obtain a posterior. Recursive state estimation algorithms using probabilistic representations receive the name of Bayesian filters (BF) and have achieved some of the most successful online perceptual algorithms in robotics so far.

3.1.1 Bayesian Recursive State Estimation: The Bayes Filter

In Bayesian recursive state estimation we assume that the state of the dynamical system we aim to estimate belongs to X, the space of all possible states. We denote as $\boldsymbol{x}_t \in X$ the random variable[3] that represents the state at time t, and $p(\boldsymbol{x}_t = x_t) = p(x_t)$ the function that defines the probability of the variable to take each concrete value, x_t, also called probability distribution function. In Bayesian terms, we consider $p(x_t)$ the prior probability, prior to the integration of knowledge from the sensor measurements and robot actions. We also assume that the measurement, $\boldsymbol{z}$, is a random variable defined over the space Z of possible measurements. The measurement acquired at time t is denoted by $z_t \in Z$. Finally, let's assume that the space of possible actions is U and an action executed at time t is denoted by $u_t \in U$.

The goal in state estimation is to determine the probability distribution over the space of possible current states conditioned on the data acquired so far (measurements, actions and, possibly, an estimate of the initial state $\boldsymbol{x}_0$)[4]: $p(x_t|z_{t:1}, u_{t:1}, x_0)$. This probability distribution is called *belief state* at time t, or posterior because is the result of the integration of the prior and the measurements. We will compute the belief in a recursive form using the Bayes rule: as the result of the integration of the previous belief(s) with the latest measurement and action. We will see in the following how to obtain a recursive form of the posterior.

To obtain a recursive solution exploiting previous estimates as priors, we first apply the

[2] In general, $p(a|b)$ indicates a conditional probability: the probability of the event a conditioned on the event b. The Bayes rule can be used to express this conditional probability as an equation of the probability of b conditioned on a:

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)} \tag{3.1}$$

$p(a)$ is called *prior probability distribution* because it represents the knowledge we have about the distribution over possible states of a before integrating knowledge about b. $p(a|b)$ is called *posterior probability distribution* because is the result of the integration of the prior with knowledge about b. When b is fixed (e.g. a given measurement), $p(b|a)$ is the likelihood function of a.

[3] Strictly speaking, X is the sample space of the random variable $\boldsymbol{x}$, and outcomes of $\boldsymbol{x}$ ($\boldsymbol{x} = x$) are elements of X. We abuse the notation by saying that $\boldsymbol{x} \in X$.

[4] Note that some algorithms do not compute the full probability distribution but focus on a specific statistical component of it, e.g. its maximum (Maximum a Posteriori estimators, MAP).

Bayes rule to the belief at time t:

$$\begin{aligned} p(x_t|z_{t:1}, u_{t:1}, \boldsymbol{x}_0) &= \frac{p(z_t|x_t, z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)p(x_t|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)}{p(z_t|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)} \\ &= \eta\ p(z_t|x_t, z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)p(x_t|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0) \end{aligned} \tag{3.2}$$

where η is a renormalization constant (independent of x_t). The role of this constant is to guarantee that the resulting function is a probability distribution by normalizing its integral over the entire state space to the unity, $\int p(x)dx = 1$.

We can simplify the first term in the following manner:

$$p(z_t|x_t, z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0) = p(z_t|x_t) \tag{3.3}$$

which indicates that the probability distribution over current measurements is independent of previous measurements, robot actions and estimated states. $p(z_t|x_t)$ is our probabilistic measurement model: the probability of acquiring a measurement z at time t assumed the state $\boldsymbol{x}_t$.

In the previous simplification we made an important **assumption**: the state is complete. A state is complete if all the necessary information to predict the future evolution of the state (and the measurements) is contained in the current state. In other words, any additional knowledge about previous states, measurements or actions does not improve our predictions. A temporally evolving physical process with a complete state is called a **Markov chain**. Many dynamical systems of interest for robot manipulation (and for the applications of this thesis, the manipulation of articulated objects) are naturally modelled as Markov chains.

In our path towards a recursive solution that exploits the previously perceived information we introduce x_{t-1} as variable within the second term of our Bayes rule equation:

$$p(x_t|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0) = \int p(x_t|x_{t-1}, z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)p(x_{t-1}|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)dx_{t-1} \tag{3.4}$$

Applying again the Markov assumption we can simplify the first term of the integral:

$$p(x_t|x_{t-1}, z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0) = p(x_t|x_{t-1}, u_t), \tag{3.5}$$

which means that the probability distribution over current state depends only on the previous state and the last robot action. $p(x_t|x_{t-1}, u_t)$ is a probabilistic forward model, also called *transition model* because it indicates how the state transitions from one step to the next one given the robot's action. This model defines the probability distribution over states at time t given the previous state and the action at time t, u_t.

Altogether we can write the probability of our estimated state given all acquired measurements and robot actions in the following recursive form:

$$p(x_t|z_{t:1}, u_{t:1}, \boldsymbol{x}_0) = \eta\ p(z_t|x_t) \int p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)dx_{t-1} \tag{3.6}$$

where $p(x_{t-1}|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0)$ is the belief distribution over the previous state.

The computation of the current belief, $p(x_t|z_{t:1}, u_{t:1}, \boldsymbol{x}_0)$, given the previous belief, $p(x_{t-1}|z_{t-1:1}, u_{t-1:1}, \boldsymbol{x}_0)$, can be considered two steps:

1. Prediction Step (Equation 3.4): Integrates information about the action to predict the state. This step increases the uncertainty about the current belief.

2. Correction Step (Equation 3.6): Integrates information about the measurement to correct the predicted state. This step decreases the uncertainty about the current belief.

Following the recursion we observe that we need to assume a prior distribution over the space of possible initial states, $p(x_0)$. If we do not have any information about the initial state, this prior can be assumed to be uniformly distributed over all possible states. The performance and convergence of the Bayesian filter improves significantly if we leverage additional information to define a sharper distribution over the initial state.

The equations presented before describe the *Bayes Filter*. Even after the application of the Markov assumption, the equations of the Bayes filter cannot be solved for arbitrary probability distributions. There are two reasons for this:

1. The probability density functions are defined over the entire space of possible values of the random variables (states, measurements,...). In other words, we have to define the value $p(x)$, $\forall x \in X$ and for all the random variables involved. Defining and operating with these distributions in an explicit form is usually intractable. For discrete and finite spaces we can define and update the probability of each state. For continuous or large discrete spaces we are advocated to 1) operate with the moments or 2) with a finite number of samples of the random variables.

2. The integral term (Equation 3.6) is very costly to compute unless we can solve it analytically. If an analytic solution cannot be computed we can evaluate the integral approximately, e.g. using Monte Carlo integration.

A way to address both aforementioned problems is to assume that all random variables involved in the Bayes filter are Gaussian distributed. In this case, the integral can be solved analytically and we can represent completely the distributions by their first two order moments, the mean and the standard deviation (or the covariance). This way of representing the distribution of a random variable is called parametric representation. A Bayes filter where we constrain the random variables to be Gaussian distributed is called a Gaussian filter. The best known solution (optimal if the Gaussian assumption holds) to the Gaussian filter is the *Kalman Filter*, which we summarize in the following.

A limitation of the Kalman filter is that, to assure that the random variables are still Gaussian distributed after passing through the dynamical system, both the measurement and the forward models have to be linear. However, many physical processes of interest in perception for robotics are non-linear. One way to extend the Kalman filter machinery to non-linear systems is to linearize the measurement and/or forward model around the current estimate. This approach receives the name of *Extended Kalman Filter*, which we will summarize after the Kalman Filter.

As explained before, in both the Kalman and the Extended Kalman Filter we will use a *parametric* representation: we represent probability distributions by their first and second order moments (mean and variance). These two moments represent completely (without loss of information) a Gaussian distribution, but any other distribution is not fully represented by just the distribution's mean and covariance. Instead of representing probability distributions parametrically, we can represent them in a *non-parametric* way: with a finite number of samples. The samples can pass directly through a non-linear measurement or forward model without any linearization, and can be recombined to approximate the posterior of the Bayes filter. The best known approach using this procedure to approximate the solution of the Bayes filter is called the *Particle Filter* and we will review its most relevant components after the parametric filters.

3.1.2 The Kalman Filter

The Kalman Filter is an optimal solution to a Bayes Filter when measurement and forward models are linear and all involved probability functions are Gaussian distributed. A dynamic process with linear forward and measurement model can be written in the form:

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + Bu_t + \boldsymbol{w}_t \tag{3.7}$$

$$\boldsymbol{z}_t = C\boldsymbol{x}_t + \boldsymbol{v}_t \tag{3.8}$$

where the first equation defines a linear forward model, $p(x_t|x_{t-1}, u_t)$, and the second equation defines a linear measurement model, $p(z_t|x_t)$.

The forward and the measurement models encode prior knowledge about the problem and the domain, i.e. the temporal evolution of the underlying dynamical system, the correlation between actions and changes in the environment, and the relationship between the state of the environment and the acquired sensor measurements. In most recursive estimation processes of this thesis we will use physical priors to define these models. Later in this thesis, we will also present a method to learn some of these models from robot experiences (see Chapter 6.8). In this section, we will assume the models to be given.

We assume that $\boldsymbol{x}, \boldsymbol{z}$ are multidimensional random variables such that $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{z} \in \mathbb{R}^m$. u is a multidimensional input action vector, $u \in \mathbb{R}^k$. For the models of Equations 3.7 and 3.8 to be linear, A, B, C have to be matrices[5], $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{k\times n}$. $\boldsymbol{w}_t \in \mathbb{R}^n$ is a random variable that represents the additive system noise. $\boldsymbol{v}_t \in \mathbb{R}^m$ is a second random variable that represents the additive measurement noise. Both system and measurement noise are zero-mean Gaussian distributed:

$$w_t \sim \mathcal{N}(0, Q_t) \tag{3.9}$$

$$v_t \sim \mathcal{N}(0, R_t) \tag{3.10}$$

with Q_t and R_t the system and measurement noise covariances, respectively[6].

As we stated before, in the Kalman Filter we assume that all distributions involved in Equations 3.7 and 3.8 are Gaussian distributed. Then, the prior belief at time t is defined by:

$$p(x_{t-1}|z_{t-1:1}, u_{t-1:1}, \boldsymbol{x}_0) = \mathcal{N}(x_{t-1}, P_{t-1}) \tag{3.11}$$

We can predict the next state by passing this prior through our previously defined forward model:

$$p(x_t|z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0) = \mathcal{N}(\hat{x}_t, \hat{P}_t) \tag{3.12}$$

$$\hat{x}_t = Ax_{t-1} + Bu_t \tag{3.13}$$

$$\hat{P}_t = AP_{t-1}A^T + Q_t \tag{3.14}$$

Based on the predicted state, we can predict the expected measurement:

$$p(z_t|x_t) = \mathcal{N}(\hat{z}_t, \hat{S}_t) \tag{3.15}$$

$$\hat{z}_t = C\hat{x}_t \tag{3.16}$$

$$\hat{S}_t = C\hat{P}_tC^T + R_t \tag{3.17}$$

[5] We have assumed that the forward and measurement models are constant over time, i.e. $A \neq A(t)$, $B \neq B(t)$ and $C \neq C(t)$, to simplify notation. The analysis and the solution we present for time-constant models also applies to time-dependent models.

[6] In this thesis, we usually represent covariances with the symbol Σ. However, the covariances in the Kalman filter have traditionally received the symbols we use in our explanation

The joint distribution of both state and measurement is then:

$$p(x_t, z_t | z_{t-1:1}, u_{t:0}, \boldsymbol{x}_0) = \mathcal{N}\left(\begin{bmatrix} \hat{x}_t \\ \hat{z}_t \end{bmatrix} , \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix} \right) \tag{3.18}$$

$$\Sigma_{xx} = \hat{P}_t \tag{3.19}$$

$$\Sigma_{zz} = \hat{S}_t \tag{3.20}$$

$$\Sigma_{zx} = \Sigma_{xz}^T = E[(\boldsymbol{x} - \hat{x})(\boldsymbol{z} - \hat{z})^T] = C\hat{P}_t \tag{3.21}$$

By applying the product rule[7] to the joint distribution we obtain a closed form solution for the desired posterior:

$$p(x_t | z_{t:1}, u_{t:1}, \boldsymbol{x}_0) = \mathcal{N}(x_t, P_t) \tag{3.22}$$

$$x_t = \hat{x}_t + K_t(z_t - C\hat{x}_t) \tag{3.23}$$

$$P_t = (I - K_t C)\hat{P}_t \tag{3.24}$$

$$K_t = \hat{P}_t C^T (C\hat{P}_t C^T + R_t)^{-1} \tag{3.25}$$

K_t is called the *Kalman Gain*. The Kalman Gain balances the estimation of the belief between the predicted state and the correction from the measurement. The balance is based on the relative uncertainty of the prediction and the measurement. This behavior of the Kalman Filter gives us a principled way to correct the estimation towards the state predicted from previously perceived information when the measurement is noisy, or towards the measurement-based estimate when the prediction is uncertain.

3.1.3 The Extended Kalman Filter

An important limitation of the Kalman filter is that both the forward and the measurement models of the system are assumed to be linear. In many physical processes of interest in robot perception, this assumption does not hold. A solution when the system and/or the measurement models are non-linear is to linearize them. This linearization of the models is the key element of the Extended Kalman filter (EKF). Once the EKF has linearized the models, the random variables will continue being Gaussian distributed after going through the equations of the dynamical system.

While the Kalman filter is an optimal estimator if the assumption (linear system and Gaussian distributed variables) hold, the EKF is in general non-optimal because of the approximation due to the linearization. However, the EKF provides a good approximation to the true state distribution for many perceptual problems in robotics.

In the EKF the dynamical system present the form:

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, u_t) + \boldsymbol{w}_t \tag{3.26}$$

$$\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{v}_t \tag{3.27}$$

where f and g are possibly non-linear but linearizable functions that represent the system and measurement models.

[7] Product rule: $p(a, b) = p(a|b)p(b)$

The EKF linearizes the possibly non-linear models using a first order Taylor expansion[8] around the expected state:

$$f(\boldsymbol{x}_{t-1}, u_t) \approx f(x_{t-1}, u_t) + f'(x_{t-1}, u_t)(\boldsymbol{x}_{t-1} - x_{t-1}) \tag{3.28}$$

$$f'(x_{t-1}, u_t) = \left.\frac{\partial f(\boldsymbol{x}, u_t)}{\partial \boldsymbol{x}}\right|_{x_{t-1}} = F_t \tag{3.29}$$

$$h(\boldsymbol{x}_t) \approx h(x_t) + h'(x_t)(\boldsymbol{x}_t - x_t) \tag{3.30}$$

$$h'(x_t) = \left.\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x_t} = H_t \tag{3.31}$$

We use the notation $\left.\frac{\partial f(x)}{\partial x}\right|_{\bar{x}}$ to indicate that we compute the derivative of function f with respect to its variable x, and substitute in the result x by a specific value $\bar{x}$.

F_t and H_t are Jacobian matrices, $F_t \in \mathbb{R}^{n\times n}$ and $H_t \in \mathbb{R}^{k\times n}$, that correlate (infinitesimally) small changes in the previous and current state, and (infinitesimally) small changes in the state to changes in the expected measurement, respectively.

Based on the linearization, the EKF predicts the distribution of the next state as:

$$p(x_t | z_{t-1:1}, u_{t:1}, \boldsymbol{x}_0) = \mathcal{N}(\hat{x}_t, \hat{P}_t) \tag{3.32}$$

$$\hat{x}_t = f(x_{t-1}, u_t) \tag{3.33}$$

$$\hat{P}_t = F_t P_{t-1} F_t^T + Q_t \tag{3.34}$$

The EKF predicts the distribution of next measurement as:

$$p(z_t | x_t) = \mathcal{N}(\hat{z}_t, \hat{S}_t) \tag{3.35}$$

$$\hat{z}_t = h(\hat{z}_t) \tag{3.36}$$

$$\hat{R}_t = H_t \hat{P}_t H_t^T \tag{3.37}$$

$$\hat{S}_t = \hat{R}_t + R_t \tag{3.38}$$

where $\hat{R}_t$ is the covariance matrix of the measurement noise (see Equations 3.9 and 3.10), and $\hat{S}_t$ is the covariance of the innovation.

Finally, based on the previous definitions and the linearizations, the EKF computes the posterior distribution as:

$$p(x_t | z_{t:1}, u_{t:1}, \boldsymbol{x}_0) = \mathcal{N}(x_t, P_t) \tag{3.39}$$

$$x_t = \hat{x}_t + K_t(z_t - h(\hat{x}_t)) \tag{3.40}$$

$$P_t = (I - K_t H_t)\hat{P}_t \tag{3.41}$$

$$K_t = \hat{P}_t H_t^T (H_t \hat{P}_t H_t^T + R_t)^{-1} \tag{3.42}$$

As commented before, the EKF is an approximation to the true posterior distribution when the models (measurement or forward) are non-linear. The quality of the approximation depends on two factors: 1) the spread of the prior distribution, and 2) the degree of non-linearity in the model around the linearization point. Therefore, if the distributions involved in the Bayes Filter are widely spread over the state space, or if the models are highly non-linear (i.e. $f(x) \not\approx f(x + \Delta x)$ for small Δx) the true posterior distribution diverges largely from

[8] Taylor Expansion: $f(x + \delta x) = \sum_{n=0}^{\infty} \frac{f'^n(x)}{n!} \delta x^n$

the approximation obtained by the EKF. The initialization of the EKF ($\boldsymbol{x}_0$) plays therefore a crucial role in its performance, since it determines the initial spread of the distribution.

A way to overcome the aforementioned limitations of the EKF, and also the limitation of both EKF and Kalman filters that assume Gaussian distributed variables, is to represent the distributions with a finite number of samples. We can pass the samples through the non-linear system and measurement models and reconstruct the distributions afterwards. Solutions based on samples are called *non-parametric* Bayesian filters to differentiate them from the Kalman and EKF filters (*parametric* filters) that operate on a parametric representation (mean and covariance) of the distributions. The best known non-parametric Bayes filter is called the Particle Filter.

3.1.4 The Particle Filter

A Particle Filter represents the estimated state (the posterior distribution given the measurements and robot actions) with a set of N samples called particles:

$$p(x_t|z_{t:1}, u_{t:1}, \boldsymbol{x}_0) \rightarrow X_t = \{x_t^0, \ldots, x_t^N\} \tag{3.43}$$

where the number of particles N can be constant or vary over time. The main idea of the Particle Filter is to link the probability of having a sample at the state $\boldsymbol{x}_t$ to the posterior probability of this state $p(x_t|z_{t:1}, u_{t:1}, \boldsymbol{x}_0)$, so that areas represented by many particles are areas of the space with high probability and vice-versa.

Assuming a recursive solution, the posterior of the previous step will be also represented by a set of particles:

$$p(x_{t-1}|z_{t-1:1}, u_{t-1:1}, \boldsymbol{x}_0) \rightarrow X_{t-1} = \{x_{t-1}^0, \ldots, x_{t-1}^N\} \tag{3.44}$$

The first step of the Particle Filter is to propagate the samples through the forward model to generate samples of the predicted state distribution. The forward model here could be non-linear and with a different noisy model than the additive Gaussian model of the (Extended) Kalman Filter. Therefore, we represent it in its more general form $p(x_t|x_{t-1}, u_t)$. The exact way to sample this distribution will depend on the forward model. For now we will assume we can sample from this distribution and generate a new set of samples:

$$x_{t-1}^n \rightarrow p(x_t|x_{t-1}^n, u_t, \boldsymbol{x}_0) \rightarrow \hat{x}_t^n \tag{3.45}$$

$$\hat{X}_t = \{\hat{x}_t^0, \ldots, \hat{x}_t^N\} \tag{3.46}$$

We now use the measurement model $p(z_t|x_t)$ to estimate the probability of the acquired measurement given each particle. This value is called *importance factor* and the set of particles with their importance factor approximates the posterior distribution $p(x_t|z_{t:1}, u_{t:1}, \boldsymbol{x}_0)$.

A crucial step in the Particle Filter is to change the representation of the posterior from a set of particles with importance factor to a new set of particles resampled based on their importance. Areas of the state space with higher probability will have a higher density of particles than areas with lower probability. This process is called *Importance Resampling*. The benefit of Importance Resampling is to cover with more samples, and therefore more accurately, the areas of the state space that are more important for the estimation, i.e. the areas where the posterior is higher. Given that we do not make use of Important Resampling in this thesis, we won't cover it in this review. For a detailed explanation of Important Resampling we recommend Thrun et al. (2005).

The Particle Filter can successfully approximate the posterior even if it presents multiple maxima (multi-modal distributions), or for complex forward and measurement models. The limitation is that the non-parametric representation of the distributions is more accurate the more samples the filter uses, but the more samples it uses, the higher the computation. The number of samples to cover the state space with an equivalent degree of detail increases exponentially with the dimensionality of the state. This problem is known as the *curse of dimensionality* in robotics, and limits the applicability of the Particle Filter to state spaces with low dimensionality.

3.2 Spatial Descriptions and Kinematics of Rigid Bodies

In the previous section we have seen algorithmic techniques to exploit the temporal structure of the perceptual problem. To apply these techniques (the family of Bayesian filters) we need prior knowledge about: 1) the underlying dynamical system that governs the temporal evolution of the state of the environment, 2) the influence of robot actions in the temporal evolution of the state, and 3) the way the state reflects into sensor signals. We will encode this prior knowledge in the recursive estimation solutions in the form of measurement and forward models.

In this section, we will review physical models that can be used as priors to solve interactive perceptual tasks in unstructured environments. These priors define mathematically the kinematic state of a rigid body –its pose– and the way the state changes over time –its motion. Such models are crucial to perceive the changes in the environment caused by robot mechanical manipulation of DoF.

3.2.1 Spatial Descriptions

The pose of a rigid body B in 3D space with respect to a reference observer O possesses six degrees of freedom (DoF): three degrees for the position and three for the orientation. While the position can be well represented with an *explicit parametrization*[9], explicit parametrizations for the orientation (e.g. Euler or roll-pitch-yaw angles) present two limitations:

- They do not represent correctly the periodic nature of orientations
- They suffer from singularities (e.g. small changes in the orientation lead to large changes in the parameters and vice-versa)

Implicit representations for the orientation (and by extension, of the pose) of a rigid body address these problems.

In robotics the most common implicit representation for a rigid body pose is a *homogeneous transformation matrix*. Homogeneous matrices are an embedding of the six dimensional space of rigid body poses into a 16 dimensional parameter space with the form of 4×4 matrices of real elements. Therefore, for this embedding the redundant dimensions of the homogeneous transformation have to be constrained and will form a manifold. In the following we will describe the general form of a homogeneous transformation and its manifold constraints.

[9] Given a space of dimensionality N a parametrization of this space is called **explicit** if it requires N parameters to cover the entire space. The parametrization is called **implicit** if it requires $M > N$ parameters and additional constraints (Lynch & Park, 2017).

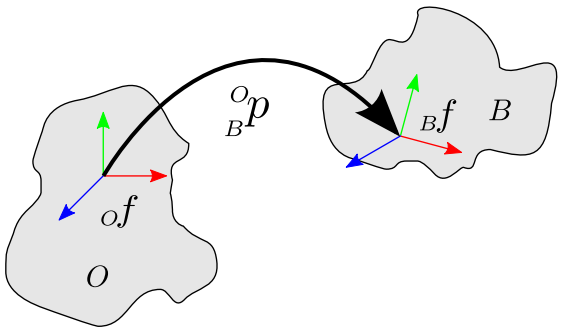


Figure 3.1: 6D pose ${}^{O}_{B}p$ of a frame ${}_{B}f$ attached to the body B with respect to the frame ${}_{O}f$ attached to the body O; this pose can be represented by a homogeneous transformation (${}^{O}_{B}T$) or any other suited representation

A homogeneous transformation representing the pose ${}^{O}_{B}p$ of a frame ${}_{B}f$ attached to a body B with respect to a frame ${}_{O}f$ attached to an observer O and defined in the coordinate system of ${}_{O}f$, as depicted in Figure 3.1, presents the form[10]:

$$ {}^{O}_{B}T = \begin{pmatrix} {}^{O}_{B}R & {}^{O}_{B}d \\ 0_{1\times 3} & 1 \end{pmatrix} \tag{3.47} $$

where ${}^{O}_{B}d \in \mathbb{R}^3$ is the position of the origin of coordinates of ${}_{B}f$ with respect to the origin of coordinates of ${}_{O}f$ and ${}^{O}_{B}R$ is a 3×3 matrix of real elements defining the orientation of the axes of ${}_{B}f$ with respect to the frame ${}_{O}f$, both expressed in the ${}_{O}f$ coordinate system[11].

The previous definition:

- constrains 4 of the 16 parameters due to the fixed last row definition
- assigns 3 of the 12 remaining parameters to represent the 3 DoF of the position of the rigid body
- dedicates the remaining 9 parameters to represent the 3 DoF of the orientation

A homogeneous transformation matrix is thus composed of an explicit parametrization of the position and an implicit parametrization of the orientation that requires to define additional constraints.

To embed the three DoF space of rigid body orientations into the 9 dimensional space of 3×3 matrices of real elements, we impose on R the following properties:

$$ \{R \in \mathbb{R}^{3\times 3} | RR^T = R^T R = I, |R| = +1\} \tag{3.48} $$

This definition constrains 6 of the 9 parameters and keeps three parameters to represent the three DoF of the orientation.

[10] This definition could seem redundant because usually some parts of it are assumed implicitly, e.g. the homogeneous transformation matrix to be defined in the coordinate system of the frame that is used as geometric reference. We will decrease gradually the verbosity of our definitions to avoid the excessive clutter. We refer the reader to De Laet et al. (2013) for a complete analysis of the constraints necessary to fully define geometric primitives and the most common implicit assumptions.

[11] We use the convention for sub-indices and super-indices as defined by Craig (2005). Sub-indices and super-indices are on the left side of the variable. The super-index indicates the reference frame in the observer body. For geometric relationships between two bodies, the sub-index indicates the frame on the second body. For geometric elements (e.g. points, vectors, frames), the sub-index indicates the body they are attached to.

The set of matrices R that fulfil the previous constraints together with the binary operation of matrix multiplication is called $SO(3)$, the Special Orthogonal group in dimension three[12]. Alternatively, we will use the symbol $\oplus$ to refer to the matrix multiplication. In general, an orthogonal group of dimension N in an Euclidean space is a group of linear transformations that preserve distance between transformed elements. The Special Orthogonal group is the subgroup that includes the identity transformation, indicated by the last constrain $|R| = +1$ (the first constraint could be also fulfilled with matrices R such that $|R| = -1$). The Special Orthogonal group corresponds to the group of all rotations about the origin, with composition (matrix multiplication) as group operation.

The result of these constraints in the rotation matrix extend to homogeneous transformations and reduce the 16 parameters of the matrix to an embedding of the 6 DoF of a rigid body pose. The set of all homogeneous transformations ${}^{O}_{B}T$ (4×4 matrices of real numbers with the aforementioned constraints on the rotational part) together with the binary operation of matrix multiplication form also a special group, the Special Euclidean group, $SE(3)$. $SE(3)$ corresponds to the group of all possible poses of a rigid body in 3D space. We will use alternatively the symbol $\oplus$ to refer to the matrix multiplication, and extend it later to refer to the composition of poses, independently of their representation.

To summarize, we defined the Special Orthogonal and Special Euclidean groups in 3D space as:

$$SO(3) = \{R \in \mathbb{R}^{3\times3} | RR^T = R^T R = I, |R| = +1\} \tag{3.49}$$

$$SE(3) = \left\{T \in \mathbb{R}^{4\times4}, T = \begin{pmatrix} R & d \\ 0_{1\times3} & 1 \end{pmatrix} | R \in SO(3), d \in \mathbb{R}^3\right\} \tag{3.50}$$

Interpretations of a homogeneous transformation matrix So far we have considered that a homogeneous matrix represents the pose of a body with respect to another. However, there are two other interpretations (or uses) of a homogeneous matrix: as an operator to transform the spatial description of geometric elements like points, vectors and other poses from one reference frame to another, and as the result of the motion of a rigid body and the operator to apply the motion to the aforementioned geometric elements. Summarizing, the uses of a homogeneous transformation matrix are:

- A representation of the pose of a rigid body in 3D space with respect to another
- An operator to change the reference frame of a geometric element (e.g. a point, a vector, a frame) in 3D space
- An operator to apply a rigid body motion to a geometric element in 3D space

In this thesis, we will alternate these three usages depending on the task at hand. We will now see how to apply homogeneous transformation matrices for the two additional usages listed above.

[12] Given a set of elements $G = g_1, g_2, \ldots$ and a binary operation between elements of the set f, G is a group under the operation f if:

1. $\forall g_1, g_2 \in G, f(g_1, g_2) = g_3 \in G$ (Closure)
2. $\exists e \in G, f(e, g) = f(g, e) = g \ \forall g \in G$ (Identity element)
3. $\forall g \in G \ \exists g^{-1} \in G, f(g, g^{-1}) = f(g^{-1}, g) = e$ (Inverse)
4. $f(a, f(b, c)) = f(f(a, b), c) \ \forall a, b, c \in G$ (Associative law)

The operation f is usually called *dot product*, $\cdot$, or in the case of $SO(3)$ and $SE(3)$, $\oplus$.

We assume we define a point q in 3D space by its location with respect to the origin of a frame ${}_Af$ attached to a body A and expressed also in the coordinates of frame ${}_Af$. The homogeneous transformation ${}^B_AT$ that defines the pose of a frame ${}_Af$ with respect to ${}_Bf$ can be used to express the coordinates of the same point with respect to the reference frame ${}_Bf$ attached to a body B (in the coordinate frame ${}_Bf$):

$$ {}^Bq = {}^B_AT\,{}^Aq \tag{3.51} $$

In order to apply the homogeneous transformation as a matrix-vector product, we have to express the coordinates of point q in the reference frame ${}_Af$ in the so-called homogeneous coordinates:

$$ {}^Aq = ({}^Aq_x, {}^Aq_y, {}^Aq_z, 1)^T \tag{3.52} $$

We will apply the same equation 3.51 when ${}^B_AT$ represents the change in the pose of a rigid body between two time steps from the pose defined by the frame B to the pose defined by the frame A. Equation 3.51 gives us the change in the kinematic state (motion) of a point q rigidly attached to the rigid body.

To transform the coordinates of a free vector v from a reference frame A to a reference frame B, we will use only the rotational part of the homogeneous transformation matrix:

$$ {}^Bv = {}^B_AR\,{}^Av \tag{3.53} $$

We will apply the same equation 3.53 when ${}^B_AT$ represents the change in the pose of a rigid body from the frame B to the frame A and we aim to compute the motion of a vector v rigidly attached to the rigid body.

Given the pose of a frame ${}_Pf$ attached to a rigid body P with respect to a reference frame ${}_Af$ attached to another body A as a homogeneous transformation ${}^A_PT$, we can express this pose with respect to another reference frame B by composing their homogeneous transformations:

$$ {}^B_PT = {}^B_AT\,{}^A_PT \tag{3.54} $$

We will apply the same equation when ${}^B_AT$ represents the change in the pose of the rigid body[13].

Finally, given the pose of two rigid bodies A and B with respect to a shared reference frame f_C by their homogeneous transformations ${}^C_AT$ and ${}^C_BT$ respectively, we can express the pose of body A with respect to body B (their relative pose) as:

$$ {}^B_AT = {}^C_BT^{-1}\,{}^C_AT \tag{3.55} $$

where we have made use of the inverse of an element of $SE(3)$ that is defined by:

$$ {}^B_AT^{-1} = {}^A_BT = \begin{pmatrix} {}^B_AR^T & -{}^B_AR^T\,{}^Bd \\ 0_{1\times3} & 1 \end{pmatrix} \tag{3.56} $$

So far, we have seen the mathematical equations to represent the pose of a rigid body as a homogeneous transformation and defined the Special Euclidean group $SE(3)$ as the group of all possible rigid body poses. We have also defined how to transform poses, points and vectors between different frames and how to apply a known displacement. These equations will be used in this thesis as physical priors to interpret sensor data in recursive state estimation processes (Chapter 4).

[13]Here we assumed that ${}^B_AT$ is expressed with respect to the reference frame ${}_Pf$ (see Footnote 10)

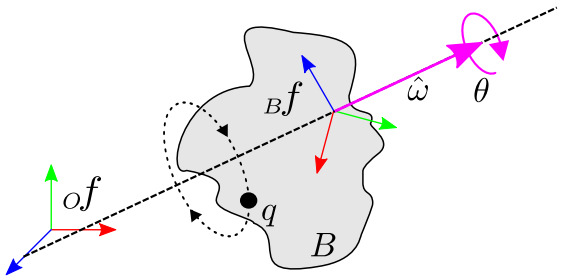


Figure 3.2: Pure rotation of a body B with respect to a body O around an axis defined by the unitary vector $\hat{\omega}$ an amount of rotation θ; the trajectory of a point q on body B is also shown; ${}_B f$ and ${}_O f$ are coordinate frames attached to the respective bodies

However, in this thesis we are interested in the changes over time of the pose of rigid bodies and other geometric elements because these changes are the result and often the purpose of robot manipulation. The robot needs to perceive motion patterns to understand the outcome of its own interactions. We will now revise notions, principles and representations for the motion of rigid bodies in 3D space.

3.2.2 Kinematics of Rigid Bodies

The temporal evolution of the pose of a rigid body in 3D space is defined by its *velocity*, which receives the name of *twist*. A rigid body twist can be used to predict poses, and is thus an important element of a recursive solution. While the pose of a rigid body is represented by an element of the spatial euclidean group $SE(3)$, twists do not belong to this group but to its associated Lie Algebra $se(3)$. We will see that, in order to integrate velocities over time to predict future poses or the trajectory of geometric elements, we need to define a mapping between $se(3)$ and $SE(3)$. This mapping receive the name of *matrix exponential*. Using this mapping, and the inverse, the matrix logarithm, we could also represent poses in exponential coordinates.

In our path towards a definition of a rigid body twists and the equations involved in the temporal evolution of the pose of a rigid body, we will begin by defining a simpler concept, the *angular velocity*, and the equations that describe the change of orientation of a rigid body over time. The concepts developed for angular velocities will extend to 6D velocities and twists.

We assume that a 3D point q is rigidly attached to a moving body B, and that the body rotates with respect to an observer frame O. The trajectory of the body B is defined by its orientation over time ${}^O_B R(t)$. The position of the point q with respect to O is given by the equation:

$$ {}^O q(t) = {}^O_B R(t) \, {}^B q \tag{3.57} $$

where the position of p with respect to B, ${}^B q$, does not depend on time because it is rigidly attached to it.

The time derivative of the previous equation gives the instantaneous velocity of the point:

$$ {}^O \dot{q} = {}^O_B \dot{R} \, {}^B q = {}^O_B \dot{R}({}^O_B R^{-1}(t) \, {}^O q) \tag{3.58} $$

Alternatively, we can represent the motion of the body B between time t and $t + \Delta_t$ as a rotation of θ around an axis defined by the unitary vector ${}^O\hat{\omega}$ (see Figure 3.2). [14] $\theta \, \Delta_t$ is

[14] We have dropped the sub-index B since in its geometric meaning ${}^O\hat{\omega}$ is just a vector defined with respect to the frame ${}_O f$. Alternatively, we can also write ${}^{O,O}_{B}\hat{\omega}$ to indicate that we are describing the velocity of B with respect to O expressed in the coordinate frame of O, but this verbose notation becomes quickly too cumbersome.

the amount of rotation per time unit. When Δ_t approaches zero, $\theta\ \Delta_t$ becomes the rate of rotation, $\dot{\theta}$, and ${}^{O}\hat{\omega}$ becomes the instantaneous axis of rotation. We can write both together as:

$$ {}^{O}\omega = {}^{O}\hat{\omega}\dot{\theta} \tag{3.59}$$

which we call *angular velocity* defined with respect to the observer frame ${}_{O}f$ and in the coordinate frame of ${}_{O}f$.

From this definition of the angular velocity, the instantaneous velocity of the point ${}^{O}q$ with respect to the body O is defined by:

$$ {}^{O}\dot{q} = {}^{O}\omega \times {}^{B}q \tag{3.60}$$

We can write the cross product as a matrix product by defining a special matrix ${}^{O}\omega^{\times}$, the *skew-symmetric* matrix representation of ${}^{O}\omega = ({}^{O}\omega_x, {}^{O}\omega_y, {}^{O}\omega_z)^T$:

$$ {}^{O}\dot{q} = {}^{O}\omega^{\times}\ {}^{B}q \tag{3.61}$$

$$ {}^{O}\omega^{\times} = \begin{pmatrix} 0 & -{}^{O}\omega_z & {}^{O}\omega_y \\ {}^{O}\omega_z & 0 & -{}^{O}\omega_x \\ -{}^{O}\omega_y & {}^{O}\omega_x & 0 \end{pmatrix} \tag{3.62}$$

From Equations 3.58 and 3.60 we see that:

$$ {}^{O}\omega^{\times} = {}^{O}_{B}\dot{R}\ {}^{O}_{B}R^{-1} \rightarrow {}^{O}_{B}\dot{R} = {}^{O}\omega^{\times}\ {}^{O}_{B}R \tag{3.63}$$

which defines the relationship between (the skew-symmetric representation of) the angular velocity with respect to the observer frame, the rotation matrix and the rate of change of the rotation matrix.

We can define the coordinates of the angular velocity with respect to the body frame ${}_{B}f$ (from Equation 3.53: ${}^{B}\omega = {}^{O}_{B}R^{-1}\ {}^{O}\omega$), the rate of change of the rotation matrix becomes:

$$ {}^{B}\omega^{\times} = {}^{O}_{B}R^{-1}\ {}^{O}_{B}\dot{R} \rightarrow {}^{O}_{B}\dot{R} = {}^{O}_{B}R\ {}^{B}\omega^{\times} \tag{3.64}$$

which defines the relationship between the rotation matrix, the rate of change of the rotation matrix, and the (skew-symmetric representation of the) angular velocity of the body B with respect to the observer O in the coordinate frame of ${}_{B}f$, .

Both ${}^{B}\omega^{\times}$ and ${}^{O}\omega^{\times}$ are elements of $so(3)$, a special set containing all skew-symmetric matrices called the *Lie Algebra* of the *Lie Group* $SO(3)$. The elements of the Lie Algebra $so(3)$ are all the possible $\dot{R}$ when $R = I$.

The coordinates of ω are also known as the **exponential coordinates** of the rotation. It is interesting to understand the origin of this name and the mapping between exponential coordinates and rotation matrices by taking a look at the Equations 3.61 and Equation 3.62. If we assume that the body B rotates with a constant velocity given by ω (rotation around the axis defined by $\hat{\omega}$ of θ per time unit), the trajectory of the point q is defined by the following first order differential equation:

$$ {}^{O}\dot{q} = {}^{O}\omega^{\times}\ {}^{O}q \tag{3.65}$$

A first order differential equation of the form $\dot{x} = Ax$ and with initial condition x_0 has a unique solution given by the *exponential* function $x(t) = exp\,(At)\,x(0)$. Analogously, for the case of the 3D trajectory of a point the solution is:

$$ {}^{O}q(t) = exp\left({}^{O}\omega^{\times}t\right)\ {}^{O}q(0) = exp\left({}^{O}\hat{\omega}^{\times}\theta t\right)\ {}^{O}q(0) \tag{3.66}$$

where ${}^{O}q(0) = {}^{O}_{B}R(0)\,{}^{B}q$ is the initial location of the point q in the observer reference frame ${}_{O}f$, and we have used that $\omega = \hat{\omega}\theta$ with $\hat{\omega}$ a unitary vector in the direction of the axis of rotation and θ the amount of rotation per time unit.

Because of the special form of the elements of the Lie Algebra (skew-symmetric matrices), the matrix exponential in the previous equation has a closed form solution:

$$exp\left({}^{O}\hat{\omega}^{\times}\theta t\right) = I + sin(\theta t)\,{}^{O}\hat{\omega}^{\times} + (1 - cos(\theta t))({}^{O}\hat{\omega}^{\times})^2 \tag{3.67}$$

which is known as the *Rodrigues' formula*.

The matrix exponential of an element of $so(3)$ is an element of $SO(3)$, a rotation matrix. Thus, the matrix exponential relates elements of the Lie Algebra to elements of the Lie Group

$$exp : \omega^{\times} \in so(3) \rightarrow R \in SO(3) \tag{3.68}$$

We will use the Rodrigues' formula to integrate angular velocities over time and obtain the equivalent rotation matrices[15].

We can define an inverse operation to the matrix exponential, the *matrix logarithm*, that computes the element of the Lie Algebra $\omega^{\times}$ associated to rotation matrix R:

$$\theta = cos^{-1}(0.5(tr(R - I))) \tag{3.69}$$

$$\hat{\omega}^{\times} = \frac{1}{2sin\theta}(R - R^T) \tag{3.70}$$

$$log\,(R) = \hat{\omega}^{\times}\theta = \omega^{\times} \tag{3.71}$$

$$log : R \in SO(3) \rightarrow \omega^{\times} \in so(3) \tag{3.72}$$

where $tr()$ is the trace of a (square) matrix, which is the sum of the elements of its main diagonal[16]. In other words, the matrix logarithm finds the angular velocity that would result in the given rotation matrix if we would integrate it one time unit.

The matrix exponential is not a bijective mapping between the Lie Algebra $so(3)$ and the Lie Group $SE(3)$: an infinite number of elements of the Lie Algebra are mapped to each element of the Lie Group. This is a consequence of the periodic nature of the rotations: $exp\,((\hat{\omega}\theta)^{\times}) = exp\,((\hat{\omega}(\theta + 2\pi)^{\times})$. Therefore, while the matrix exponential provides a unique solution for each element of the Lie Algebra, the matrix logarithm has infinite solutions for each element of the Lie Group with the same $\hat{\omega}$ and $\theta \pm 2k\pi$ for k any natural number of complete turns. It is common to restrict the solution to the interval $\theta \in [0, \pi]$ and maintain externally a count of the turns of a trajectory. We will use this technique in Chapter 4 to correctly estimate the amount of rotation around a revolute axis.

We can extend these definitions from rotations to the case of 6D rigid body motions. We first compute the result of the Equation 3.64 for the case of homogeneous transformations. We define a rigid body velocity, also known as twist, as:

$${}^{O}_{B}T^{-1}\,{}^{O}_{B}\dot{T} = \begin{pmatrix} R^T & -R^T d \\ 0_{1\times 3} & 1 \end{pmatrix}\begin{pmatrix} \dot{R} & \dot{d} \\ 0_{1\times 3} & 0 \end{pmatrix} = \begin{pmatrix} R^T\dot{R} & R^T d \\ 0_{1\times 3} & 0 \end{pmatrix} = \begin{pmatrix} {}^{B}\omega^{\times} & {}^{B}v \\ 0_{1\times 3} & 0 \end{pmatrix} \tag{3.73}$$

where ${}^{B}\omega^{\times}$ is the angular velocity of the body B and ${}^{B}v$ is the linear velocity of the origin of the frame ${}_{B}f$, both expressed in body frame coordinates.

[15] The Rodrigues' formula generates also the rotation matrix associated to a rotation defined in axis-angle representation (amount of rotation θ around an axis $\hat{\omega}$)

[16] There are two singular points of these equations for the cases a) $R = I$, and b) $tr(R) = -1$. The first case indicates that there is no rotation, $\theta = 0$, and $\hat{\omega}^{\times}$ is undefined. The second case is a rotation of π, $\theta = \pi$, and there are three valid solutions for $\hat{\omega}^{\times}$.

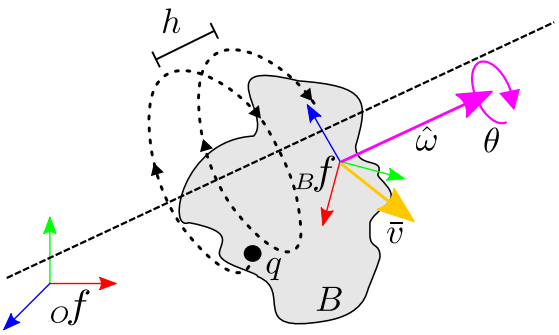


Figure 3.3: Screw motion of a body B with respect to a reference frame ${}_Of$; body B rotates around and translates along the screw axis defined by the unitary vector $\hat{\omega}$; the amount of rotation θ and the pitch of the screw, h, define the amount of translation, $h\theta$; The trajectory of a point q on body B is also shown; ${}_Bf$ and ${}_Of$ are coordinate frames attached to the respective bodies

Even though the previous matrix is not skew-symmetric, it is common to extend the notation $^\times$ and summarize the previous result as:

$$
\begin{aligned}
{}^B\eta^\times &= \begin{pmatrix} {}^B\omega^\times & {}^Bv \\ 0_{1\times 3} & 0 \end{pmatrix} \\
{}^B\eta &= ({}^B\omega, {}^Bv)^T
\end{aligned}
\tag{3.74}
$$

${}^B\eta^\times$ is an element of $se(3)$, the Lie Algebra of the Lie Group $SE(3)$, and ${}^B\eta$ are the **exponential coordinates** or *twist* of the rigid body motion, both expressed in body frame coordinates.

We can similarly define:

$$
{}^O\eta^\times = {}^O_B\dot{T}\,{}^O_BT^{-1} = \begin{pmatrix} {}^O\omega^\times & {}^Ov \\ 0_{1\times 3} & 0 \end{pmatrix} \tag{3.75}
$$

$$
{}^O\eta = ({}^O\omega, {}^Ov)^T \tag{3.76}
$$

where ${}^O\omega^\times$ is the angular velocity of the body B and ${}^Ov$ is the linear velocity of the origin of the frame B, both expressed in the coordinates frame of the observer O.

The relationship between twists in different coordinate frames is defined by the *Adjoint Transformation* matrix:

$$
{}^O\eta = {}^O_BAd\,{}^B\eta \tag{3.77}
$$

$$
{}^O_BAd = \begin{pmatrix} {}^O_BR & 0 \\ d^\times\,{}^O_BR & {}^O_BR \end{pmatrix} \tag{3.78}
$$

The trajectory of a point attached to the body B is given by the equation:

$$
{}^Oq(t) = exp\left({}^O\eta^\times t\right)\,{}^Oq(0) = exp\left(\hat{\eta}^\times_O\theta t\right)\,{}^Oq(0) \tag{3.79}
$$

If both the angular and the linear velocity terms are non-zero the point follows a *helical* trajectory (see Figure 3.3). In that case the rigid body B follows a *screw* trajectory, where the linear and angular velocities are related by a constant value called pitch, h. A mechanism restricting the relative motion between two rigid bodies to a constant twist is called *screw joint*.

In the special cases of zero angular velocity or zero linear velocity we have a pure rotation or pure translation. We will see later that the mechanisms restricting the relative motion between two rigid bodies to these special cases are called *revolute joint* and *prismatic joint*.

Analogously to the rotation case, we can define a matrix exponential and logarithm to integrate twists into homogeneous transformations and "derive" homogeneous transformations into twists.

$$exp : \eta^{\times} \in se(3) \rightarrow T \in SE(3) \tag{3.80}$$

$$log : T \in SE(3) \rightarrow \eta^{\times} \in se(3) \tag{3.81}$$

Geometric interpretation of a twist Before, we interpreted an angular velocity ω as a line (the rotation axis) with orientation represented by the unit vector $\hat{\omega} = \frac{\omega}{\|\omega\|}$ passing through the origin, and a rate of rotation around this line $\|\omega\| = \dot{\theta}$. Analogously, a twist can be interpreted as a line (the twist axis) in 3D Euclidean space, and a rate of rotation around it and translation along it. Equivalently, we can define a single rate of motion (twist velocity, $\dot{\theta} \in \mathbb{R}$) associated with the line, and the factor relating amount of rotation and translation. The line, together with the factor between rotation and translation, is also known as the screw axis, and the factor is the pitch of the screw.

Given a twist $\eta = (\omega, v)^T$ we can compute the screw axis as the line $\lambda = (\hat{l}_{ori}, l_{pos}, h)$ parallel to the unitary vector $\hat{l}_{ori}$, passing the three-dimensional point l_{pos} with the pitch h, defined by the equations:

$$\hat{l}_{ori} = \frac{\omega}{\|\omega\|} \tag{3.82}$$

$$l_{pos} = \frac{\omega \times v}{\|\omega\|^2} \tag{3.83}$$

$$h = \frac{\omega \cdot v}{\|\omega\|^2} \tag{3.84}$$

In the special case of a pure rotation ($\|v\| = 0$), the line of the screw axis passes through the origin ($l_{pos} = \bar{0}$) and the pitch is zero ($l_{pitch} = 0$), reducing the screw axis to the revolute axis we studied before.

In the special case of a pure translation ($\|\omega\| = 0$) the orientation of the screw axis is defined as the direction of the translation, $\hat{l}_{ori} = \frac{v}{\|v\|}$ and the pitch is infinite ($h = \infty$). The location of the screw axis of a pure translation is undefined, meaning that any line parallel to $\hat{l}_{ori}$ represents the axis of motion.

Inversely, given a screw axis $l = (\hat{l}_{ori}, l_{pos}, h)$ and a twist velocity $\dot{\theta}$, the complete twist is given by

$$\eta = \begin{pmatrix} \hat{l}_{ori}\dot{\theta} \\ -\hat{l}_{ori}\dot{\theta} \times l_{pos} + h\hat{l}_{ori}\dot{\theta} \end{pmatrix} \quad \text{if} \quad h \neq \infty \tag{3.85}$$

or

$$\eta = \begin{pmatrix} 0_{3\times 1} \\ \hat{l}_{ori}\dot{\theta} \end{pmatrix} \quad \text{if} \quad h = \infty \tag{3.86}$$

In the general case ($h \neq \infty$) the linear velocity is the sum of two components, a motion along the screw axis $h\hat{l}_{ori}\dot{\theta}$ and a motion on the perpendicular plane resulting from the rotation $-\hat{l}_{ori}\dot{\theta} \times l_{pos}$.

We will use the equations and definitions of this section to perceive from interactions the motion of rigid bodies in the environment. These equations will be useful to define models for recursive state estimation processes so that the processes can exploit physical priors. However, as stated in Chapter 1, our goal is to apply our approach for interactive perception for robotic

manipulation to the estimation of properties of articulated objects. These objects are defined by the constraints in the motion of their composing rigid bodies. So far we have seen only mathematical definitions for the unconstrained motion of a rigid body. In the next section we will provide mathematical models for motion constraints that we will use within recursive state estimation processes to perceive the kinematic structure of an articulated object, and also its dynamic properties.

3.3 Articulated Objects and their Kinematics

Articulated objects are mechanisms composed of rigid parts, called links, and connections between them, called joints. The joints restrict the relative motion between links to less dimensions than the six generally possible between disconnected rigid bodies. From a dynamics point of view, the joint mechanisms return forces applied in the constrained dimensions, while forces in the allowed dimension (if sufficient to overcome friction and other dynamic effects of the mechanism) will generate motion of the links and a change in the kinematic state of the articulated object.

Humans exploit these kinematic and dynamic properties to create articulated objects with desired restrictions in their motion. Many tools and everyday human objects are articulated mechanisms, e.g. scissors, pliers, books, drawers, doors, boxes, or faucets. Humans can easily manipulate them using compliant interactions, which reveal and allow to perceive their constraints.

Mechanically restricting the relative motion between components of an articulated object provides an important advantage for manipulation: forces applied to the object induce motion **only** along the allowed (desired) dimensions. This property is exploited in the design of mechanisms so that a large variety of applied forces result in the same restricted motion of the mechanism and undesirable areas of the space of relative motion are avoided. The kinematic structure of an articulated objects acts as a funnel guiding (restricting) the motion towards the desired subspace.

Being such a common type of object, perceiving these objects (their kinematic, geometric and dynamic properties) is crucial for robots that aim to understand and manipulate human environments. And to do so, interacting with them is the best way to reveal their properties and their functionalities.

In the previous sections we studied the free motion of rigid bodies, how to represent and operate 6D poses and trajectories. We will now study constraints in this motion: how to represent them and how to relate them to 6D rigid body motion. In the following we will review the most relevant types of kinematic constraints in articulated mechanisms and their representation.

We call *kinematic structure* the model that defines the motion constraints and degrees of freedom within rigid components of an articulated object. The kinematic structure defines a submanifold in the space of all possible combinations of 6D relative poses between links. Elements of this submanifold are possible configurations of the articulated mechanism. The coordinates to uniquely define each element of the submanifold are called *generalized coordinates* and represent the state of each joint of the object. The set of all joint states is also called *kinematic state* of the articulated object. Given the kinematic structure and state, the pose of each link can be estimated unequivocally in an operation that is called *forward kinematics*. The inverse operation, obtaining the kinematic state given the kinematic structure and the pose of one/several links of the mechanism is called *inverse kinematics*.

TYPES OF LINKS Links are classified based on the number of joints they connect to, called the order of the link. We talk about a *binary* link when it connects with two joints to other links, a *ternary* link if it connects to three joints, and a *quaternary* link if it connects to four joints. In this thesis, we will encounter mostly articulated objects with binary links but the proposed methods apply to links of any order. Most articulated objects possess binary links, while higher order links are used to create complex mechanisms like motors.

TYPES OF JOINTS We define analogously the order of a joint as the number of links they connect to: a *binary* joint connects two links, a *ternary* joint connects three links and a *quaternary* joint connects four links. The approaches presented in this thesis are restricted to binary joints and thus we will only review this type of joints in this section. However, we do not consider this limitation to be relevant since most articulated objects in human environments possess only binary joints.

The number of *degrees of freedom (DoF)* of a joint represents the dimensionality of the manifold of the 6D space of relative poses defined by the joint. It corresponds to the minimum number of independent variables required to span the manifold and fully define the relative pose. Based on their number of degrees of freedom we further classify binary joints into the following most common linkages:

- 0-DoF joints: rigid joints
- 1-DoF joints: *revolute* (also called hinge or pin), *prismatic* (also called slider) and *screw joints* (also known as helical joints).
- 2-DoF joints: *cylindrical* joints
- 3-DoF joints: *planar* and *spherical* joints

Revolute and prismatic joints are the most common in human environments, and therefore, the perceptual systems we will present in the following chapters will focus on the perception of these types of motion constraints.

In the following we will review the most common parametric representations and properties of 0, 1 and 2-DoF joints. We will assume that each link connected to the joint have a reference frame rigidly attached to it. Without loss of generality we will consider one of the links to be the reference, also known as parent link, and the other to be the dependant or child link. We will specify the joint parameters with respect to the frame of the parent link.

Given the parameters of a joint, we will define the forward kinematic equations to describe the pose of the child link frame with respect to the parent link frame. The forward kinematic equations depend on the joint parameters and the kinematic state of the joint: ${}^{parent}_{child}T(\lambda^{joint}, q^{joint})$.

We will decompose the forward kinematics equation into a constant element given by the relative pose of the child link with respect to the parent link when the joint state is zero, and a variable component representing the change in relative pose due to joint actuation:

$$ {}^{parent}_{child}T(\lambda^{joint}, q^{joint}) = {}^{parent}_{child}\Delta T(\lambda^{joint}, q^{joint})\, {}^{parent}_{child}T(q^{joint} = 0) \tag{3.87} $$

RIGID JOINT A rigid joint does not allow any relative motion between the connected links. Therefore, there are no parameters nor joint state variable required to define the joint. The pose of the child link frame with respect to the parent link is constant over time and does not depend on any joint state variable, ${}^{parent}_{child}T = cte$.

REVOLUTE JOINT A revolute joint allows a single degree of freedom of rotation between the connected links. The axis of rotation fully defines the constraints of motion of a revolute joint. This axis corresponds to a line in 3D Euclidean space (see Figure3.4). We can choose any line parametrization to define the rotation axis. In the next chapters, we will use a point-vector parametrization:

$$\lambda^{rev} = (\hat{l}^{rev}_{ori}, l^{rev}_{pos}) \tag{3.88}$$

where $l^{rev}_{pos} \in \mathbb{R}^3$, and $\hat{l}^{rev}_{ori} \in \mathbb{R}^3$ and $\|\hat{l}^{rev}_{ori}\| = 1$, which constrains $\hat{l}^{rev}_{ori}$ to lay on the unit sphere S^2 in $\mathbb{R}^3$. We will often represent $\hat{l}^{rev}_{ori}$ by its spherical coordinates $\hat{l}^{rev}_{ori} = (\phi, \theta, r = 1)$, where we could drop r. Compared to the equations of a screw axis, we can represent a revolute axis as a screw axis with zero pitch, $h = 0$.

Given a rate of rotation $\dot{q}^{rev}$, a revolute joint constrains the velocity twist to present the form:

$$\eta^{rev}(\dot{q}^{rev}) = \begin{pmatrix} \hat{l}^{rev}_{ori}\dot{q}^{rev} \\ -\hat{l}^{rev}_{ori}\dot{q}^{rev} \times l^{rev}_{pos} \end{pmatrix} \tag{3.89}$$

Similarly, the change in relative pose for a given joint state q^{rev} is constrained by the revolute joint to

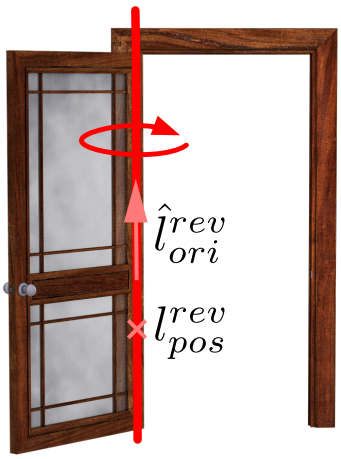


Figure 3.4: An articulated object, a door, with a revolute joint; the blade of the door rotates with respect to the door frame around the joint axis; the joint axis is shown in red color; the point-vector parametrization of the axis is shown in salmon color

$$^{parent}_{child}\Delta T(\lambda^{rev}, q^{rev}) = exp\,(\eta^{rev}(q^{rev})) \tag{3.90}$$

PRISMATIC JOINT A prismatic joint allows only one degree of freedom of translation between the connected links. To fully define the motion constraints of a prismatic joint we need to define the axis of translation. This axis is a free-floating line in 3D Euclidean space (see Figure3.5). Any parametrization for free-floating lines can be used to define the translation axis. We use a vector parametrization:

$$\lambda^{pri} = \hat{l}^{pri}_{ori} \tag{3.91}$$

$\hat{l}^{pri}_{ori} \in \mathbb{R}^3$ and $\|\hat{l}^{pri}_{ori}\| = 1$, which constrains $\hat{l}^{pri}_{ori}$ to lay on the unit sphere S^2 in $\mathbb{R}^3$. As with the orientation of a revolute axis, we will often represent $\hat{l}^{pri}_{ori}$ by its spherical coordinates, azimuth ϕ and elevation θ. The above prismatic axis definition is equivalent to an screw axis with the same line parameters and infinite pitch, $h = \infty$.

Given a rate of translation $\dot{q}^{pri}$, the prismatic joint constrains the velocity twist to:

$$\eta^{pri}(\dot{q}^{pri}) = \begin{pmatrix} 0_{3\times 1} \\ \hat{l}^{pri}_{ori}\dot{q}^{pri} \end{pmatrix} \tag{3.92}$$

The change in relative pose due to prismatic joint actuation for a given joint state q^{pri} is

$$^{parent}_{child}\Delta T(\lambda^{pri}, q^{pri}) = exp\left(\eta^{pri}(q^{pri})\right) \tag{3.93}$$

Screw Joint A screw joint allows only one degree of freedom of motion (rotation and translation with a fixed scale factor) between the connected links. To fully define the motion constraints of a screw joint we need to define the axis of motion. This axis is a line in 3D Euclidean space. Any parametrization for lines can be used to define the screw axis. We use again a point-vector parametrization:

$$\lambda^{scw} = (\hat{l}^{scw}_{ori}, l^{scw}_{pos}) \tag{3.94}$$

where $l^{scw}_{pos} \in \mathbb{R}^3$, and $\hat{l}^{scw}_{ori} \in \mathbb{R}^3$ and $\|\hat{l}^{scw}_{ori}\| = 1$.

Given a velocity of screw motion $\dot{q}^{scw}$, the screw joint constrains the velocity twist to:

$$\eta^{scw}(\dot{q}^{scw}) = \begin{pmatrix} \hat{l}^{scw}_{ori}\dot{q}^{scw} \\ -\hat{l}^{scw}_{ori}\dot{q}^{scw} \times l^{scw}_{pos} \end{pmatrix} \tag{3.95}$$

The change in relative pose due to screw joint actuation for a given joint state q^{scw} is

$${}^{parent}_{child}\Delta T(\lambda^{scr}, q^{scr}) = exp\,(\eta^{scr}(q^{scw})) \tag{3.96}$$

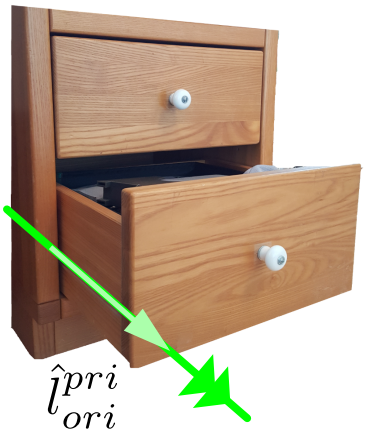


Figure 3.5: An articulated object, a drawer, with a prismatic joint; the drawer translates with respect to cabinet along the joint axis; the joint axis is shown in green color; the orientation vector of the axis is shown in light green color

Cylindrical Joint A cylindrical joint allows two degrees of freedom of motion (independent rotation and translation) between the connected links. To fully define the motion constraints of a cylindrical joint we need to define the axis of motion. This axis is a line in 3D Euclidean space. Any parametrization for lines can be used to define the screw axis. We use a point-vector parametrization:

$$\lambda^{cyl} = (\hat{l}^{cyl}_{ori}, l^{cyl}_{pos}) \tag{3.97}$$

where $l^{cyl}_{pos} \in \mathbb{R}^3$, and $\hat{l}^{cyl}_{ori} \in \mathbb{R}^3$ and $\|\hat{l}^{cyl}_{ori}\| = 1$.

Given the joint velocity $\dot{q}^{cyl} \in \mathbb{R}$, the cylindrical joint constrains the velocity twist to:

$$\eta^{cyl}(\dot{q}^{cyl}) = \begin{pmatrix} \hat{l}^{cyl}_{ori}\dot{q}^{cyl}_{ang} \\ -\hat{l}^{cyl}_{ori}\dot{q}^{cyl}_{ang} \times l^{cyl}_{pos} + \hat{l}^{cyl}_{ori}\dot{q}^{cyl}_{lin} \end{pmatrix} \tag{3.98}$$

where $\dot{q}^{cyl}_{ang}$ and $\dot{q}^{cyl}_{lin}$) are the angular and linear velocities, respectively.

The change in relative pose due to cylindrical joint actuation for a given joint state q^{cyl} is

$${}^{parent}_{child}\Delta T(\lambda^{cyl}, q^{cyl}) = exp\left(\eta^{cyl}(q^{cyl})\right) \tag{3.99}$$

We will use the previously presented kinematic models of joints to build, estimate and track articulated objects from interactions.

3.4 Mathematical Notation

The following tables give a list of variables, operators, and functions used throughout the thesis. We will diverge from this notation in individual cases to maintain the explanations unambiguous, or to improve the readability of the text. The correct meaning of the symbols in those cases should be obvious from the context.

Notation	
$\boldsymbol{x}$	Random variable
$p(x)$	Probability distribution of the random variable $\boldsymbol{x}$
x	Mean of the random variable $\boldsymbol{x}$
Σ_x	Covariance of the random variable $\boldsymbol{x}$
$\mathcal{N}(x, \Sigma_x)$	Gaussian distribution of mean x and covariance Σ_x
$\boldsymbol{x}_t$	Current state in a recursive estimation process
$\boldsymbol{z}_t$	Current measurement in a recursive estimation process
u_t	Current action in a recursive estimation process
$\hat{x}_t$	Predicted state in a recursive estimation process
$\hat{z}_t$	Predicted measurement in a recursive estimation process
P_t	Covariance of the current state in a parametric Bayes filter
w_t	Additive Gaussian system noise
Q_t	Covariance of the system noise in a parametric Bayes filter
v_t	Additive Gaussian measurement noise
R_t	Covariance of the measurement noise in a parametric Bayes filter
F_t	First order derivative of the forward model with respect to the state variable in an EKF
H_t	First order derivative of the measurement model with respect to the state variable in an EKF
Δ_t	Time interval between state estimations
$\frac{\partial f(x)}{\partial x}\Big\rfloor_{\bar{x}}$	Derivative of function f with respect to its variable x, followed by the substitution $x = \bar{x}$

Table 3.1: Mathematical notation used in this thesis: probability theory and Bayesian filtering

Notation (cont.)	
${}^O_B p$	6D pose of (a frame attached to) a body B with respect to (a frame attached to) a body O (undetermined parametrization)
${}^O_B R$	Rotation matrix representing a) the 3D orientation of a frame B with respect to O, b) a transformation of the coordinates from frame B to O, or c) the operation when a body moves from the orientation of frame O to B
${}^O_B d$	Translation vector representing a) the 3D position of a frame B with respect to O, b) a transformation of the coordinates from frame B to O, or c) the operation when a body moves from the location of frame O to B
${}^O_B T$	Homogeneous transformation matrix representing a) the 6D pose of a frame B with respect to O, b) a transformation of the coordinates from frame B to O, or c) the operation when a body moves from the location of frame O to B
$\oplus$	Composition of poses (product of matrices in homogeneous form)
$\ominus$	Composition of poses with pre-inversion of the second element (product of matrices in homogeneous form, with pre-inversion of the second)
${}^O_B \omega$	Angular velocity of a body B with respect to O, expressed in the coordinate frame of O
${}^O_B \omega^\times$	Skew-symmetric matrix representation of the angular velocity of a body B with respect to O
${}^O_B \hat{\omega}$	Unitary vector in the direction of the angular velocity of a body B with respect to O
${}^O_B v$	Linear velocity of a body B with respect to O
${}^O_B \hat{v}$	Unitary vector in the direction of the linear velocity of a body B with respect to O
${}^O_B \eta = ({}^O_B \omega, {}^O_B v)$	6D spatial velocity of a body B with respect to O (twist in exponential coordinates)
${}^O_B \eta^\times$	Matrix representation of the 6D spatial velocity of a body B with respect to O
$exp\left({}^O_B \omega^\times \Delta_t\right),\ exp\left({}^O_B \eta^\times \Delta_t\right)$	Matrix exponential
$log\left({}^O_B R\right),\ log\left({}^O_B T\right)$	Matrix logarithm
${}^O_B Ad$	Adjoint transformation matrix from frame B to frame O

Table 3.1: (Continued) Mathematical notation used in this thesis: spatial descriptions and kinematics of rigid bodies

4

Perceiving Kinematics of Articulated Objects from RGB-D Streams

Robots achieve tasks by manipulating their environment. This manipulation is the deliberate change of the configuration of objects. When it comes to articulated objects, the change in the configuration is a change of the kinematic state of the joints of the mechanism. To perform such manipulation successfully, the robot must be able to detect and track degrees of freedom (DoF) and constraints in the environment, defined by the kinematic joints. Detection includes the characterization of DoF based on joint type and joint axis. Tracking implies the continuous perception of DoF state in order to monitor manipulation progress, recognize completion, or detect failure. These perceptual capabilities are a fundamental prerequisite for successful manipulation in unstructured environments with unknown objects (Figure 4.1).

Perceiving kinematic properties of articulated objects is intrinsically an interactive task. Interactions generate motion and reveal the constraints imposed by the kinematic structure on the links of the object. The insight that interaction should be an intrinsic component of a perceptual solution is at the core of the family of algorithms called interactive perception, which includes the system presented in this chapter.

The perception of DoF of articulated objects poses the challenges we discuss in Chapter 1.1: the robot needs to understand the changes caused by interaction from changes in the sensor signals (CH1), this understanding has to be quick and based only on sensor signals acquired so far (CH2), and the perceptual skill has to be versatile and applicable to many different objects and environmental conditions (CH3). In this chapter, we present an *online* interactive perception (online IP) system to estimate parametrized kinematic models of unknown objects from streaming RGB-D data addressing the aforementioned challenges. The key of our approach is to leverage the structure of the perceptual problem: the information from interactions, physical priors that model the underlying processes, the temporal correlation in the perceived information, and the interdependency between perceptual subproblems. To exploit these structural properties we propose an online interactive perception system based on three interconnected levels of recursive estimation: *1)* the estimation of 3D feature motion based on the 2D motion of tracked RGB features, *2)* the estimation of rigid body motion based on the estimated feature motion, and *3)* the estimation of the kinematic model based on the rigid body motion (Figure 4.2). The probabilistic representations used for estimation yield a kinematic model with uncertainty estimates.

Our system exploits structural properties inherent to perceptual problems related to robot

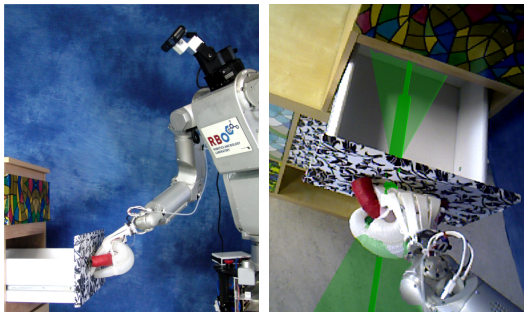


Figure 4.1: Example of online interactive perception: The robot pulls on the drawer using an anthropomorphic soft hand built in our lab (Deimel & Brock, 2014) and perceives the prismatic joint (joint axis shown as narrow green cylinder, joint value shown as wider green cylinder), including an estimate of the uncertainty (transparent green cone) [© 2014 IEEE]

manipulation of DoF in unstructured environments, leading to the observed robustness, accuracy, and generality. First, the online IP system exploits and integrates **interaction as part of the perceptual solution**. The system focusses on the interpretation of the changes in the environment revealed from interactions, and the understanding of the underlying structure in the world that governs these changes (the kinematic constraints). This working principle classifies our proposed approach into the family of interactive perception methods.

Second, our solution uses **recursion** as algorithmic implementation to **exploit the temporal structure** in the perceptual problem. Using recursion our solution turns detection into tracking using the previously perceived state as prior to restrict the possible next states. This contributes to the online capabilities of our solution.

Third, the factorization of the overall perceptual problem into three levels enables the use of highly relevant, level-specific **physical priors**, namely motion continuity, rigid body physics (based on the assumption that the environment is composed of rigid parts) and kinematics of rigid bodies. The physical priors effectively improve the quality of data at each level.

And fourth, the three levels of the recursive estimation **problems are interconnected**, leveraging synergistically the interdependencies between subtasks. The information improved by the level-specific priors is passed to other levels, thereby also improving the effectiveness of the estimation process on other levels. The overall effect is that the combined estimation process is informed not only by sensor data but also by three specific process models, each containing task-relevant information to help interpret the uncertain data.

In the following, we will first review previous approaches from computer vision and robotics that tackle the problem of perceiving kinematic properties of articulated objects. We will see that our system is the first solution that addresses the full perceptual problem (from raw sensor data to kinematic model) for previously unknown objects and in an online manner. Then, we will present our approach based on coupled recursive estimation, followed by an experimental evaluation on different articulated objects and environmental conditions. We will end with a discussion of the limitations and the implications of this work, also in the context of the challenges and opportunities for perception discussed in this thesis.

4.1 Related Work

The earliest approaches to perceive articulated objects were proposed by the computer vision community. These approaches perceived kinematic constraints in the motion of multiple rigid bodies from video sequences. Approaches of this "first generation" are based on the seminal work by Costeira & Kanade (1998). The authors propose an approach to reconstruct shape and motion for multiple moving bodies. Their approach builds and analyzes the structure of

a matrix containing the trajectories of a set of point features tracked along the entire video sequence. This matrix is called the *measurement matrix* (Tomasi & Kanade, 1992) and, correctly interpreted, leads to information about the (sparse) shape and the motion of the bodies. The method to interpret the measurement matrix is spectral clustering, and generates groups of features that move with correlated trajectories and the motion parameters of these trajectories in 3D space.

Tresadern & Reid (2005) proposed an integrated approach for segmentation and joint detection based on an analysis of the dependencies in the motion subspaces obtained with spectral clustering. They find intersecting dimensions in these subspaces that indicate constraints in the relative motion between pairs of bodies. Based on the intersecting dimensions their method can classify joints into disconnected, universal and revolute/hinge, and estimate the joint parameters: the axis of rotation of the revolute joint and the point of rotation of the universal joint. This method is limited to one-joint structures.

Later on, Yan & Pollefeys (2006) extended the idea of Tresadern and Reid to more complex kinematic structures. They build a fully connected graph where the nodes are moving bodies and the edges are weighted by the motion dependency observed between bodies, measured as the minimum principle angle between their motion subspaces (Golub & Van Loan, 2012). Pairs of bodies moving with high dependency are connected by joints. The complete kinematic structure of the articulated object is defined as the minimum spanning tree in the motion dependency graph. Their method can deal with multiple articulated objects based on an upper threshold for the minimum principle angle that indicates that two bodies are disconnected, but it cannot deal with closed kinematic chains.

The methods based on spectral clustering demonstrated that it is possible to perceive visually the kinematic constraints between moving bodies and infer the kinematic structure of an object. However, they require to accumulate large motion data to estimate correctly the clusters and the motion subspaces of the point features, and are thus inherently offline algorithms. While this limitation is not important for video analysis, it restricts the application of spectral clustering-based methods to perception that aims to support ongoing robot manipulation of articulated objects.

More recent solutions to perceive kinematic structures adopted a probabilistic approach. They posed the perceptual problem as the estimation of the model (the kinematic structure, and possibly also its dynamic state) that maximizes the likelihood of the observations. Ross et al. (2008) proposed a generative model as solution to the underlying multi-body structure from motion (SfM) from point feature trajectories and joint estimation problems. In an iterative process they first assign point features to links, run SfM, and estimate the points that belong to a pair of links and that do not change their 3D location. These points indicate possible locations of a revolute or a universal joint. The generative model obtained with this method is used to evaluate how well the hypothetical kinematic structure model fits the observed point feature trajectories, and select the most likely. This method cannot cope with prismatic joints nor with multi-joint structures.

Sturm et al. (2009) (Sturm et al., 2010b, 2011) presented a probabilistic approach to joint classification and parameter estimation. Their method uses as input the 6D pose trajectories of the moving bodies to build and maintain four models of possible motion constraints between pairs of bodies. The method estimates through optimization the set of parameters maximizing the likelihood of the observed trajectories. The authors propose three possible parametric models for joints – rigid, prismatic, or revolute joints – and one non-parametric model – a Gaussian process joint. Similar to Yan & Pollefeys (2006), they compute the minimum spanning tree on a graph structure as final kinematic structure of the object. But differently, in their method the weight of the nodes is inversely proportional to the posterior

probability of the estimated best joint models, and therefore finding the minimum spanning tree is equivalent to a maximum a posteriori computation (MAP).

While the method by Sturm et al. (2009) is elegant, completely defined using probabilistic algebra, and applicable in real-time, it does not tackle the full perceptual problem. The method does not address a crucial subtask when perceiving kinematics of articulated objects: detecting and tracking unknown rigid bodies from raw sensor data. The approach assumes the number of rigid bodies to be known beforehand, and their poses to be tracked reliably, delegating this task to a visual tracker based on AprilTag-like fiducial markers (Wang & Olson, 2016). The exclusion of the "lowest" part of perception (the interpretation of the noisy sensor stream) from the problem is a missed opportunity to link and exploit high level reasoning and low level signal processing.

In two later extensions Sturm et al. extended their approach to reduce the dependency on the fiducial visual markers. In a first extension (Sturm et al., 2010b) they proposed to obtain the body trajectories from a plane tracker based on depth images. This method is limited to planar objects, and its serial processing procedure does not leverage the information about the kinematic structure that could help the plane tracker. In a second extension (Sturm et al., 2010a) the authors use as perceptual signal the robot's end-effector trajectories generated with a model predictive controller. This approach, while nicely linking perception and action, is strongly limited because it can only perceive one DoF objects rigidly attached to the environment (see Section 6.2.2).

Another group of methods perceive the kinematic model from a geometrical analysis of the rigid body trajectories. Huang et al. (2012) present an offline method to extract 3D models of articulated rigid objects using interactive perception. This method requires multiple object views to first generate a full point cloud of an object, which is then used to estimate the kinematic state by matching the configurations before and after the interaction.

Katz et al. (2014) propose an RGB-based, offline solution for the perception of three-dimensional, rigid kinematic structures. In their approach, the authors apply bundle adjustment to groups of point feature trajectories to estimate their 3D motion, and fit the computed 3D trajectories to joint hypothesis based on their geometrical properties. To group features into rigid bodies the authors apply a series of min-cuts (Matula, 1987) to a graph where the nodes are point features, and the edges are weighted based on a set of feature similarity estimators (color, relative motion, ...). Subsequently, this method was adapted for RGB-D sensors (Katz et al., 2013b). The use of RGB-D sensing avoids the costly bundle adjustment computation and is therefore more accurate and computationally more efficient, but still offline since it requires large feature trajectories, and thus suffering some inherent limitations, for example for newly appearing objects (see Section 4.3.3).

Conclusions and comparison to the proposed approach: The existing methods in the literature estimate the kinematic model in a batch offline manner: first they collect enough motion observations, and then they analyse the motion to infer the kinematic constraints. The robot cannot use the information acquired in this form to support and steer ongoing interactions. Storing all measurements and running one of these methods over the entire memory when a new signal arrives becomes quickly computationally infeasible since the computation complexity grows with the number of measurements.

An exception is the algorithm by Sturm et al. (2009): it runs in an online manner using a sliding window approach that selects a subset of the entire series of acquired observations. This is possible because the perception of motion of the rigid parts is not part of the tackled problem, but obtained from fiducial markers, which solve the segmentation, matching, and

pose estimation problems. The sliding window approach fails if there are large time periods without any motion.

There are two possibilities to improve over the batch and the sliding window approaches. The first solution is to design a measurement selection method to reduce the amount of data for the analysis, losing as few information as possible. Such a method would need to evaluate the information contained on each measurement *before* the analysis, which usually requires complex heuristics.

The second solution is to take an online recursive approach: reuse at each step the result of the kinematic computation at the previous step and refine it based on the latest observation. Using recursion we keep the amount of data to process constant: ideally only the latest observation. As explained in Chapter 3.1.1 this is possible if we assume that the state is complete and the dynamic system is Markovian. Finding a factorization of the problem where the subcomponents can be assumed to fulfill these properties would allow us to use recursive estimation for the solution. Moreover, if we characterize probabilistically the uncertainty over the measurements and the predictions generated by the recursive solution we can apply a Bayesian filter implementation to balance correctly between the previously perceived information and the newly acquired observation.

Compared to the existing systems, the system we present in this chapter advance the state of the art in interactive perception of kinematics of articulated bodies in three respects. First, as explained before, existing IP methods are offline systems and therefore cannot inform the ongoing action of the robot, originally the goal of interactive perception. The proposed online method overcomes this, and integrates the perception process into the execution of actions. Second, the offline setting lead to failure cases that are properly addressed with our online method. Third, existing offline methods are not probabilistic and hence do not include an estimate of model uncertainly. We deem it to be important to reason about uncertainty when manipulating in unstructured environments and this reasoning is a crucial component of our recursive solution.

4.2 Online Visual Perception of Kinematics from Interactions

Our proposed online system factorizes the interactive perception of articulated objects into three recursive state estimation levels: estimating feature motion, rigid body motion, and the overall kinematic model. The structure and interactions of these levels is depicted in Figure 4.2. The system instantiates the general approach for IP we proposed in Section 1.3. Each level exploits a level-specific physical prior: motion continuity, rigid body physics (assuming the objects are composed of rigid parts), and the kinematics of rigid bodies. These priors improve convergence of the state estimate. The resulting state information is passed as a measurement to the next-higher level (blue arrows). The predicted measurement of each level is also fed back as the predicted state to the next-lower level (red arrows). The information passed to the next-higher and next-lower levels is now informed by the prior and improves convergence at the other levels. These design choices (factorization, recursion, use of priors, feedback to lower levels), based on the opportunities for perception for robot mechanical manipulation, are crucial to achieve the effectiveness, robustness, accuracy, and versatility of the proposed online IP system.

We will now explain in detail the three recursive state estimation levels that constitute our proposed online IP system.

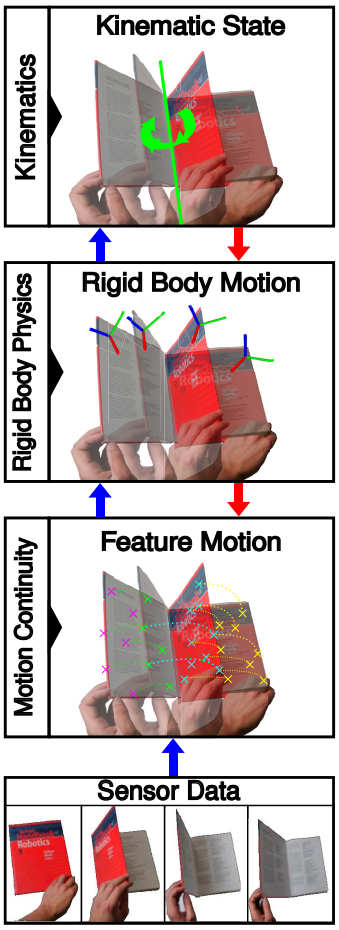


Figure 4.2: Multi-level recursive estimation of kinematic models: (from bottom to top) an RGB-D sensor data stream provides information about a scene, feature motion is estimated, from the feature motion rigid body motion is estimated, from the rigid body motion the kinematic model is estimated; the estimations from each level are passed as measurements to the next-higher level (blue arrows) and the predicted measurements from one level are passed to the next-lower level as state predictions (red arrows); level-specific physical priors to help the estimation process are a key feature of the proposed system (vertical text on the left side of the boxes); the system instantiates the general approach of Section 1.3 [© 2014 IEEE]

4.2.1 Recursive Estimation of Feature Motion

The first level of recursive processing tracks the motion of a set of salient point features in an RGB-D sensor stream using a recursive procedure. The state of this filter at time t, x_t^{fm} (fm = feature motion) presents the form:

$$x_t^{fm} = \{f_t^n = (x_t^n, y_t^n, z_t^n, l^n)\}_{n\in\{1,\dots,N\}} \tag{4.1}$$

where $x_t^n, y_t^n, z_t^n \in \mathbb{R}$ are the coordinates of the salient point feature n in the 3D Euclidean space relative to the sensor frame at time t, and $l^n \in \mathbb{N}$ is a time-constant label that identifies uniquely the feature. N is the number of tracked points that we maintain constant by detecting new salient point features when necessary (see Section 4.2.1).

In the following, we will use the operator $\mathrm{Loc}(f^n) = (x^n, y^n, z^n)^T$ to build a 3D vector of the location of a feature f^n. Sometimes, we will abuse the terminology and use the same operator to build the vector of homogeneous coordinates of the feature location, $\mathrm{Loc}(f^n) = (x^n, y^n, z^n, 1)^T$. The difference will be clear from the context of the operation.

The measurements for this salient point feature tracking process at time t present the form:

$$z_t^{fm} = \{q_t^n = (u_t^n, v_t^n, l^n)\}_{n\in\{1,\dots,N\}} \tag{4.2}$$

where $u_t^n, v_t^n \in \mathbb{R}$ are the 2D coordinates of the salient point feature n in the image plane at time t, and $l^n \in \mathbb{N}$ is the feature label of the corresponding 3D point. To obtain these measurements we align the surroundings of the salient point features between consecutive RGB images using the iterative registration approach by Lucas & Kanade (1981). Applying this registration method to track points in consecutively images of a video was first proposed by Tomasi & Kanade (1991). This point feature tracking procedure is known as the Kanade-Lucas-Tomasi (KLT) salient point feature tracker. In the following we will summarize the characteristics of the KLT tracker that are relevant to our method.

The KLT tracker estimates the displacement $d = (du, dv)$ (also known as *flow*) of an image point (pixel) from an initial location q in a first image at time $t-1$ to its corresponding location $q+d$ in a second image at time t. To estimate the displacement, the KLT tracker considers a window W surrounding the point and minimizes the following energy (error residue) term:

$$\epsilon = \int_{q\in W} [I_{t-1}(q) - I_t(q+d)]^2 dq \tag{4.3}$$

$I_t()$ is the video intensity function that defines the intensity of the image points time t, $I : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$.

The estimation of the flow of a point is solved iteratively from an initial estimate d_0 using the first element of the Taylor series of the video intensity function assuming that:

$$I_t(q + d) \approx I_t(q) + gd_i \tag{4.4}$$

where g is the 2D vector of the gradient of the image I_t and d_i the currently estimated displacement. This iterative process composed of a linearization (first order Taylor expansion) and a minimization is equivalent to a Newton-Raphson optimization procedure. The Newton-Raphson process estimates the displacement that minimizes the difference between image intensities ϵ.

Being an iterative gradient-based method, the solution depends on the initial estimate of the displacement d_0. Different initializations could converge to different local minima of the error ϵ. This sensitivity to the initialization is a known problem of the KLT tracker. In our method we will leverage different priors to initialize the KLT based on predictions of the motion of the point features (see Figure 4.3).

Prediction in Feature Motion Estimation

We propose two forward models to predict the motion of the features. The first model is an internal forward model within the recursive process that assumes that the tracked 3D points do not move from their previous location. This forward model generates a first prediction for the next state (I = first prediction):

$$\hat{x}_t^{fm,I} = \{\hat{f}_t^{n,I} = (x_{t-1}^n, y_{t-1}^n, z_{t-1}^n, l^n)\}_{n \in \{0,\dots,N\}} \tag{4.5}$$

Therefore, the internal forward model encodes *motion continuity* as physical prior: the current 3D location of a point is close to its previous location.

The second model leverage information from the next-higher level (the recursive Bayesian estimation of rigid body motion, see Section 4.2.2) as prior to generate a second prediction for the next state, $\hat{x}_t^{fm,II} = \{\hat{f}_t^{n,II}\}_{n \in \{0,\dots,N\}}$ (II = second prediction). We predict the location of a point feature f^n on a body B that moves with a predicted velocity $\hat{\eta}_t^B$ as:

$$Loc(\hat{f}_t^{n,II}) = exp\left(\Delta_t \hat{\eta}_t^B\right) Loc(f_{t-1}^n) \tag{4.6}$$

where Δ_t is the time elapsed between $t-1$ and t,

This second forward model leverages physics of rigid bodies as prior: the motion of the point features on a rigid body must be consistent with the motion of that rigid body. This second prior allows us to leverage information determined by the next-higher level, the estimated motion of rigid bodies. The next-higher level effectively acts as the forward model of the recursive estimation of feature motion. These more informed predictions lead to a better initialization of the KLT feature tracker of the measurement update, as we will see next.

Measurement Update in Feature Motion Estimation

To predict the measurements we project the two sets of predicted 3D locations into the image plane and obtain two sets of predicted 2D locations, $\hat{z}_t^{fm,I}$ and $\hat{z}_t^{fm,II}$. In this process, our measurement model leverages projective geometry as physical prior to interpret sensor data as evidences of the state.

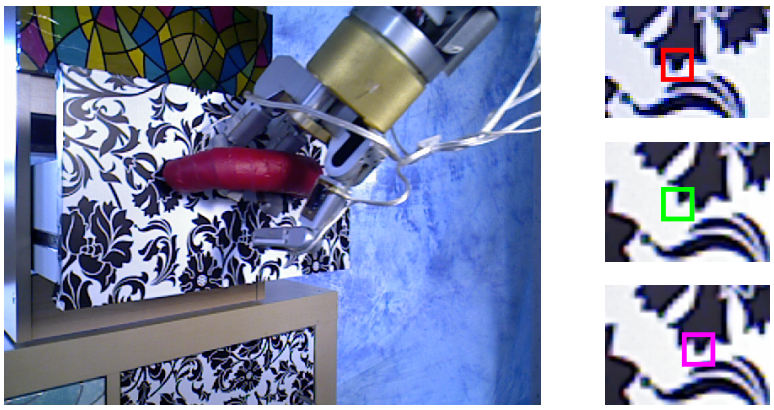


Figure 4.3: Estimating feature motion; left: RGB image input to our perceptual system ; right top: detail on the surface of the moving drawer and location and window (red) of one tracked point feature; right, middle and bottom; same area of the drawer in the next processed RGB image and corrected point feature location from the first initialization (middle, green window) and from the second initialization (bottom, magenta window); the initialization with priors from the next-higher level guides the search to the right location

We use the predicted 2D locations to initialize the KLT salient point tracking algorithm. The KLT tracker corrects these predicted 2D locations finding the displacement that minimizes the registration error.

The two sets of predicted 3D locations lead to different initialization values for the KLT feature tracker. The first set, from the internal forward model based on motion continuity, leads to the standard zero initial displacement of the iterative KLT process, $d_0 = (0, 0)$. The KLT tracker then searches for the optimal registration of the intensity window of the point in the first image starting by the window around the same location in the second image. This standard initialization restricts the capabilities of the KLT to track large motions between frames since the initial displacement could lay in the region of convergence of different point. This problem is depicted in Figure 4.3.

The second set of predicted 3D locations leads to an initialization of the KLT tracker informed by the motion of the rigid bodies. Predictions about the 3D location of the point features based on the expected motion of the bodies guide the salient point KLT tracker to a different area of the image that is closer to the right location.

We compare the feature tracking residues based on the two initializations, $\epsilon^{n,I}$ and $\epsilon^{n,II}$, and select the correction with lowest residue. As we explained before, the residue measures the quality of the matching between image patches after applying the estimated optimal flow. The assumption is that the best correction is the one that best aligns the image patches around the salient point. Finally, we update the state of the recursive process, the 3D location of the point features, by querying the value of the depth map at the corrected tracked 2D locations.

Our recursive estimation schema (prediction and measurement update) in feature motion improves the tracking accuracy and robustness of the salient point KLT feature tracking algorithm by initializing the iterative process using information from the next-higher level.

Feature Initialization and Maintenance

The above presented procedure estimates recursively the 3D location of a set of N points associating them to 2D point features in the image. To initialize the recursion we need to find a set of points that we can track reliably. Additionally, we will need to find new points to maintain a constant number of N tracked points when previous features are lost or actively removed, as we will explain later.

We have seen before that our measurement update uses the KLT iterative solution to the point feature registration problem (Tomasi & Kanade, 1991). We also saw that this solution is based on the computation of the image gradient g in the window W around the point feature. We will now see that, concretely, the computation depends on the second order moments of the intensity gradient in the search window. Based on this dependency, Shi & Tomasi (1994) proposed a method to find the *good features to track* with the KLT. We will now summarize the method to find good features to track by Shi & Tomasi (1994) and the characteristics that are relevant to our system.

To understand the dependency on the second order moments of the intensity gradient we first replace the first order Taylor expansion of the image intensity (see Equation 4.4) in the equation of the residue error:

$$\epsilon = \int\limits_{q \in W} [I_{t-1}(q) - I_t(q) - gd]^2 dq \tag{4.7}$$

The previous equation is quadratic in d and we can solve it analytically by setting its derivative with respect to d to zero:

$$\int\limits_{q \in W} [I_{t-1}(q) - I_t(q) - gd] g dq = 0 \tag{4.8}$$

We can factorize the previous equation into:

$$Gd = e \tag{4.9}$$

where e is the projection of the difference between images along the direction of the gradient, $e = \int\limits_{q \in W} [I_{t-1}(q) - I_t(q)] g dq$ and G is the matrix of second order moments of the gradient in the window W (Hessian):

$$G = \int\limits_{q \in W} g g^T dq = \begin{pmatrix} \int\limits_{q \in W} I_u^2(q) dq & \int\limits_{q \in W} I_u(q) I_v(q) dq \\ \int\limits_{q \in W} I_v(q) I_u(q) dq & \int\limits_{q \in W} I_v^2(q) dq \end{pmatrix} \tag{4.10}$$

with $I_{uu}(q)$, $I_{vv}(q)$, and $I_{uv}(q)$ are the image gradients in the window in the different directions.

We can solve the Equation 4.9 computing $d = G^{-1}e$. This computation depends on the 2×2 matrix G to be well-conditioned so that we can invert it. For G to be invertible its two eigenvalues should be non zero. In practice, obtaining a numerically stable inverse is only possible if both eigenvalues are over a discriminative threshold $min(\lambda_1, \lambda_2) > \lambda_{min}$. This minimum eigenvalue is called the *saliency* of the point.

We can give an intuition of the significance of the salience of a point. If only one of the eigenvalues is large, the intensity of the image varies along one direction. This is the case for

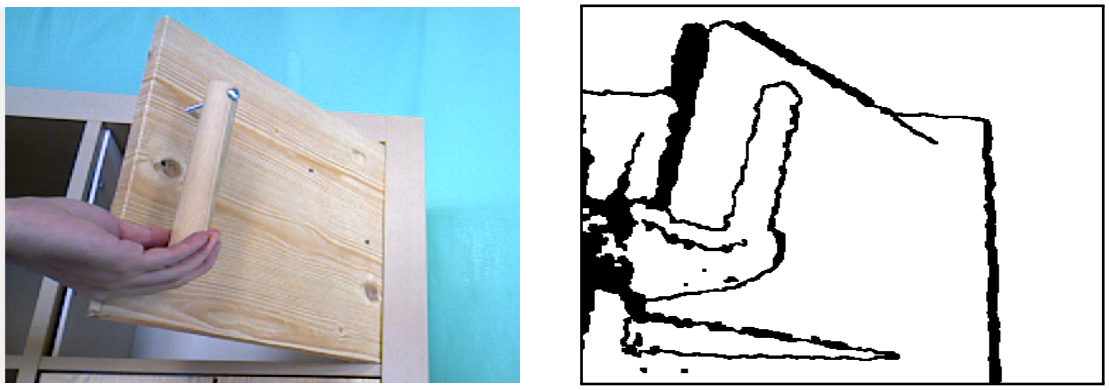


Figure 4.4: Rejection of features on the depth discontinuities; left: original image to our perceptual system; right: depth discontinuities mask (black frame added for visualization); point features on the black edges are rejected

image windows depicting an edge. Image locations where the matrix of second order moments of the gradient in the window is well-conditioned are locations where the image intensity changes abruptly along both directions. This correspond to a corner-like structure in the window. This structures are clearly distinctive compared to points along an edge or on a uniform surface. Therefore, these points are called *salient point features* or *corner features* and they are tracked more robustly and unequivocally that pixels on uniformly textured surfaces.

Based on the previous definition of *good features to track* (Shi & Tomasi, 1994) with the KLT procedure, we detect an initial set of N salient point features where the second order moments of the gradient of the RGB image are maximum and over a discriminative threshold, $\lambda_{min}^{detecting}$.

We compute their corresponding 3D coordinates to initialize the state of the process based on the associated depth value in the fourth channel of the registered RGB-D frame. Often, our approach detects salient points features in sub-optimal locations like depth edges or shadows that do not actually move with the motion of the rigid body. Using our prediction-correction mechanism informed by the motion of the rigid bodies that we explained before we can compensate for these points.

We further improve the reliability of feature tracking by actively rejecting features based on four criteria. First, we remove features when they move out of the field of view because we do not have sensor measurements to update their location. Second, we reject features lying close to depth discontinuities in the RGB-D image (see Figure 4.4). In the presence of sensor noise, these features change their depth drastically, negatively affecting the estimation of rigid body motion. We estimate discontinuities in the depth image using a Canny edge detector (Canny, 1986) and use them to reject point features. Third, when the robot arm enters the field of view, we reject features tracked on its surface (see Figure 4.5). We determine these features by projecting a geometric model of the robot into the image plane using the joint angles of the robot's arm and forward kinematics. In this way we focus the attention of the perceptual system into the degrees of freedom of the unknown articulated objects, and not on the known robot arm. And fourth, we reject points if their tracking error residue increases over a maximum value ϵ_{max}, or if their saliency (minimum eigenvalue of the matrix of second order moments of the gradient image) falls under a discriminative threshold $\lambda_{min}^{tracking}$.

Due to the mechanisms explained above, features get lost often. To compensate for this

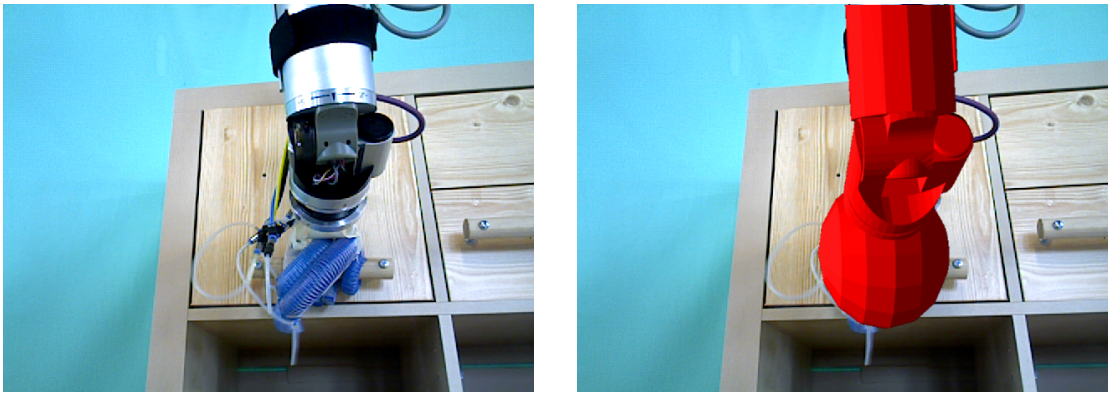


Figure 4.5: Rejection of features on the surface of the robot manipulator; left: original image to our perceptual system from a camera on the robot; right: same image with an overlay (red) of the projected geometric model of the robot; point features on the red area are rejected; the soft hand is represented by a sphere because its exact geometry after the inflation is unknown

loss and to continuously be able to extract useful information from the sensor stream, we increasingly add novel points (based on the approach explained above to detect salient points) to constantly maintain a set of N features.

4.2.2 Recursive Bayesian Estimation of Rigid Body Motion

The second level of recursive state estimation is responsible for detecting and tracking the motion of rigid bodies, based on the feature motion estimated by the next-lower estimation level and the kinematic model estimated by the next-higher estimation level (see Section 4.2.3).

The online solution to this problem requires to solve three interdependent problems. First, we have to continuously associate salient point features to existing or novel rigid bodies. Second, we have to detect when a novel rigid body begins to move. And third, given the association of features to rigid bodies, we have to estimate the motion of each rigid body based on the feature motion.

The motion of one single rigid body is estimated with a recursive Bayesian filter (RBF). We instantiate and maintain one independent RBF for each moving rigid body. In the following, we first suppose that a set of salient point features have been correctly associated to one RBF and describe its prediction and measurement update steps. The RBF to estimate the motion of one rigid body is implemented as an extended Kalman filter (EKF). The detection of rigid bodies and the assignment of features to rigid bodies will be described later in this section.

We represent the kinematic state of a rigid body by its 6D pose and velocity relative to the sensor frame, which we assume to be Gaussian distributed. Representing a Gaussian distribution over 6D poses is not trivial. We adopt the formalism of Barfoot & Furgale (2014) and represent the pose distribution as a mean pose, p (represented in exponential coordinates), perturbed with noise in the tangential Lie algebra space, $\Sigma_t^{p_t}$. The resulting state is ($rbm =$

rigid body motion):

$$\boldsymbol{x}_t^{rbm} = (\boldsymbol{p}_t, \boldsymbol{\eta}_t) \sim \mathcal{N}((p_t, \eta_t), P_t^{rbm}) \tag{4.11}$$

$$p_t \in se(3) \tag{4.12}$$

$$\eta_t \in se(3) \tag{4.13}$$

where the upper-left 6×6 block of the state covariance matrix, $P_t^{rbm} \in \mathbb{R}^{12\times12}$, corresponds to the pose uncertainty in the tangential Lie algebra space as commented before[1].

The measurements of this RBF is the set of M point features $f_t^0, \ldots, f_t^{M-1}$ in 3D Euclidean space associated to this rigid body. We stack their 3D locations to compose a measurement vector for the rigid body RBF:

$$z_t^{rbm} = (\mathrm{Loc}(f_t^0), \ldots, \mathrm{Loc}(f_t^{M-1}))^T \in \mathbb{R}^{3M} \tag{4.14}$$

Prediction in Single Rigid Body Motion Estimation

We use three different process models in parallel to predict the next rigid body state. The first model predicts the next pose of the rigid body based on its current pose and velocity and the elapsed time, Δ_t. The second process model handles the special case when a rigid body stops moving abruptly (for example, when closing a door), setting the current velocity to zero and the predicted pose to be the current pose. The third process model uses the current kinematic model, estimated by the next-higher estimator, to predict an alternative next pose and velocity for the rigid body.

The first forward model is a constant velocity model with random walk in acceleration (I = first prediction):

$$\boldsymbol{x}_t^{rbm,I} = f^{rbm,I}(\boldsymbol{x}_{t-1}^{rbm}) + \boldsymbol{w}_t^{rbm} \sim \mathcal{N}((\hat{p}_t^I, \hat{\eta}_t^I), \hat{P}_t^{rbm,I}) \tag{4.15}$$

$$\hat{p}_t^I = \Delta_t \eta_{t-1} \oplus p_{t-1} \tag{4.16}$$

$$\hat{\eta}_t^I = \eta_{t-1} \tag{4.17}$$

where $\oplus$ is the composition of poses.

The system noise is a zero mean Gaussian distributed random variable defined as:

$$\boldsymbol{w}_t^{rbm} = \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix} \boldsymbol{a}^{rbm} \tag{4.18}$$

where $T = I_{6x6} \cdot \Delta_t$, I_{6x6} is the identity matrix of size 6×6 and $\boldsymbol{a}^{rbm} \sim \mathcal{N}(0, \Sigma_a)$ is a 6D zero mean Gaussian distributed random variable that represents the unknown rigid body acceleration. The covariance of its distribution is given by:

$$\Sigma_a = \begin{pmatrix} a_x & 0 & 0 & 0 & 0 & 0 \\ 0 & a_y & 0 & 0 & 0 & 0 \\ 0 & 0 & a_z & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{rx} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{ry} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{rz} \end{pmatrix} \tag{4.19}$$

[1]Strictly speaking, the vectors of their exponential coordinates p_t and η_t are not elements of the Lie algebra, but the matrices $p_t^\times, \eta_t^\times \in se(3)$. Here we are simplifying the notation.

where the diagonal elements correspond to possible accelerations of the rigid body in the different 6D dimensions. Larger values in the diagonal allow the RBF to adapt to fast motions at the cost of being more sensitive to point feature noise.

The second process model handles the special case when a rigid body stops moving abruptly (for example, when closing a door), setting the current velocity to zero and the predicted pose to be the current pose (II = second prediction):

$$\boldsymbol{x}_t^{rbm,II} = f^{rbm,II}(\boldsymbol{x}_{t-1}^{rbm}) + \boldsymbol{w}_t^{rbm} \sim \mathcal{N}((\hat{p}_t^{II}, \hat{\eta}_t^{II}), \hat{P}_t^{rbm,II}) \tag{4.20}$$

$$\hat{p}_t^{II} = p_{t-1} \tag{4.21}$$

$$\hat{\eta}_t^{II} = 0_{6\times 1} \tag{4.22}$$

The third process model uses the current kinematic model, estimated by the next-higher estimator (see Section 4.2.3), to predict an alternative next pose and velocity for the rigid body (III = third prediction):

$$\boldsymbol{x}_t^{rbm,III} = f^{rbm,III}(\hat{\boldsymbol{z}}_t^{joint}) + \boldsymbol{w}_t^{rbm} \tag{4.23}$$

where the function $f^{rbm,III}$, independent of the current state, predicts the pose and velocity of the body with respect to the sensor frame based on the relative pose between links predicted by the kinematic model. The next-higher level is used therefore as alternative process model of the recursive estimation of rigid body motion.

We will select the prediction among the three alternatives that best predicts the motion of the point features, as we will see later in this section.

Measurement Update in Single Rigid Body Motion Estimation

The measurement input consists of the 3D feature locations estimated on the next-lower level. We predict the future locations of features based on the predicted state of the rigid body and the following observation model:

$$\boldsymbol{z}_t^{rbm,i} = h(\boldsymbol{x}_t^{rbm,i}) + \boldsymbol{v}_t^{rbm} \tag{4.24}$$

$$z_t^{rbm,i} = \begin{pmatrix} \mathrm{Loc}(\hat{f}_t^0)^i \\ \mathrm{Loc}(\hat{f}_t^1)^i \\ \vdots \\ \mathrm{Loc}(\hat{f}_t^{M-1})^i \end{pmatrix} = \begin{pmatrix} exp\left(\hat{p}_t^{rbm,i}\right) \mathrm{Loc}(f_{init}^0) \\ exp\left(\hat{p}_t^{rbm,i}\right) \mathrm{Loc}(f_{init}^1) \\ \vdots \\ exp\left(\hat{p}_t^{rbm,i}\right) \mathrm{Loc}(f_{init}^M) \end{pmatrix} \tag{4.25}$$

where $exp\left(\hat{p}_t^{rbm,i}\right) \in SE(3)$ is the homogeneous transformation obtained from the predicted rigid body pose, $\mathrm{Loc}(\hat{f}_t^m)^i$ is the predicted feature location based on that predicted pose, $i \in \{I, II, III\}$ indicates one of the three alternative predictions, and $\mathrm{Loc}(f_{init}^m)$ are the homogeneous coordinates of the 3D location of the features when the body was initially detected. We predict the new observations relative to this reference to take advantage of increased precision with larger feature motion. While the result of the row operations in the matrix of Equation 4.25 are also homogeneous coordinates, i.e. $(x, y, z, 1)^T$, we remove the constant 1 at the end and stack the remaining elements to obtain a $3M$ dimensional predicted measurement vector.

The noise associated to the measurement, $\boldsymbol{v}_t^{rbm}$, is a zero mean Gaussian distributed variable with $3M \times 3M$ covariance matrix, R_t. We assume that the measured locations are uncorrelated between features, and that the uncertainty about the measured location of a feature f^m is defined by the measurement covariance:

$$R_t^m = f_\sigma(\lambda_t^m, z_t^m) \cdot I_{3\times 3} \tag{4.26}$$

where z_t^m is the z-coordinate of the feature, λ_t^m is the salience of the feature given by the KLT tracker, and $f_\sigma(\lambda_t^m, z_t^m)$ is a function that characterizes the uncertainty about a feature location based on its depth and its salience.

For a measured point feature at the location (x, y, z) with salience value of λ the uncertainty about its location, $f_\sigma(\lambda, z)$, is defined as:

$$f_\sigma(\lambda, z) = \min\left(\sigma_{min}, \frac{\alpha_\lambda}{\lambda - \lambda_{min}}, \alpha_z z^2\right) \tag{4.27}$$

where σ_{min} is the minimum uncertainty value about the point feature measurements, $\frac{\alpha_\lambda}{\lambda - \lambda_{min}}$ assigns higher uncertainty to features tracked in visual areas with low texture (see Section 4.2.1 for an explanation of the salience of a point feature and the minimum value λ_{min}), and $\alpha_z z^2$ represents the quadratic dependency of the measurement uncertainty to the depth of the point in f_σ. This dependency is based on the statistical analysis of RGB-D sensors by Khoshelham & Elberink (2012).

The three alternative process models in our RBF generate different state and measurement predictions, $\text{Loc}(\hat{f}_t^m)^i$. To select the best prediction, the Bayes filter measures the distance between predicted and measured location per feature and adds a vote to the state prediction that leads to the shortest distance. The prediction with most votes is selected as the best state prediction and is used for correction.

Our system exploits in the measurement update the assumption that the environment is composed of rigid bodies, and that the motion of a rigid body is governed by known kinematic relationships. The generated predictions for the feature locations (which our system propagates down to the feature motion estimation level) are thus informed by these two priors.

Extended Kalman Filter for Rigid Body Motion

The estimation of rigid body motion as presented above presents non-linearities in the state and measurement updates. We implement an extended Kalman filter for the recursive solution of this state estimation problem (Section 3.1.3). The EKF linearizes the system using a first-order Taylor expansion of the forward and measurement models around $\boldsymbol{x}_{t-1}^{rbm}$, the previous state estimate. The EKF uses the linearization to estimate the correction and the covariance matrix of the next state of the rigid body.

The linearization of the first forward model corresponds to the following Jacobian matrix:

$$F_t^{rbm,I} = \left.\frac{\partial f^{rbm,I}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,I}} = \begin{pmatrix} \text{Ad}_{\Delta_t \mathrm{v}_{t-1}} & \Delta_t I_{6\times 6} \\ 0_{6\times 6} & I_{6\times 6} \end{pmatrix} \tag{4.28}$$

where $\text{Ad}_{\Delta_t \mathrm{v}_{t-1}}$ is the adjoint transformation corresponding to the kinematic update based on the estimated velocity (see Section 3.2.2 for the definition of the adjoint transformation).

We can give an intuition of the role of the adjoint transformation in this linearization. In Chapter 3 we introduce the adjoint transformation as an operation to transform twist velocities from one reference frame to another. In the linearization of the first forward model,

the adjoint is playing another role. Here, the adjoint transforms the uncertainty about the pose of the body from the frame of the previous estimate to the frame of the predicted body pose. In general, when we transform a random 6D pose that we assume Gaussian distributed with mean p and covariance defined in the tangential Lie space $\Sigma_p \in \mathbb{R}^{6\times6}$ applying a second pose p', the result will not be Gaussian distributed. However, we can approximate the result to Gaussian distribution of mean $p_{new} = p \oplus p'$ and covariance $\Sigma_{new} = \mathrm{Ad}_p \Sigma_p \mathrm{Ad}_p^T$, correct to first order (Barfoot & Furgale, 2014). Observe the similarity between this equation and the Equation 3.34 of the EKF, which is now transforming the uncertainty about the pose of the rigid body to the predicted new location.

The linearization of the second forward model (assuming an abrupt break event in the motion) corresponds to the following Jacobian matrix:

$$F_t^{rbm,II} = \left.\frac{\partial f^{rbm,II}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,II}} = \begin{pmatrix} I_{6\times6} & 0_{6\times6} \\ 0_{6\times6} & 0_{6\times6} \end{pmatrix} \tag{4.29}$$

In practice, what the RBF does if the second forward model generates the best predictions is to consider only the pose of the rigid body, p_t, as the state of the body and correct it.

The third forward model (based on information from the next-higher level) is independent of the state of the filter. To correct the state we use the linearization of the first forward model, $F_t^{rbm,III} = \left.\frac{\partial f^{rbm,I}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,III}}$, using the prediction from the higher level as a different point for the linearization. If this point is closer to the true mean of the posterior, the result of the EKF correction based on the third prediction generates a better approximation of the true current state.

The linearization of the measurement model with respect to the state yields the following Jacobian matrix:

$$H_t^{rbm} = \left.\frac{\partial h^{rbm}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,i}} = \begin{pmatrix} H_t^{f^0,i} \\ H_t^{f^1,i} \\ \vdots \\ H_t^{f^M,i} \end{pmatrix} \tag{4.30}$$

where $H_t^{f^m,i}$ correspond to the linearization of the model for an individual feature of the form around the best predicted state from the model $i \in \{I, II, II\}$:

$$H_t^{f^m,i} = \begin{pmatrix} & 0 & \mathrm{Loc}_z(\hat{f}_t^m)^i & -\mathrm{Loc}_y(\hat{f}_t^m)^i & \\ I_{3\times3} & -\mathrm{Loc}_z(\hat{f}_t^m)^i & 0 & \mathrm{Loc}_x(\hat{f}_t^m)^i & \mathbf{0}_{3\times6} \\ & \mathrm{Loc}_y(\hat{f}_t^m)^i & -\mathrm{Loc}_x(\hat{f}_t^m)^i & 0 & \end{pmatrix} \tag{4.31}$$

Sequential processing of measurements: In the equations of the EKF (see Equation 3.42) we observe that the covariance of the innovation, $S_t = H_t \hat{P}_t H_t^T + R_t$, has to be inverted to compute the Kalman gain. In our case this involves the inversion of a $3M \times 3M$ matrix, with M the number of features assigned to the RBF. If M is large (many features are associated to the rigid body) the matrix inversion can be computationally expensive and affect the online capabilities of our system. Therefore, based on the assumption that the measured feature locations are uncorrelated to each other, we process the point features sequentially and avoid the costly inversion of the full matrices (Bar-Shalom et al., 2001).

In the sequential procedure we initialize the corrected state and its covariance with $\boldsymbol{x}_t^0 = \hat{\boldsymbol{x}}_t$ and $P_t^0 = \hat{P}_t$. We then compute a correction based on each feature m. For the feature m, the corrected state becomes (we dropped some super-indices to make the equations easier to read):

$$x_t^m = x_t^{m-1} + K_t^m(\text{Loc}(f_t^m) - \text{Loc}(\hat{f}_t^m)^i) \tag{4.32}$$

$$P_t^m = (I - K_t^m H_t^{f^m}) P_t^{m-1} \tag{4.33}$$

$$K_t^m = P_t^{m-1}(H_t^{f^m})^T [H_t^{f^m} P_t^{m-1} (H_t^{f^m})^T]^{-1} \tag{4.34}$$

In Equation 4.32, the first part of the state vector x represents a 6D pose in Euclidean space. To correctly integrate the corrections of this first part of the state vector we use the composition of poses, $\oplus$, instead of the normal vector sum.

The final correction is then:

$$x_t = x_t^{M-1} \tag{4.35}$$

$$P_t = P_t^{M-1} \tag{4.36}$$

Recursive Bayesian Estimation of Multi-Body Motion

To track the motion of multiple rigid bodies, we have to match point features to the corresponding rigid body RBF and use them to update the state of the filters. This matching process is called data-association. We associate features to those rigid bodies that best predict their motion. We measure the Euclidean distance between the observed and the predicted feature location from the filters and assign the features to the filter with the lowest prediction distance. We assume that the existing filters cannot predict the motion of a feature if all predicted locations are further than d_{max}^f from measured location.

If the motion of a set of features cannot be accurately predicted by any of the existing rigid body Bayes filters it could be necessary to instantiate a new filter. We instantiate a new filter if a set of non-assigned features move coherently. To evaluate if a set of non-assigned features move coherently we use RANSAC and try to find a rigid body transformation describing their motion. If a rigid body transform explains the motion of at least f_{min} features, a new RBF is created using this rigid body transform as the initial state. Based on this procedure the proposed system works for an arbitrary number of moving rigid bodies in the scene, as long as f_{min} visual features can be tracked on each body.

The overall perceptual process begins with a single Bayes filter that represents the static background. We assume that the static background does not move ($\eta_{bg} = 0_{6\times1}$), although it would be easy to integrate an algorithm that provides motion estimates of the camera with respect to the static background (Nistér et al., 2004, Forster et al., 2014). New detected point features are initially assigned to the static rigid body, until they begin to move and their location cannot be predicted by the static background filter. These features are either assigned to another filter (if it can predict their motion) or used to create a new filter for a newly moving body.

4.2.3 Recursive Bayesian Estimation of Kinematic Model

The third level of our system estimates and tracks the kinematic model of the scene, based on the motion of rigid bodies obtained on the next-lower estimation level. We assume a pair of rigid bodies to be related in one of four possible ways: *(i)* prismatic joint, *(ii)* revolute

joint, *(iii)* rigid connection, or *(iv)* disconnected, the latter being a special case defined as the absence of relationships *(i)–(iii)*. We model these relationships with different types of RBF, each type modeling the necessary parameters for that relationship (joint axis, joint variable, etc.) in the random state variable $\boldsymbol{x}_t^{joint}$. We instantiate and maintain one RBF of every type for each pair of rigid bodies in the scene.

The measurements $z_t^{joint} \in \mathbb{R}^6$ are obtained from the next-lower estimation level and correspond to the change in relative pose (in exponential coordinates) between the two rigid bodies attached to the joint, defined with respect to one of the bodies:

$$\boldsymbol{z}_t = {}^{parent}_{child}\Delta p_t + \boldsymbol{v}_t^{joint} = {}^{parent}_{child}p_t \ominus {}^{parent}_{child}p_{init} + \boldsymbol{v}_t^{joint} \tag{4.37}$$

The body that acts as reference is called *parent link*, and the second body is called *child link*. In the previous equation $\ominus$ represents the subtraction between poses, ${}^{parent}_{child}p_t$ is the current pose of the child link with respect to the parent link (in parent link frame), and ${}^{parent}_{child}p_{init}$ is the pose of the child link with respect to the parent link when the joint starts to be tracked. The terms parent and child link refer to the common terminology for tree structures of kinematic mechanisms in the literature.

The covariance, R_t^{joint}, of the measurement model noise, $\boldsymbol{v}_t^{joint}$ is also obtained from the next-lower level:

$$R_t^{joint} = {}_{parent}Ad(\Sigma_{{}_{parent}p})\ {}_{parent}Ad^T + {}_{parent}Ad(\Sigma_{{}_{child}p})\ {}_{parent}Ad^T \tag{4.38}$$

As explained in Section 4.2.2, we are applying the adjoint operator here to transform covariances between reference frames (Barfoot & Furgale, 2014). In this case, we transform the covariance of the pose of both parent and child link from the sensor reference frame to the reference frame of the parent link.

In the following we will explain the state representation, prediction, measurement update and EKF solution of the three different RBF types. Each RBF type uses a different kinematic prior which defines its state and measurement model. As before, the priors enable the estimation and tracking of kinematic models, but also the prediction of the next state of the next-lower level (feedback). Following this, we explain how we estimate the most likely joint type between two bodies and the overall kinematic structure, which completes the description of this estimation level.

Prismatic Joint Estimation

The state of a prismatic joint is parametrized by the orientation of its axis (azimuth ϕ and elevation θ in spherical coordinates), its joint variable $q^p \in \mathbb{R}$ (translation along the joint axis), and the velocity of the joint variable $\dot{q}^p \in \mathbb{R}$, which we represent with a multidimensional Gaussian distributed random variable. In the prediction step, we use the joint velocity to update the joint state.

To predict the change in pose of the child link relative to the parent link, we use the following measurement model:

$$\hat{z}_t^{joint,p} = \begin{pmatrix} 0_{3\times 1} \\ q^p \cdot o^p \end{pmatrix} \tag{4.39}$$

where $0_{3\times 1}$ is a three dimensional null vector that indicates that the orientation between the links is constrained by the prismatic joint, and $o^p \in \mathbb{R}^3$ is the axis orientation (unit vector) estimated from ϕ and θ as $o^p = (\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta))^T$.

Extended Kalman Filter for Prismatic Joint Estimation While the forward model of the prismatic joint is linear, the measurement model is non-linear with respect to the joint parameters. Therefore, we implement an EKF to correct the state based on the acquired measurement.

The matrix of derivatives of the measurement model with respect to the state, $H_t^{joint,p}$, is defined by:

$$H_t^{joint,p} = \begin{pmatrix} -q^p \sin(\phi)\sin(\theta) & q^p \cos(\phi)\cos(\theta) & \cos(\phi)\sin(\theta) & 0 \\ q^p \cos(\phi)\sin(\theta) & q^p \sin(\phi)\cos(\theta) & \sin(\phi)\sin(\theta) & 0 \\ 0 & -q^p \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.40}$$

where we have made use of the equivalence between spherical (ϕ and θ) and Cartesian (o^p) representations of the orientation vector.

Revolute Joint Estimation

The state of a revolute joint is parametrized by the orientation of its axis (azimuth ϕ and elevation θ in spherical coordinates), a point on the axis $p^r \in \mathbb{R}^3$, its joint variable $q^r \in \mathbb{R}$ (rotation about the joint axis), and the velocity of the joint variable $\dot{q}^r \in \mathbb{R}$, which we represent with a multidimensional Gaussian distributed random variable. We use the joint velocity to predict the next joint state as process model.

To predict the change in pose of the child link relative to the parent link, we use the following measurement model:

$$\hat{z}_t^{joint,r} = \begin{pmatrix} q^r \cdot o^r \\ t^r \end{pmatrix} \tag{4.41}$$

where $o^r \in \mathbb{R}^3$ is the axis orientation (unit vector) estimated from ϕ and θ and $t^r = (-q^r \cdot o^r) \times p^r$ is the linear relative motion between rigid bodies.

Extended Kalman Filter for Revolute Joint Estimation As in the case of a prismatic joint, the filter for the revolute joint model contains a non-linear measurement model. We implement an EKF that linearizes this model and corrects the state from the acquired measurement. The linearization of the measurement model, $H_t^{joint,r}$, is defined as:

$$H_t^{joint,r} = \begin{pmatrix} -q^r c_\phi s_\theta p_z^r & -q^r(s_\theta p_y^r + s_\phi c_\theta p_z^r) & 0 & \dots \\ -q^r s_\phi s_\theta p_z^r & q^r(c_\phi c_\theta p_z^r + s_\theta p_x^r) & 0 & \dots \\ q^r s_\theta(c_\phi p_x^r + s_\phi p_y^r) & q^r c_\theta(s_\phi p_x^r - c_\phi p_y^r) & -q^r c_\theta & \dots \\ -q^r s_\phi s_\theta & q^r c_\phi c_\theta & q^r s_\phi s_\theta & \dots \\ q^r c_\phi s_\theta & q^r s_\phi c_\theta & 0 & \dots \\ 0 & -q^r s_\theta & 0 & \dots \end{pmatrix}$$
$$\begin{pmatrix} \dots & q^r c_\theta & -q^r s_\phi s_\theta & c_\theta p_y^r - s_\phi s_\theta p_z^r & 0 \\ \dots & 0 & q^r c_\phi s_\theta & c_\phi s_\theta p_z^r - c_\theta p_x^r & 0 \\ \dots & -q^r c_\theta & -q^r c_\phi s_\theta & s_\theta(s_\phi p_x^r - c_\phi p_y^r) & 0 \\ \dots & 0 & 0 & c_\phi s_\theta & 0 \\ \dots & 0 & 0 & s_\phi s_\theta & 0 \\ \dots & 0 & 0 & c_\theta & 0 \end{pmatrix} \tag{4.42}$$

where $s_\phi = \sin\phi$, $c_\phi = \cos\phi$, $s_\theta = \sin\theta$, and $c_\theta = \cos\theta$, and $p^r = (p_x^r, p_y^r, p_z^r)^T$.

Rigid Joint Estimation

A rigid joint does not allow for relative motion between rigid bodies. Therefore, it has no parameters nor variables to estimate. The measurement model of a rigid joint predicts that there is no change in the relative pose between bodies, i.e. $\hat{z}_t^{joint,rigid} = 0_{6\times1}$. Because there are no parameters to estimate, we do not need to implement an EKF for this type of joint.

Recursive Bayesian Estimation of Multi-Type Kinematic Model

After evaluating the RBF of every type for each pair of rigid bodies, we select the RBF that is most consistent with the observed rigid body motion. We base this selection on the likelihood of the measurements given the estimated models. The likelihood of the observed data is defined as

$$p(z_t^{joint}|M, \boldsymbol{x}_t^{joint,M}) = \mathcal{N}(z_t^{joint}; \hat{z}_t^{joint,M}, \hat{R}_t^{joint,M}) \tag{4.43}$$

where M are the considered joint models, $M \in \{\text{Prism}, \text{Rev}, \text{Rigid}\}$, $\boldsymbol{x}_t^{joint,M}$ is the current estimate of model M, and $\hat{z}_t^{joint,M}$ and $\hat{R}_t^{joint,M}$ are the predicted measurement mean and covariance, respectively.

Instead of selecting the model that best explains only the latest measurement, we select the one that explains all past measurements. This makes the selection of the most likely joint more stable. We consider that large measured relative motion is more informative to find the most likely joint, since large motions are more difficult to predict randomly. Therefore, we assign a weight to the estimated likelihood at each step proportional to the amount of change in relative motion between links, measured as the norm of the vector of exponential coordinates $\|z_t^{joint}\|$, and compute the mean of these weighted likelihood values over the trajectory.

We select the model with the maximum accumulated weighted likelihood as the joint that best explains the motion between a pair of bodies. We consider that none of the models can explain the motion with sufficient reliability if any of their accumulated weighted likelihoods is over a minimum threshold, L_{disc}. In this case, we declare this pair of rigid bodies to be *disconnected*.

From all pairwise selected joint types and parameters, we build the kinematic model of the scene. Because joints are always determined considering only pairs of rigid bodies, our system can naturally determine the kinematic model of branching mechanisms and closed kinematic chains.

4.3 Experiments

We conducted four sets of experiments. In the first set we study the sensitivity of the system to the number of tracked features, N. We evaluate if the computation time, the accuracy and the robustness depend on N. To measure the accuracy we compute the error between the estimated rigid body poses and the ground truth obtained with a motion capture system (Motion Analysis, 2017). We evaluate also the contribution of the predictions from higher levels in the performance of the system.

In the second set of experiments we evaluate the performance of the online IP system with different articulated objects. We measure the robustness, quality, and convergence of the kinematic model estimation by comparing to ground truth. To obtain the ground truth for the joint parameters, we placed artificial markers that are not used by the system to estimate the kinematic model. We then manually measured the joint parameters in the RGB-D stream using the markers.

Parameters for the Estimation of Kinematic Properties		
Parameter	Description	Value(s)
N	Number of tracked features	150–250*
λ_{min}	Threshold for the smallest eigenvalue of the second order moments of the gradient around a point feature	0.005
σ^2_{min}	Minimum standard deviation of the feature location measurements	1 cm
α_λ	Constant factor for the uncertainty of a feature due to its saliency	5×10^{-4}
α_z	Constant factor for the uncertainty of a feature due to its depth	$2.58\,\mathrm{mm/m^2}$
a^2_x, a^2_y, a^2_z	Linear acceleration noise in rigid body motion estimation	$0.02\,\mathrm{m\,m^{-1}}$
$a^2_{rx}, a^2_{ry}, a^2_{rz}$	Angular acceleration noise in rigid body motion estimation	$0.2\,\mathrm{m\,m^{-1}}$
d^f_{max}	Maximum prediction error for the feature-to-body data association	1.5 cm
L_{disc}	Minimum likelihood for joint models	0.1

Table 4.1: Parameters in our system for the online estimation of kinematic models (* indicates the selected values after their evaluation)

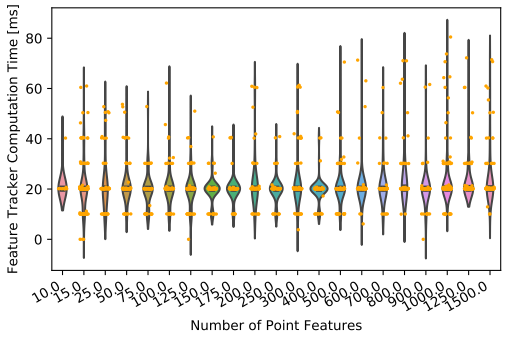
In the third set of experiments, we test our system in scenarios were offline systems fail. And in the fourth set, we make use of the online abilities of the system to control the motion of a robot, closing the loop between perception and action. This demonstrates that the perceived information is relevant for the robot manipulation of DoF.

In all experiments, the input is an RGB-D stream, provided either by a Kinect or a Carmine RGB-D sensor. The articulated objects are of different size, color, texture, and with different kinematic structures (number and type of joints). The only constraint for the objects is that they have some visible texture. We also vary lighting conditions and the relative pose between the objects and the sensor. In these experiments we use N between 150 and 250. Our system computes at a frame rate of 30 frames per second, running on real-time on an Intel Xeon E5520 PC at 2.27 GHz. Table 4.1 contains the value of the most relevant parameters used in the experimental evaluation.

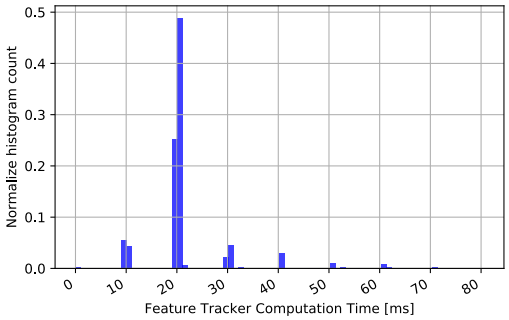
4.3.1 Parameter Sensitivity Analysis

The computation complexity in our system increases with the number of features N. This parameter is involved in the feature motion and the rigid body motion estimation. Therefore, we first evaluate if the number of features influences the computation time at these levels.

Figure 4.6 shows the computation time of the feature motion for different values of N. We observe that the computation at this level is independent of the number of tracked features. In most of the iterations our system spends approximately 20 ms in the computation of the motion of the features. This time includes the tracking of the features with the KLT algorithm (Tomasi & Kanade, 1991) and the detection of new features to maintain their number on N using the approach by Shi & Tomasi (1994). If the detection process does not generate enough new features, our system retries to detect features. Each detection increases the time by approximately 10 ms. In some iterations our system tracks successfully all features and does not need to detect new features. This iterations last approximately 10 ms. The com-

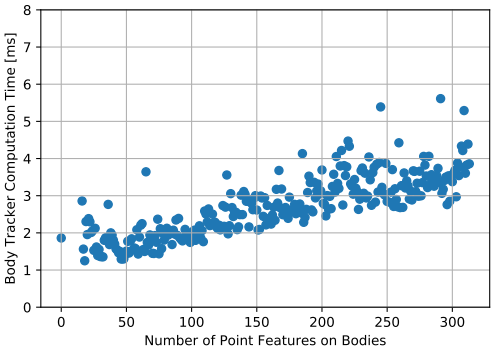


(a) Distribution of computation time as function of the number of tracked features, N; orange dots indicate the computation time of each iteration

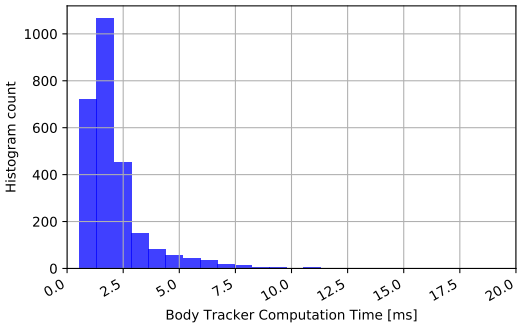


(b) Histogram of computation times; most iterations require around $20\,\mathrm{ms}$; the detection of new features requires around $10\,\mathrm{ms}$ that cause the periodically spaced peaks

Figure 4.6: Computation time of the feature motion level; the time is independent of the number of features N; most of the iterations consume approximately $20\,\mathrm{ms}$ to track and detect features to maintain N; each additional detection process adds approximately $10\,\mathrm{ms}$



(a) Computation time for the rigid body motion level at each iteration and associated number of features



(b) Histogram of computation times at the rigid body level

Figure 4.7: Computation time of the feature motion level; the time slowly increases with the number of features assigned to a rigid body; most of the iterations consume approximately $2\,\mathrm{ms}$

putation time of this level allows us to estimate at 30 fps (approximately 33.33 ms between frames).

Figure 4.7 depicts the computation time of the rigid body motion for different values of assigned features. The computation time increases from 2 ms to 4 ms when the associated features increase from 30 to 320. These times do not restrict the performance of the system.

Until now, we did not find any strong limitation on the number of tracked features N from the analysis of the computation times. However, if the number of tracked features is high, some of them are placed in less distinctive locations. Features in non-distinctive locations will

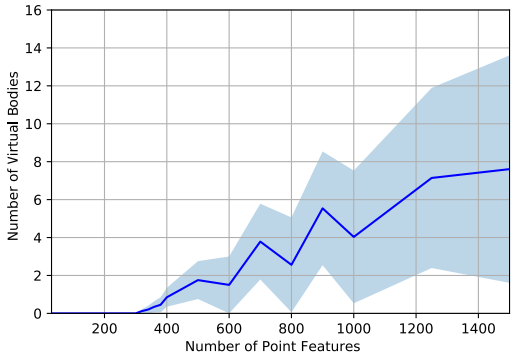


Figure 4.8: Generation of virtual (wrong) rigid body hypotheses as a function of the number of tracked features N; imposing a large number of features to track increases the amount of noisy trajectories and the probability of creating a virtual body hypothesis

produce noisy trajectories. When the proportion of noisy trajectories grows, subsets of features will randomly move in a coherent manner and create virtual rigid bodies. Even though the "life-span" of this virtual bodies is short, we consider them harmful for the perceptual process and we will try to reduce their appearance. Figure 4.8 depicts the number of virtual (error) rigid bodies created as function of the tracked features N. When N is under 300 there are almost no virtual bodies. Over this value, the number of virtual bodies increases.

In a last experiment, we evaluate the contribution of the predictions from other levels to the overall performance of the system. To evaluate this contribution, we compare the accuracy in the estimated rigid body motion in the fully integrated system (with predictions) versus the system without predictions from kinematics to rigid body motion estimation, or without predictions from rigid body motion to feature motion estimation. In the experiment the system perceives the motion of a drawer from human interactions during 6.2 s. The ground truth of this motion is obtained with a motion capture system Motion Analysis (2017). Figure 4.9 shows the results of the experiment.

The combined system using all predictions outperform the other two variants. Significantly, the predictions about next feature locations help to reject noise, especially when the number of tracked features is large. In this case, many of the features are noisy and of low quality: the predictions from the rigid body level helps to reject them and track them more stable.

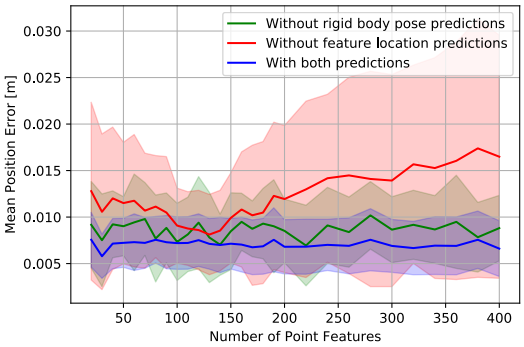
4.3.2 Experimental Evaluation

We measured the accuracy and convergence of our online IP system for kinematic properties on four articulated objects. Figure 4.10 shows initial, intermediate (after 1 s), and final frames of these experiments. The figure also includes graphs of the estimation error including estimated uncertainty over time. In some of the experiments, the observed motion was produced by human interaction, in some by a robot interacting with the environment, and in some the environment moved autonomously. We recorded all interactions and made them publicly available[2]. In the following, we discuss each of the experiments from Figure 4.10.

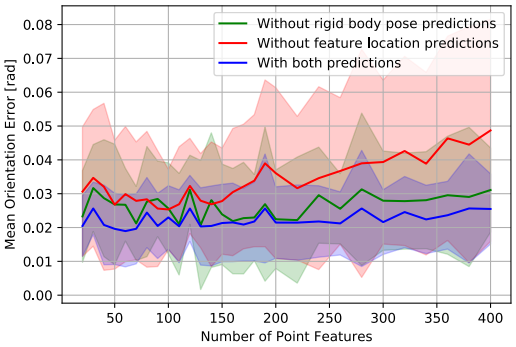
Book Experiment

The book is opened 60° and closed again (120° of accumulated motion) in 14 s. The joint is correctly classified from the first frame and converges within 1 s to a stable set of parameters. Point features are correctly assigned to the moving book cover. The error remains under 4°

[2] https://tinyurl.com/onlineIPdata



(a) Mean error in position over the entire video sequence (6.2 s) for different number of tracked features



(b) Mean error in orientation over the entire video sequence (6.2 s) for different number of tracked features

Figure 4.9: Error in the estimated rigid body pose; the nominal system (with predictions from kinematics to rigid motion estimation and from rigid motion to feature motion estimation) outperforms the two variants without one of the two predictions

for the orientation and under 2 cm for the position of the revolute axis. We used artificial markers to obtain the ground truth of the revolute axis.

Umbrella Experiment

The umbrella is extended by 40 cm in a motion lasting 10 s. The joint is continuously estimated correctly as prismatic. The features on the umbrella are correctly assigned. Some features on the hand are also assigned to the umbrella since they move coherently with it. The error of the estimated joint axis remains under 5° during the entire experiment. We used artificial markers to obtain the ground truth of the prismatic axis.

PUMA 560 Experiment

In a motion lasting 15 s, the shoulder joint of the PUMA 560 robot moves 90° and the elbow joint moves 140°. Initially, our system detects both links as a single moving rigid body. When the motion of the two links of the robot arm is different enough (0.7 s), the system succeeds at separating them. Once both moving rigid bodies are detected, the features are correctly assigned. The revolute axis between base and upper arm and the revolute axis between upper arm and forearm are quickly classified as revolute, and their parameters converge fast to a stable accurate value. The joint between the base and the forearm is initially classified as revolute, but the system quickly detects that there is no direct connection (disconnected joint). The estimation error of the first revolute axis (shown in the graph) remains under 6° for orientation and 5 cm for position; for the second joint the error remains under 8° and 8 cm after convergence. The estimates of joints connecting two moving bodies are usually less accurate, as the errors in motion estimation for both bodies add up. The robot does not have sufficient texture to reliably track features at this distance; we attached checkerboards to it to remedy this problem. In this experiment, the RGB-D sensor is pointing parallel to the joint

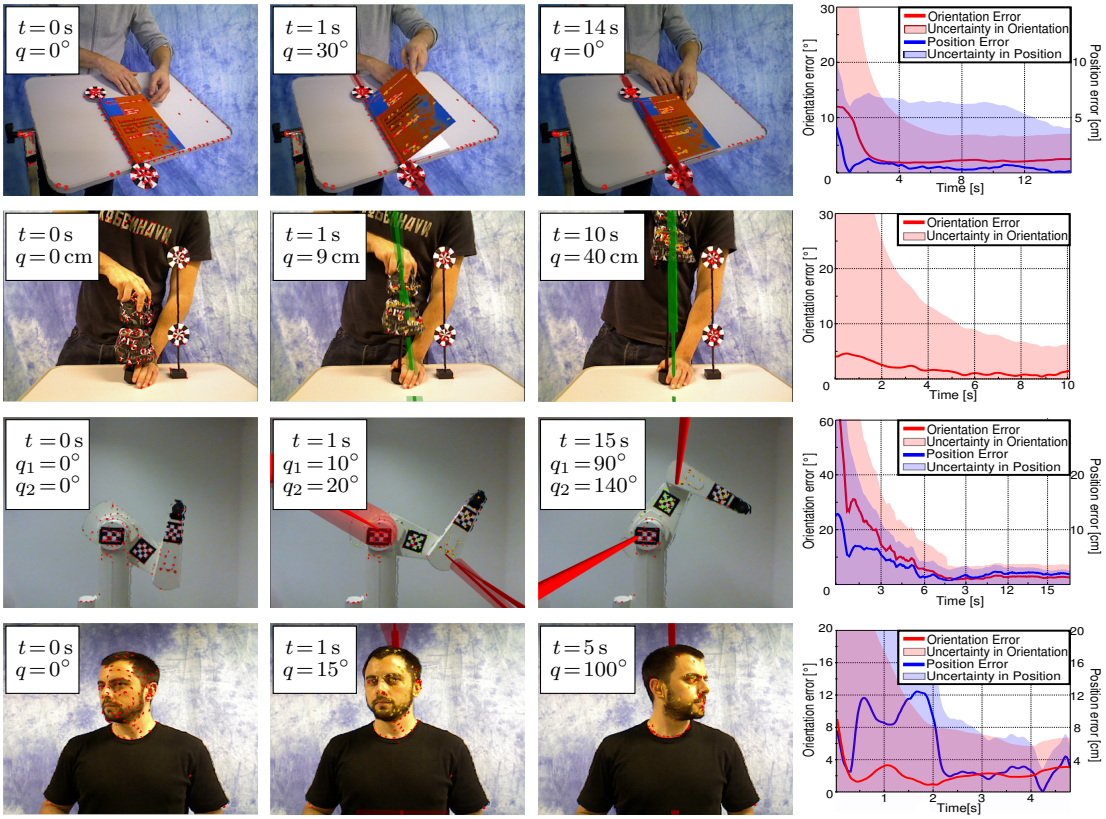


Figure 4.10: Experiments with online IP (each row represents a different experiment): initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model, including error plot (fourth column) of joint configuration estimation, relative to ground truth, including uncertainty (shaded areas); the insets in the three images show the time t and the estimated joint variable q; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation; red dots are features assigned to the static background; dots of other colors are features assigned to moving rigid bodies [© 2014 IEEE]

axes of the robot to simplify ground truth estimation. The experiments demonstrates the system's ability to determine multiple DOF of a kinematic chain at the same time.

Human Head Experiment

The system estimates the neck joint of a human shaking his head. The human rotates his head 100° in 5 s. The joint is correctly classified from the beginning of the motion, all features are correctly assigned, and the error of the axis after convergence remains under 5° and 4 cm. The RGB-D sensor is pointing perpendicular to the orientation of the joint to simplify ground truth estimation. The joint position is manually measured in the point cloud. This experiment demonstrates the performance of the system on large semi-rigid articulated bodies.

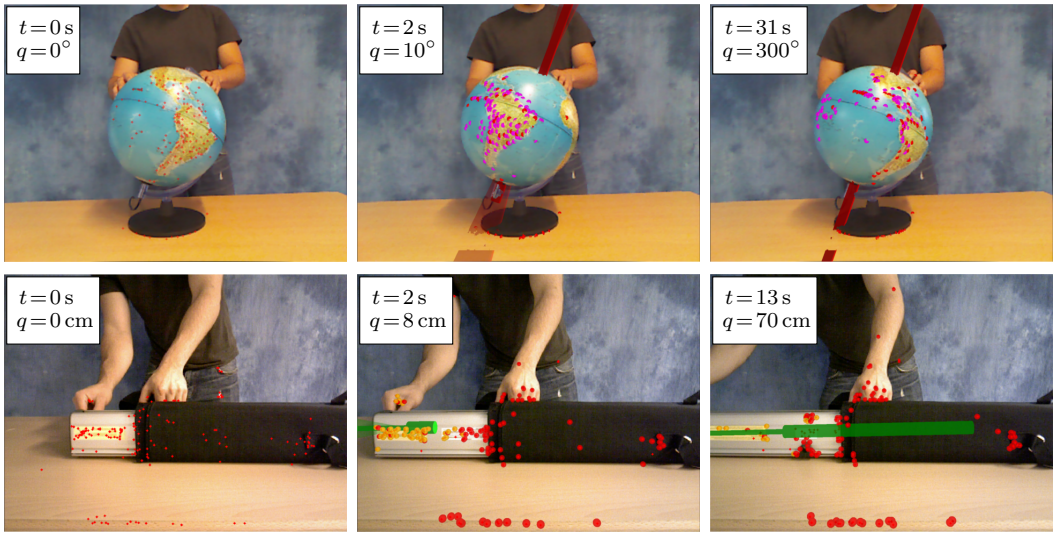


Figure 4.11: Experiments on a failure case of previous approaches: point features disappear from the view due to occlusions and/or large displacements; each row represents a different experiment; initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model; the insets show the time t and the estimated joint variable q; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation [© 2014 IEEE]

4.3.3 Failure Cases of Previous Offline Algorithms Solved With Online IP

In this section, we show three situations that can only be handled by an online incremental IP system. Existing offline methods would fail in the following scenarios.

Disappearing Features

The motion of the object may cause all features obtained at the beginning of the motion to disappear by moving out of visual field or simply due to tracking error. Offline IP methods would fail, as they cannot find matching features between the initial and the final frame. We use a rotating globe and a portable projection screen with casing (see Figure 4.11) to demonstrate that the incremental nature of our online IP system aims to overcome this problem. We rotate the globe 300° in 31 s and open the poster hanger 70 cm in 13 s. Our online system quickly detects the moving bodies and incrementally assigns new features to them as they appear. This allows us to successfully track the motion of the rigid body, even when the initially visible parts of the object get obstructed (globe) or leave the field of view (projection screen).

Appearing Objects

The articulated object may not be visible at the beginning of the analysis. To demonstrate how our online IP system can address this, we use a book in a cabinet and a Pioneer mobile base (see Figure 4.12). The cabinet has to be opened to perceive the book. We then open the book 30° in 3 s. Once the book is visible, new features are detected on its surface, and

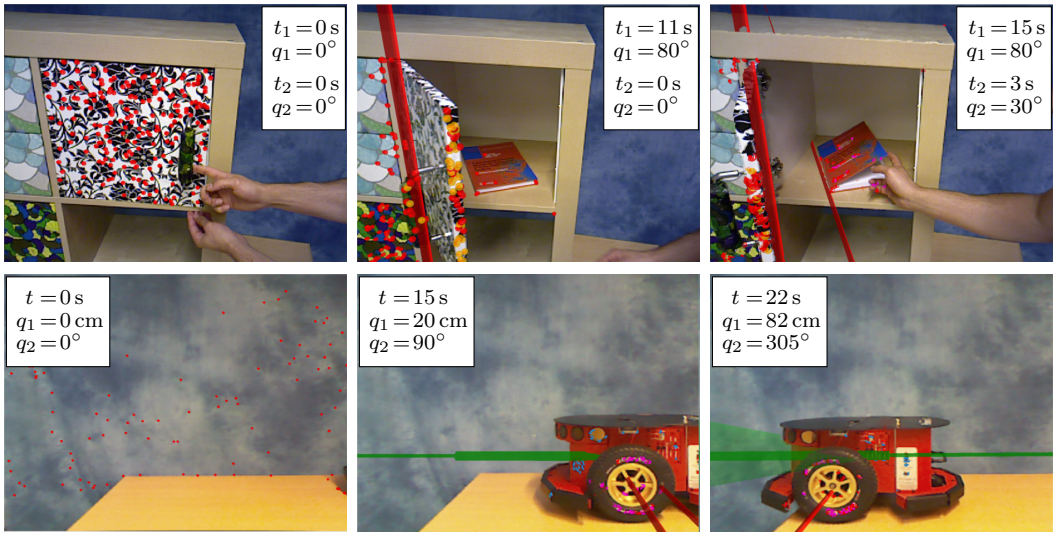


Figure 4.12: Experiments on a failures case of previous approaches: objects are not visible at the beginning of the interaction; each row represents a different experiment; initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model; the insets show the time t and the estimated joint variable q; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation [© 2014 IEEE]

the joint can be perceived when the book is opened. The Pioneer base enters the field of view from the right. The base moves 82 cm in 22 s after entering the scene. The revolute joint connecting the wheel to the base as well as the prismatic joint between the robot base and the background are correctly estimated. At the end of the experiment the uncertainty about the prismatic joint increases because the robot base slightly changes its orientation.

Identical Initial and Final Configuration

When the initial and final configuration of the object performing the motion are identical, a comparison of these poses will not reveal information about the kinematic model. To show that our online IP system overcomes this problem of some offline IP methods, we experiment with a cabinet door and a drawer. The drawer is opened and closed (50 cm of accumulated motion) in 6 s, and the door is opened and closed (80° of accumulated motion) in 7 s. The proposed online IP system estimates accurately the kinematic model. The model remains converged after the object returns to its initial configuration.

4.3.4 Monitoring Interaction With Online IP

One of the main advantages of an online IP system is the ability to use the kinematic model to control the robot's interaction with the environment. By using the information for an ongoing interaction we demonstrate that the perceived information is relevant for the mechanical manipulation of DoF. In this section, we demonstrate the utility of online perception in two experiments with two objects each (door and drawer). The goal of the first experiment is to obtain a kinematic model with a specified uncertainty bound (5° in orientation and 5 cm in

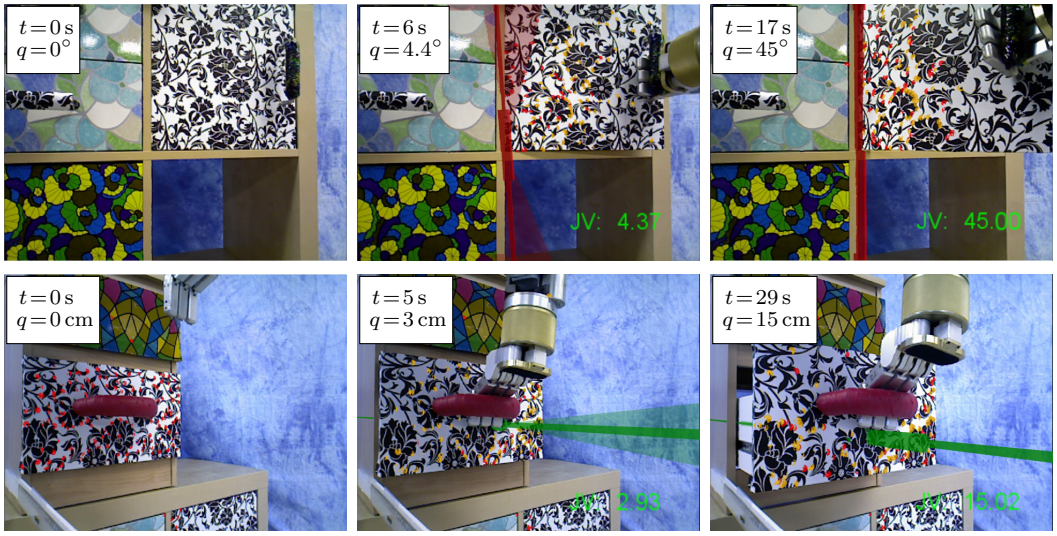


Figure 4.13: Experiments on the usability of the online perceived kinematic model to steer robot manipulation of DoF; each row represents a different experiment; initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model; the insets show the time t and the estimated joint variable q; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation [© 2014 IEEE]

position of the joint axis). The goal of the second experiment is to move one of the joints to a specific configuration. Each experiment is repeated ten times. Figure 4.13 shows initial, intermediate, and final frames of two trials of these experiments, with the online estimated joint variable in the bottom right corner.

In the first experiment we measure the amount of interaction necessary for the system to reduce the uncertainty below a required level, and the deviation of the estimated kinematic model to ground truth (manually measured in the point clouds). In the case of the drawer, our controller stops, due to the attained uncertainty bounds, after a mean amount of motion of 5.07 cm. The mean error of the estimated axis is 4° with a single value above 5° (5.08°). In the case of the door our controller stops due to the attained uncertainty bounds after a mean amount of motion is 8.4°, with a maximum value of 26°. The mean error of the estimated axis is 2.95° with a maximum of 4.47°. The mean error in the estimated joint axis position is 7.03 cm with a maximum of 49.71 cm for a failed trial. Without this value the mean position error is 2.28 cm (under the 5 cm threshold).

In the second experiment, the robot manipulates the same objects as before so as to attain a certain value of a joint variable. In the case of the drawer, this value is 15 cm. The robot stops, when its model indicates this amount of motion. We measure the ground truth motion manually. The mean value of the measured joint value is 15.55 cm and the maximum and minimum are 15.9 cm and 15.2 cm, respectively. In the case of the door, the desired joint configuration is 45°. The mean value of the measured rotation is 44.8° with a minimum of 44° and a maximum of 46°.

The results of these experiments demonstrate that our online IP can be used to monitor and control interactions with articulated objects in the environment. We showed that it is possible to adjust the robot's action based on a desired uncertainty bound for the accuracy

during the estimation of a kinematic model. This demonstrates that the estimated uncertainty reflects the correctness of the estimated kinematic structure. We also showed that the online estimation of joint values can be used to monitor and attain manipulation goals, expressed in terms of specific joint configurations.

4.4 Discussion and Limitations

We will begin this section by discussing the strengths and limitations of the presented online IP system. We will discuss them in the context of the four opportunities for perception for robot manipulation of DoF presented in Chapter 1 (OP1-OP4). We will also discuss whether the system overcomes the challenges for perception (CH1-3), and possible future directions and extensions.

Exploiting Interactions (OP1) Our system depends on motion to perceive the kinematic model. This dependency is solved by the interactions from the robot or from another agent. These interactions create information-rich sensor signals and reveal the underlying motion constraints. Thus, our system exploits the information revealed by an interacting agent. However, due to the lack of the necessary interactive models, our system cannot fully exploit information about the concrete interaction that create the sensor signals. We will address this limitation later in this thesis in Chapter 6.

Another limitation of the system is that the interactions have to be dexterous enough to cause motion in the articulated object. In our robot experiments we overcome this limitation teaching the robot kinesthetically how to actuate the mechanism. We will reduce the dependency on fully taught robot trajectories implementing force/torque impedance controllers for our robot in Chapter 6. And in Chapter 7 we will propose a method for the robot to generate and select the interactions that promise to reveal most information.

Exploiting Physical Priors (OP2) Each level of our perceptual system is based on task-specific priors encoding physical regularities: spatial consistency and projective geometry for the estimation of feature motion, physics of rigid bodies for the estimation of body motion, and kinematics of articulated objects to build and update the kinematic model. Based on these physical priors our system interprets the input signals as evidences of a known underlying physical process.

A limitation of this approach (and of the system) is that the environment should be well represented by the physical priors. This restricts the application of our approach to objects composed of rigid (or semi-rigid) bodies, since we exploit priors about rigid body physics. Perceiving motion in non-rigid bodies require other types of models, like physics models for liquids (Schenck & Fox, 2017) or motion fields (Newcombe et al., 2015).

Another way of overcoming this limitation on predefined and accurate physical priors is to extract the physical models from the sensor data with machine learning techniques. The physical priors we use in our system allow us to 1) predict the changes in the environment from previous states and interactions, and 2) correlate this changes to sensor signals. Learning these types of forward and measurement models (or a combination thereof) is a currently active and very promising research field. An example of this idea for the context of kinematic models can be seen in the work by Sturm et al. (2011), where one of the joint models is a Gaussian process that can predict body motion from the previously seen data. The system presented in this chapter does not show such an adaptive behavior.

Exploiting Temporal Structure (OP3) Recursion is a crucial element of each of the levels of our solution. The perceptual system restricts the space of possible solutions for each newly acquired measurement based on the results of the previous analysis. Combining temporal and with physical priors, the system can predict the upcoming events and adapt the perceived models when the sensor signal arrives, balancing the correction to the relative reliability between the predictions and the measured signals. Temporal recursion is a crucial element to the online capabilities of our system.

Exploiting Interdependencies between Perceptual Subtasks (OP4) At the core of our proposed perceptual system is a factorization of the original problem into subproblems that can be solved with recursive estimation, and their synergistic interconnection. Each of this subproblems represents a perceptual subtask. Information flowing bottom-up (input measurements) and top-down (predictions) enabled the online capabilities and contributed to the robustness and accuracy of our system by reusing priors of one level into the other levels.

In our first set of experiments we evaluated the contribution of the predictions to the overall performance of the system. The predictions of rigid body poses from the kinematic model, and of the feature locations from the rigid body motion, help to reject noise and stabilize the estimation at all levels.

A current caveat of our approach is that finding the right factorization (and what to represent and how to represent it) is a human engineering process. We think that this difficult design task is a limitation to apply our general approach for perception to other problems. Current approaches in artificial perception has shown improved performance applying machine learning techniques to find the most suited representation for a perceptual task (Krizhevsky et al., 2012b).

Technically, the proposed factorization and representation present also limitations. For example, we chose deliberately to not represent and update a "map" of each rigid body, avoiding the full SLAM problem. Maintaining several maps that grow and shrink dynamically (based on the association of feature) is a complex problem. However, representing the map in the state could have benefits, e.g. a possible improvement in accuracy and robustness by using the interdependencies between feature locations in the map that we assume independent. Extending our RBF for rigid body motion to full SLAM solutions would be a promising extension.

We will now conclude this chapter discussing whether the proposed online IP system achieves the goal of extracting information that is relevant to support robot mechanical manipulation of DoF, and to what degree the system addresses the three challenges in perception for robot manipulation (CH1-3) presented in Chapter 1.

Applying the Information for Manipulation In our experimental section (Section 4.3.4) we showed that the information acquired online can be used to monitor and steer a predefined robot interaction. This information is relevant to support ongoing mechanical manipulation of DoF. However, we still need to develop additional methods to generate and adapt ongoing robot motion based on the information perceived online. We will address this problem later in this thesis (see Chapter 6).

Extracting Information from Changing Sensor Signals correlated to Interactions (CH1) The presented system exploits the changes in the sensor signals as source of information. The system focusses its attention (and computation) on the moving bodies in the environment, creating filters on demand to estimate their motion. These moving bodies

are the most important since the goal is to perceive kinematics of articulated objects, and to support robot mechanical manipulation tasks that aim to change the pose of the bodies.

The kinematic model, together with the intercommunication mechanism between levels, correlates changes in the sensor signals and interactions. The model constrains the motion of the bodies. Through the intercommunication between levels, our system transforms these constrains in body motion into predictions of feature motion, that are effectively predictions about the appearance of the windows around features in the next image. The process links actions represented as changes in the kinematic state of the articulated object to changes in the sensor signals. However, the system does not link robot actions and changes in the articulated object. Later in this thesis (Chapter 6) we will investigate how to perceive and learn interaction models to bridge this gap, and link more intimately actions to changes in the environment and in sensor signals.

Perceiving Quickly and Online (CH2) Our presented system is fully online, using only previous measurements to interpret current signals. Moreover, thanks to our Bayesian filter approach the system does not need to memorize sensor signals and uses only the last acquired measurement. There are computational limitations, but we do not deem these severe, given the results of our first set of experiments. To be able to integrate into the robot's action loop, our system must perform at reasonably high frame rates. In all our experiments, we track between 150 and 250 features at 30 Hz, independent of the number of moving rigid bodies.

Versatile Perception in Unstructured Environments (CH3) We evaluated our proposed system in articulated objects of different sizes, structure, color and shape, demonstrated experimentally its robustness and versatility. Now, only objects with sufficient trackable texture can be perceived. As a result, our method inherits the limitations of the salient point feature KLT tracker, including the requirement of good features, relatively stable lighting conditions and bounded object acceleration. Note that we explicitly address the case of high deceleration to zero velocity (see Section 4.2.2) and high velocities using predictions from the estimation of rigid body motion. Alleviating these limitations will be the goal of the perceptual system we will present in the next chapter, integrating and exploiting information about the geometry of the object.

An occasional failure in our approach is that the system instantiates multiple rigid body filters to track the motion of the same rigid body. We evaluated this problem in the experiments of Section 4.3.1. We saw that the problem is more acute when the system tracks a high number of point features, increasing the probability of tracking features in low-textured areas that generate noisy trajectories. Our system alleviates the problem using the predictions from higher levels (informed by their priors) to reject noise. The rigid joint model is a second way of correcting for this failure. However, since a rigid joint indicates the same rigid body, the best strategy would be to inform the rigid body motion estimation level of this connection and merge the RBFs to improve the tracking accuracy. This further exploitation of kinematic information in the estimation of rigid body motion would be a good extension to our method.

4.5 Conclusion

In this chapter, we presented an online system for the interactive perception of kinematic properties of articulated bodies. It receives as input an RGB-D stream and outputs, at interactive frame rates, a kinematic model of the observed scene, including joint configuration

values. This perceptual capability supports and facilitates robot mechanical manipulation of DoF in unstructured environments.

Our perceptual system exploits the four problem regularities discussed in Chapter 1.2 (OP1-OP4) based on a coupled recursive estimation structure. This structure is composed of three interconnected recursive estimation processes, successively estimating feature motion, rigid body motion, and kinematic model of moving objects in the scene. The composition of these three processes and the bidirectional flow of information between them result in a highly robust system that exploits the interdependencies between the subproblems. This robustness is a result of level-specific physical priors that help to interpret the data and to reject measurement noise. The connectivity between the levels passes valuable information among the levels, further improving the convergence of the overall system.

5

Integrating the Perception of Shape and Kinematics of Articulated Objects

In the previous section we presented a perceptual system that builds kinematic models of previously unseen articulated objects from visual (RGB-D) input. The versatility of the system to different objects presents a limitation: the objects' surface have to present enough color texture. A way to overcome this limitation is to exploit the shape of the objects to compensate for uniformly textured-surfaces. This shape is unknown if the objects have not been seen before. Therefore, in order to support the feature-based tracking of objects the previous chapter, with shape-based tracking, we need to tackle the additional perceptual subproblem of building a model of the object's geometry, what is known as *shape reconstruction*. The result of the reconstruction process is useful, not only to support the perception of motion of uniformly textured objects, but also for other processes in robotics like motion planning, grasping, or action selection, as we will see in Chapter 7.

The majority of the existing approaches in robot perception, addresses the perception of pose, shape, and kinematic relationship in isolation, as we will see in the next section. This procedural approach neglects the interdependencies between these subtasks and their possible synergies. Shape reconstruction, pose tracking, and kinematic structure estimation naturally complement each other. To reconstruct the shape of an object from the information from the RGB-D sensors, it is necessary to integrate multiple views of the object under the assumption that the relative poses of the views are known. Most approaches, therefore, require knowledge of the pose (Krainin et al., 2011). On the other hand, to track the pose of an object, methods commonly rely on the knowledge of the object's shape and its segmentation in the image (Wuthrich et al., 2013, Choi & Christensen, 2013, Schmidt et al., 2014). Similarly, the estimation of the kinematic structure of an unknown object is facilitated by knowing the poses of its rigid parts (Sturm et al., 2011)—but knowing the kinematic structure can also improve pose estimation, as we saw in the previous chapter. Since each of these problems requires input that is provided by the others, we propose to combine them in a synergistic manner so that each subproblem provides helpful information to the others. Therefore, the solution we present in this chapter, while leveraging the four structural properties presented in Chapter 1.2 (OP1-OP4), delves deeper into the synergistic exploitation of the interdependencies between perceptual subtasks. We show that the integrated solution achieves better results than solutions for the individual problems.

Since the objects we are interested in are articulated and composed of parts that move

Figure 5.1: Our robot perceiving an articulated object using our integrated approach; it interacts with the drawer and detects the moving body, tracks it and incrementally reconstructs its shape (yellow layer); the robot estimates and tracks the kinematic model, including the joint axis (narrow green cylinder) and joint state (wider green cylinder), and an estimate of the uncertainty (transparent green cone) [© 2016 IEEE]

differently, we cannot build a geometric model using methods that assume static environments (Gonzalez-Aguirre et al., 2011, Kerl et al., 2013, Endres et al., 2014, Newcombe et al., 2011a). In fact, we need to identify the areas of the RGB-D images that move coherently in order to process them separately and build separate models of the links. Interestingly, in the combined problem, object motion segmentation serves as the connection between shape reconstruction and pose tracking: each of the two subcomponents passes information about its current object segmentation hypothesis to the other in order to improve the estimation (Figure 5.2). We will address object motion segmentation as an additional subtask in our perceptual system.

The method we will present in this chapter extends to two methods that follow the same insight (Stückler & Behnke, 2015, Ma & Sibley, 2014). Most importantly, it includes and exploits the estimation of kinematic structures of the interacted articulated objects (Figure 5.1). We will also provide a thorough experimental evaluation to analyze the improvements afforded by an integrated solution. We will analyze the contribution of each component to the final result. The evaluation includes difficult cases that are unsolvable when the subproblems are not tightly integrated. Our experimental evaluation will demonstrate the benefits of combining problems in robot perception and solving them in an integrated manner.

5.1 Related Work

In this section, we first review related approaches that address pose estimation, shape reconstruction, and segmentation independently. The scientific literature on these topics is vast; we focus on the most prominent approaches and the methods that directly relate to ours. At the end of this section, we will turn to combined approaches that integrate these problems.

5.1.1 Visual Pose Estimation

Visual pose estimation is the problem of inferring an object's pose from an image; the problem is called *visual tracking* if performed on a stream of images exploiting the temporal structure in the problem (using the previous pose to estimate the current pose). We have distinguished two main approaches to visual tracking: based on a *known shape* model of the object (Wuthrich et al., 2013, Schmidt et al., 2014, Choi & Christensen, 2013, Garcia Cifuentes

et al., 2017), and based on *sufficient surface-texture* on the objects using point features (Choi & Christensen, 2012, Lepetit & Fua, 2006, Collet et al., 2011) or dense optical flow (Stückler & Behnke, 2015, Ochs et al., 2014). While tracking with a known model is more accurate, texture-based approaches are also applicable to unknown objects. In this chapter, we present a method that exploits the advantages of both approaches by bootstrapping the system with feature-based tracking, and subsequently combining it with shape-based tracking.

5.1.2 Shape Reconstruction

Shape reconstruction acquires a 3D appearance model of an object by merging a set of partial object views into a coherent shape model using information about the relative object poses with respect to the camera. Partial object views can be obtained using image segmentation (Matsuyama et al., 2004), and pose information by the controlling camera and object motion (Krainin et al., 2011) or estimating this pose visually. Our approach automatically generates both, partial object views and their pose information, and uses them to reconstruct the shape.

5.1.3 Image Segmentation

The segmentation problem consists of finding the region in the visual input occupied by the object. We distinguish between *single-image-based segmentation* and *motion-based segmentation* operating on image streams. For segmenting single images, a wide variety of different approaches has been proposed, e.g. assuming *surface continuity* as exploited by conditional random fields (Lafferty et al., 2001) graph-cuts, (Felzenszwalb & Huttenlocher, 2004), and supervoxel region growing (Papon et al., 2013a), or exploiting *object location* as in active segmentation (Mishra et al., 2009). Motion-based segmentation exploits the notion of "objectness" by assuming that all points on a rigid body move together. To detect which points moved due to object motion, image differencing (Chien et al., 2002, Kenney et al., 2009) can be applied. To reject changes caused by background motion, image differencing can be combined with information from a tracker to select only points that move consistently with the object (Stückler & Behnke, 2015, Ochs et al., 2014). In our approach, we have applied motion-based segmentation to generate sparse segments and extend them using supervoxel region growing, and we have used single-image segmentation by using the continuously updated reconstruction of the shape.

5.1.4 Integrated Approaches

A prominent *combined approach* to tracking and shape reconstruction is visual SLAM (Gonzalez-Aguirre et al., 2011, Kerl et al., 2013, Endres et al., 2014, Rusinkiewicz et al., 2002, Weise et al., 2009, Newcombe et al., 2011a,b). However, visual SLAM reconstructs an entire scene assuming it is static, and does not segment and reconstruct single objects. An extension for deformable objects was presented by Newcombe et al. (2015). This method considers only one object and it does not build kinematic models of articulated objects.

This *object* perception problem has recently been addressed in a combined manner. Ren et al. (2013) present a method to simultaneously track and reconstruct 3D objects by refining an initial primitive shape model; in contrast to our method, it can only reconstruct and track one moving object, and the initial location of this object must be manually provided. Walsman et al. (2017) present an approach that tracks articulated objects and refines an initial coarse model of the shape of the links. This method needs to know the kinematic model

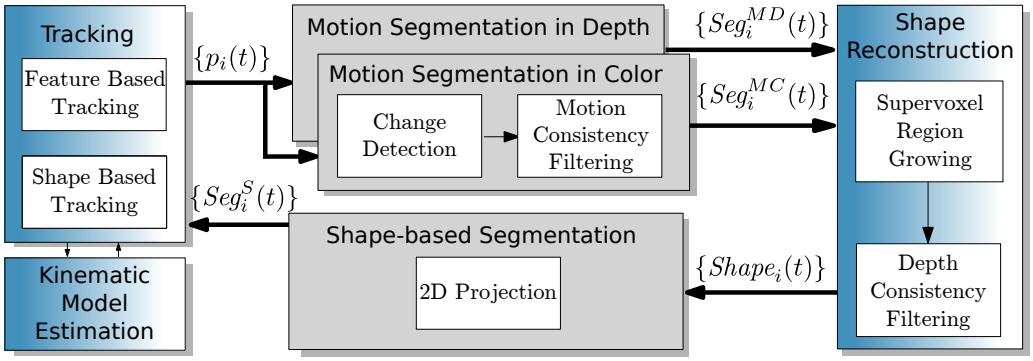


Figure 5.2: Our tightly integrated shape, pose and kinematic structure estimation system, using segmentation as an intermediate process; tracking information is used to segment changing parts of the RGB-D images into coherently moving segments; the segments are extended to larger regions of continuous color and curvature (supervoxels) and accumulated into shape models; the models of the shape are used to find a more complete segmentation of the RGB-D images, and to support the tracking of the objects; the refined pose information improves the estimation of the kinematic models [adapted from © 2016 IEEE]

beforehand and cannot be applied to build models of previously unseen objects. Stückler & Behnke (2015) suggest a method that combines object tracking, segmentation, and reconstruction using an Expectation-Maximization (EM) algorithm. Their method differs from ours, as it relies on an initial oversegmentation and groups the segments using motion and surface clues, making it sensitive to a wrong initial segmentation. Other methods (Ma & Sibley, 2014, Herbst et al., 2010, Xu et al., 2015) build on top of KinectFusion (Newcombe et al., 2011a). These methods build a model of the environment and consider any part that becomes inconsistent with this model as a new object. In contrast to these methods, the approach presented here combines object segmentation and tracking with the generation of a kinematic model and thus is able to track articulated objects.

Conclusions and comparison to the proposed approach: Most existing methods in the literature address the estimation of kinematics, shape, and pose of articulated objects in isolation. This factorization of the problem does not leverage the interdependencies and synergies between the subtasks. A few methods integrate some of the subtasks into unified perceptual systems but none of them address the shape reconstruction, pose tracking, and kinematic structure estimation for previously unseen objects. The system we present in this chapter will integrate these three subtasks and the segmentation of images based on motion.

5.2 Integrating Shape Reconstruction, Segmentation and Kinematic Modeling

In this section, we describe our integrated method for pose tracking, object segmentation, shape reconstruction, and kinematic structure estimation (Figure 5.2). Our feature and shape trackers (Section 5.2.1) provide the motion information to segment RGB-D frames into objects (Section 5.2.2). These object segments are used to reconstruct the shape of the object over time (Section 5.2.3). To close the loop, we have used result from shape reconstruction to find better object segments (Section 5.2.4) and use them to refine tracking (Section 5.2.1).

5.2.1 Sensing and Tracking

As in the previous chapter, the input to our method is an RGB-D stream that we represent as a sequence of color images $I(t)$ and depth maps $D(t)$ for every time step t. Some parts of our method directly operate on point clouds $P(t)$, which combine color and depth.

We pre-process the raw depth images by applying a joint bilateral filter (Le et al., 2014) that fills the depth-missing areas based on their surrounding depth and color information. Similar to the original bilateral filter (Tomasi & Manduchi, 1998), the joint bilateral filter extends information over regions delimited by edges. The main difference is that while the bilateral filter detects edges and extends to regions of the same image, the joint bilateral filter detect edges in the color image and uses it to extend information of the depth image. The assumption is that a region of uniform color possesses uniform depth. This process fills the areas without range measurements that appear frequently in depth images from RGB-D sensors based on projected light due to occluding shadows.

Feature-Based Tracking and Kinematic Model Estimation

To bootstrap our pipeline, we obtain information about object motion and location from the combined perceptual system for motion tracking and kinematic model estimation that we presented in the previous chapter (see Chapter 4). The system of Chapter 4 is composed of three interconnected processes. The first process estimates the motion of a set of 3D point features. The location of these features is passed to the second process, a *feature-based tracker*, that groups coherently moving features into rigid bodies and tracks the motion of the bodies. Finally, the third process estimates a kinematic model that explains the motion constraints between the rigid bodies, and defines the articulated object. The key of this system is to leverage the four problem structural regularities we identified in Chapter 1: temporal structure, physical priors, interactions as source of information, and interdependencies between subprocesses. In this chapter, we further exploit interdependencies between perceptual subprocess, passing information between them to mutually improve each others results.

The system presented in the previous chapter estimated the 6D pose $\{p_t^i\}_{i\in\{1,\dots,N\}}$ of the N currently tracked objects, their 6D velocities $\{\eta_t^i\}_{i\in\{1,\dots,N\}}$, a sparse set of M tracked 3D point features on each object $\{f_t^m\}_{m\in\{1,\dots,M\}}$, and kinematic constraints, i.e. joints, between the objects. We will use the output from the system presented in the previous chapter (and later of the combined tracker) in our motion segmentation component (Section 5.2.2), which will generate inputs for the shape reconstruction.

Shape-Based Tracking

We use the reconstructed shape to improve the pose estimation based on point features. First, we estimate the part of the reconstructed shape that is visible from the current view projecting the model into the image plane (previously transformed based on the initial pose estimate predicted from the previous pose and velocity). Then, we align this partial view to the current point cloud using the iterative closest point (ICP) algorithm (Pomerleau et al., 2011). To reduce the complexity of this process, which depends on the number of points to be aligned, we focus the search to the area of the current point cloud where we expect the object to be found. We use the results of the shape-based segmentation (Section 5.2.4) to delimit the area of the field of view where the object should be located. Using only the visible part of the reconstructed shape, the segmented part of the current point cloud and the initialization to the predicted pose, we reduce the computation time and facilitate convergence to a favorable pose.

This shape-based tracking overcomes the limitations of the system presented in the previous chapter that could only track texturized objects, as shown in our experiments.

5.2.2 Motion Segmentation

We build increasingly complete models of the geometry of the moving objects b integrating partial views. To obtain these partial views, we use the information from the pose tracker and compute a motion segmentation of the objects. The pose information and the object segments are combined to reconstruct the entire object as detailed in Section 5.2.3.

The general idea of motion segmentation is to first detect changes in the depth and color images of two consecutive time steps, and then use the tracked 6D poses to identify areas that change consistently with the motion of the object. These areas are the motion segments. Using the pose for motion segmentation is beneficial, even in cases where only one object moves, because it allows to reject the false positives found by change detection.

In the following section, we will explain the two similar processes we follow to detect and classify changing image segments into moving objects. Each processes is based on changes in a different visual channel: depth and color images.

Motion Segmentation in Depth

Algorithm 1 Motion Segmentation in Depth

1: $\Delta D = D(t) - D(t-1)$
2: $M_{\mathrm{E}} := \Delta D < -\gamma_{\mathrm{motion}}$
3: $M_{\mathrm{L}} := \Delta D > \gamma_{\mathrm{motion}}$
4: $M_{\mathrm{AccE}} := M_{\mathrm{AccE}} \wedge M_{\mathrm{E}} \wedge \neg M_{\mathrm{L}}$
5: **for all** p^i **do**
6: $\quad \Delta p^i := p_t^i \ominus p_{t-1}^i$
7: $\quad \mathrm{Seg}_i^{\mathrm{AccE}} := \mathrm{1NN}(\Delta p^i \cdot P(t-1), P(t)|_{M_{\mathrm{AccE}}})$
8: $\quad \mathrm{Seg}_i^{\mathrm{L}} := \mathrm{1NN}(\Delta p^i \cdot P(t-1)|_{M_{\mathrm{L}}}, P(t))$
9: $\quad \mathbf{Seg_i^{MD}} := \mathrm{Seg}^{AccE} \cup \mathrm{Seg}^{L}$

We first detect changes in the scene by computing a difference image ΔD of depth maps from subsequent computation steps (line 1 in Algorithm 1). Assuming for a moment that every change in ΔD has been caused by the motion of the body, we know that the body has abandoned some part of the image and/or entered some other. In a particular region of the image, we can discriminate between these two cases by looking at the sign of ΔD and computing a binary motion mask for each case (lines 2-3): the *entering-motion mask* M_E contains distance decreasing pixels (the sign of the difference is negative), the *leaving-motion mask* contains distance increasing pixels (the sign of the difference is positive), considering only absolute differences over a noise threshold γ_{motion}. To handle small motions between subsequent time steps we accumulate points in M_E over time in the *accumulated entering-motion mask* M_{AccE}.

Next, we discriminate which of the detected changes are consistent with each object's tracked motion Δp^i (line 6). The basic idea is to apply Δp^i to the previous point cloud $P(t-1)$ and compare the result to the current $P(t)$ using nearest-neighbor search (1NN, line 7). By first filtering $P(t)$ with M_{AccE} we only take into account points that changed in depth. In a

similar fashion we use M_L to find points that belong to the object in $P(t-1)$ (line 8), and add the two point clouds together to obtain the final depth-based motion segment Seg_i^{MD}.

Motion Segmentation in Color

Our depth-based motion segmentation method is rather conservative as it does not add points to the segments that have not changed their depth value, even if they are consistent with Δp^i. Although reducing the risk of adding false positives, the approach fails if no change in depth is present, e.g. a rotating globe (see Section 5.3). We, therefore, add a color-based motion segmentation, which works similar to the depth-based version presented in Algorithm 1. The main differences are that we compute the image difference in HSV color space, using only hue (H) values for all pixels with sufficient saturation (S $>$ 90) and that we do not discriminate between image regions that the body has left or entered (since depth information is required for this). We discriminate into coherent moving bodies in the exact same way as in the segmentation in depth: moving the previous point with the motion and comparing to the current point cloud, but filtering first using the parts of the image that changed their color.

5.2.3 Shape Reconstruction

We use the information from the pose tracker (p^i) to reconstruct the shape based on the motion-based segments (Seg_i^{MD} and Seg_i^{MC}). We transform the points of the segments to the initial object pose applying the inverse of the current pose, and accumulate the result into a shape model. We represent the shape by a point cloud, which is resampled using a voxel-grid filter at every time step in order to keep the required memory constant and to address the inhomogeneous point distribution resulting from the depth measurements. To deal with regularly shaped objects with uniform color, which usually generate sparse motion segments, we first extend these segments by exploiting *surface continuity* and *known object location*, and finally filter out points that are inconsistent with the current view of the scene, as we will explain in the following section.

Supervoxel Region Growing

We extend the partial motion segments with a region growing procedure on a supervoxel segmentation. First, we apply a supervoxel segmentation to the RGB-D point cloud. Then, for each moving object, we seed region growing with the supervoxel that contains most of the points of object segment. Region growing then extends from a supervoxel (A) to a neighboring supervoxel (B) if (B) fulfills one of the following criteria: (i) most of its points demonstrated coherent motion (resulting from the segmentation based on motion 5.2.2), (ii) most of its points are part of the reconstructed shape so far (resulting from the segmentation based on shape 5.2.4), or (iii) the mean color and mean surface normal of (B) are very similar to the mean color and mean surface normal of (A). We also extend the segment if the neighboring supervoxel contains multiple point features of the rigid body. The result is then merged into the shape model.

Depth Consistency Filtering

All previous steps are adding points to the shape model. However, sensor noise, errors in tracking and overly optimistic supervoxel extensions can lead to wrong points being added to the model. We can remove many of these points by verifying whether they are consistent

with the current depth map $D(t)$ when projecting the model to the image plane. We therefore remove every point for which the projection calculates a lower depth value than observed in $D(t)$ – this means that we could see a background point in the image where we expected a point of the object. We do not remove points where the measured depth is lower than the expected from the projection of the model, since they could be produced by occlusions with other objects.

5.2.4 Shape-Based Segmentation

Using the results from the previous shape reconstruction steps, shape segmentation becomes trivial. The shape segment at the current time step $\text{Seg}_i^S(t)$ is the result of transforming the shape model using its tracked pose p^i and projecting the result into the image plane. The projection of the model into the image plane does not include points that are outside the current viewing area or points that are occluded by the object itself or by other moving objects. We feed the shape segment to the shape tracking component to restrict the alignment of the visible part of the model to the area of the point cloud occupied by the object. This helps in the computation of the ICP algorithm and thus, in the kinematic structure estimation, and the procedure starts over in the next time step.

5.3 Experiments

5.3.1 Experimental Setup

We evaluate our approach in eight different experiments, each carefully selected to verify the contribution of each component of our method[1]. In each experiment, a human or a robot actuates one or more objects in the scene. During robot manipulation, we exploit additional information (forward kinematics and a known shape model of the robot arm) to infer the part of the image that corresponds to the robot and to exclude it from tracking and segmentation. We recorded each experiment using a statically mounted Asus Xtion RGB-D sensor. We run our algorithm on an Intel Xeon E5520 CPU at 2.27 GHz, reaching 3 to 10 frames per second for the shape-based tracker, depending on the size and number of moving objects. Segmentation and shape reconstruction are running at a lower rate of 0.8 s due to the computationally demanding supervoxel segmentation and to clearly discriminate changes in the image differencing steps. But since tracking is running at a high rate, slow reconstruction time does not affect the capability to track fast motions, and the overall online capabilities to estimate kinematic models.

The experiments consist of three scenes containing only rigid objects and five scenes with articulated ones. Three of the RGB-D sequences of the experiments were part of the evaluation of the online IP system of the previous chapter and we include them in this evaluation to examine the influence of the new perceptual subtasks on the overall system. In the following we will describe each experiment and the challenges it presents for perception.

Box: A box with little texture moves parallel to the viewing-plane for about 50 cm (duration: 9 s). We expect approaches that do not exploit surface continuity to require longer time to reconstruct the shape.

Two Bodies: In this experiment two bodies move freely on a table-top at the same time (duration: 14 s). We want to verify that the method can cope with multi-body settings and to which extent the hand of the experimentator is added to the reconstructions.

[1] Our datasets are publicly available under http://tinyurl.com/o3bu7pd.

Red Figure: A red figure is moved freely on a table-top (duration: 10 s). This experiment is designed to test how the pipeline performs when the quality of the point-features abruptly degrades. We therefore manually force the textured part of the object to become occluded after 3 s, and evaluate the contribution of shape-based tracking.

Drawer: A robot opens a drawer (duration: 2 s). An easy articulated object case.

Globe: A globe rotating 360 degrees around its revolute axis (duration: 18 s). We expect that pure feature-based tracking is inaccurate due to the large uniformly colored areas, and we expect incomplete reconstructions if surface continuity is not used.

Head: The first author rotates his head left and right (duration: 8 s). This experiment evaluates to which extent semi-rigid objects pose a problem for our method.

Cabinet and Drawer: A cabinet moves freely on the floor (duration: 15 s). At some point, a drawer is pulled out of the cabinet and pushed inside again. We evaluate the performance when objects partially get out of the field of view.

Laptop: A laptop is moved freely on a table-top and then being closed and opened (duration: 9 s). We evaluate the effect of purely rotational motion on the reconstruction.

5.3.2 Evaluation Criteria

We evaluate the contribution of each component of our system using two criteria. First, we quantitatively assess the object segmentation results provided by each component. This gives an indirect means of comparing the impact of the different parts, as the accuracy of the pose tracker directly influences the quality of the motion-based segmentation, and the correctness of the shape reconstruction affects the shape-based segmentation. Secondly, we evaluate the quality of the reconstructed shape and the estimated kinematic structure (using the online IP system of previous chapter extended with the shape-based tracking) by visual inspection of the results.

To evaluate the segmentation results, we manually annotated each video sequence with the ground truth every 0.8 seconds. We compute precision, recall and $f_{0.5}$-score[2] for the *full pipeline* (i) and five additional variants of our algorithm: to assess how integrating tracking and shape reconstruction affects the result, we evaluate the *full pipeline without feedback from shape tracking* (ii); to evaluate the contribution of the tracker to the pipeline we look at *depth-based motion segmentation* (iii) and *color-based motion segmentation* (iv); finally, we assess the contribution of shape-base tracking by evaluating *shape-based segmentation using only depth* (v) and *shape-based segmentation using only color information* (vi). We additionally compare our results to a baseline, a dense optical flow approach using RGB presented by Ochs et al. (2014) (using the recommended standard parameters). We compare against this method as it is the only relevant approach for which code was available at the time of the evaluation.

5.4 Results

The segmentation results are depicted in Figure 5.5 (rigid objects) and Figure 5.6 (articulated objects). We observe that, at the end of each experiment for all except two cases (two bodies: statue; cabinet: drawer), the full pipeline with tracking (solid black curve) outperforms all other variants, attaining $f_{0.5}$-scores above 0.8. Thus overall, our method detects most of

[2]The $f_{0.5}$-score is a standard variant of the f-score which weighs precision higher. We use this variant since points wrongly attributed to the object – which only affect precision – have a more significant negative impact on tracking and thus reconstruction performance than missed points. In our system, we primed precision over completeness on the model.

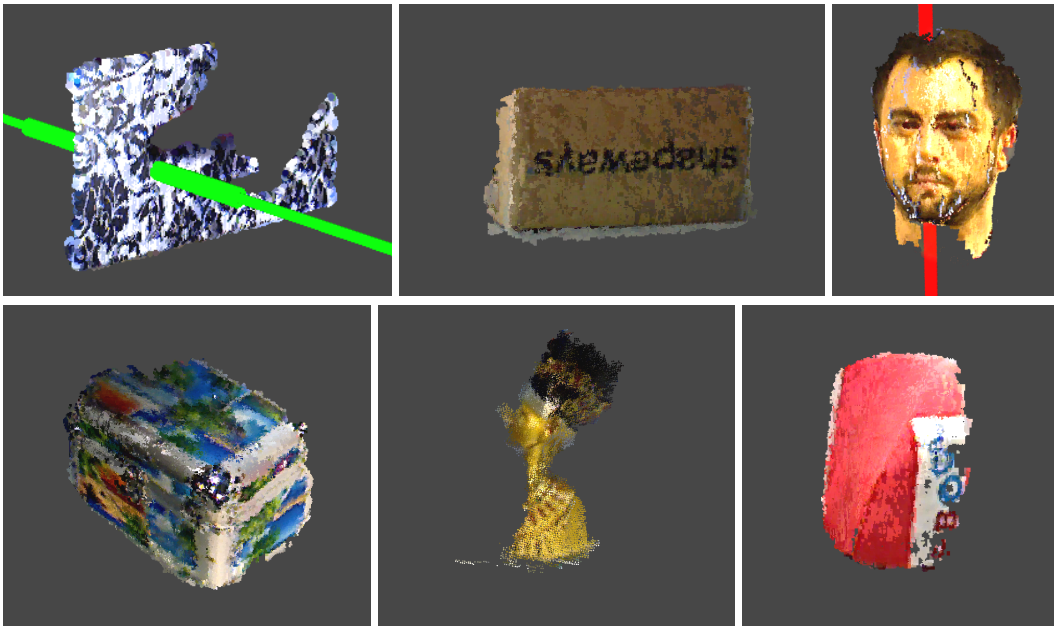


Figure 5.3: Results of the shape reconstruction in combination with motion tracker; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; joint value shown as wider cylinder [© 2016 IEEE]

the area occupied by each object (high recall) while adding few false points (high precision). Secondly, we observe that both full pipeline variants converge very quickly to their final results. The reason is the effective combination of the different priors: whereas the pure motion segmentation variants (blue and red curves) usually require long time to obtain the full segment and reach high recall, extending the motion segment with region growing on supervoxels allows to quickly obtain a complete segment. The baseline by Ochs et al. (2014) often fails to find the correct segments and gives competitive results only in the head experiment.

The reconstruction and kinematic structure estimation results are shown in Figure 5.3 and Figure 5.4. The results are in line with the segmentation results since all objects except for the statue and the drawer are reconstructed correctly. We also observe that the joint estimation is much more accurate when including shape tracking, which indicates that the combined tracker provides higher quality pose estimates.

We will now turn to a detailed analysis of every scene.

Box: Most variants perform well in segmentation and reconstruction, but close to the object borders, some variants add wrong points that belong to the background. The reason is inaccurate registration of depth and RGB pixels by the sensor, causing wrong depth measurements at the objects borders.

Two Bodies – Metal case: The full pipeline succeeds in quickly segmenting the metal case and reconstructing all three visible sides of the object. The motion-based segmentation methods fail because they do not exploit knowledge about object location, and add spurious points on the arm.

Two Bodies – Statue: This is the only case where the full pipeline without shape tracking (dashed black curve) outperforms the other strategies. The reason is that early during the experiment, the hand of the experimenter is added to the segment. When the hand starts

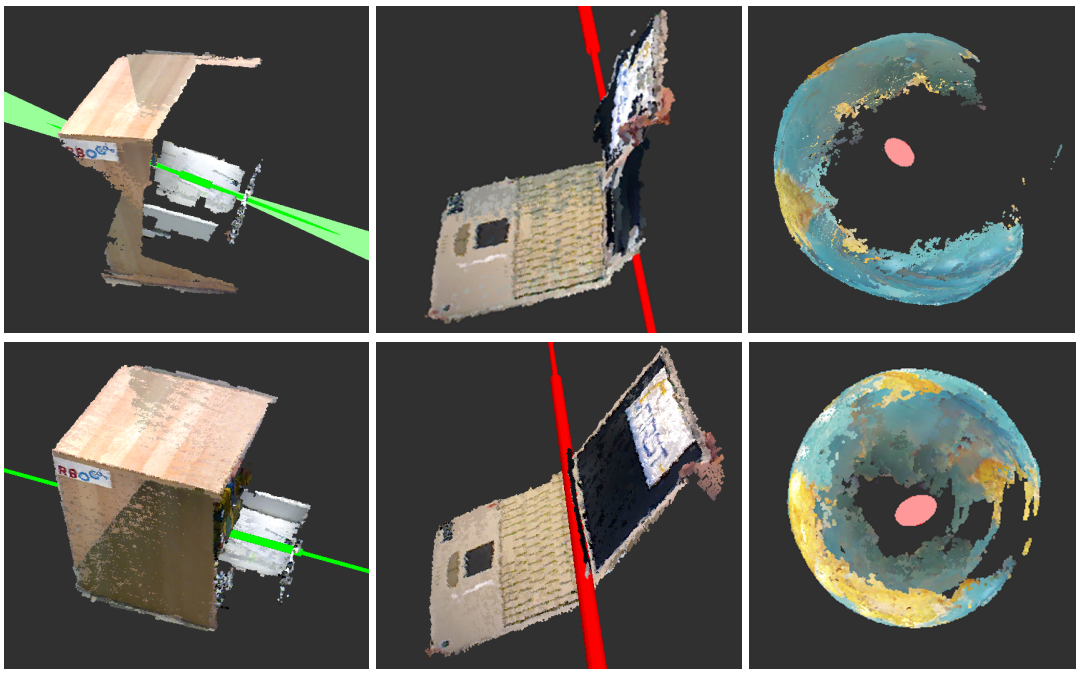


Figure 5.4: Results of the shape reconstruction and kinematic structure estimation; each column represents a different articulated object; from top to bottom: results when shape-based motion tracker is not integrated, results when shape-based motion tracker is integrated; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; joint value is shown as wider cylinder; uncertainty about the joint is represented as transparent cones [© 2016 IEEE]

retracting at $t = 9.8s$ till the end, the full pipeline wrongly biases the feature tracker to pay attention to the motion of the hand, whereas the variant without shape tracker removes the hand from the model. This effect is a limit of our motion segmentation approach: two bodies moving similarly during enough time are perceived as the same body. This effect also biases the reconstruction result: when using shape tracking, the arm motion causes the consistency filter to remove the right half of the statue. If we omit the feedback to the tracker, a smaller part of the arm is initially added, but eventually removed in the consistency filtering step.

Red Figure: The red figure is best segmented by the full pipeline. Without shape-based tracking, the performance drops drastically when the point-features disappear because the textured part becomes occluded. However, even without the shape tracker the pipeline can partially recover because enough changes in depth are visible. The reconstructed shape is almost complete when using feedback from shape tracking, in contrast to the partial reconstruction of the other variants.

Drawer: Almost all variants segment and reconstruct the front lid of the drawer (which is the only visible part) and detect the prismatic joint quickly. Color-based segmentation fails because the drawer contains many dark areas and we ignore points where the saturation in the HSV space is low.

Globe: Since there is no change in depth and little change in color, pure motion segmentation fails to segment large parts, indicated by the low recall values. The variant using region growing on supervoxels obtains the most complete reconstruction and segmentation. The quality of the reconstruction and joint estimation depends on the tracker (Figure 5.4). There is

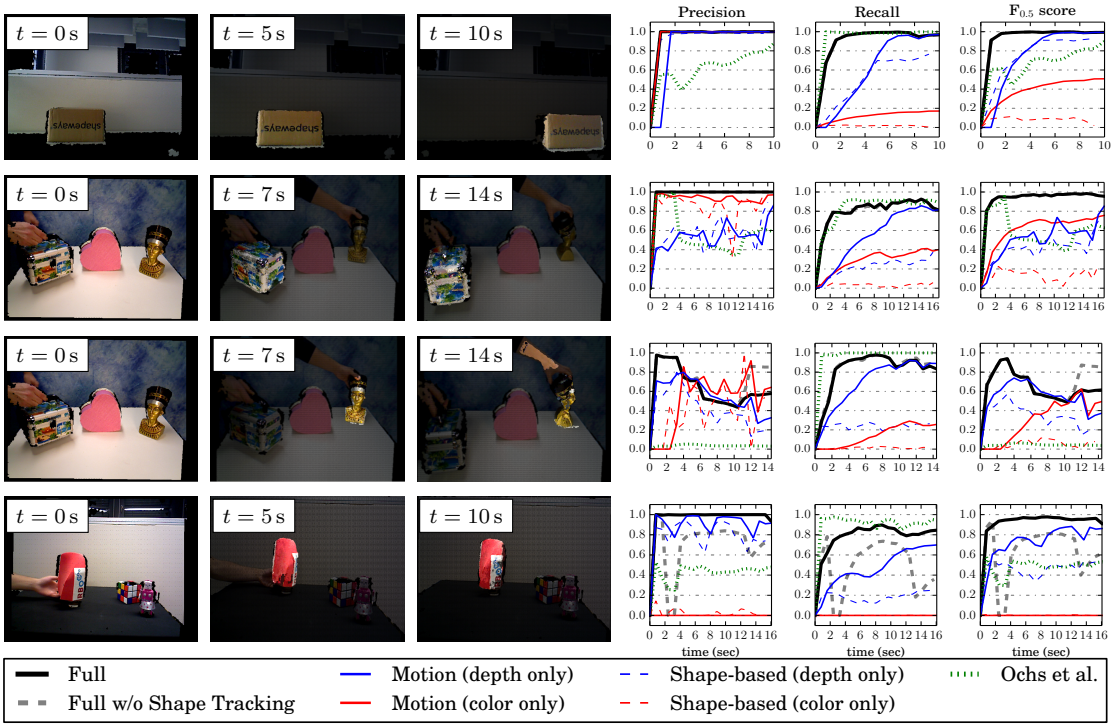


Figure 5.5: Results of the segmentation (each row represents a different object); from left to right: full initial scene (RGB-D point cloud projected to image plane), result after first segmentation, and final segmentation result (solid color indicates the segment), precision, recall, F-Score; we compare our full pipeline to subparts of it and to the segmentation generated by Ochs et al. (2014); the insets in the three images show the time t [© 2016 IEEE]

a large part of the globe that does not exhibit sufficient texture to accurately track a large number of features and hence tracking is as accurate as in the other parts, resulting in a non-spherical shape reconstruction. By constantly integrating the feedback from shape-based tracking, we obtain a much more spherical reconstruction.

Head: Similar to the globe, the head is only well segmented by the full pipeline (and the baseline). Some misclassifications of the neck and the hair lead to minor segmentation errors. The reconstruction is accurate but exhibits some abrupt color changes due to non-uniform lighting.

Cabinet – Frame: The cabinet frame is quickly segmented by the full pipeline which is also reflected in the reconstruction. Without shape tracking, the feature-based tracker loses the frame object at $t = 7$ seconds after the drawer moved for 3 seconds. This is because there are more features on the drawer, so the tracker assumes that the remaining features on the cabinet are outliers and drops them. In contrast, by taking into account the shape of the bodies the tracker correctly splits the cabinet and the drawer into two rigid objects. In this case, the cabinet is correctly reconstructed and remains stable even when it leaves the scene.

Cabinet – Drawer: The drawer is only partially segmented by the full pipeline. This is because the motion segmentation provides two disconnected components: the inner part and the lateral part. The supervoxel growing is seeded from the inner part and, due to the gap in

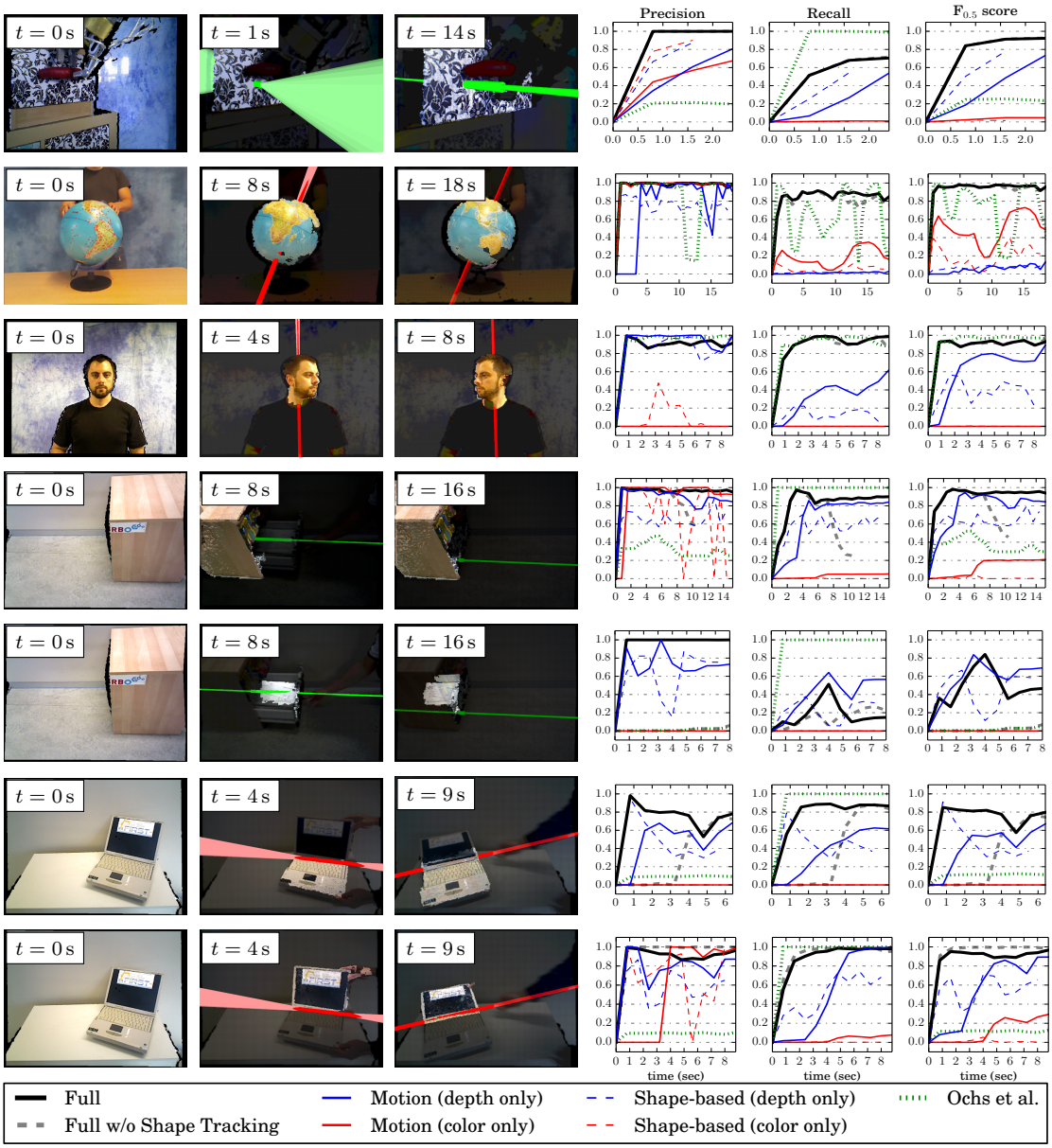


Figure 5.6: Results of the segmentation of articulated objects (each row represents a different rigid body); from left to right: full initial scene, result after first segmentation, final segmentation result, precision, recall, F-Score; the insets in the three images show the time t; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation [© 2016 IEEE]

depth, does not extend to the lateral part. The purely motion-based segmentation performs better because it does not use this location information (which however leads to degraded performance in the box experiment as mentioned earlier). The segmentation reduces the quality of the reconstruction. Still, using the partially reconstructed drawer for shape-based

tracking largely improves the reconstruction and the estimation of the prismatic joint, as shown in Figure 5.4, first column. Moreover, our method correctly handles the occlusion of the drawer when it closes and remembers its shape.

Laptop – Bottom and Lid: Both rigid bodies of the laptop are correctly segmented by the full pipeline, with only some points incorrectly added. These points are close to the revolute axis and present small errors when the body motion is applied to them. We consider this a limitation of our current algorithm that could be solved by reconstructing jointly all parts of the articulated object, instead of independently for each part.

Again, shape reconstruction and tracking is much more accurate when the shape-based tracker is used. Without shape-tracking, the orientation of the estimated joint in the kinematic structure diverges by approximately 5°, as visible in Figure 5.4, second column.

To conclude, in all experiments the full pipeline with tracking feedback provides good segmentation and reconstruction results, and outperforms the other variants in all but two experiments.

5.5 Discussion and Limitations

In this section we discuss the benefits and weaknesses of our system. We first analyze its technical limitations. Then, we discuss the role of each of the four opportunities for perception for robot manipulation (OP1-OP4) in the performance of our presented system. We conclude by discussing whether the presented system overcomes the challenges of perception (CH1-CH3).

The presented system obtains precise motion segmentation results that generate shape models to support texture-based pose estimation. However, the reconstructed models are not at the quality level (e.g. level of detail, completeness) of other state-of-the-art shape reconstruction approaches (Newcombe et al., 2011a, 2015, Sturm et al., 2013, Xu et al., 2015). We think the gap is largely caused by our representation of shapes. We represent shapes as point clouds. This forces us to a costly maintenance of the shape models (voxel-grid filtering, see Section 5.2.3), and to use point-to-point ICP implementations.

On the other hand, state-of-the-art reconstruction approaches use (truncated) signed distance functions, SDF (Curless & Levoy, 1996). An SDF is an implicit surface representation. It represents the space as a voxel grid and stores the distance from each voxel to the object's surface. Implicitly, the object surface is defined by the zero-distance voxels. SDFs naturally capture the information from an RGB-D sensor, like the Kinect sensor, with an efficient ray-tracing operation. The SDF also allows for quick rendering of virtual depth maps. SDFs enable the efficient computation of point-to-plane ICP alignments of the shape models to the RGB-D sensor data, based on the quick estimation of distance gradients and surface normals. For these reasons, we think that an optimal implementation of our shape reconstruction approach should be based on an SDF. In spite of the suboptimal representation, our system delivers fairly complete and accurate geometric models that support robot manipulation, as we will demonstrate in Chapter 7.

A second technical limitation on which we commented before (see Section 5.4) arises from the independent reconstruction of the geometric models for each moving body. Because of the independent process, the same points can be added simultaneously to different rigid bodies. We think that the points should be exclusively integrated into the rigid body that best predicts their motion.

After the discussion of the most critical technical limitations, we will now discuss how the system exploits the four proposed opportunities for robot perception. The role of interactions (OP1), physical priors (OP2) and temporal structure (OP3) did not change significantly between the systems of Chapter 4 and this chapter. In a nutshell, the system of this chapter

1) uses interactions (only) as generators of information-rich motion cues, 2) leverages priors about projective geometry and rigid body physics to segment and reconstruct the shape of the links, and 3) segments images using the shape models that result from the temporal accumulation of previously segmented partial views. Differently from the system of the previous chapter, in this chapter, our system exploits additional physical priors to extend the segmentation results. The system assumes that objects have continuous color and curvature.

The most important difference between the systems of previous and current chapters is in the way they exploit interdependencies between perceptual subtasks (OP4). The system of this chapter integrates segmentation and shape reconstruction as subtasks into the perceptual process. These subtasks and the ones of our system of Chapter 4 (especially the pose tracking) are strongly correlated and complement each other naturally, as we have shown in our experimental evaluation. The pose of the moving parts is the necessary information to integrate partial views correctly into a shape model. The shape improves the pose estimation through tracking and helps to obtain more accurate kinematic model estimates. The segmentation connects both the subtasks, since information of the pose is needed to identify new extensions of the shape models, and the shape models restrict the parts of the current point cloud for tracking.

We will now turn to a discussion of the limitations of the presented system in the context of the three challenges of perception for robot manipulation. We will focus on the main differences with respect to the system presented in Chapter 4.

Extracting Information from Changing Sensor Signals correlated to Interactions (CH1) The proposed system focusses on signal changes using image differencing at constant intervals. This contrasts to our procedural approach to estimate motion of point features and rigid bodies, where we focus on signal changes using a predicting-correcting recursive procedure. For segmentation and shape reconstruction we chose image differencing to reduce the high computational cost of predicting and correcting motion of entire shape models. As explained before, this high cost is a consequence of our inefficient representation of the shapes, as sets of points. To operate efficiently on entire shapes we would need a different representation, e.g. signed distance functions (Newcombe et al., 2011a) or groups of 3D points (Stückler & Behnke, 2015)

Thanks to the integration of segmentation and shape reconstruction with the other subprocesses in the perceptual system, the robot can predict better changes in the sensor stream. Based on the generated dense shape models, the tracking of the rigid bodies improves and the robot can predict more accurately the motion of the point features and their surroundings. Additionally, while we do not use this capabilities for perception, the robot can use the shape models to predict the appearance of the parts of the image where the rigid bodies project. However, it is still not possible to link robot actions to changes in the kinematic state because of the lack of the necessary interaction models. Therefore, the current model does not entirely link actions to changes in sensor signals. We will overcome this limitation in the next chapter, Chapter 6.

Perceiving Quickly and Online (CH2) The segmentation and the reconstruction subprocesses run every 0.8 s. The robot cannot rely on shapes reconstructed at this rate to support highly dynamic and fast manipulations. The main reason for this low rate is the computationally demanding supervoxel segmentation, and that the image differencing needs to observe enough change between the input images. The model projections and the subsampling of the accumulated point cloud model also contribute to the large computation time. As we

discussed before, a way to reduce the computation time would be to use a more efficient shape representation, e.g. a truncated SDF. However, since the shape-based tracking is performed at a higher rate, the system can still perceive online kinematic models.

Versatile Perception in Unstructured Environments (CH3) The dependency on image differencing to generate candidate regions for segmentation makes our approach more suited for setups where the camera is static. Motion of the camera would result in our system detecting the entire field of view as moving, and it would increase the computational burden significantly. While this is an important limitation in the versatility of the system, in many real world robotic manipulation scenarios, the camera is static when the robot needs to perceive a kinematic model from an interaction.

The initial motivation for us to build shape models was to support and improve feature-based tracking, and thus, improve the versatility of our system to perceive articulated objects without highly textured surfaces. We have shown in the experimental evaluation that the synergistic integration of shape reconstruction improves pose estimation (OP4), and that the overall system is more robust and generate more accurate kinematic model estimates. The versatility of the system improved as well, since the system compensates with shape information for the lack of texture to perceive motion. However, the system is still dependant on vision to contain sufficient information about the articulated object. The system fails if the visual conditions are suboptimal, e.g. due to occlusions of the articulated object or adversarial lighting conditions. We will see in the next chapter that we can exploit interdependencies between perceptual subprocesses in different sensor modalities to alleviate these limitations.

5.6 Conclusion

In this chapter, we presented a combined perceptual system for estimating pose, shape, and kinematic structure of articulated objects. The system exploits the synergies between the subprocesses and integrates them online. The subprocesses provide information to each other, leading to performance improvement and eliminating the dependency on a priori knowledge about the objects. We demonstrated the benefits of the combined system by comparing its performance with that of subsystems with less integrated subprocesses, in several challenging perceptual tasks. Our algorithm perceives the shape and the pose of multiple moving objects, and estimates a kinematic model of the articulation.

6

Perceiving Articulated Objects From Multi-Modal Streams

In chapters 4 and 5, we presented and evaluated two interactive perceptual systems for articulated objects. The systems are based on our proposed approach to tackle perceptual problems in robot manipulation. Our approach aims to address the challenges (CH1-CH3) in perception for robot manipulation by exploiting the structure of the problem (the opportunities, OP1-OP4). The systems of Chapters 4 and 5 are based on a single sensor modality, vision, provided by an RGB-D sensor. From the evaluation and the discussion of the properties of the previous systems we concluded that:

- *Concerning the versatility of perception to different unstructured environments (CH3):* The robustness and versatility of the systems are limited because the systems are solely based on visual information. This sensor modality does not contain sufficient information for the perceptual task in certain environment and manipulation conditions, e.g. in adversarial lighting conditions or when the object is visually occluded.
- *Concerning the use of action for perception (CH1):* The systems do not fully exploit the information about the interaction (and its correlation to changes in the sensor signals) that is available when the robot manipulates articulated objects, e.g. haptics and robot's motion.
- *Concerning the use of perceived information to support ongoing manipulation of DoF:* While we have shown that the robot can monitor an ongoing DoF manipulation based on the information perceived online, we did not demonstrate yet that the information can be applied to control and generate new interactions, increasing the relevance of the information for the task.

In this chapter we will address these three limitations.

First, we will increase the versatility and the robustness of robot perception with a novel perceptual system that integrates multiple sensor modalities in a cross-modal manner. Cross-modal perception is a form of multi-modal perception that leverages the information obtained from one modality to facilitate the interpretation of signals of another modality. We will leverage cross-modal information using our proposed approach based on coupled recursive estimation for perception for robot manipulation. The goal is to create a system that robustly

perceives kinematics of articulated objects in challenging unstructured environments, e.g. when the lighting conditions are adversarial or the properties of the task impedes the direct visualization of the articulated objects. We will combine vision and proprioception to perceive dynamic properties of the joints, like the force to overcome stiction and kinetic friction.

Second, given that one of the modalities we will integrate –proprioception– contains direct information about the interaction, we will propose a simple method to learn interaction forward models, i.e. models relating the interaction to changes in the sensor signals. These models can be exploited for perception and for controlling the robot towards a manipulation goal as well.

And third, we will present and evaluate motion generation methods to explore unknown articulated objects safely and to exploit the information acquired online to generate new robot motion trajectories.

6.1 Cross-Modal Integration of Sensor Information for Interactive Perception

We will begin this chapter presenting a perceptual system for the cross-modal perception of articulated objects. In cross-modal integration the information of one modality is used as prior to interpret signals in another modality. Cross-modal integration is able to leverage regularities in the combined multi-modal signal space, whereas traditional multi-modal perception interprets the individual sensor signals independently and then combines the results.

An example of cross-modal perception in humans is the *McGurk effect* (McGurk & MacDonald, 1976). In this perceptual illusion, a subject watches a video of a person pronouncing the same syllable repeatedly but dubbed with different audio utterances, such as ba-ba or ga-ga. The subject is convinced to see a change in the lip motion when the dubbed sound changes. The subject misjudges the visual information because the video in fact depicts the person saying the exact same syllables, but the sound has been altered to play the different phonemes (see Figure 2.4 in Section 2 for an illustration of the McGurk experiments). Interestingly, the inverse effect –the wrong perception of changing auditory phonemes due to changes in the facial motion in a dubbed video– has also been reported. These illusions occur because the auditory cue influences the perception of the facial motions, and vice-versa. As a result, the identical facial motions (or sounds in the inverse effect) are perceived as being different. The illusions demonstrate the dependency on visual cues of hearing, as well as the influence of auditory cues on seeing. This cross-modal interpretation of multiple modalities is necessary for the robust perception of speech (Rosenblum et al., 2007). If humans were simply merging modalities, the subject would notice the contradiction in the visual and audio signals.

We will leverage cross-modality in robot perception using our proposed approach based on coupled recursive estimation processes. As in previous chapters, each estimation process in our proposed system addresses a perceptual subproblem. The coupling of these components allows us to use the estimated value of one recursive estimation loop as a prior for others, even across different modalities. For example, our system predicts motion from proprioception and uses it to interpret visual perception. The system also combines proprioception and vision to perceive the type of grasp achieved by the robot hand, then uses this information as a prior to disambiguate proprioceptive signals. Using information about the kinematic structure, the system can interpret the wrenches as evidences of the dynamic properties of the articulated object. These examples illustrate that information from multiple modalities and multiple perceptual subproblems propagates through the network, leading to robust online perception.

We will also evaluate experimentally if the information perceived during an interaction can

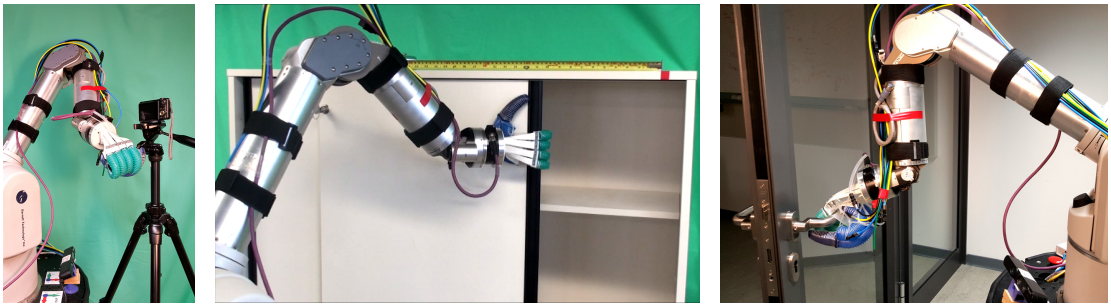


Figure 6.1: Our robot manipulating three articulated objects (a cupboard door, a glass door, and a camera tripod) and perceiving their kinematic structure; the robot uses a RBO2 soft-hand (Deimel & Brock, 2016) for safe interactions; the exploratory interaction is steered using our velocity-impedance controller; our online perceptual system integrating vision (RGB-D stream) and proprioception (joint encoders, force-torque and air-pressure signals) acquires information from the exploration and generates robot trajectories for new manipulation tasks [© 2017 IEEE]

be used to monitor the manipulation and to generate new trajectories (see Figure 6.1).

6.2 Related Work

The work we present here is 1) a new interactive perception system integrating multiple sensor modalities in a cross-modal manner. On the path towards a multi-modal system, we also propose 2) a novel perceptual approach to perceive kinematic structures based only on proprioception. Finally, using the information about the kinematic structure as prior, we propose 3) a system to the following infer dynamic properties of the articulated objects: the wrench to overcome stiction and initiate an actuation, and the wrench to maintain an ongoing actuation. We will now discuss these thee areas of related work: multi-modal perception (in the context of interactive perception), proprioception-based perception of kinematic structures, and the estimation of dynamic properties of articulated objects.

6.2.1 Multi-Modal Perception

Multi-modality has been applied previously in recursive filters to overcome limitations of uni-modal robotic perceptual systems. The common methodology is to estimate a correction by fusing the multi-modal signal into a single estimate (Ilonen et al., 2014, Hebert et al., 2012). This approach does not leverage information from one modality to help interpret the other. Instead, we exploit the results from one recursive filter as priors in the others to obtain more information. This cross-modal exploitation was applied successfully by Garcia Cifuentes et al. (2017) to track a robot arm and an object from a multi-modal stream. However, their method requires models of the arm and the object, and cannot be applied to perceive previously unseen articulated objects.

Previous interactive perception methods applied to object segmentation and recognition (van Hoof et al., 2012, Sinapov et al., 2011), shape reconstruction (Xu et al., 2015), and the perception of dynamic (Endres et al., 2013) and kinematic properties (Hausman et al., 2015) of articulated objects are based on a single modality, or use multiple sensor modalities, but they apply one independently to each perceptual subtask. This neglects the benefits of

a tighter integration and exploitation of the interdependencies between subtasks. The multi-modal interactive perception system we present in this chapter improves robustness and versatility over previous approaches and the systems we presented in previous chapters, by using cross-modal communication to acquire priors from one modality for the interpretation of the other.

6.2.2 Perceiving Kinematic Models From Proprioception

Previous approaches show that the kinematic properties of an articulated object can be perceived from end-effector trajectories (Sturm et al., 2010a) and applied wrenches (Karayiannidis et al., 2016) during interaction. These methods are based on two assumptions that limit their applicability: 1) There is only one moving part connected with a joint to the static environment, and 2) there is no translation between the end-effector and the moving part during the interaction. We leverage information from vision to correctly interpret proprioception and to overcome these limitations, estimating the correct grasp model and perceiving more complex kinematic structures.

6.2.3 Perceiving Dynamic Models of Articulated Objects

Atkeson et al. (1986) presented an approach to estimate the inertia properties of a grasped object from interactions, based on the information from the robot's encoders. When it comes to the estimation of dynamic properties of articulated objects most existing approaches generate models of the dynamics of a robot arm, based also on the robot's encoders signals (Xinjilefu et al., 2014, Ma & Hollerbach, 1996). However, few methods addressed the estimation of dynamics of external articulated objects for which internal joint sensors are not available. Endres et al. (2013) presented an approach to learn dynamic models of doors with a force/torque sensor on robot's wrist. Their model is composed by a parametric component representing the moment of inertia, and a non-parametric model (a Gaussian process) representing the deceleration of the mechanism due to friction (some kind of viscous friction model). To obtain the kinematic information necessary to estimate the dynamics (e.g. velocity of the actuation of the joint), the authors employed the method by Sturm et al. (2011). The authors demonstrate that the learned dynamic model is useful for manipulation by planning and executing swing interactions on the door that bring it to a predefined goal. Our model differs from theirs since we are focussed on controlled interactions while grasping the articulated object rather than dynamic swinging. In our manipulation scenario, the inertia and viscous friction effects are negligible and we focus on the force necessary to initiate and to maintain the joint actuation.

In a different type of work, Jain et al. (2010) presented a study of doors and drawers from human interactions. They estimated kinematic and dynamic properties of several everyday objects in human environments. The mechanisms of their study contained springs that create dynamic effects depending on the configuration of the joint. They also observed that the highest forces are required to initiate the actuation. This supports our proposed simple model of the joint dynamics that also acknowledges the importance of the force to initiate the actuation, the force to overcome stiction.

Conclusions and comparison to the proposed approach: Most existing interactive perception approaches are based on a single sensor modality, or use one modality independently for each perceptual subtask. These methods fail if the environmental conditions are adversarial for the modality they use. Differently, we aim to exploit the interdependencies

between modalities passing information across subprocesses so that the resulting perceptual system is versatile to cope with different environment and task conditions.

While some initial work demonstrated that it is possible to perceive kinematics of articulated objects from proprioception, these methods made strong assumptions about the problem (e.g. the object possesses only one joint and it that connects it to the environment) that we will relax combining multiple sensor source. We will additionally address the estimation of the dynamic properties of articulated objects, which was not studied extensively in the literature.

6.3 Proprioception-Based Perception of Kinematic Properties

Our goal is to integrate vision and proprioception into a single multi-modal system that exploits cross-modal information. We will integrate the system presented in Chapter 4 to a novel perceptual system based on proprioception. We leave out the extension of Chapter 5 to simplify the evaluation of the benefits of the cross-modal integration. The integrated system with its most relevant recursive filters is depicted in Figure 6.2: on the left, the visual system of Chapter 4 and in the middle and the right, the novel system of this chapter. In this section, we will present a novel perceptual system for kinematic models based on proprioception. In the next section (Section 6.4), we will explain how to integrate vision and proprioception such that both systems leverage cross-modal information, and how to exploit their combination to perceive the dynamic properties of the articulated objects.

Proprioception refers to sensory information about the configuration of the robot's own body (kinesthetics) and the forces it exerts (haptics). Our robot obtains proprioceptive signals from a force-torque sensor on its wrist, from the air-pressure sensors monitoring the chambers of its pneumatic soft hand, and from the joints encoders of its arm. The goal is to use these signals to perceive the motion of the object the robot is interacting with as well as its motion constraints, leading to the object's kinematic model.

The motion of the interacted body and the robot's end-effector are coupled, as their relative motion is constrained by their contact. Because our robot uses a soft hand for the interaction, the relative motion between the hand and the object depends on the deformation of the hand and on the remaining degrees of freedom of the contact interaction (grasp).

We factorize the perception of articulated bodies into the following five subproblems: The estimation of A) the motion of the end-effector, B) the bending state of the soft-hand, C) the kinematic model of the grasp, D) the motion of the interacted body, and E) the constraints in the motion of the interacted body. Figure 6.2 depicts the recursive filters addressing these subproblems, together with the filters of the vision-based system. The estimation of motion of the interacted body is subsumed with the estimation of other bodies from vision in the box "Rigid Body Motion".

Blue arrows in the figure represent estimated states passed as measurements to the next process. Thus, the originating process acts as a virtual sensor for the second process. We used this communication pattern to "inject" more priors at each filter until we solve the entire perceptual problem.

Red arrows in the figure represent predicted measurements passed as state predictions to the next process. Exploiting this communication pattern, we restrict the space of possible solutions of one subproblem using the other processes (and their priors) as alternative forward and measurement models. We will exploit similar intercommunication patterns in the perceptual system based on proprioception, and will exploit cross-modal information in the multi-modal system.

In the following, we will explain how we solve the subproblems of the proprioception-based system using coupled recursion estimation (the last subproblem is solved the same way as the

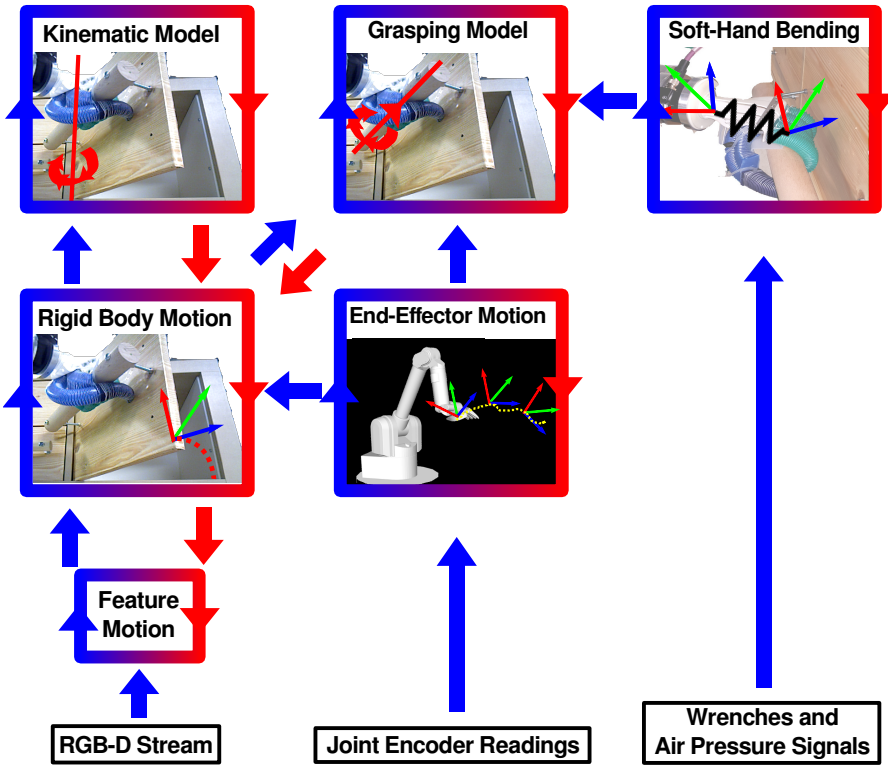


Figure 6.2: Our proposed system for interactive perception of kinematic properties of articulated objects based on cross-modal information between coupled recursive filters; bottom: input sensor signals; arrows: information flow between filters and across modalities (blue: input measurements, red: alternative predictions) [© 2017 IEEE]

estimation of kinematic models from rigid body motion in Chapter 4).

6.3.1 Estimation of End-Effector Motion

The first recursive filter estimates the motion of the end-effector. The state of the end-effector is represented by the end-effector's pose and velocity, $\boldsymbol{x}_t^{ee} \sim \mathcal{N}((_{ee}p_t, _{ee}\eta_t), P_t^{ee})$. To predict the next state based on the previous estimate, we use a velocity-based kinematic update:

$$_{ee}\hat{p}_t = \Delta_t \, _{ee}\eta_{t-1} \oplus _{ee}p_{t-1} \tag{6.1}$$

$$_{ee}\hat{\eta}_t = _{ee}\eta_{t-1} \tag{6.2}$$

The measurements for the estimation of the end-effector motion are the pose and velocity of each robot joint provided by the robot's joint encoders, $z^{ee} = (q_j, \dot{q}_j)$, $j \in \{0, \ldots, J-1\}$, where J = robot's number of joints. Predicting this measurement based on the state would require to solve an inverse kinematics problem. Instead, we combine the measurements on the robot's joints' poses and velocities with prior knowledge about the robot's embodiment and forward kinematics. This way, we obtain a direct measurement of the end-effector's pose and velocity, which we integrate recursively:

$$z'^{ee} = (_{ee}p_z, _{ee}\eta_z) \tag{6.3}$$

Figure 6.3: Effect of the deformation of the soft-hand; left: hand in the nominal state; middle and right: hand in the bent state after a motion of the end-effector without motion of the interacted body (a door handle) [© 2017 IEEE]

where the sub-index z indicates that they are measurements (direct observations of the state).

With this measurement model, the estimation of end-effector motion corresponds to a filtering of the proprioceptive measurements, weighted by the uncertainty of the observations, R_t^{ee}, that we set proportional to the velocity (fast end-effector motion corresponds to highly uncertain pose measurements).

6.3.2 Estimation of Hand Bending

When the robot interacts with an object, the soft hand deforms (bends). This changes the relative pose between the hand and the object (see Figure 6.3). In the second recursive filter, we estimate the consequences of this bending effect.

We represent the bending state of the soft hand as the relative transformation between the *nominal* end-effector pose (estimated by the filter described above) and the pose of a virtual body we call *bent* end-effector (defining the hand's physical pose),

$$\boldsymbol{x}_t^{bent} \sim \mathcal{N}({}^{ee}_{bee}p_t, P_t^{bent}) \tag{6.4}$$

$${}^{ee}_{bee}p = {}^{bee}p \ominus {}_{ee}p \tag{6.5}$$

where $\ominus$ is the inverse composition of poses. We assume that the bending state remains constant between consecutive time steps, $\hat{x}_t^{bent} = x_{t-1}^{bent}$.

We use as measurements the signals of the proprioceptive stream that correlate to the bending of the hand. These are the wrenches measured at the robot's wrist and the pressure values in the four air chambers of the soft-hand:

$$z^{bent} = (w, a) \tag{6.6}$$

where the wrenches are $w \in \mathbb{R}^6$, $w = (f, \tau)^T$, and the air pressure signals $a \in \mathbb{R}^4$.

Defining an analytic measurement model relating bending and proprioceptive signals for a complex soft-manipulator as the RBO Hand 2 (Deimel & Brock, 2016) is a difficult problem (Smoljkic et al., 2015). We will adopt a data-driven approach and learn from experiences

a model that transforms the proprioceptive signals into direct observations of the bending state:

$$f(w, a) = \boldsymbol{z}'^{bent} \sim \mathcal{N}({}^{ee}_{bee}p_z, R_z^{ee-bee}) \tag{6.7}$$

where the sub-index z indicates that they are measurements (direct observations of the state).

We approximate the model f using an artificial neural network. To obtain labeled data to train the model, we execute 15 interactions of the robot grasping an object that is rigidly attached to the environment. We record the wrenches and the pressure signals at different relative poses of the bent soft hand with respect to the nominal pose during these interactions. We then train a multi-layered perceptron regressor (MLPR1) to map from wrenches and pressure signals to the 6D relative pose observations.

To integrate the observations recursively, we also need to learn their uncertainty, R_z^{ee-bee}. Following the approach proposed by Rojas (1996), we train several partial MLPRs, leaving out groups of two trials, and computing the standard deviation between predictions from these partial MLPRs and the fully trained MLPR. We then train a second MLPR (MLPR2), mapping wrenches and pressure signals to the standard deviation of the regressor. With this procedure, the second MLPR learns the difficulty of the transformation problem for each input signal and allows us to filter proprioceptive signals into a robust estimate of the hand bending state.

6.3.3 Estimation of Interaction-Grasp Model

In the third recursive filter, we estimate a kinematic model of the grasp. The grasp model explains the kinematic constraints between the motion of the bent end-effector and the interacted body. We maintain and estimate independently the parameters of four filters for the grasp models, one for each type of grasp that our anthropomorphic soft-hand can perform: *(i)* perfect grasp (no relative motion), *(ii)* revolute grasp (allowing rotation around the grasping axis), *(iii)* cylindrical grasp (allowing rotation around and translation along the grasping axis), and *(iv)* failed grasp (no motion constraint).

For revolute and cylindrical grasps, the state of the filter is parametrized by the orientation of the axis (azimuth $\phi^{gr,r}$ or $\phi^{gr,c}$, and elevation $\theta^{gr,r}$ or $\theta^{gr,c}$ in spherical coordinates), and by a point on the axis ($p^{gr,r} \in \mathbb{R}^3$ or $p^{gr,c} \in \mathbb{R}^3$). For the perfect grasp, the state is parametrized by a fixed 6D pose between the bent end-effector and the interacted body (${}^{bee}_{ib}p$). The failed grasp does not impose any motion constraints and therefore does not have any parameters to estimate, $x^{gr,f} = \emptyset$. We initialize these parameters based on the morphology of the hand and an initial low uncertainty, indicating that this initial estimate for the parameters of the grasping models should be trusted.

The estimation of the grasp model leverages the coupling between filters to obtain measurements. The estimates of the pose of the bent hand (from the previous two filters) and the interacted body (from the next filter) are combined to generate a measurement:

$$z^{gr} = f(x^{ee}, x^{bent}, x^{ib}) = {}_{ib}p \ominus ({}_{ee}p \oplus {}^{ee}_{bee}p) = {}^{bee}_{ib}p \tag{6.8}$$

The estimation of the parameters and the most likely type are performed similarly to the estimation of joint parameters of a kinematic model in Chapter 4. A difference with respect to our approach to estimate joints of kinematic models of articulated objects is that the grasping model estimation does not include the estimation of the joint state. The predicted measurements (relative poses) are a function of this joint state. For each measurement, we compute the current joint state of each model that minimizes the difference between the predicted

relative pose (a function of the joint state) and the measured relative pose. We will use this minimum difference to evaluate the most likely model.

Given the low uncertainty of the initial estimates of the grasping parameters, the method presented here can be seen as a model-selection approach (among a set of predefined models). Later in this chapter (Section 6.8), we will present a method to replace the model selection by a model-learning approach. We will learn a full interaction-grasp model in the form of a Jacobian matrix from experiences (pairs of interactions and correlated changes in the environment). This second approach reduces the dependency on a good initial estimate of the parameters of the grasping model and can be applied to end-effectors of unknown morphology.

6.3.4 Estimation of Interacted Body Motion

The fourth recursive filter estimates the motion of the body the robot interacts with. The state of the interacted body is represented by its pose, $x^{ib} = {}_{ib}p$. The prediction of its next state also leverages the coupling between filters: the change in pose depends on the motion of the end-effector, corrected with the bending effect and propagated through the grasping model,

$$ {}_{ib}p_t = (\bar{x}^{gr} \, {}^{ee}_{bee}Ad \, {}_{ee}\eta\Delta_t) \oplus {}_{ib}p_{t-1} \tag{6.9} $$

where $\bar{x}^{gr}$ is a 6×6 matrix representation of the kinematic constraints of the grasping model and ${}^{ee}_{bee}Ad$ is the adjoint transformation associated with the bending effect.

None of the proprioceptive signals can be used as observations of the motion of the interacted body, and thus the predicted distribution over the next state becomes the current belief.

Limitations of Proprioception-Based estimation of Kinematic Models The first limitation of the system based only on proprioception is due to the mutual dependency between the estimation of the interacted body motion and the grasp model. The motion of the interacted body is estimated based on the current belief over the grasp model. In turn, the grasp model is updated based on the estimated motion of the interacted body. This mutual dependency effectively reaffirms the initial prior distribution over the grasp model. The accuracy of the estimated interacted body motion depends thus on the accuracy of this grasp model prior.

The second limitation is that the proprioceptive signals, because of their limited range, only provide measurements about the state of the robot and the responses from the interacted body. The system can only perceive a single body connected by a joint to the environment, defining the kinematic model. Overcoming both limitations will require additional prior knowledge that our integrated system will obtain from vision by leveraging the cross-modal information.

6.4 Integration of Vision and Proprioception

6.4.1 Perceiving Kinematic Properties

Once we have explained how to extract information from each modality –from vision in Chapter 4; from proprioception in the previous section– we will explain how to leverage information from one modality to help interpret the other. The proposed multi-modal system exploits cross-modal information to overcome the limitations of a uni-modal perception system.

Predictions about the motion of the interacted body from proprioception are leveraged to correctly assign visual point features, even under challenging visual conditions, e.g. with very low lighting or large occlusions. The features can be used as observations to correct the proprioceptive predictions, $z^{ib} = x^{fm|ib}$, where $x^{fm|ib}$ are the visual point features assigned to the interacted body. The cross-modal predictions from proprioception to vision and the corrections from vision to proprioception lead to a new estimate that breaks the mutual dependency of the proprioception-only system, $x^{ib} = {}_{ib}p$.

Using the interacted body motion perceived from cross-modal information, our system can correctly interpret the constraints in the bent end-effector motion perceived from proprioception, and retrieve the kinematic grasp model, $\boldsymbol{x}^{gr}$. The type and parameters of the grasp model are inferred from the relative motion between the bent end-effector and cross-modal estimates of the interacted body motion (Section 6.3.3):$z^{gr} = {}_{bee}p \ominus {}_{ib}p$.

The system can use grasp model estimates from cross-modal information as prior to further interpret proprioceptive signals when the visual modality degenerates (e.g. the object goes out of the field of view, or is occluded, or due to extremely bad lighting conditions or not enough visual texture). The prior obtained from cross-modal information is sufficient to estimate the kinematic model of the interacted body using only proprioceptive signals.

The integrated system correctly interprets the constraints in the motion of the interacted body perceived from proprioception, leveraging information from vision. The system perceives from vision the motion of other bodies apart from the directly interacted one and uses this prior to analyze the motion constraints of the interacted body from proprioception. The integrated system based on cross-modal information can perceive complex kinematic models with multiple joints or when the interacted body is not connected to the static environment, $\boldsymbol{x}^{joint}$.

6.4.2 Perceiving Dynamic Properties

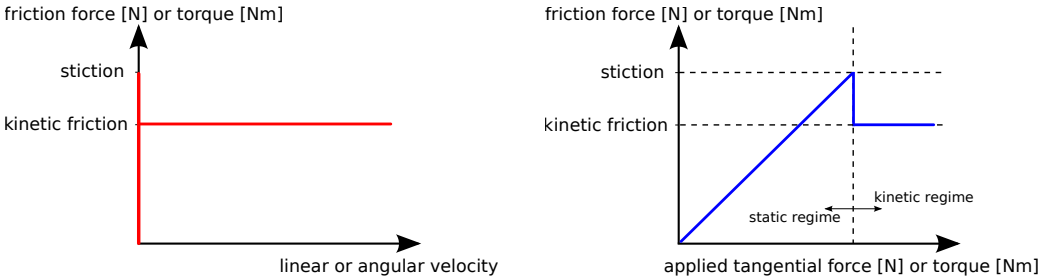
The combination of vision and proprioception allows the robot to infer new information about an actuated articulated object: its dynamic properties. The dynamic properties relate the forces and torques applied to an object with their kinematic effects.

Because in this thesis we are interested in controlled and safe robot interactions with constrained mechanisms, the wrenches the robot applies on the objects are bounded, and so are the joint accelerations they generate. In these conditions, we can neglect the inertia effects from our analysis of the dynamics, and we apply a *quasi-static* analysis, where the dominating term is the friction. We also deem the effect of other dynamic processes (e.g. damping and viscous friction, inertia) to be negligible for the objects and the safe contact interactions we consider.

Roboticists have developed multiple models to explain the friction effects in articulated mechanisms. These models vary in complexity and the number of parameters. In our estimation method, we use the Coulomb friction model and estimate two parameters: *stiction* and *constant kinetic friction* (Dupont, 1990). Both values and their relationship to the actuation of the mechanism are depicted in Figure 6.4, and explained below.

The contact surface between two bodies (e.g. within a kinematic joint) creates friction forces/torques[1]. We can distinguish between two dynamic regimes with different friction effects, depending on the relative motion between the surfaces: If the bodies do not move

[1]In the rest of the text we will use the terms *force/torque* (instead of wrench) and *linear/angular velocity* to keep the explanation general for any type of kinematic constraint; however, please note that for prismatic joints, we should only consider forces and linear velocities, and for revolute joints, consider torques and angular velocities.



(a) Friction as a function of the velocity; without motion (static regime), friction ranges between zero and *stiction*; with motion (kinetic regime), friction is independent of the velocity and equal to the *constant kinetic friction*; if in motion the applied tangential force/torque decays under the *constant kinetic friction* the actuation stops

(b) Friction as function of the applied tangential force/torque; without motion (static regime), friction is equal to the applied tangential force/torque; with motion (kinetic regime), friction is constant and equal to the *constant kinetic friction*; if in motion the applied tangential force/torque decays under the *constant kinetic friction* the actuation stops

Figure 6.4: Coulomb model of friction; motion of a joint begins when the applied tangential force (for prismatic joints) or torque (for revolute joints) overcomes *stiction*; friction during motion is constant and equal to *kinetic friction*; if the applied tangential force/torque decreases under the *constant kinetic friction*, the motion of the joint stops

with respect to each other, the equilibrium of forces/torques is in the static regime, and the force/torque opposing the applied force/torque is called *static friction*. Static friction is a force/torque equal in magnitude and opposite in direction to the applied force/torque in the allowed dimension by the joint, the so-called *tangential force/torque*. Forces/torques in the dimensions constrained by the joint, the so-called *normal forces/torques*, do not generate motion and are always counteracted by the mechanism. Therefore, we do not need to consider the normal component of the applied force/torque in our dynamic analysis.

When the applied tangential force/torque overcomes a threshold, the two bodies begin to move with respect to each other. This threshold is called *stiction* and is one of the parameters we estimate in our model since it is relevant information for the manipulation.

During motion, the equilibrium of forces/torques is in the kinetic regime, and the force/torque opposing the motion is called *kinetic friction*. We assume this force/torque to be approximately constant and independent of the relative velocity (Coulomb model). We call this value *(constant) kinetic friction* and it is the second parameter we estimate in our model. If the applied tangential force/torque decays under the (constant) kinetic friction, the motion decelerates and stops quickly, and the equilibrium of forces/torques returns to the static regime. Knowledge about the force to overcome stiction and kinetic friction allows to plan safe interaction, as we will see in the next chapter.

Given the previous definitions, we propose to estimate the parameters of the friction model recursively using a particle filter. The state of the filter is a set of particles representing the distribution over dynamic parameters of the joint: $\boldsymbol{x}^{dyn} = \{p^{dyn,i}\}, i \in \{1 \dots N_{dyn}\}$. Each particle contains a hypothesis of the dynamic parameters, $p^{dyn,i} = (S^i, KF^i)$.

The observations to update the state of the filter, z^{dyn}, are joint velocities, $\dot{q}$, and the magnitude of applied tangential force/torque, $\|ft^{tan}\|$. We use here the term "force/torque" and the symbol *ft* to indicate either force or torque (depending on the type of joint) and not full 6D wrenches.

As explained before, our dynamics (friction) model is independent of the magnitude of the joint velocity. We will simplify the measurements and consider $\dot{q}$ a binary variable, indicating if the joint is moving or not, $\dot{q} \in \{0, 1\}$.

In the following, we will first assume that the measurements for the estimation, z^{dyn}, are given and explain how to update the state of the filter. Then, we will explain how we obtain the measurements leveraging cross-modal information from other subprocesses of the perceptual system.

The way we use the measured tangential force/torque to update the filter state (the measurement model) depends on the current dynamics regime: static or kinetic, or boundary state. To evaluate the current dynamics regime, we compare the current and previous observations of the joint motion, and distinguish four cases:

- The joint was not moving before and is not moving now: This case indicates that in both the previous and the current steps, the tangential force/torque is not enough to overcome stiction. The current tangential force/torque is *under* stiction.
- The joint was not moving before and is moving now: The tangential force/torque now is enough to overcome stiction and initiate motion. The current tangential force/torque is *over* stiction.
- The joint was moving before and is moving now: The tangential force/torque is enough to maintain motion. The current tangential force/torque is *over* kinetic friction.
- The joint was moving before and is not moving now: Kinetic friction dominates the quasi-static scenario and impedes the motion now. This effect indicates that the current tangential force/torque is *under* the kinetic friction.

These four cases lead to four measurement updates, with different importance functions for the particles.

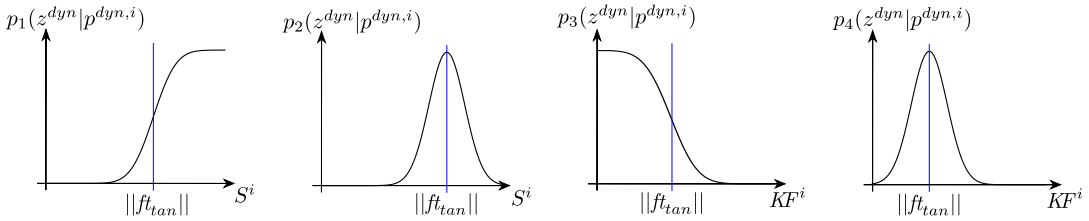
In the first case, the particles predicting motion should receive a lower importance factor than the particles correctly predicting no motion, especially the ones that assume that the threshold to initiate motion (stiction) was largely overcome. For a given measured tangential force/torque, $\|ft_{tan}\|$, we define the importance factor of a particle $p^{dyn,i} = (S^i, KF^i)$ with the function

$$p_1(z^{dyn}|p^{dyn,i}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{S^i - \|ft_{tan}\|}{\sigma_{ft}\sqrt{2}}\right)\right] \tag{6.10}$$

The equation above is the accumulative density function of a Gaussian distribution with mean $\|fm_{tan}\|$ and covariance σ_{ft} evaluated at the stiction value of the particle, S^i. We will see later how to obtain these mean and covariance values that represent the measured tangential force/torque and its uncertainty. $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x exp\left(-t^2\right) dt$ is the *error function*, the probability of a random variable normally distributed with mean 0 and variance 1/2 being in the range $[-x, +x]$.

The previous importance factor function is depicted in Figure 6.5a. While this is not a well-defined probability density function (its integral over the entire space is not equal to one), the renormalization of the particles before the resampling step assures that the filter is probabilistically consistent. The function penalizes the particles where the stiction was largely surpassed by the applied tangential force/torque.

In the second case, the particles that predict correctly that the motion starts in the current step should receive a higher weight than any other, i.e. a higher weight than particles



(a) Case 1: $\dot{q}_{t-1} = 0$ and $\dot{q}_t = 0$

(b) Case 2: $\dot{q}_{t-1} = 0$ and $\dot{q}_t = 1$

(c) Case 3: $\dot{q}_{t-1} = 1$ and $\dot{q}_t = 1$

(d) Case 4: $\dot{q}_{t-1} = 1$ and $\dot{q}_t = 0$

Figure 6.5: Four different importance factor (likelihood) functions for the four cases, depending on whether the joint was actuated or not in the previous and current steps; the functions are centered at the mean applied tangential force/torque, μ_{FT} and "spread" accordingly to its covariance, σ_{FT}; the functions are applied to the stiction (S) or the kinetic friction (KF) of the particles, $\{p^i_{dyn}\}$

predicting no motion because the tangential force/torque is under stiction, and than particles predicting that the threshold to initiate motion (stiction) was largely overcome. For a given magnitude of the tangential force/torque, $\|ft_{tan}\|$, we define the importance factor of a particle $p^{dyn,i} = (S^i, KF^i)$ as

$$p_2(z^{dyn}|p^{dyn,i}) = \frac{1}{\sqrt{2\pi\sigma_{FT}^2}} exp\left(-\frac{S^i - \|ft_{tan}\|}{2\sigma_{FT}^2}\right) \tag{6.11}$$

This importance factor function is depicted in Figure 6.5b. The function benefits the particles that correctly predicted that the motion of the joint should begin now (values of stiction close to the measured $\|ft_{tan}\|$).

The importance factor of the particles in the third and fourth cases are analogous to the first and second cases, but based on the particle's kinetic friction value:

$$p_3(z^{dyn}|p^{dyn,i}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{KF^i - \|ft_{tan}\|}{\sigma_{ft}\sqrt{2}}\right)\right] \tag{6.12}$$

and

$$p_4(z^{dyn}|p^{dyn,i}) = \frac{1}{\sqrt{2\pi\sigma_{ft}^2}} exp\left(-\frac{KF^i - \|ft_{tan}\|}{2\sigma_{ft}^2}\right) \tag{6.13}$$

The importance factor functions are depicted in Figure 6.5c and Figure 6.5d.

At each step we use the previous and current observations of the motion of the joints, $\dot{q}_{t-1}$ and $\dot{q}_t$, to select the right importance factor function to update the filter state. These observations are provided by an existing subprocess of the perceptual system, the estimation of the kinematic model from visual data. In an example of cross-modal integration, information from the estimation of kinematics is used as a prior to correctly interpret the measured tangential force/torque, $\|ft_{tan}\|$. The subprocess estimating kinematics provides also prior information to decompose the applied force/torque into the tangential and the normal components. In the rest of this section, we will explain how to obtain these measurements of the tangential force/torque relative to a given kinematic joint.

We compute the magnitude of the tangential force/torque applied by the robot in a two-step process. First, we need to compute the force/torque the robot applies on the object, and second, we need to decompose the applied force into the tangential and normal components.

To compute the applied force/torque we use measurements from a sensor attached to the robot's end-effector. We account for the effect of the gravity on the end-effector by subtracting the end-effector's weight from the raw force/torque readings, w, using the end-effectors mass, center of mass, and pose. The remaining force/torque signal is the applied force/torque by the robot on the object, $w_{app} = (f_{app}, \tau_{app})^T$.

To obtain the tangential force/torque, we geometrically compute the decomposition of the applied force/torque given the joint axis definition. Because in our perceptual system the joint axis are defined by probability distributions over joint parameters, we sample multiple joint axes from these distributions and project the applied force/torque onto the hypothesized axes. Then, we collect the tangential projections, compute their norm and fit a Gaussian to the resulting norm samples. The result of this process, $\|ft_{tan}\|$ and σ_{ft}, is a probability distribution over the applied tangential force/torque, grounded in the uncertainty about the kinematic model. We will explain in detail the geometric decomposition for a prismatic and revolute joint.

For a prismatic joint, the tangential component corresponds directly to the projection of the applied force onto the direction of the axis f_{tan} (see Figure 6.6a).

For a revolute joint, we first compute the applied force/torque at the sampled axis, $w'_{app} = (f'_{app}, \tau'_{app})^T$, from the applied force/torque at the point of contact, $w_{app} = (f_{app}, \tau_{app})^T$. For this transformation we assume that both locations are on the same rigid body, i.e. the robot applies forces/torques on one of the two links connected the joint. The transformation of the force/torque to another point on the same rigid body is given by

$$(f'_{app}, \tau'_{app}) = (f_{app}, \tau_{app} + \bar{r} \times f_{app}) \tag{6.14}$$

where $\bar{r}$ is the vector connecting the point of application of the force/torque and one point on the sampled axis. Finally, we project the transformed torque, τ'_{app}, onto the direction of the sampled revolute axis and obtain the tangential τ_{tan} (see Figure 6.6b).

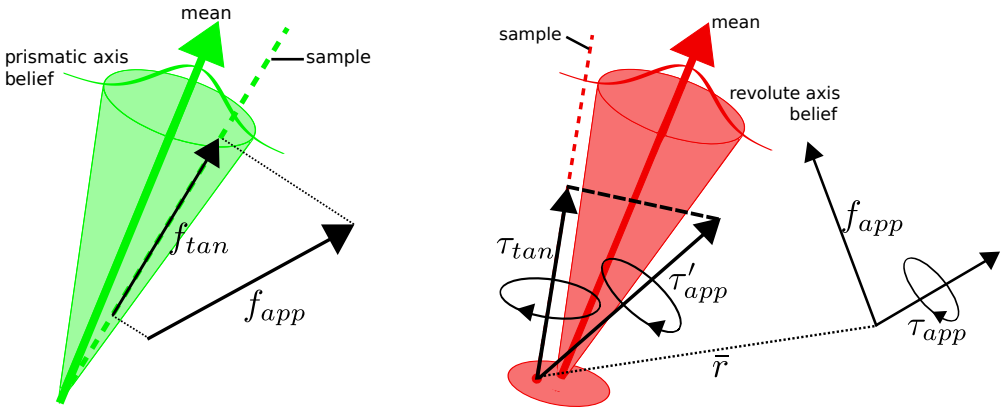
To obtain measurements of the tangential applied force/torque our method integrates information from two other subprocesses in the perceptual system: the estimation of kinematic models (to project the applied force/torque into the joint axis) and the estimation of end-effector motion (to transform the measured wrenches from the reference frame of the robot's wrist to the spatial configuration of the joint).

Our particle filter method increasingly reduces the uncertainty over the dynamic parameters as more haptic measurements are acquired, especially when the joint starts or stops moving. An example of the evolution of the estimated dynamic parameters from continuously arriving haptic measurements is depicted in Figure 6.8.

6.5 Robot Motion Generation and Control

Generating a multi-modal stream rich in information depends on the strategy to control the robot's interaction. Our goal is to generate motion in the dimensions allowed by the (initially unknown) kinematic structure. This adaptive behavior can be achieved using a compliant controller based on the force-torque signals (Bruyninckx & Schutter, 1996).

We use an operational space impedance controller on the Lie Group $SE(3)$ (Part, 1985, Park & Kim, 2014) to adapt a desired trajectory of the 6D pose of the end-effector, ${}_{ee}p^{des}(t)$, based on the signals from the force-torque sensor. The operation space framework relates robot's joints motion to end-effector motion. Impedance control relates deviations from a given end-effector trajectory to reactive forces. Impedance control is thus a crucial technique



(a) Geometric projection (f_{tan}) of the applied force (f_{app}) onto the direction of a sample from the belief over the prismatic joint axis

(b) Geometric projection (τ_{tan}) of the applied torque onto the direction of a sample from the belief over the revolute joint axis; the applied wrench is transformed from the point of application (f_{app}, τ_{app}) to one point in the sampled axis (τ'_{app}) assumed they are points on the same rigid body

Figure 6.6: Geometric projection of the applied wrench onto a sample of a prismatic **(a)** or a revolute **(b)** joint; translucent cones indicate one standard deviation to the mean of the axis orientation; the translucent sphere indicates one standard deviation to the mean position of the axis; the projection of the applied wrench decompose it into a tangential components (f_{tan} and τ_{tan}) and normal components (not shown)

for robots to interact with constrained mechanisms in a safe manner, since it allows the robot to adapt to the constraints of the mechanism with controlled forces.

The behavior of this controller is parametrized by three 6×6 matrices –stiffness (K), damping (D), and mass (M)– that transform virtually the end-effector into a spring-mass damped system with different reactive behavior for each dimension. In a nutshell, the impedance controller reacts with the following end-effector wrench to a deviation from the desired trajectory:

$$ {}_{ee}w^{imp} = M({}_{ee}\dot{\eta}^{meas} - {}_{ee}\dot{\eta}^{des}) + B({}_{ee}\eta^{meas} - {}_{ee}\eta^{des}) + K({}_{ee}p^{meas} \ominus {}_{ee}p^{des}) \tag{6.15} $$

where ${}_{ee}\dot{\eta}$ is the end-effector's acceleration, ${}_{ee}\eta$ its velocity and ${}_{ee}p$ its pose (in exponential coordinates), and the super-indices *meas* and *des* indicate the measured and the desired values. The former equation indicates that the robot compensates a deviation from a given trajectory (desired values) exerting a wrench that is defined by the stiffness, damping and mass parameters. To compute the robot commands (robot's joint torques) to generate the desired wrenches, we use the operational space formalism (Khatib, 1987). The operational space control method is an approach to implement inverse dynamics control of ${}_{ee}w$ directly in task space. While we did not make any novel contribution to operational space or impedance control, we developed the necessary robot skills to implement them.

The aforementioned controller can adapt an initial exploratory trajectory. To generate such a trajectory for articulated objects we propose a velocity-based controller that sets a constant goal in velocity and, at each step, a new goal for the end-effector pose, ${}_{ee}p_t^{des}$, based on the error between the measured and desired velocity twists:

$$ {}_{ee}p_t^{des} = k_p({}_{ee}\eta^{meas} - {}_{ee}\eta^{des}) \oplus {}_{ee}p_{t-1}^{des} \tag{6.16} $$

$$ {}_{ee}\eta_t^{des} = {}_{ee}\eta_{t-1}^{des} \tag{6.17} $$

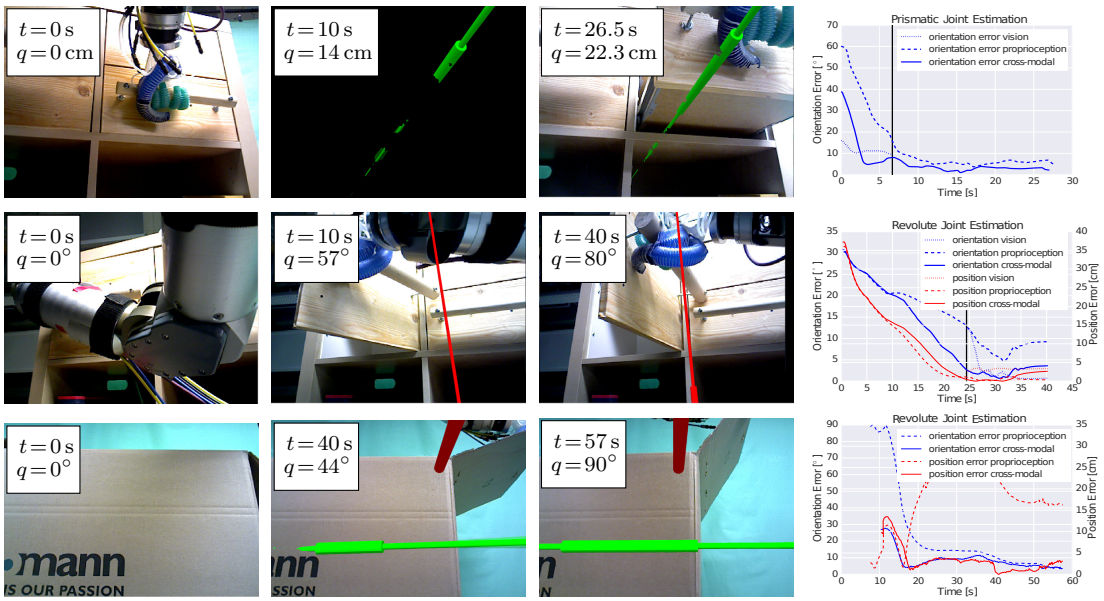


Figure 6.7: Experiments of the estimation of kinematic models (each row represents a different object): initial (first column), intermediate (second column), and final frame (third column) of the estimation, including error plot (fourth column) of estimated joint parameters relative to ground truth; the insets in the three images show the time t and the estimated joint configuration q using the cross-modal variant; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders [© 2017 IEEE]

We will define the desired velocity and the impedance parameters such that the robot performs pulling operations while being compliant in the other dimensions. Combining both the controllers the robot can explore articulated objects with different dynamic properties and create rich multi-modal signals for perception.

Once the kinematic structure has been revealed and perceived leveraging cross-modal information, the robot should be able to use this information to improve interaction or generate new manipulations. We implemented this skill as an online trajectory generator that computes an end-effector operational space trajectory to achieve a manipulation task, i.e. reaching a desired joint configuration. The trajectory generator uses the perceived kinematic model of the object and interpolates the object's joint configuration towards the desired state. Then the trajectory generator computes the motion of the interacted body necessary to obtain the desired object's joint configuration, and from that, the trajectory of the end-effector that generates the interacted body motion.

6.6 Experiments on Cross-Modal Integration

We conducted three sets of experiments. In the first set, we evaluate quantitatively the performance of our system when perceiving different articulated objects and compared the use of 1) only vision, 2) proprioception, or 3) the multi-modal stream leveraging cross-modal information. We measure the robustness, accuracy, and convergence of the kinematic model estimation by comparing the estimates to ground truth for the joint parameters and state.

In the second set, we make use of the online information about kinematics from the cross-modal system to control the robot's motion and fulfill a manipulation task. The robot explores an articulated object until it discovers a joint and perceives that it reaches a desired joint configuration. Then, the robot exploits the perceived information to plan a new trajectory to return the object to its initial configuration. We measure the accuracy of the execution (final joint state) of both the explorative and the exploitative interactions.

In the third set, we apply our approach for the estimation of dynamic properties to different articulated objects. We analyze the properties of our method, first in isolation, using kinematics information from a motion capture system, and second, in integration with the rest of the online IP system for articulated objects.

6.6.1 Experimental Setup

In our robot experiments, we use a robot manipulator composed of a Barrett WAM arm and a RBO Soft Hand 2 (Deimel & Brock, 2016). The joint configurations of the arm are measured at 200 Hz by encoders placed at the motors controlling the cables. The stretching of the cables introduces uncertainty about the end-effector's pose that we model with a covariance of 1 cm and 3° in the end-effector pose measurements, resulting from an offset calibration. The visual input is an RGB-D stream (640×480 pixels at 30 Hz) provided by a Carmine sensor rigidly attached and registered to the robot's base. The force-torque signals are provided by an ATI 6DoF sensor mounted on robot's wrist delivering signals at 100 Hz. Air pressure in the chambers of the soft-hand are delivered at 100 Hz. To compensate for the disparity in sensor frequencies we accumulate signals and process them at 15 Hz. This estimation rate can be maintained on an Intel Xeon E5520 PC at 2.27 GHz.

The estimated states of each filter are assumed to be Gaussian distributions. Both process and measurement models are of the form $\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, u_t) + \boldsymbol{w}_t$ and $\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{v}_t$, where f and h are possibly non-linear (but linearizable) forward and measurement models, and $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are the process and the measurement additive Gaussian noises. This allows us to implement the recursive estimation filters as Kalman filters or their variant for non-linear models, extended Kalman filters.

The neural network regressors (MLPR) have a topology of three layers with 10-10-10 fully connected neurons. This topology was selected in a hyperparameter search by a leave-one-out cross-validation process, selecting between 1 and 100 neurons per layer in networks of one, two, or three layers. The vision-based system tracks $N = 200$ point features. To focus the attention on the estimation of the kinematic model of the articulated object and not on the robot's arm, we project a model of the robot on the camera plane and subtract this part from the visual analysis.

In our experiments, we parametrized the controller to be compliant in all dimensions (main diagonal elements of $K = 0.1$, $D = 1$, and $M = 1$) except in the pulling direction of the end-effector (main diagonal elements of $K = 400$, $D = 200$, and $M = 1$). In the pulling direction the robot will attempt to follow the trajectory, while adapting with low forces in the other dimensions. These parameters perform well in all objects we evaluated. To generate an exploratory behavior we guide the robot's end-effector into a grasping distance of the object, command the robot to close the hand and command a desired velocity, ${}_{ee}\eta^{des}$, of $1\,\mathrm{cm\,s^{-1}}$ in the pulling direction ($k_p = 0.1$). As a result, the robot reveals the kinematic structure by pulling with an increasing force and adapting to the dimension of allowed motion of the articulated object.

For the experiments where we evaluate the estimation of dynamic properties in isolation, we use a motion capture system to obtain ground truth about the kinematic information (Mo-

tion Analysis, 2017). We use the motion capture system to obtain ground truth poses of the links of the mechanisms and the joint parameters. The interactions are performed with a force-torque sensor attached to a stick, whose pose is also tracked using the motion capture system. This data is part of a dataset of articulated objects and sensor data of interactions that we have released and made public for other researchers[2].

We use 1000 particles in our particle filter for the estimation of the dynamic parameters. The initial prior for the stiction and kinetic friction distribution is a uniform prior between 0 N and 10 N for prismatic joints, and between 0 N m and 3 N m for revolute joints. These values cover the dynamics of all articulated mechanisms in human environments that our robot can actuate with the soft hand. We impose in the particles the additional physical constraint that stiction must be higher or equal the kinetic friction.

We evaluate our system on articulated objects with different types of joints, size, color, and surface properties. We did not add artificial visual markers that could facilitate the visual perception. The objects are placed at different pose with respect to the robot and the sensors. In some experiments we also change abruptly the lighting conditions to evaluate the robustness of the perceptual systems. To obtain the ground truth for kinematic properties, we manually measured the joint parameters and the final joint state. To obtain the ground truth of the dynamic properties we measured with a force gauge the minimum tangential force/torque to initiate joint actuation and to maintain it. We average over three ground truth measurements to obtain an accurate estimate.

6.6.2 Experimental Evaluation

Uni-Modal vs. Cross-Modal Perception

We evaluate the accuracy and convergence of the kinematic model estimates from the three perceptual systems: 1) only vision, 2) only proprioception and 3) cross-modal integration. Figure 6.7 shows three images from the RGB-D sensor (initial, middle, and final steps) and graphs of the estimation error to ground truth over time.

In the first experiment the robot interacts with a drawer. After 6.5 s (indicated with a vertical line in the plot) we change abruptly the lighting conditions by switching off the lights. The vision-only system stops perceiving the object while the proprioception-only and the cross-modal system continue the estimation. The final joint state estimated by the cross-modal system is the most accurate (22.3 cm, ground truth 22.5 cm), followed by the proprioception-only (23 cm). The vision-only system stops tracking at (9.8 cm). The cross-modal system achieves the best performance because it leverages vision to estimate a more accurate grasping model, which lead to more accurate body motion estimates and robustness against vision failures from the interpretation of proprioception.

In the second experiment, the robot interacts with a door that rotates around a revolute joint. The robot almost completely occludes the object during the first 25 s of interaction (indicated with a vertical bar in the plot). The proprioception-only and the cross-modal system perceive the object during the entire interaction. The vision-only system perceives the interacted object only when it becomes clearly visible. The final joint state estimation from the cross-modal system (80°, ground truth 85°) is the most accurate, followed by the proprioception-only (78°). The final estimation of the visual system (43°) is affected by the delayed start. The cross-modal system achieves the best performance because it uses the proprioceptive signals to interpret the visible motion in the small non-occluded parts of the object.

[2]Our dataset is publicly available under https://tu-rbo.github.io/articulated-objects/.

Object	Error and std. dev. at the end of the exploration phase	Error and std. dev. at the end of the exploitation phase
Sliding Door	2.2 cm±1.6 cm	1.8 cm ±1.6 cm
Camera Tripod	7.8°±2.3°	2.6°±2.24°
Glass Door	1.3°±0.73°	0.6°±0.5°

Table 6.1: Error at the end of the exploration and the exploitation phases of the robot interaction based on the cross-modal perceived information

In the third experiment, the robot interacts with a cardboard box and closes one of its lids. As a result from the explorative interaction, the entire box translates. We focus the analysis on the estimation of the relative revolute joint between the box and the lid. Both uni-modal systems fail to detect this joint. The vision system only perceives the lower part of the box, while the proprioception system detects only the motion of the lid and interprets it as a revolute joint with respect to the environment. The cross-modal system correctly perceives the relative joint between the box and the lid because it uses the motion of the box perceived from vision to correctly interpret the motion constraints of the lid perceived from proprioception. The final joint state estimate from multi-modality is 90° (ground truth 100°).

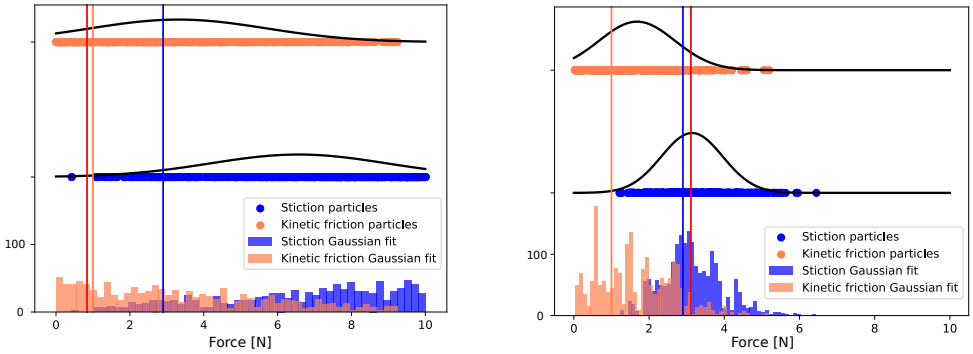
Controlling Interaction with Online Interactive Perception

We tested our cross-modal perceptual system and online trajectory generator for the manipulation of three previously unseen objects (see objects in Figure 6.1): opening a glass door (GD) 20°, turning camera tripod (CT) 45° and opening a sliding door (SD) 30 cm. These objects are challenging because they do not present strong textured surfaces and because the hand cannot grasp them perfectly. We repeated the interactions 5 times on each object with different initial robot-object pose. The results (mean and standard deviation on the error to ground truth) are depicted in Table 6.1. The interaction succeeded in the 15 trials (see video attachment) indicating that the information perceived online can be used to generate new successful trajectories. Our proposed perceptual system leverages information between vision and proprioception to estimate accurately the joint state at the turning point and the end of the manipulation, which indicates that the system can be applied to monitor ongoing interactions.

Perceiving Dynamic Properties of Joints

The results of the first set of experiments, where we use ground truth poses and kinematics and evaluate only the estimation of dynamic properties, are summarized in Table 6.2. The table compares the ground truth values of the parameters to the mean and standard deviation of the estimated values. The standard deviation of the ground truth indicates that even among the three controlled manual measurements, the observed frictional values are slightly different, which illustrates that estimating the frictional properties is a hard perceptual task. The estimated value is an average over five estimation processes from different interaction sequences. In these sequences we vary the pose of the object, the force/torque sensor, and the contact point on the link of the object. The evolution of the estimation from continuously arriving signals in one of the experiments is depicted in Figure 6.8.

The method estimates both the threshold in force to initiate motion (stiction) and the minimum force to maintain the motion (kinetic friction) with sufficient accuracy, and the error to ground truth is within the uncertainty bounds. However, the estimated values diverges



(a) Estimation after several measurements without joint motion ($t = 0.2\,\mathrm{s}$)

(b) Estimation when the joint starts to move ($t = 1.7\,\mathrm{s}$)

(c) Estimation after several measurements with joint motion ($t = 1.75\,\mathrm{s}$)

(d) Estimation when the joint stops moving ($t = 4.2\,\mathrm{s}$)

Figure 6.8: Four steps of the estimation of the dynamic properties in a prismatic joint (Ikea); blue vertical line: stiction ground truth ($2.9\,\mathrm{N}$); orange vertical line: minimum kinetic friction ground truth ($1.0\,\mathrm{N}$); red vertical line: current tangential applied force; each plot depicts the histogram of particles (bottom axis), the particles and Gaussian fit for the stiction parameter (middle axis) and the particles and Gaussian fit for the kinetic friction parameter (top axis)

strongly in two cases: the cabinet drawer and the microwave door. The cabinet drawer is quite heavy and possesses high quality bearings. This means that while a high force is necessary to initiate motion (stiction), once the drawer is moving its inertia (an effect that we don't model) is considerable and helps the drawer to keep moving. The motion continues with low friction thanks to the bearings, even without much additional force. In this conditions, measuring the force to maintain the actuation (constant kinetic friction) is difficult. The microwave door is very light and presents low friction. In these conditions, the noise in the measured wrenches strongly affect the estimation.

In the second set of experiments we evaluate the integration of our dynamics estimation method with the system that estimates kinematic properties from a cross-modal integration of sensor signals. We evaluated the estimation of frictional properties on a cabinet door and two drawers. The results are depicted in Table 6.3. While the integration in the system increases the uncertainty about the working wrench, the estimated dynamic parameters are still close to the ground truth values. However, the robot consistently overestimates the dynamic parameters because our method does not discount the part of the tangential force that the soft-hand

Object (Joint Type)	Parameter	Estimation ± SD	Ground Truth ± SD
Ikea (prismatic)	Stiction	2.94 N±0.756 N	2.9 N±0.12 N
	Kinetic Friction	0.94 N±0.45 N	1.0 N±0.3 N
Ikea (revolute)	Stiction	1.73 N m±0.54 N m	1.8 N m±0.4 N m
	Kinetic Friction	1.69 N m±0.63 N m	1.7 N m±0.6 N m
Cabinet (prismatic)	Stiction	8.56 N±0.85 N	8.31 N±1.2 N
	Kinetic Friction	0.62 N±0.41 N	1.83 N±0.52 N
Microwave (revolute)	Stiction	1.34 N m±0.71 N m	1.2 N m±0.35 N m
	Kinetic Friction	0.48 N m±0.82 N m	0.65 N m±0.36 N m
Ikea small (prismatic)	Stiction	3.23 N±0.47 N	3.12 N±0.65 N
	Kinetic Friction	1.78 N±0.42 N	1.98 N±0.82 N
Laptop (revolute)	Stiction	9.38 N m±2.31 N m	9.58 N m±0.84 N m
	Kinetic Friction	7.94 N±0.83 N	8.40 N m±0.63 N m

Table 6.2: Estimation of stiction and kinetic friction from human interaction and ground truth kinematics; SD stands for standard deviation; the standard deviation of the ground truth values corresponds to differences within our three ground truth measurements; the estimated values are averaged over five estimation processes from different interactions

Object (Joint Type)	Parameter	Estimation ± SD	Ground Truth ± SD
Ikea (prismatic)	Stiction	3.4 N±0.77 N	2.9 N±0.12 N
	Kinetic Friction	1.35 N±0.73 N	1.0 N±0.3 N
Ikea (revolute)	Stiction	2.07 N m±0.61 N m	1.8 N m±0.4 N m
	Kinetic Friction	1.93 N m±0.79 N m	1.7 N m±0.6 N m
Ikea small (prismatic)	Stiction	3.83 N±0.88 N	3.12 N±0.65 N
	Kinetic Friction	2.51 N±1.36 N	1.98 N±0.82 N

Table 6.3: Estimation of stiction and kinetic friction from robot interaction in the integrated perceptual system; SD stands for standard deviation; the standard deviation of the ground truth values corresponds to differences within our three ground truth measurements; the estimated values are averaged over five estimation processes from different interactions; the robot consistently overestimates the stiction and kinetic friction due to the unaccounted effect of the soft-hand

absorbs and transforms into deformation. The estimated values are accurate enough to predict approximately their effect on the interaction, as we will see in the next chapter.

6.7 Discussion and Limitations of the Cross-Modal Integration

We will begin this section by comparing the perceptual systems of previous chapters and this chapter. We will analyze the different exploitation of problem structure. Due to the cross-modal integration of modalities, the system of this chapter exploits further the interdependencies between perceptual subtasks (OP4). And because one of the modalities characterizes the robot's actions, the cross-modal system also deepens the exploitation of interactions for the interpretation of changes in sensor signals (OP1). We will focus on these two aspects.

Exploiting Interactions (OP1) While in the systems of previous chapters the interactions served only as generators of information-rich signals, in this chapter the system uses information about the robot's actions to interpret the changes in the sensor stream. This is possible because part of the input sensor-action stream of the cross-modal system –the state of the robot's joints and the forces and pressures the robot applies on the environment– reports about robot's actions. Now, as we argued in previous chapters, a system that uses information about the interaction to interpret the changes in the sensor signals needs some sort of *interaction model*. The model relates actions and changes in the sensor signals, the structure of the combined $S \times A \times t$. We defined four possible interaction-grasp models correlating the motion of the robot's end-effector and its effect into the pose of the parts of the articulate object. Our perceptual system identifies the model that explains best the interaction and applies it to interpret upcoming sensor signals. However, since our method depends on predefined interaction-grasp models, it cannot generalize to different robots and end-effectors.

Thanks to the interaction grasping models, our cross-modal system improves robustness and versatility, exploiting actions as source of information. In fact, we think that interaction models are a crucial element for interactive perception and robot manipulation. There is a justified interest from the robotics community on methods to estimate, learn and acquire forward models (Agrawal et al., 2015, 2016, Byravan & Fox, 2017). In the next section of this chapter we will investigate a method to learn interaction models online from experiences, thereby reducing the dependency on predefined grasping models.

Exploiting Interdependencies between Perceptual Subtasks (OP4) The system we have presented in this chapter implements our general approach for robot perception based on coupled recursive estimation processes presented in Section 1.3. We have shown that this algorithmic approach, when applied to multiple sensor modalities, leads to a system that exploits the concept of cross-modality: when information from one sensor modality is applied to interpret signals from another. This approach to multi-modal perception is present in human perception (McGurk & MacDonald, 1976) and as we showed it is also beneficial for robot perception.

In our system the models that define the interdependencies between perceptual subtasks are given a priori. As part of the system description we define how subtasks interact with each other and how to exploit information from one subtask for another. This dependency on predefined engineered interdependencies is a limitation of our system. We believe that combining our algorithmic approach with methods to infer the models of the interdependencies between subtasks could be the key to apply our approach to a broader range of problems in perception for robot manipulation.

We will now discuss to what extent the system presented in this first part of the chapter accomplishes the goal and overcomes the challenges of perception for robot manipulation. We will repeat the discussion at the end of the chapter, after we present our online learning method for interaction models. Compared to the systems of Chapter 4 and Chapter 5, the online IP system of this chapter 1) demonstrates that the perceived information is task-relevant applying it to control ongoing interactions and generate new motion, 2) links further actions and their consequences in the sensor signals (CH1), and 3) increases the robustness and versatility of the robot perceptual skills to a broader range of unstructured environments and task conditions (CH3).

Applying the Information for Manipulation As part of the experimental evaluation we have developed and tested methods that exploit the information perceived online in our

system to monitor ongoing interactions and to generate new manipulation actions. These methods demonstrate that the perceived information is relevant for the task: the manipulation of articulated objects. However, the trajectory generation method we presented in this chapter only considers kinematic constraints of the articulated object, neglecting other properties like shape and dynamics. In Chapter 7 we will present a method that considers also these characteristics to generate and select informative actions.

Extracting Information from Changing Sensor Signals correlated to Interactions (CH1) As we have argued previously in this section, the interaction-grasp models link actions and changes in the environment. The system we presented estimates the most likely interaction-model as part of the perceptual problem. The result of this additional perceptual subtask bridges the gap we identified in previous chapters: the robot can now predict changes in the sensor signals as consequences of its own actions. This allows the robot to extract more accurate information from the changing signals using information about the interaction.

Versatile Perception in Unstructured Environments (CH3) Our cross-modal system builds kinematic models of articulated objects in adversarial environmental conditions, and overcomes limitations of uni-modal systems. This increases the versatility of the IP system to cope with different unstructured environments and tasks. The perceptual system uses cross-modal information to also build simple models of the dynamics of the joints (frictional properties). However, the system can only perceive the dynamic properties of one DoF articulated mechanisms. While mechanisms with only one DoF are the most commonly found in human environments, we consider this characteristic a current limitation on the versatility of our system. The estimation of the dynamics also assumes that the friction is independent of the configuration, which does not hold for mechanisms with springs.

6.8 Learning Interaction Forward Models from Experiences

6.8.1 Motivation

We have seen in Chapters 4 and 5 and in the previous part of this chapter that we can build uni-modal and multi-modal interactive perception systems as implementations of our general approach based on coupled recursive estimation. These systems acquire task-relevant information and address, with some limitations, the challenges of perception by leveraging the structure of the problem. One of the structural properties they leverage is the interdependencies between interactions and changes in the sensor signals, the key idea behind interactive perception.

The systems of Chapter 4 and 5 exploit interactions to create information-rich signals and reveal hidden structures, e.g. kinematic constraints. The system presented in the first part of this chapter goes beyond that and uses information about the interaction (i.e. proprioceptive signals) as prior to interpret changes in the sensor signals. This second type of exploitation of interactions requires some form of interaction model that relates actions and changes in the environment and in the sensor stream.

In the first part of this chapter we predefined four interaction-grasp models for a specific end-effector that linked robot actions and changes in the environment. We proposed, as part of the perceptual system, a method to select the most likely grasping model among these four hypotheses. Previous approaches in interactive perception methods also assumed a priori specified correlations between actions and changes in the sensor signals (Barragán et al., 2014,

van Hoof et al., 2012, Hausman et al., 2015). These correlations represent the structure of the combined space $S \times A \times t$ of sensor signals S and actions A over time t that is relevant for the perceptual task.

In the remainder of this chapter we will take a closer look to the problem of estimating interaction forward models. An interaction forward model predicts the changes in the sensor signals due to the robot's actions (Levi & Kernbach, 2010). Our goal is to learn these models online from robot's experiences to reduce the dependency on predefined models. If the robot can autonomously find the right structure in $S \times A \times t$ this will improve the versatility of the interactive perceptual system.

To be able to learn interaction forward models online, we will assume that the combined space $S \times A \times t$ is strongly structured, which means that the actions and the changes in the sensor signals are intimately correlated. If our hypothesis is right and $S \times A \times t$ is strongly structured, it should be possible to find this structure for a large group of manipulation tasks by interacting and observing the resulting changes in sensor signal.

We propose to find the relevant structure in this combined space by estimating the (possibly dynamic) correlations between A and changes in S. We assume that the relationship between actions and changes in sensor signals is sufficiently smooth to be estimated recursively from pairs of actions and observed changes. We will exploit the acquired model that relates A and changes in S to address perceptual tasks that cannot be solved passively and to achieve tasks defined as goals in S.

Methodologically, we will explore this hypothesis by extending the system we developed in Chapter 4 with the integration and exploitation of knowledge about the robot interaction. We will present a method to learn online the interaction forward model that relates robot actions to changes in the environment and the sensor signals, the structure in the combined $S \times A \times t$ space. Using this model the robot can exploit the knowledge about the interaction to complete missing sensor information and to make the perceptual process more robust. To remove the dependency on predefined models and because such interaction models are usually dynamic and task specific, we will propose a learning method to estimate the interaction forward model from ongoing interactions (CH2).

We will evaluate if the learned model can be applied to improve perception and manipulation in two aspects: a) to predict the motion of controllable degrees of freedom even under occlusions, and b) to generate actions that fulfill a manipulation task.

In the following, we will first present our method to estimate recursively interaction models. Then, we will evaluate the improvement in perception and manipulation when using the online learned interaction forward models.

6.8.2 Bayesian Recursive Estimation of Interaction Forward Models

The prevalent approach to learn interaction forward models is to restrict the problem to a specific robot task and generate enough experimental data (changes in sensor signals) using physical models (Barragán et al., 2014, Battaglia et al., 2013) or continuous executions of the task (Lenz et al., 2015, Levine et al., 2016, Agrawal et al., 2015, 2016). This approach requires large amount of data due to the high dimensionality of the state space for any realistic robotic manipulation task. The approach cannot be applied to learn a model from ongoing interactions.

In this work, we take a different approach: We exploit prior knowledge about the interaction forward model by assuming that the model changes smoothly with respect to time and robot's configuration space. This assumption is reasonable for those tasks where an action of the robot causes a proportional reaction in the environment, and there are no abrupt disconti-

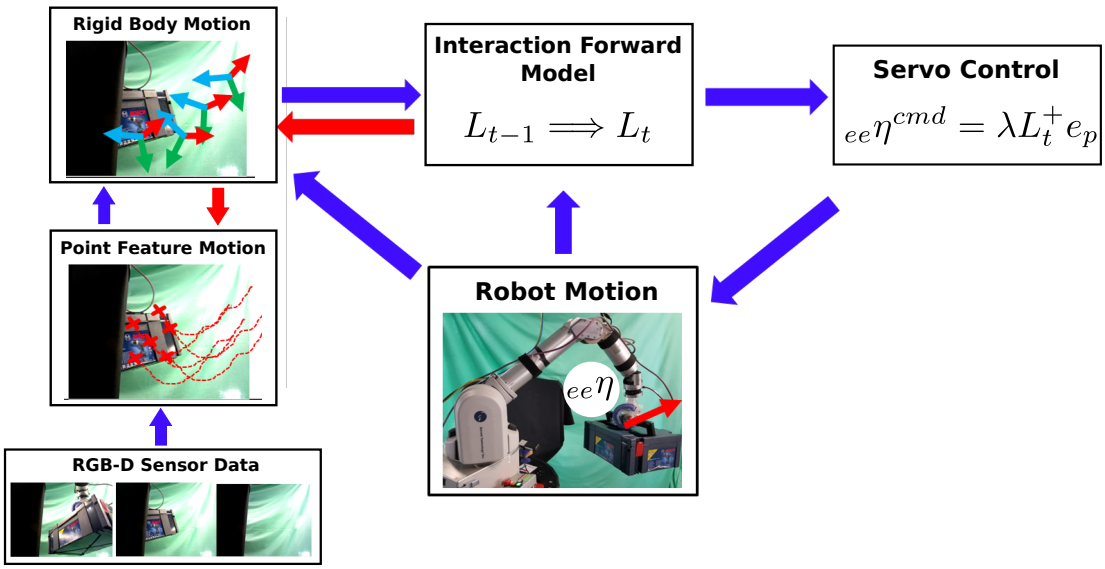


Figure 6.9: System to learn online forward models of the interaction; left column: original vision-based system to estimate motion of rigid bodies; based on pairs of measurements of robot motion (${}_{ee}\eta$) and the associated rigid body motion the robot learns a model that correlates both; the online learned model can be used to improve perception (red arrow entering the estimation of rigid body motion) and to control the manipulation (servo control loop)

nuities (e.g. due to changes in the contact interface between the robot and the environment). This prior assumption allows us to approximate locally the forward model by a linear model L that maps changes in the sensor stream, $\dot{s}$, to robot actions u:

$$\dot{s} = L \cdot u \tag{6.18}$$

where the matrix $L \in \mathbb{R}^{k \times m}$, k is the dimensionality of the action vector and m the dimensionality of the measurement vector. In our method we consider as action the changes in the configuration of the end-effector of the robot, i.e. the end-effector spatial velocity:

$$u = {}_{ee}\dot{p} = {}_{ee}\eta \tag{6.19}$$

This information is contained in the proprioceptive signals we integrate in this chapter and processed in Section 6.3.1.

The Equation 6.18 appears often in the context of robot visual control and servoing tasks (Corke, 2011). In these tasks, the robot's goal is to cause changes in a set of visual features ($\dot{s}$) towards a feature goal. For visual control tasks, the linear model correlating actions and changes in sensor signals is known as the *interaction matrix* (Chaumette & Hutchinson, 2006). Given the known general structure of the interaction matrix and the assumption that it changes smoothly, this matrix can be learned online from pairs of robot actions and changes in the visual sensor signals. Jägersand et al. (1997) presented an efficient update rule for visual servoing tasks that estimates the elements of the interaction matrix when the changing sensor signals are assumed to be visual point features.

Similar to the approach by Jägersand et al. (1997), we propose to estimate the interaction matrix L recursively based on data of robot motion and corresponding feature change. Therefore, the state we aim to estimate is the interaction forward model (*ifm* = interaction forward

model):

$$\boldsymbol{x}^{ifm} = \boldsymbol{L} \tag{6.20}$$

We transform our linear model relating changes in the sensor signals and robot actions to obtain L in vector form:

$$\dot{s}_t = L_t u_t = H_t l_t \tag{6.21}$$

$$l_t = (L_t^{1,1}, L_t^{1,2}, \ldots, L_t^{1,k}, L_t^{2,1}, L_t^{2,2}, \ldots, L_t^{m,k})^T \tag{6.22}$$

$$H_t = \begin{pmatrix} {}_{ee}\eta^T & & 0 \\ & \ddots & \\ 0 & & {}_{ee}\eta^T \end{pmatrix} \tag{6.23}$$

where $L_t^{i,j}$ is the (i,j) element of the matrix L_t. The resulting state vector l_t contains all the elements of the interaction matrix. We reformulate our problem to the estimation of $\boldsymbol{x}_t'^{ifm} = \boldsymbol{l}_t$. The use of the symbol H in the equation below is not casual: This matrix will act as measurement model to estimate the interaction forward model, as we will see later.

We assume that the estimated interaction forward model does not change between time steps:

$$\boldsymbol{l}_t = \boldsymbol{l}_{t-1} + \boldsymbol{w}_t^{ifm} \tag{6.24}$$

$$\hat{l}_t = \hat{l}_{t-1} \tag{6.25}$$

The covariance of the system noise, Q_t^{ifm}, is a free parameter of our approach that governs the sensitivity of the estimation: with large Q_t^{ifm} the estimation would adapt the model quickly to new relationships between robot actions and changes in sensor signals, at the cost of increasing the sensitivity to noise in the measured sensor changes.

We use as measurements to estimate the interaction model, observations of the changes in the sensor signals:

$$z_t^{ifm} = \dot{s}_t \tag{6.26}$$

$$\dot{s}_t = H_t \hat{l}_t + \boldsymbol{v}_t^{ifm} \tag{6.27}$$

The measurement noise, R_t^{ifm}, is based on the uncertainty of the sensor signals used for the estimation.

The previously presented recursive estimation of a linear forward model can be applied to any set of features in sensor space, s, that changes in correlation to the robot's action. We use as features the poses of the moving rigid bodies estimated from visual information, $\{{}_i p\}_{i \in \{1 \ldots N\}}$, with N the number of currently tracked bodies. The measurements for the estimation of the interaction forward model are thus $\dot{s} = \{{}_i \dot{p}\} = \{{}_i \eta\}$, the velocities of the N rigid bodies. We create and maintain a separate interaction forward model for each moving body, L^i. To define the uncertainty of the measurements for the interaction forward model estimation we use the visual uncertainty about the motion of the rigid bodies. The usage of information from vision as explained above constitutes another example of cross-modal integration, where the information perceived from vision enables a novel interpretation of the proprioceptive signals.

The interaction forward models that we obtain from the recursive process defined are called *position-based image Jacobian matrices* in the visual servoing literature (Chaumette & Hutchinson, 2006). These matrices correlate motion of the end-effector, ${}_{ee}\eta$, to visually

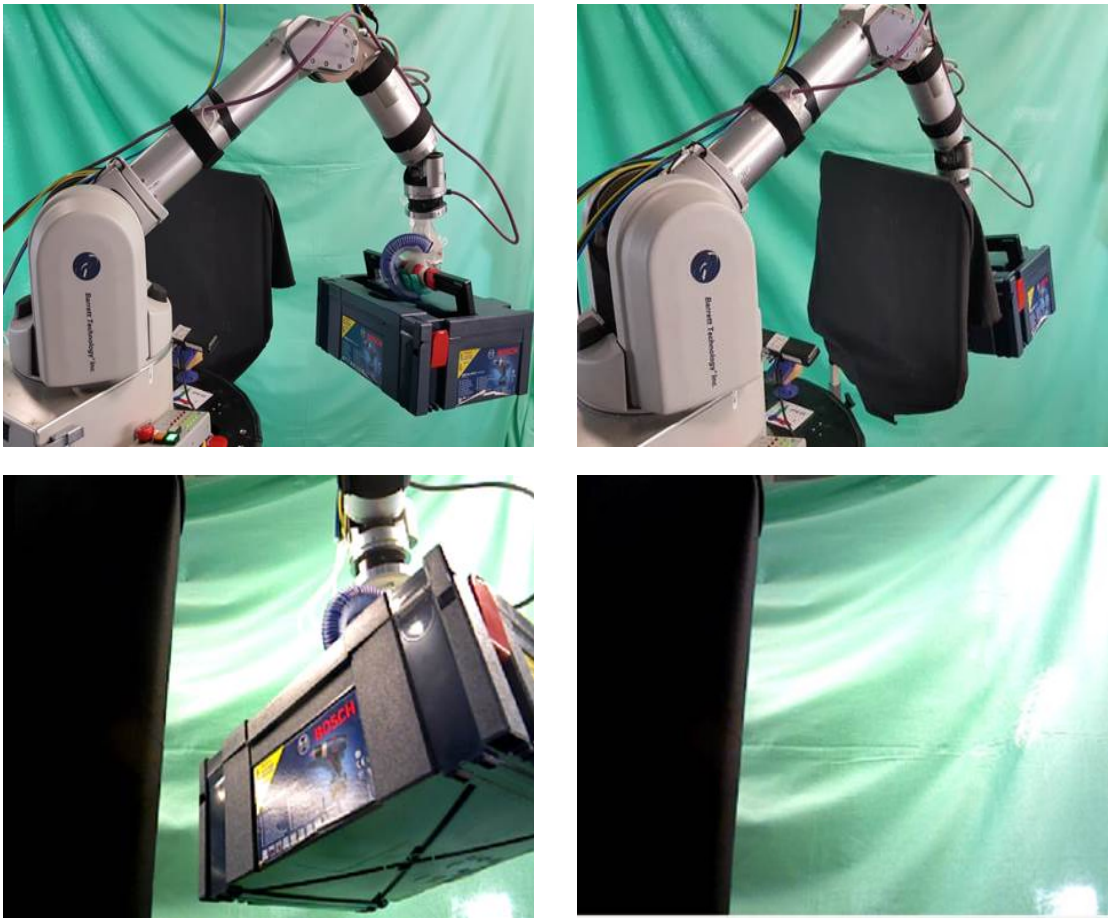


Figure 6.10: The robot grasps an object with a cylindrical grasp and manipulates it; rotations along the main axis of the cylindrical grasp are not transmitted to the object; first row: external view of the experiment; second row: robot view from an RGB-D sensor; left column: the robot observes the outcome of its actions and learns an interaction forward model; right column: the object is occluded and its pose is predicted using the forward model

perceived motion of the rigid bodies, $\{{}_i\eta\}$. Nevertheless, the method is general enough to estimate an interaction forward model for any other set of features that fulfill our working assumption: the correlation between changes in the features and robot actions should be linear or linearizable and change smoothly over time and configuration space.

6.9 Experiments on Learning Interaction Models

In the following experiments we initialize the interaction forward model $\boldsymbol{x}_0'^{ifm} = \boldsymbol{l}_0$ with zeros to indicate that the objects are usually uncontrollable until the robot perceives that they move in correlation with its own actions. The estimation problem is solved using a Kalman filter. We found experimentally that with a covariance matrix for the system noise of $Q_t^{ifm} = 0.05 I_{36\times 36}$ the estimation of the models converges quickly and is robust against noise in the

sensor input.

6.9.1 Online Interactive Perception Using Interaction Forward Models

First, we evaluate if the online learned models improve perception. We compare the tracking capabilities for rigid bodies of our online IP system based only on vision (Chapter 4) to a system that predicts the motion of the rigid bodies based on the robot actions using the online learned interaction forward model.

In the online IP system of Chapter 4, we predicted the motion of the rigid bodies based on the estimated velocity. Alternatively, we predicted motion assuming that the body was abruptly stopping, or using predictions based on the kinematic structure of the articulated object. None of these predictions used information about the action of the robot.

We create an alternative online IP system that predicts the next pose of the rigid bodies in the following manner:

$$ {}_{ib}p_t = {}_{ib}p_{t-1} \oplus \Delta_t(L^i_t \, {}_{ee}\eta) \tag{6.28} $$

In the alternative IP system, this model substitutes the three alternative forward models in the system presented in Chapter 4.

We will evaluate whether the robot is capable of tracking rigid bodies more accurately and robustly using the alternative IP system, even when there is no visual information to correct the prediction, e.g. due to visual occlusions.

Experiment: To evaluate the integration of the estimated forward models into the alternative IP system, we command the robot to grasp and move randomly a tool-box with a broad handle. The model describing this interaction is not trivial because some robot motions have no effect on the pose of the object: The robot grasps the handle with a *cylindrical grasp* that does not transmit rotations nor translations along the main axis of the cylinder (see Figure 6.10).

Approximately after 65 s (enough time for the robot to learn the interaction forward model), the robot moves the object to a region where the object becomes visually occluded. And approximately 15 s later the robot moves the object outside of the occlusion area. We will compare the pose estimation from the original IP system and the alternative IP system using the learned interaction forward model. We obtain ground truth of the pose of the box attaching AprilTags fiducial markers (Wang & Olson, 2016) and a second RGB-D sensor that we calibrate externally. The markers are placed so that they are not visible to the RGB-D sensor of the robot. The results are depicted in Figure 6.11.

During occlusion there are no visual signals to correct: both the online IP systems use the predictions from their forward models as perceived object motion. The IP system that does not use the interaction forward model predicts the object motion based on the last estimated velocity. These predictions starts to drift as soon as the robot action changes the object's trajectory. The alternative IP system that uses the learned model can predict the object pose correctly based on the robot's actions, even though the error to ground truth in position initially increases due to the unmodeled effect of the soft hand

When the object reappears in the visual field, only the IP system that uses the interaction forward model can re-identify the object and assign correctly visual features. The IP system that does not use the learned model cannot re-identify the object and continues drifting. We think that the ability to re-identify objects after complete occlusions is a positive benefit of the integration of the interactive forward models into the perceptual system of Chapter 4.

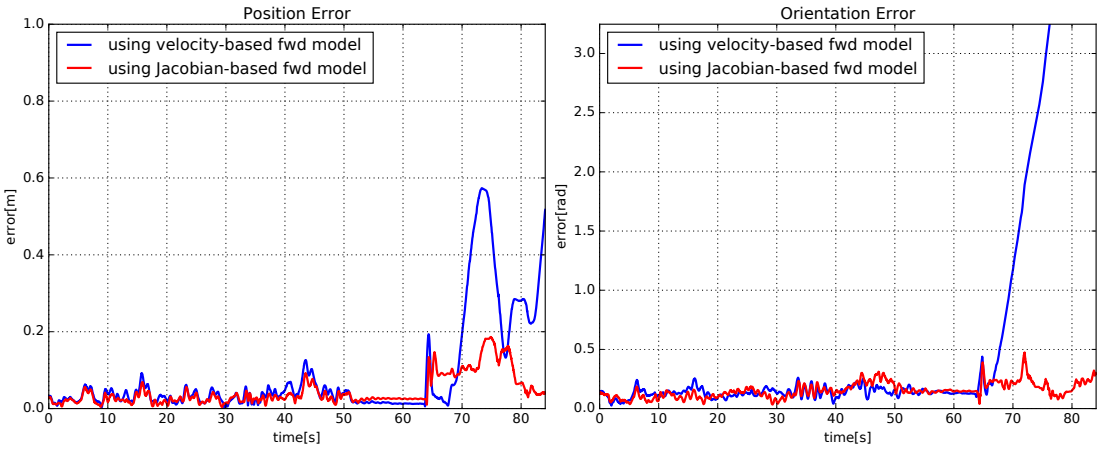


Figure 6.11: Position and orientation (unrolled) error of the object pose using the interactive and the passive forward models; the online IP system without the interaction forward model (blue) generates predictions based only on the estimated body velocity and fails when the object is occluded (after 65 s); the modified online IP system (red) uses the online learned interaction forward model to predict the motion of the body based on robot's actions and estimates its pose even without visual signals

6.9.2 Visual Servoing

Given the similarity of the equations to estimate forward models and the visual servoing control, we can "invert" the equation and formulate a servo control law to achieve desired changes in the sensor signals from robot actions. In our case, the goal is to control the motion of a manipulated object without prior knowledge about the contact, the hand morphology and the camera configuration.

Similar to classical visual servoing control, we use a pseudo-inverse of the estimated interaction forward model, $(L_t^i)^+ = L_t^+$, and a proportional gain λ, to define a control velocity twist ${}_{ee}\eta_t^{cmd}$ that minimizes the error e_p between the objects current and desired goal pose:

$$ {}_{ee}\eta_t^{cmd} = -\lambda \hat{L}_t^+ e_p \tag{6.29}$$

The error e_p corresponds to the difference in exponential coordinates between the current pose of the object and the goal.

We will evaluate if the robot can use the learned model to move an object to a desired pose based on the RGB-D images from an uncalibrated camera (at an unknown pose with respect to the robot).

Experiments: In our experiment, two objects move in front of the robot's camera, one grasped by the robot and the other moved by an experimenter (Figure 6.12). In the initial phase, the robot moves randomly the object and observes the motion of both objects. The robot estimates the interaction forward models correlating its own actions and the motion of both objects. Then, the robot compares the interaction models to identify the controllable object. The robot considers that one of the objects is controllable if the predictions based on the learned interaction forward model are accurate under an error threshold e_{th}.

In the second phase, after approximately 20 s, the robot is confident enough that one of the bodies is controllable and uses the online learned interaction forward model to move this object to the desired goal pose. The desired goal pose is at the center of the field of view of the robot in the uncalibrated camera. We repeat the experiment three times with different initialization of the robot and distractor. In the three trials, the robot successfully identifies

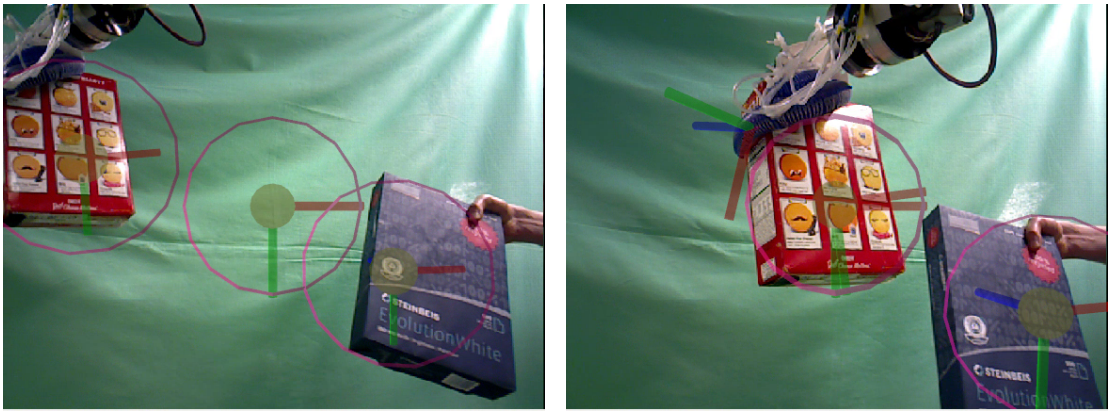


Figure 6.12: Robot view of two objects moving, one controlled by itself the other controlled by an experimenter; left: the robot identifies the controllable object based on the online learned interaction forward model, and uses the model to bring the controllable object to the goal (circle and frame at the center of the image); right: the pose of the controllable object converges to the goal

the controllable object and moves it to the desired goal pose with a final error under 3 cm in position and 7° in orientation. The robot can successfully estimate the model that correlates its own actions and the visually perceived motion of the objects, even without prior knowledge of the camera pose and the grasping.

6.10 Discussion and Limitations of Learning Interaction Models

In this section we will discuss the strengths and limitations of the method we presented to learn interaction models, and its applications for perception and control.

Applying the Information for Manipulation As part of the experimental evaluation, we showed that the robot could use the online learned interaction model to control the motion of an object towards a predefined goal. It would be necessary to carry out a more extensive evaluation to assess thoroughly the convergence and robustness of the proposed approach in different environmental and task conditions. However, we think that our experiments showed that the robot can learn online how its own actions influence the environment, and use this information to control the manipulation. Learning interaction models online requires to leverage additional problem structure, in this case the assumption that the interaction model changes smoothly with respect to time and robot's configuration space.

Extracting Information from Changing Sensor Signals correlated to Interactions (CH1) Our proposed approach to learn interaction forward models uses the coupling between changes in the sensor signal and interactions as input. Therefore, the method is focus on the information contained in the *changes*, rather than in the static sensor signals. We also presented an extended variant of the online IP system of Chapter 4 that integrates the learned interaction forward model. This variant of the online IP system can predict the changes in the sensor signals (in body poses and from that, in feature motion) using robot's actions as input.

Perceiving Quickly and Online (CH2) The method we presented estimates interaction forward models from ongoing interactions. However, there are some limitations in the method's online capabilities. Given that we represent the interaction forward model as a Jacobian, the dimensionality of the problem is the dimensionality of the combined *sat* space. In our implementation, we assumed that the motion of the end-effector is the action signal and the visually perceived motion of the rigid bodies are the sensor features. The dimensionality of the interaction forward model estimation problem in this case is $6 \times 6 = 36$. Our method requires some time to converge to a good estimation of these 36 values. For more complex sensor features, the dimensionality of the problem increases.

Versatile Perception in Unstructured Environments (CH3) The main limitation of our methods derives from our working assumption: The interaction forward model changes smoothly with respect to time and configuration space. This assumption allowed us to assume that the model is linearizable and solve the problem as an online Jacobian estimation problem. However, if the interaction model was not linear or cannot approximated by a slowly changing linearization (i.e. if the model changes abruptly), the method we presented would fail. Therefore, our method cannot be applied to tasks with discontinuities like grasping/ungrasping operations, or abrupt contact changes. For those scenarios, a model representation that can cope with non-linearities (e.g. an artificial neural network) should be preferred (Levine et al., 2016, Agrawal et al., 2016).

6.11 Conclusion

In the first part of this chapter, we presented an online IP system that perceived articulated objects (kinematic and dynamic properties) integrating multiple sensor modalities in a cross-modal fashion. Using cross-modality, our system leveraged information from one modality as prior to interpret signals from the other. Exploiting these interdependencies between perceptual subtasks, our IP system overcame limitations of previous uni-modal systems. We complemented the perceptual system with a velocity-impedance controller that generated information-rich signals from safe interactions. We also proposed an online trajectory generator that used the perceived information to bring the object to a new configuration.

Since one of the input modalities to our system contained information about the robot's actions, we developed interaction models correlating interactions and changes in the sensor signals. In the first part of this chapter, we predefined several interaction (grasping) models and proposed a method for our system to select the most likely one. In the second part of this chapter, we investigated a simple method to reduce the dependency on predefined interaction models with an online-learning approach. We demonstrated that for tasks where the interaction model changes smoothly, an online Jacobian estimation approach provided models that could be used to support and control ongoing robot interaction.

7

Action Selection for Interactive Perception

In the previous chapters we presented three systems that perceive kinematic, geometric and dynamic (frictional) properties of articulated objects from visual and proprioceptive signals of interactions. To generate information-rich signals, we implemented an impedance controller that guides the robot interaction (pushing and pulling actions) along the dimensions allowed by the constraints of the object, based on measurements from a force-torque sensor on the robot's wrist. Using the controller, the robot actuates successfully the articulated mechanisms, assuming a decent controller initialization. However, the robot still needs the initial parameters for the interaction (e.g. where and how to grasp, in which direction to pull or push), which we provided using kinesthetic teaching or manual specification. This is a strong limitation for robots that aim to explore autonomously the environment and acquire information about the articulated objects in it.

In this chapter we overcome these limitations with a method to autonomously generate and select the most informative actions to feed the interactive perception methods presented in previous chapters. Combining the action selection method and the perceptual systems the robot builds incrementally richer models with task-relevant information of the articulated objects. Our goal is thus twofold: first, we aim to increase the autonomy of the robot to create and select actions that will serve for the interactive perception of the articulated objects in the environment. And second, we close the loop and evaluate if the information perceived with the systems of previous chapters enables autonomous robot manipulation of DoF, the goal for perception we defined in Chapter 1.

As one of our goals is to generate informative actions, we need to define which actions will create the information-richest sensor signals for the interactive perception systems of the previous chapters. Usually, information gain is measured directly as the reduction of entropy of the belief state of the environment. In our presented perceptual systems, the actuation of the articulated mechanisms reduces the entropy about the articulated object. Therefore, in our action selection method, we advocate for the use of induced motion as an simple but effective proxy for information gain in the context of perceiving models of articulated objects. In this problem, motion is a good proxy as it reveals the articulation of the mechanism and the relationship between the motion of rigid bodies and the forces applied to them.

A crucial bottleneck in selecting informative actions is the model that is used to predict action outcomes. Manipulations of articulated objects are contact-rich interactions. The large variety of possible kinematic structures and their dynamic properties make it difficult to find general predictors of the real world. This is even harder for soft end-effectors as the one

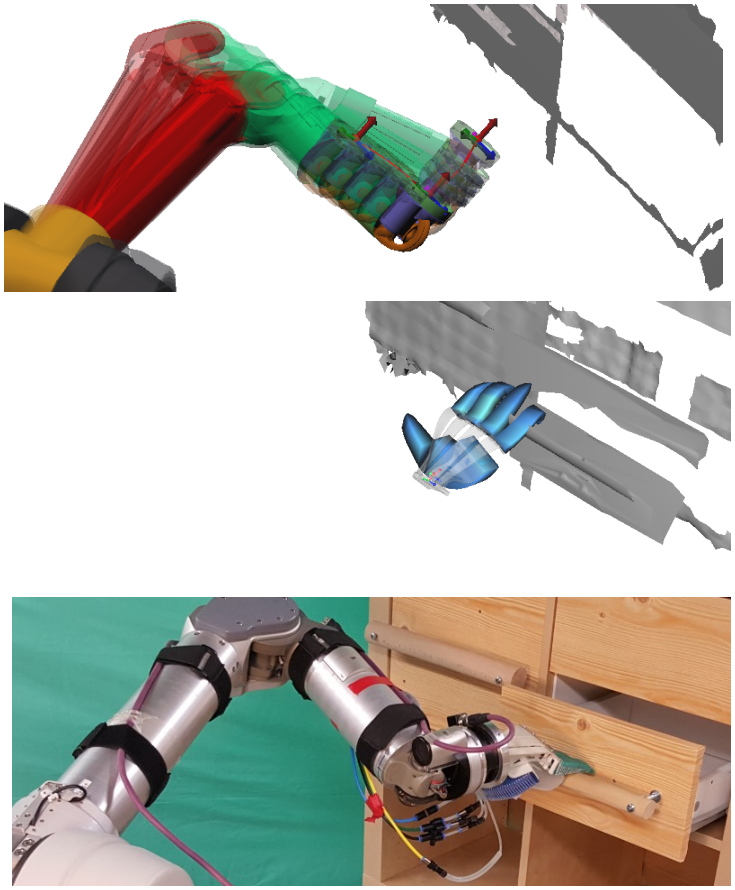


Figure 7.1: The selection of information-revealing actions for interactive perception of articulated objects is split into two subproblems;(1) constraints due to robot kinematics, collisions, and kinematics of the articulated object are satisfied via sequential convex optimization (Schulman et al., 2013a) on a kinematic model (*top*); (2) the complex contact interactions between end-effector and object (*center*) are evaluated with a dynamic physics simulation (Allard et al., 2007); the execution of the selected motion (*bottom*) reveals information about the object, which improves the model and in turn affects the next action selection

equipped on our robot. We therefore propose the use of a physical simulation for predicting and evaluating the outcome of actions. The proposed simulations are grounded in the real world because they are based on estimated articulated models obtained during interactive perceptual. To alleviate the accompanied computational costs of such simulations, we compare different sampling methods to select informative actions.

As an action generation and selection approach, the contributions of this chapter are twofold: We show that motion can be used as a proxy for information gain and that the gained knowledge allows for riskier and more tailored manipulations. Second, we present a method to find informative actions by sampling physics simulations and splitting the search into kinematic and dynamic aspects. We integrate our proposed action selection method and the perceptual systems of previous chapters into a real-world robot system that perceives and interacts with articulated objects.

7.1 Related Work

Previous methods that select actions for interactive perception differ in *1)* how they assess the information gain of an action (some kind of cost or objective function), and *2)* how they explore the space of possible interactions to find the most informative one. One of the first methods in interactive perception (Tsikos & Bajcsy, 1988)) proposed an approach to map

the content of a tray into a graphical representation that encodes the spatial distribution of objects. This representation is directly mapped into the best next action (e.g. shake the tray, pick and remove) to clear the tray. In a similar vein, Gupta & Sukhatme (2012) proposed an approach to perceive the "amount of clutter" of objects on a table. The amount of clutter maps directly into the best next action (e.g. pick an object, push the clutter) to clear the table. Hermans et al. (2012) presented an action selection method to push objects on a table and singulate them. Their method is based on the insight that pushing along the direction of visual edges between image regions would maximally help to separate objects. These methods generate an intermediate representation that maps heuristically to the most informative action. The set of possible actions is predefined and their outcome is not explicitly predicted. Differently, we do not use a representation limited to the action selection task, but the result of the perceptual systems of previous chapters that, as we already shown, supports robot manipulation of articulated objects. Also, when interacting to perceive articulated objects, the complexity of the manipulation do not allow for a simplification of the outcome as the one of the previous approaches, and requires to better predict the effect of the interaction in the constrained mechanism.

A second group of action selection methods use entropy-based information gain criteria to select the action of (expected) largest reduction on the uncertainty about the environment. van Hoof et al. (2012) presented a method to select the best pushing action to segment a cluttered scene. Their probabilistic model contains hypotheses about the regions that belong to the same object and serves as simple forward model. Our model contains more detailed kinematic and dynamic information that we use to obtain more descriptive action consequences and to generate and select more complex grasp-and-interact sequences. Hausman et al. (2015) presented a method to select the best action to gain knowledge about the kinematic constraints of an articulated object. Similar to our approach, they require an initial human interaction. They assume a known grasping pose and select the best pulling direction. Otte et al. (2014) proposed a similar entropy reduction method based on a physics simulator. Their method considers several single-joint articulated objects and selects to interact with the one that will reveal more information about the overall structure of the environment. Different to these methods, ours generates and selects autonomously complete actions –including grasping pose and manipulation trajectory– and incrementally incorporates and exploits information including dynamic properties.

Entropy-based methods require to predict 1) the outcome of an action, and 2) the influence of the outcome on the belief (through a perceptual system). Because both predictions are costly to compute, previous approaches generate a finite set of possible actions from the continuous space of action parameters based on a heuristic, and computes the most informative one. Our method addresses differently the challenges of searching for the most informative action: First, given that our perceptual system reduces entropy by accumulating motion evidences about the articulated object, we avoid the costly computation of the exact belief change for each action and predict instead the amount of actuation of the articulated mechanism. Second, we do not predefine a discrete set of actions but explore the parameter space of actions to find the most informative one.

The motion planning community has also addressed the problem of generating and planning interactions with articulated objects using knowledge about its kinematic constraints. These methods exploit the definition of the task (the manipulation of an articulated object) to simplify the generation and/or selection of actions (Prats et al., 2007, Boutselis et al., 2014, Tovar & Suárez, 2016, Pflueger & Sukhatme, 2015). We also aim to obtain task-aware actions but do not rely on given models; on the contrary, our method integrates the action generation, selection and the perceptual problem into a single process and provides interactions that

reveal more information to build a richer model. Stilman (2007) use the constraints of the articulated object to guide the search of robot trajectories in joint space. Instead of searching in the space of joint trajectories, we search in a simpler task-related action space and enforce the feasibility of the manipulation using trajectory optimization. Our goal is not to find one solution for the overly constrained motion planning problem, but rather to find the optimal solution to actuate the mechanism *and* reveal information about it.

Finally, the idea of using a physics simulator as a model for motion planning or action selection has been previously explored (Otte et al., 2014, Dogar et al., 2012). We think this is a well suited approach to avoid having to assume simplified action effects that cannot be predicted for complex articulated objects. However, our approach is essentially different to the literature because we integrate a perceptual algorithm to ground the simulation to the real world, leading to more realistic simulated action effects.

Conclusions and comparison to the proposed approach: Most previous approaches to generate informative actions for interactive perception did not tackled the manipulation of constrained mechanisms. The approaches that selected actions for articulated objects predefined a small set of actions and predicted the information gain based on simple forward models. The motion planning community tackled the interaction with complex constrained mechanisms only as a manipulation task, not as an information-gaining problem. They also assume perfect models of the constraints. We will merge concepts from interactive perception and motion planning and propose a method to generate and select informative actions autonomously. These actions are robust to uncertainties in the model of the articulated object and help to reduce them.

7.2 Physics-Based Action Selection

7.2.1 Modeling Articulated Objects

We use the perceptual systems presented in previous chapters to estimate a (partly) probabilistic model of an unknown articulated object. Based on this model we present a method that selects the action that reveals most information to improve this estimate.

We integrate the kinematic, geometric and dynamic properties perceived in previous chapters into an undirected graph, $x_{ao} := (L,\ J)$, where the set of nodes L are links and the set of edges J represent joints. A link $l_i \in L$ is represented with a triangular mesh of its shape computed from the model $Shape_i$ perceived with the method of Chapter 5. A joint $j_k \in J$ is the most likely kinematic constraint between a pair of links, and it is represented with random variables of its kinematic and dynamic properties perceived with the methods of Chapter 4 and Chapter 6:

$$\boldsymbol{j}_k := (\boldsymbol{\lambda}_k, \boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \boldsymbol{S}_k, \boldsymbol{KF}_k), \tag{7.1}$$

where $\boldsymbol{\lambda}_k$ are the joint-specific parameters, $\boldsymbol{q}_k$ is the joint's configuration, $\dot{\boldsymbol{q}}_k$ is the velocity of the joint, $\boldsymbol{S}_k$ is the force to overcome stiction (force required to initiate joint motion), and $\boldsymbol{KF}_k$ is the constant kinetic friction (force required to maintain joint motion).

The prerequisite of the perceptual systems of previous chapters is a forceful interaction with the articulated object that generates *motion* and information-rich sensor-action signals. While kinematic and shape properties can be estimated both from observing another agent interacting or (more easily) from self-interaction, the estimation of dynamic properties requires the robot to contact the object to obtain haptic sensor signals. In the previous chapter this prerequisite was circumvented by predefining contact-rich interactive manipulations, e.g.

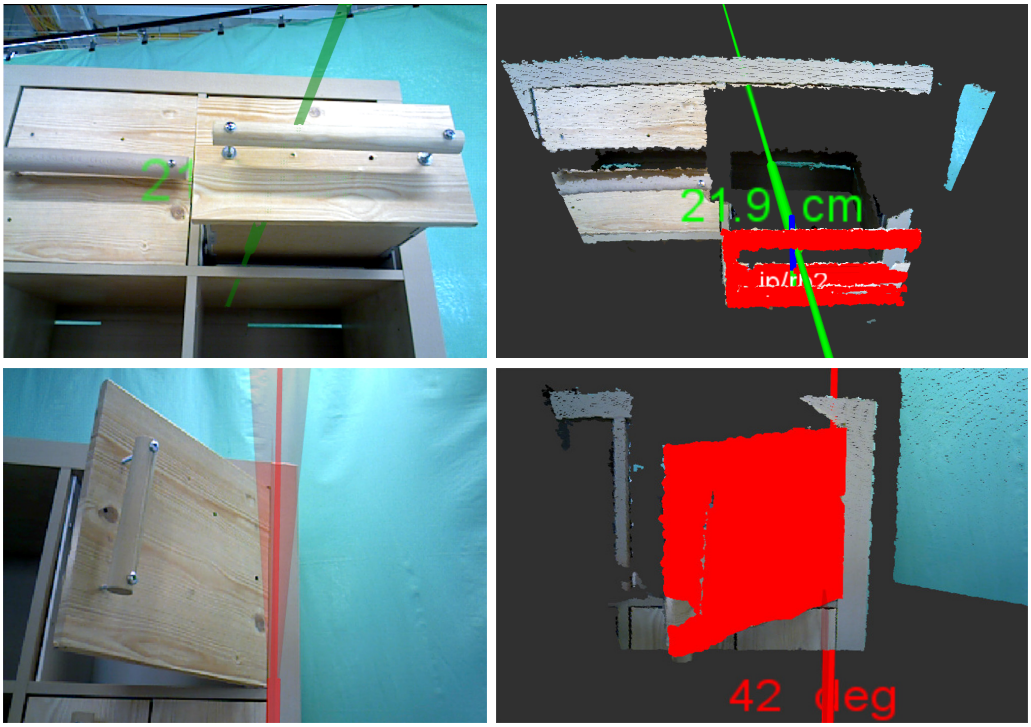


Figure 7.2: Left: robot view at the end of a human interaction with the articulated objects (estimated kinematic structure overlaid: prismatic joints in green, revolute joints in red, uncertainty indicated with translucent cones); Right: 3D visualization of the RGB-D input and the estimated kinematic model and state (reconstructed shape of the movable link in red)

with kinesthetic teaching or providing a good initialization for an impedance controller. In this work we address the fully autonomous generation and selection of the most informative interactions to be used by the interactive perceptual systems of previous chapters.

7.2.2 Selecting Actions for Articulated Objects

Our goal is to generate and select robot actions that learn as much about the articulated object as possible, i.e. decrease the uncertainty of the estimated model of the articulated object, $\boldsymbol{x}_{ao}$. To achieve this uncertainty reduction we use a task-specific objective – maximizing the motion of the articulated object – since this is the main source of information for our interactive perception method. However, when revealing information of articulated objects there are additional (kinematic) constraints that the action needs to satisfy. And because our goal is to generate actions to be executed by a real robot, the specific robot manipulator additionally restricts the actions: they have to be achievable given the kinematics of the manipulator and should not lead to collisions of the robot with the environment. Considering these require-

ments, we are looking for an action

$$a^* = \underset{a \in A}{\operatorname{argmax}}\, \Delta q(a) \tag{7.2}$$
$$\text{subject to} \quad \begin{aligned} &\text{valid\_robot\_kinematics}(a), \\ &\text{valid\_object\_kinematics}(q), \\ &\text{collision\_free}(a) \end{aligned}$$

where $\Delta q(a)$ is the change of the object's kinematic configuration induced by the robot action a.

To maximize the amount of motion and actuation of the mechanism we parametrize a by assuming three phases: reach towards a grasping/pushing pose, close the hand and move it along the estimated DoF of the mechanism. The first part is fully characterized with a grasping/pushing frame (that we assume to be on the surface of the movable link) and an approach vector towards this frame. We use a soft hand (the RBO Hand 2 presented in Deimel & Brock (2016)) in our interactions that simplifies the search problem because it adapts morphologically to the environment during the closing phase and avoids having to define additional grasping parameters. The last phase is a motion of the hand along the dimension of allowed motion of the articulated object. To avoid reaching the joint limits of the mechanism we generate motion between the borders of the joint state range observed so far. Therefore, an action a is defined as $a \in \mathbb{S}^2 \times SE(3)$.

The effect of an action a in terms of the motion $\Delta q_k(a)$ induced on the articulated object is predicted using the physics dynamic simulation SOFA (Allard et al., 2007). SOFA is a simulator that provides physically coherent interactions between an articulated object and a soft-manipulator like the RBO Hand 2. The simulation is spawned with the current estimate x_{ao} by including the reconstructed triangular meshes for each rigid body, $Shape_i$, the estimated kinematic constraints $\lambda_k, q_k, \dot{q}_k$, poses, and frictional properties S_k, KF_k. To account for the probabilistic components of x_{ao}, we draw $N_{model} = 3$ samples for each simulated action. Because the simulation of contact and interaction of the soft-manipulator with the articulated object is computationally expensive we pre-impose the kinematic constraints due to the robot manipulator on the action. We enforce that the robot's, object's kinematic constraints and collision constraints are fulfilled using a sequential convex optimization (Schulman et al., 2013a). We simulate the robot-consistent actions on the physics simulator and estimate the expected actuation of the mechanism over the samples of the belief of the environment $\Delta q_k(a')$. The action selection process is summarized in Algorithm 2.

Sampling the space of action parameters and evaluating the induced motion to find a^* is costly. We compare three sampling schemes with increasing exploitation of previous sample quality: a random mesh-based sampling (pure exploration), an evolution strategy with Gaussian moves, and a sequential sampling based on batch Bayesian optimization. The assumption of the exploitative methods is that the similar actions will result in similar outcomes. The goal is to derive a sampling strategy that requires as few samples as possible to find informative actions avoiding costly simulations. To reduce the time required in the simulation of sampled actions, we parallelize them within batches, i.e. we evaluate $N_{batches}$ batches of $N_{batchsize}$ actions. In our experiments we use $N_{batches} = 10$ and $N_{batchsize} = 100$, totalling 1000 actions.

Random Sampling

The random sampling scheme uniformly selects a point on the mesh surface, a hand orientation and approach vector. In contrast to the two other schemes it is a pure exploration

Algorithm 2 Physics-Based Action Selection

Input: $\boldsymbol{x}_{ao}$ ▷ The current estimate of the articulated object
$A \leftarrow \emptyset$, $Q \leftarrow \emptyset$ ▷ The set of all available actions and the corresponding induced articulated object motion
for $i = 1..N_{batches}$ **do**
  $A^{new} \leftarrow \text{sample}(A)$ ▷ Sample $N_{batchsize}$ actions
  $A^{new} \leftarrow \text{constrain}(A^{new})$
  for $a \in A^{new}$ **do**
    for $j = 1..N_{model}$ **do**
      $o \leftarrow \text{sample}(\boldsymbol{x}_{ao})$
      $\Delta q_k^j \leftarrow \text{simulate}(a, o)$ ▷ Simulate an action on a current articulated object sample (SOFA)
    $A \leftarrow A \cup \{a\}$, $Q \leftarrow Q \cup \{\frac{1}{N_{model}} \sum_j \Delta q_k^j\}$
  $a^* \leftarrow \underset{a \in A}{\text{argmax}}\, Q_a$
return a^*

strategy, without taking past samples and their performance into account.

Evolution Strategy

For each new batch the evolution strategy uses $N_{batchsize}$ of all best performing past actions and mutates them by adding normally distributed noise. The standard deviation of the noise decreases linearly in the number of batch iterations. This creates the effect of going from an initially exploratory behavior towards an exploitative one, similar to the temperature decrease in simulated annealing. The very first batch uses only uniformly distributed random actions, as in the random sampling strategy.

Bayesian Optimization

In vanilla Bayesian optimization, samples are drawn sequentially based on an acquisition function which is estimated from known data. We use upper confidence bounds as our acquisition function. Since we want to sample entire batches of actions instead of single ones, we use a batch Bayesian optimization approach (González et al., 2016). In this approach, samples within one batch are chosen iteratively as maximizers of the acquisition function. In each iteration a penalizing function is applied which discourages new samples in the local neighborhood of existing ones. The influence of the local penalizer depends on an estimate of the Lipschitz constant of the acquisition function which represents the smoothness of the function over the entire domain.

7.3 Experiments

We evaluate our approach by first supporting our assumption that motion indicates the amount of information gained. Based on this result, we show that the informative actions

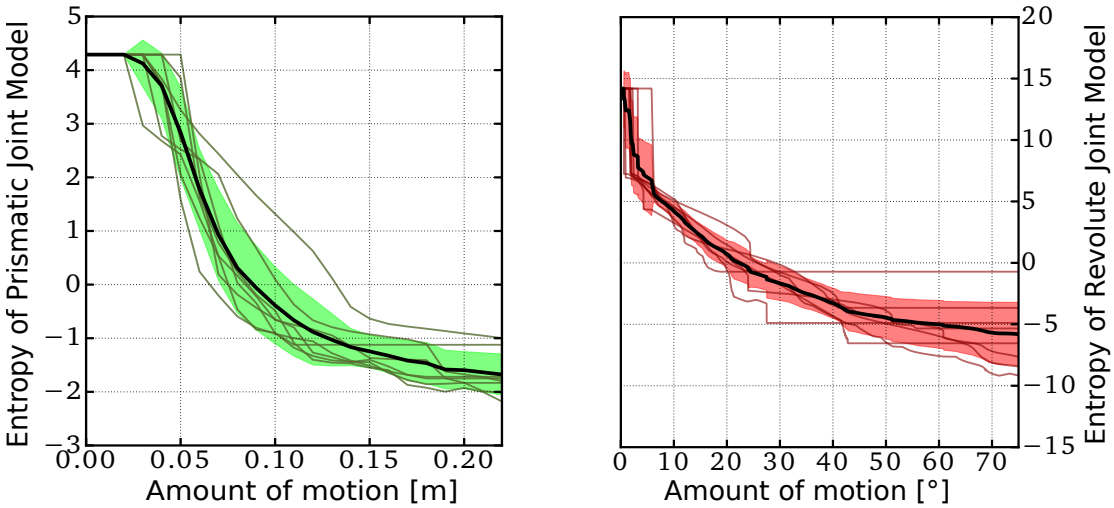


Figure 7.3: Entropy of the probabilistic model of the articulated object as a function of the amount of motion in prismatic (left) and revolute (right) joints; black curve: mean entropy of eight interactions with different objects; light green/red: standard deviation of the entropy of eight interactions with different objects; dark green and red curves indicate individual entropy reduction in each experiment; the entropy monotonically decreases with the amount of actuation of the kinematic mechanism

incrementally improve the estimated model of the environment in real world experiments, and lead to more robust actions. Finally, we find that Bayesian batch optimization is the most efficient sampling strategy.

7.3.1 Induced Motion Correlates with Information Gain

The perceptual systems we use to update the belief about the state of the environment (from previous chapters) recursively integrate sensor evidences about the constraints of motion. These systems decrease the uncertainty about the belief over the state by observing motion in the articulated object. We analyze the entropy reduction of our estimation algorithm on 16 examples of interactions with drawers and cabinet doors. This data was recorded from different point of views and contains human as well as robot interactions. Figure 7.3 depicts the mean and standard deviation of the entropy as a function of the amount of induced actuation. Since all estimations begin with the same prior belief, the initial entropy is always the same. Our experiment confirms that the entropy of the estimate decreases monotonically as more motion of the mechanism is observed.

7.3.2 Acquiring Dynamic Information Improves Interactions

To show that our method selects informative actions which allow to plan more robust manipulations, we conduct two experiments including a drawer and a cabinet door. We use a 7-DoF Barrett WAM, equipped with the pneumatically actuated RBO Hand 2 (Deimel & Brock, 2016), an Asus RGB-D sensor and an ATI FTN-Gamma force-torque sensor on the wrist (Figure 7.1).

Our approach requires an initial human interaction, since it starts with the assumption that the environment is a single static rigid body. Once the mechanism has been articulated by a human, an initial kinematic model with significant certainty can be estimated (see Fig-

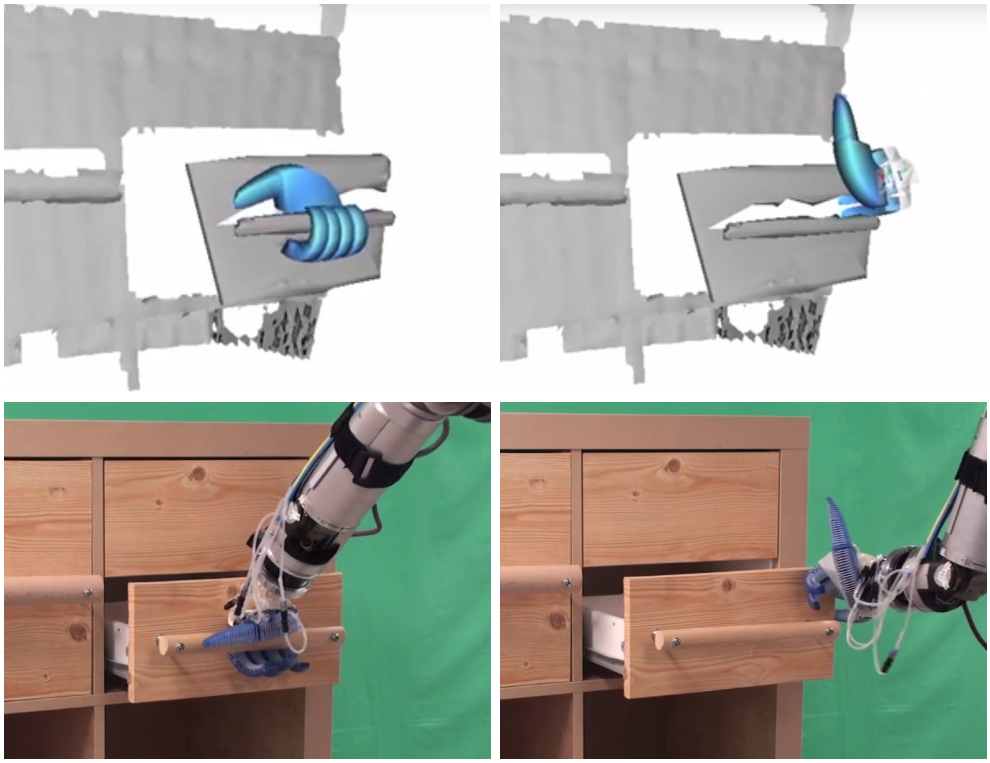


Figure 7.4: Result of the action generation and selection for different levels of uncertainty about the articulated object; First row: simulated actions; Second row: real robot execution; Left: best action for highly uncertain articulated model (action robust against uncertain kinematics and dynamics); Right: best action after the reduction of uncertainty from the execution of the robust action

ure 7.2). In contrast, the estimates of stiction and kinetic friction of the joints are still uncertain. Based on this model, our method generates and selects an interaction that maximizes the expected articulation (see Figure 7.6). By executing this action the robot gathers additional visual and haptic data to infer the joint's dynamic properties. In the drawer experiment, the estimates after this interaction are $\boldsymbol{S} \sim \mathcal{N}(3.3\,\mathrm{N}, 0.6\,\mathrm{N}^2)$ and $\boldsymbol{KF} \sim \mathcal{N}(1.1\,\mathrm{N}, 0.2\,\mathrm{N}^2)$.

Figure 7.4 shows how certainty in the estimation of the drawer's dynamic parameters affects the selected interaction. During the first interaction our method finds a rather conservative handle grasp to generate the most motion in the face of unknown joint stiction and friction. After the first action, a riskier but more tailored manipulation is selected. Actuating the drawer by pulling the edge of its front part only works because of the low known joint resistance. This action would fail if the drawer was filled (see attached video). The effect of higher certainty in the estimated model of the drawer and cabinet door is also shown in Figure 7.5. In both cases known dynamics lead to more solutions that cause large motions of the articulated object. In the cabinet door experiment the robot's haptic observation was noisier, leading to a less pronounced benefit compared to the drawer experiment.

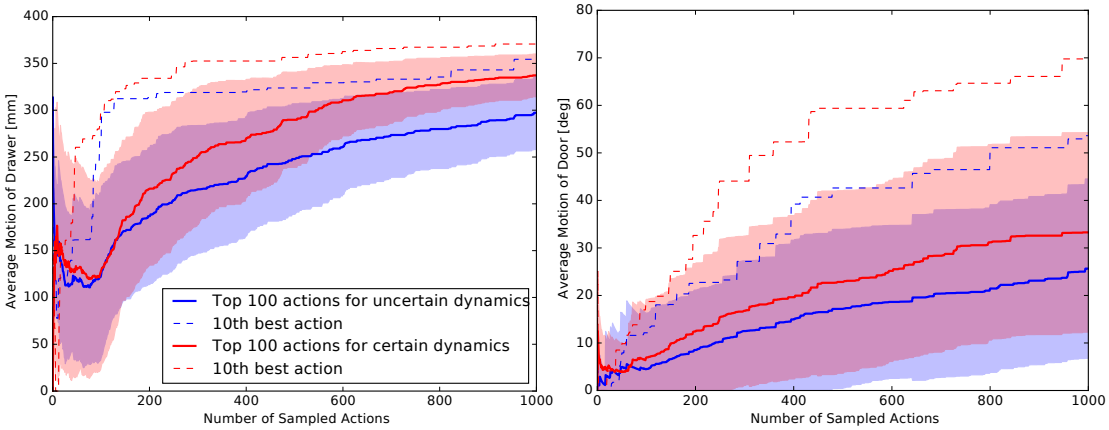


Figure 7.5: Comparison of the random mesh-based sampling strategy on the models with uncertain (blue) and certain (red) dynamic parameters; after acquiring information about the dynamics the algorithm generates and selects interactions that lead to larger motion

7.3.3 Comparison of Action Sampling Schemes

We compare our three proposed action selection strategies (random sampling, an evolution strategy, and batch Bayesian optimization) to evaluate how many samples they require to approximate the optimal action. We ran those strategies on the drawer example and selected a total of 1000 actions in ten consecutive batches. The results in Figure 7.6 show that focussing the search on promising actions – as done by the evolution strategy and Bayesian batch optimization – helps to find informative actions more quickly. The Bayesian optimization already finds multiple good solutions after 5 batches, while the evolution strategy becomes overly exploitative in the later stages.

7.4 Discussion and Limitations

We will now discuss the most severe technical limitations of our proposed approach. We will also discuss how the integration of the action selection method and the interactive perceptual systems of previous chapters, changes the way the combined system leverages the opportunities in perception for the manipulation of DoF.

The current method does not show generalization to new articulated objects but it can be applied to articulated objects with first order joints of any shape, appearance and size, as far as it can be perceived by the perceptual systems and actuated by the robot. This is a result of the versatility we pursued when developing the perceptual systems in previous chapters. To improve generality, an simple object recognition method could be used to transfer the estimated information and the successful interactions between instances of articulated objects.

All three evaluated sampling schemes are initialized with a uniform sampling. An initial sampling based on heuristics exploiting shape or kinematic information, or by the same object classification to transfer information about object classes would reduce the amount of initial exploration.

Beyond the aforementioned limitations and properties of the action selection method, the combination of action selection and the perceptual systems of previous chapters leads to changes in the way these systems exploit the opportunities in perception for robot manipula-

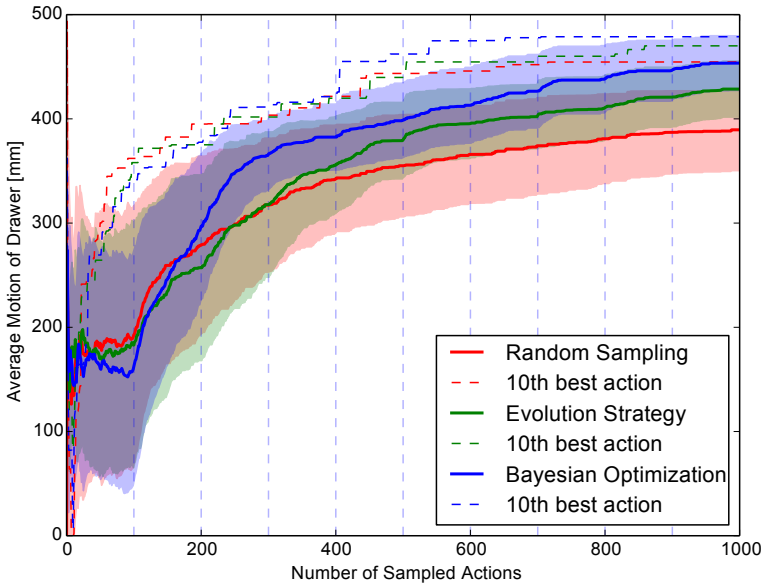


Figure 7.6: Comparison of three sampling schemes, showing the mean (solid) and standard deviation of the induced motion of the top 100 actions and the 10th best action (dashed); dashed vertical lines depict ten batches; exploitative methods find an optimal set of actions more efficiently (with less samples)

tion of DoF. From the side of the opportunities the presented method improves the way the perceptual systems exploits interactions (OP1), and physical priors (OP3). Also, the method presented here further demonstrates that the perceived information is useful for the manipulation task and to increase the autonomy of robots in unstructured environments. In the following we will analyze these changes.

Exploiting Interactions (OP1) Our action generation and selection method inherits the need for an initial interaction from our systems to perceive articulated objects, Chapter 4, 5 and 6. Without any initial information the amount of possible actions is too large to be searched randomly. However, the integration of action selection removes the need of a predefined robot interaction to improve the initial model and perceive dynamic properties. The robot can autonomously generate and select the most informative action, given the current belief over the articulated object. This represents a new level in the exploitation of interactions for perception since the overall system is not limited to interpret the action-sensor stream, but it can actively change the action-sensor stream to increase the information it contains.

Exploiting Physical Priors (OP2) While not directly for perception, the combined system exploits in a different manner known physical regularities of the environment: through the physics simulator. The simulator predicts accurately the effect of the robot interaction with a soft manipulator, acting as forward model.

We think that a remarkable property of our combination of action selection and perceptual system is that the predictive capabilities of the simulator greatly improve because we ground the physics simulator to the real world with perception. However, using the Sofa physics simulator as forward model presents a limitation: the simulations are very expensive computationally and in terms of time. This limitation, which not severe when planning next actions,

is a handicap to use the physics simulator directly as forward model for our online perceptual system.

Applying the Information for Manipulation The action selection method of this chapter operates directly on the information provided by the perceptual systems of previous chapters. Based on this information, the method of this chapter generates possible interactions with a newly encountered articulated object. These interactions are free of collisions and executable for the given robot platform and soft end-effector. We can conclude that the information provided by the perceptual systems is thus allowing to generate and select robot manipulation actions with the articulated objects.

7.5 Conclusion

We presented a method to generate and select actions for interactive perception, exploiting the insight that for a class of interactive perception methods, information gain correlates with the magnitude of the resulting motion (i.e. actuation of the articulated mechanism). The action selection method is based on the information perceived with the systems presented in the previous chapters. Based on the proposed action selection, the robot closes the loop and builds increasingly rich and accurate models of articulated objects through interactions. We presented and evaluated different action sampling schemes to reduce the costly step of predicting the effects of the contact-based interactions while still finding the optimal action parameters. We validated our approach in real-world experiments with two articulated objects of different joint types, demonstrating that the method applies to both revolute and prismatic joints.

8

Discussion and Conclusion

In this thesis we investigated how to design robot perceptual systems that acquire information to support ongoing robot mechanical manipulation. We focussed on a specific type of manipulation, the purposeful change of kinematic degrees of freedom (DoF) of the environment, and the special case when the environment contains articulated objects with constrained DoF. We now briefly summarize our main insights and then revisit the open challenges in robot perception for manipulation of DoF presented in Section 1.1.

We started this thesis (Chapter 1) identifying several challenges that perception for mechanical manipulation must overcome (CH1-CH3), and opportunities, in the form of structural properties of the problem, that perception can exploit (OP1-OP4). These structural properties are the correlation between interactions and (changes in) sensor signals (OP1) , the physical structure of the environment and the sensor signal formation (OP2), the temporal structure in the manipulation processes and its influence on the sensor stream (OP3), and the interdependencies between information extraction subprocesses (OP4). We proposed a general approach based on coupled recursive estimation processes to exploit these opportunities and overcome the challenges of robot perception for mechanical manipulation.

In Chapter 4 we presented a first interactive perceptual system to build kinematic models of articulated objects from a visual stream. The system exploited the opportunities for perception (OP1-OP4) implementing our general approach for interactive perception. We showed that the exploitation of the problem structure allows our perceptual system to build online models of the constrained DoF of the objects, and that these models can be used to monitor and steer ongoing interactions. However, the proposed system presented a limited versatility because it can perceive only articulated objects with enough color texture.

In Chapter 5 we investigated how to overcome the aforementioned limitation in versatility by integrating additional perceptual subtasks to the estimation of kinematics and leveraging their interdependencies (OP4). We proposed a new perceptual system that includes the perception of geometry (the shape of the objects) and uses the results of this process to improve the estimation of kinematic models. We showed that the integrated system overcomes the limitation of the system of Chapter 4. However, this second interactive perception system was also based solely on visual input and failed if this sensor modality did not contain enough information.

In Chapter 6 we investigated if our general approach for interactive perception can be used to integrate information from different sensor modalities in a way that overcomes their limitations. We presented a system that combines vision and proprioception in a cross-modal

manner: using information from one modality as prior to interpret the other. Cross-modality allows to leverage the interdependencies between perceptual subtasks (OP4) that use different sensor signals. We showed that the cross-modal system overcomes the limitations of vision-based interactive perception of kinematics, and obtains novel information about the dynamic properties of the articulated objects.

To exploit proprioceptive information we formulated and selected the most likely model among a set of possible grasp models that define the correlation between robot motion and motion in the environment. To define these grasp models we used prior knowledge about the morphology of the hand mounted on our robot. Therefore, our perceptual system can only be used by our robot or robots with similar hand morphology. To alleviate this limitation, we investigated how the robot can learn the interaction models correlating actions and changes in the environment and in sensor signals online. We proposed a simple online learning mechanism combining visual and proprioceptive information to generate linear models (Jacobian matrices) of the interaction that can be used for perception and for control.

In Chapter 7 we presented an approach to select robot actions to explore articulated objects based on the information acquired by the perceptual systems. Our motivation was two-folded: 1) increase the autonomy of the robot to explore its environment, and 2) demonstrate that the information obtained from our perceptual systems is relevant and useful for robot manipulation. The combination of the action selection approach and the perceptual systems allows the robot to build incremental models of the articulated objects in the environment.

We complemented the perceptual systems of chapters 4, 5 and 6 and the action selection approach of Chapter 7 with robot control and motion generation approaches to interact safely with constrained mechanisms. We proposed simple velocity-impedance controllers to guide the robot motion along the DoF of the articulated objects using force-torque signals.

8.1 Challenges in Perception for Robot Manipulation Revisited

Extract Information from Changing Sensor Signals in Correlation to Actions (CH1)

The perceptual problems tackled in this thesis and that related to the perception and manipulation of articulated objects illustrated the benefits of using interactions as part of the perceptual solution. Interactions reveal information about the objects in the form of changing sensor signals, information that is often very difficult to obtain passively, like the DoF or the dynamic properties of articulated objects. The signals that the interaction creates are correlated to the specific robot action and therefore knowledge about the interaction can be used to interpret them.

We have argued that to fully exploit these interdependencies, perceptual systems need interaction models. The models can be defined based on physical priors, but defining these models a-priori is complex for manipulation tasks involving rich contact with the environment. We explored two possibilities to tackle this problem: 1) learning models from correlated action-changing sensor signals, and 2) using a physics simulator as interaction model.

Learned interactive models can be very accurate but require large amount of training data (Agrawal et al., 2015, 2016). A sensitive solution is to assume certain properties on the model to be learned, e.g. the linear structure of the online Jacobian estimation, or the $SE(3)$ structure as demonstrated in the approach by Byravan & Fox (2017). The limitation of the learned models is that their applicability is restricted to the domain represented by the training data. How to increase the generalization of these models without increasing the amount of data is one the current research questions in the field. The most promising approach is to

incorporate physical priors into the learning process (Byravan & Fox, 2017, Jonschkowski & Brock, 2015)

Physics simulators overcome the aforementioned limitations of learned models. They are valid to any environment and task that involve physical processes modelled in the simulation, and do not require training data. However, the quality of the simulation depends strongly on the accuracy of the parameters of the physical models. A way to overcome this limitation is to ground the simulation to the real world through perception, as we showed in Chapter 7.

SUMMARY: Perception becomes easier if it exploits the structure in the combined $S \times A \times t$ space. We need to develop new methods to learn this structure and obtain interaction models correlating actions and changes in sensor signals, or link physics simulators to the real world through perception.

ONLINE PERCEPTION FROM CONTINUOUSLY ARRIVING SENSOR STREAMS (CH2)

Perception for the type of robot tasks we consider in this thesis, the mechanical manipulation of kinematic DoF, has to deliver information quickly. The robot needs to perceive and understand the consequences of its own actions on the DoF of the environment while the interaction is being performed. Based on the information perceived online the robot can monitor the task execution, detect failures and correct for them. This online requirements apply to perceptual tasks to support other types of robot manipulation.

We have seen that turning batch solutions into online perception could require a complete different perceptual approach. In our case we compensated for the loss of future sensor data using additional problem structure, e.g. using temporal recursion, leveraging physical priors and exploiting information from other perceptual subtasks.

SUMMARY: The online requirements of perceptual tasks in robot manipulation can be fulfilled if the solution leverages the structure of the problem.

BE VERSATILE: PERCEIVE IN UNSTRUCTURED ENVIRONMENTS (CH3)

We started this thesis arguing that the difficulties for perception for robotics increase when the robot needs to acquire task-relevant information in unstructured environments. The reason is that these environments are uncontrolled, dynamic and very different from one another. The robot needs to perceive in the large variety of environments we would like it to help us.

Increasing the versatility of the robot perceptual systems to different environments and tasks reduces the specificity of assumptions we can make about the environment. But as we have shown in this thesis, there is always some structural properties that are general to many environments, like the ones we exploit in our approach (OP1-OP4).

SUMMARY: To design versatile perceptual systems for robots to manipulate in unstructured environments we need to identify the regularities of the perceptual problem and propose methods to exploit them.

8.2 The Future of the Four Perceptual Opportunities Leveraged in This Thesis

The field of artificial and robot perception has changed drastically in the last years. Machine learning techniques are gaining importance in perceptual systems, replacing engineered models by statistics obtained from the data. This trend has gained momentum especially since the revival of artificial neural networks. Neural networks of many layers (known as deep neural networks, DNNs) have shown that they can extract regularities from a large amount of data thanks to the improvement in algorithms and computation, e.g. by the use of graphics processing units (GPUs) for training.

Solutions based on DNNs have reached new levels of performance in perceptual tasks like image classification, speech recognition, or even when applied to problems like reinforcement learning. The question is then, is a large amount of training data and artificial neural networks all we need to solve perception for robot manipulation? Is there any future for solutions exploiting the four problem regularities we employed in this thesis? We will conclude this thesis by discussing these important questions.

The use of neural networks does not eliminate the dependency on interactions to reveal information that cannot be perceived passively (OP1). However, we envisage a future where perceptual systems like the ones presented in this thesis provide labeled data about interactions that can be used to train a neural network. For example, labeled images of articulated objects and their properties (kinematics, dynamics) could allow a robot to learn to predict these properties without interactions, in a similar way as humans can predict the structure of a door or a laptop after interacting with some of them. Such a procedural approach using a model based system to generate data for a DNN has been successfully applied by Schmidt et al. (2017) to address the data association problem.

We already argued that one of the problems of machine learning approaches, and especially of DNNs, is that they require a large amount of training data, and that the research community is trying to reduce this dependency encoding physical priors (OP2) into the network structure. The question is not if physical priors are useful, but rather how to leverage them in artificial neural networks. This is currently an open question in artificial perception.

There are already neural network architectures that try to exploit the temporal structure of the perceptual problem (OP3). This is the goal of recurrent neural networks (RNNs) that feed the output as an additional input. Solutions based on RNNs have shown improvements in some areas of perception like semantic labeling (Xiang & Fox, 2017).

In this thesis we proposed to factorize complex perceptual problems into subtasks, and interconnect the subtasks so that they can leverage their interdependencies. This modular approach is opposed to the monolithic structure of DNN solutions. However, some recent approaches have shown that the factorization and interconnection idea can be applied to neural network architectures (Kosiorek et al., 2017). Researchers are developing new algorithmic approaches to modularize and compose DNNs in a way that leverages the interdependencies between subunits (Sabour et al., 2017).

8.3 Epilogue

Robot perception should not focus on generating complex and complete models of the environment, but on extracting the information that is relevant for the task (Aloimonos, 1990, Ballard, 1991). Based on this information robots could reach new levels of autonomy and versatility to manipulate in human environments. As Rodney Brooks said:

> If we were only able to provide the visual capabilities of a 2-year old child, robots would quickly get a lot better. (Tobe, 2015)

But we have seen that robot perception is still a hard task, even for highly constrained tasks and environments as the Amazon Picking Challenge (Correll et al., 2016). We think that increasing the perceptual capabilities of robots is possible by leveraging the right problem structure. In this thesis we identify four problem regularities that apply to many robot perceptual tasks and environments, and presented an approach to leverage them. We believe that the conceptual and technical contributions of this thesis could help to build robust and versatile robot perceptual systems that will support and enhance their manipulation capabilities.

References

Agrawal, P., Carreira, J., & Malik, J. (2015). Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 37–45).

Agrawal, P., Nair, A. V., Abbeel, P., Malik, J., & Levine, S. (2016). Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 5074–5082).

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delingette, H., & Grisoni, L. (2007). SOFA–An Open Source Framework for Medical Simulation. In *MMVR 15 - Medicine Meets Virtual Reality*, volume 125 of *Studies in Health Technology and Informatics* (pp. 13–18). Palm Beach, USA: IOP Press.

Aloimonos, J. (1988). Shape from texture. *Biological Cybernetics*, 58, 345–360.

Aloimonos, J. (1990). Purposive and qualitative active vision. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume i (pp. 346–360 vol.1).

Aloimonos, J., Weiss, I., & Bandyopadhyay, A. (1988). Active vision. *International Journal of Computer Vision*, 1(4), 333–356.

Aloimonos, Y. (1993). *Active Perception*. Psychology Press.

Aloimonos, Y. & Fermüller, C. (1995). Vision and Action. *Image and Vision Computing*, 13(10).

Atanasov, N., Sankaran, B., Ny, J. L., Pappas, G. J., & Daniilidis, K. (2014). Nonmyopic view planning for active object classification and pose estimation. *IEEE Transactions on Robotics*, 30(5), 1078–1090.

Atkeson, C. G., An, C. H., & Hollerbach, J. M. (1986). Estimation of Inertial Parameters of Manipulator Loads and Links. *The International Journal of Robotics Research*, 5(3), 101–119.

Baillargeon, R., Spelke, E. S., & Wasserman, S. (1985). Object permanence in five-month-old infants. *Cognition*, 20(3), 191–208.

Bajcsy, R. (1988). Active perception. *in Proceedings of the IEEE*, 76(8), 966 –1005.

Bajcsy, R., Aloimonos, Y., & Tsotsos, J. K. (2016). Revisiting active perception. *Autonomous Robots*, (pp. 1–20).

Ballard, D. H. (1991). Animate vision. *Artificial Intelligence*, 48.

Bar-Shalom, Y., Li, X., & Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc.

Barfoot, T. D. (2017). *State Estimation for Robotics*. Cambridge University Press.

Barfoot, T. D. & Furgale, P. T. (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3), 679–693.

Barragán, P. R., Kaelbling, L. P., & Lozano-Pérez, T. (2014). Interactive Bayesian identification of kinematic mechanisms. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 2013–2020).

Barrow, H. & Tenenbaum, J. (1978). *Recovering Intrinsic Scene Characteristics from Images*. Technical report, SRI International.

Battaglia, P. W., Hamrick, J. B., & Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *In Proceedings of the National Academy of Sciences*, 110(45), 18327–18332.

Baum, M., Bernstein, M., Martín-Martín, R., Höfer, S., Kulick, J., Toussaint, M., Kacelnik, A., & Brock, O. (2017). Opening a lockbox through physical exploration. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)*.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 346–359.

Beale, D., Iravani, P., & Hall, P. (2011). Probabilistic models for robot-based object segmentation. *Robotics and Autonomous Systems*, 59(12), 1080–1089.

Benno Heigl, Joachim Denzler, H. N. (2000). Combining computer graphics and computer vision for probabilistic visual robot navigation. In *Proceedings of the International Conference on Enhanced and Synthetic Vision* (pp. 4023–4033). Orlando, USA.

Bergström, N., Ek, C. H., Björkman, M., & Kragic, D. (2011). Scene Understanding through Autonomous Interactive Perception. In J. L. Crowley, B. A. Draper, & M. Thonnat (Eds.), *Computer Vision Systems*, number 6962 in Lecture Notes in Computer Science (pp. 153–162). Springer Berlin Heidelberg.

Blake, A. & Yuille, A. (1993). *Active Vision*. MIT Press.

Bogo, F., Black, M. J., Loper, M., & Romero, J. (2015). Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 2300–2308).

Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., & Sukhatme, G. S. (2017). Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*. to appear.

Boutselis, G. I., Bechlioulis, C. P., Liarokapis, M. V., & Kyriakopoulos, K. J. (2014). Task specific robust grasping for multifingered robot hands. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 858–863).

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1), 14–23.

Bruyninckx, H. & Schutter, J. D. (1996). Specification of force-controlled actions in the "task frame formalism" - a synthesis. *IEEE Transactions on Robotics*, 12(4), 581–589.

Byravan, A. & Fox, D. (2017). Se3-nets: Learning rigid body motion using deep neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 173–180).

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679–698.

Chang, L., Smith, J. R., & Fox, D. (2012). Interactive singulation of objects from a pile. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 3875–3882).

Channoufi, I., Bourouis, S., Hamrouni, K., & Bouguila, N. (2016). Deformable models based object tracking: Challenges and current researches. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS)* (pp. 35–40).

Chaudhary, K., Au, C., Chan, W. P., Nagahama, K., Yaguchi, H., Okada, K., & Inaba, M. (2016). Retrieving unknown objects using robot in-the-loop based interactive segmentation. In *IEEE/SICE International Symposium on System Integration (SII)* (pp. 2474–2325). Sapporo, Japan.

Chaumette, F., Boukir, S., Bouthemy, P., & Juvin, D. (1996). Structure from controlled motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5), 492–504.

Chaumette, F. & Hutchinson, S. (2006). Visual servo control. I. Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4), 82–90.

Chien, S.-Y., Ma, S.-Y., & Chen, L.-G. (2002). Efficient moving object segmentation algorithm using background registration technique. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(7), 577–586.

Choi, C. & Christensen, H. I. (2012). Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features. *The International Journal of Robotics Research*, 31(4), 498–519.

Choi, C. & Christensen, H. I. (2013). RGB-D object tracking: A particle filter approach on GPU. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1084–1091).

Collet, A., Martinez, M., & Srinivasa, S. S. (2011). The MOPED framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10), 1284 –1306.

Cooley, J. W. & Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90), 297–301.

Corke, P. (2011). *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer.

Corke, P. (2017). *Robotics, Vision and Control - Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition, Second Edition*, volume 118 of *Springer Tracts in Advanced Robotics*. Springer.

Correll, N., Bekris, K. E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J. M., & Wurman, P. R. (2016). Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, (pp. 1–17).

Costeira, J. P. & Kanade, T. (1998). A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3), 159–179.

Craig, J. J. (2005). *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River.

Curless, B. & Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques* (pp. 303–312).: ACM.

De Laet, T., Bellens, S., Smits, R., Aertbelien, E., Bruyninckx, H., & De Schutter, J. (2013). Geometric relations between rigid bodies (part 1): Semantics for standardization. *IEEE Robotics and Automation Magazine*, 20(1), 84–93.

Deimel, R. & Brock, O. (2014). A novel type of compliant, underactuated robotic hand for dexterous grasping. In *Proceedings of Robotics: Science and Systems (RSS)* (pp. 1687–1692).

Deimel, R. & Brock, O. (2016). A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35, 161–185.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (pp. 248–255).

Dogar, M., Hsiao, K., Ciocarlie, M., & Srinivasa, S. (2012). Physics-based grasp planning through clutter. In *Proceedings of Robotics: Science and Systems (RSS)*.

Dupont, P. E. (1990). Friction modeling in dynamic robot simulation. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 1370–1376).

Endres, F., Hess, J., Sturm, J., Cremers, D., & Burgard, W. (2014). 3D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1), 177–187.

Endres, F., Trinkle, J., & Burgard, W. (2013). Learning the dynamics of doors for robotic manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3543–3549). Tokyo, Japan.

Eppner, C., Martín-Martín, R., & Brock, O. (2017). Physics-based selection of actions that maximize motion for interactive perception. In *RSS workshop: Revisiting Contact - Turning a problem into a solution*.

Ernst, M. & Banks, M. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415, 429–433.

Ersen, M., Oztop, E., & Sariel, S. (2017). Cognition-enabled robot manipulation in human environments: Requirements, recent work, and open problems. *IEEE Robotics and Automation Magazine*, (pp. 108–122).

Everitt, B. & Skrondal, A. (2002). *The Cambridge dictionary of statistics*, volume 106. Cambridge University Press Cambridge.

Faugeras, O. D. (1992). What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of the European Conference on Computer Vision* (pp. 563–578).: Springer.

Felzenszwalb, P. F. & Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2), 167–181.

Fitzpatrick, P. (2003). First contact: an active vision approach to segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3 (pp. 2161–2166).

Fitzpatrick, P., Metta, G., Natale, L., Rao, S., & Sandini, G. (2003). Learning about objects through action-initial steps towards artificial cognition. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*, volume 3 (pp. 3140–3145).

Fitzpatrick, P. M. & Metta, G. (2002). Towards manipulation-driven vision. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1 (pp. 43–48).

Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). Svo: Fast semi-direct monocular visual odometry. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 15–22).

Fox, D., Burgard, W., Dellaert, F., & Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349), 2–2.

Furch, J. & Eisert, P. (2012). Robust key point matching for dynamic scenes. In *Proceedings of the European Conference on Visual Media Production (CVMP)*.

Gallagher, S. (2006). *How the body shapes the mind*. Clarendon Press.

Garcia Cifuentes, C., Issac, J., Wüthrich, M., Schaal, S., & Bohg, J. (2017). Probabilistic articulated real-time tracking for robot manipulation. *IEEE Robotics and Automation Letters (RA-L)*, 2.

Gibson, J. J. (1966). *The Senses Considered as Perceptual Systems*. Greenwood Press Reprint, 1 edition.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Routledge.

Gilden, D. L. & Proffitt, D. R. (1989). Understanding collision dynamics. *Journal of Experimental Psychology: Human Perception and Performance*, 15(2), 372.

Golub, G. H. & Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

González, J., Dai, Z., Hennig, P., & Lawrence, N. (2016). Batch bayesian optimization via local penalization. In *Artificial Intelligence and Statistics* (pp. 648–657).

Gonzalez-Aguirre, D., Hoch, J., Röhl, S., Asfour, T., Bayro-Corrochano, E., & Dillmann, R. (2011). Towards shape-based visual object categorization for humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 5226–5232).

Grossberg, S. & Rudd, M. E. (1992). Cortical dynamics of visual motion perception: short-range and long-range apparent motion. *Psychological review*, 99 1, 78–121.

Gupta, M. & Sukhatme, G. S. (2012). Using manipulation primitives for brick sorting in clutter. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 3883–3889).

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference* (pp. 10–5244). Manchester, UK: Alvey.

Hartley, R. & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Hausman, K., Balint-Benczedi, F., Pangercic, D., Marton, Z.-C., Ueda, R., Okada, K., & Beetz, M. (2013). Tracking-based interactive segmentation of textureless objects. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 1122–1129).

Hausman, K., Niekum, S., Osentoski, S., & Sukhatme, G. S. (2015). Active articulation model estimation through interactive perception. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 3305–3312). Seattle, USA.

Hayman, E. (2000). *The use of zoom within active vision*. PhD thesis, University of Oxford.

Hebert, P., Hudson, N., Ma, J., Howard, T., Fuchs, T., Bajracharya, M., & Burdick, J. (2012). Combined shape, appearance and silhouette for simultaneous manipulator and object tracking. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 2405–2412).

Held, R. & Hein, A. (1963). Movement-produced stimulation in the development of visually guided behavior. *Journal of Comparative and Physiological Psychology*, 56(5), 872–876.

Herbst, E., Henry, P., & Fox, D. (2010). Towards online 3D object segmentation and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 2610–2617). Hong Kong, China.

Hermans, T., Rehg, J. M., & Bobick, A. (2012). Guided pushing for object singulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4783–4790).

Hespos, S. J. & vanMarle, K. (2012). Physics for infants: Characterizing the origins of knowledge about objects, substances, and number. *Wiley Interdisciplinary Reviews: Cognitive Science*, 3(1), 19–27.

Hjørland, B. & Christensen, F. S. (2002). Work tasks and socio-cognitive relevance: A specific example. *Journal of the American Society for Information Science and Technology*, 53(11), 960–965.

Hsiao, K., Kaelbling, L. P., & Lozano-Perez, T. (2007). Grasping POMDPs. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 4685–4692).

Huang, X., Walker, I., & Birchfield, S. (2012). Occlusion-aware reconstruction and manipulation of 3D articulated objects. *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*, (pp. 1365–1371).

Ilonen, J., Bohg, J., & Kyrki, V. (2014). Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing. *The International Journal of Robotics Research*, 33(2), 321–341.

International Federation of Robotics (2016). Press conference 2016 - robotics world market. https://tinyurl.com/ifr-market, Accessed: 2017-09-01.

Jägersand, M., Fuentes, O., & Nelson, R. (1997). Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*, volume 4 (pp. 2874–2880).

Jain, A., Nguyen, H., Rath, M., Okerman, J., & Kemp, C. C. (2010). The complex structure of simple devices: A survey of trajectories and forces that open doors and drawers. In *Proceedings of the IEEE, RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)* (pp. 184–190).

Jentzsch, S., Gaschler, A., Khatib, O., & Knoll, A. (2015). MOPL: A multi-modal path planner for generic manipulation tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6208–6214). Hamburg, Germany.

Jonas, J. B., Schmidt, A. M., Müller-Bergh, J., Schlötzer-Schrehardt, U., & Naumann, G. (1992). Human optic nerve fiber count and optic disc size. *Investigative ophthalmology & visual science*, 33(6), 2012–2018.

Jonschkowski, R. & Brock, O. (2015). Learning state representations with robotic priors. *Autonomous Robots*, 39(3), 407–428.

Kaelbling, L. P., Littman, M., & Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101.

Karayiannidis, Y., Smith, C., Barrientos, F. E. V., Ögren, P., & Kragic, D. (2016). An adaptive control approach for opening doors and drawers under uncertainties. *IEEE Transactions on Robotics*, 32.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4), 321–331.

Kato, Z. & Pong, T.-C. (2001). A markov random field image segmentation model using combined color and texture features. In *Computer Analysis of Images and Patterns* (pp. 547–554).: Springer.

Katz, D. & Brock, O. (2007). Interactive perception: Closing the gap between action and perception. In *IEEE International Conference on Robotics and Automotion (ICRA). Workshop: From features to actions-Unifying perspectives in computational and robot vision.*

Katz, D. & Brock, O. (2008). Manipulating articulated objects with interactive perception. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 272–277).

Katz, D. & Brock, O. (2011a). A factorization approach to manipulation in unstructured environments. In *Proceedings of the International Symposium on Robotics Research (ISRR)* (pp. 285–300). Springer.

Katz, D. & Brock, O. (2011b). Interactive segmentation of articulated objects in 3D. In *IEEE International Conference on Robotics and Automation (ICRA). Workshop on Mobile Manipulation: Integrating Perception and Manipulation.*

Katz, D., Kazemi, M., Bagnell, J., & Stentz, A. (2013a). Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 5003–5010).

Katz, D., Kazemi, M., Bagnell, J. A., & Stentz, A. (2013b). Interactive segmentation, tracking, and kinematic modeling of unknown articulated objects. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 5003–5010).

Katz, D., Orthey, A., & Brock, O. (2014). Interactive perception of articulated objects. In O. Khatib, V. Kumar, & G. Sukhatme (Eds.), *Experimental Robotics*, volume 79 of *Springer Tracts in Advanced Robotics* (pp. 301–315). Springer Berlin Heidelberg.

Katz, D., Venkatraman, A., Kazemi, M., Bagnell, J. A., & Stentz, A. (2013c). Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. In *Proceedings of Robotics: Science and Systems (RSS)* Berlin, Germany.

Kemp, C. C., Edsinger, A., & Torres-Jara, E. (2007). Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics and Automation Magazine*, 14(1), 20–29.

Kenney, J., Buckley, T., & Brock, O. (2009). Interactive segmentation for manipulation in unstructured environments. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 1377–1382).

Kerl, C., Sturm, J., & Cremers, D. (2013). Dense visual slam for RGB-D cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2100–2106).

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1), 43–53.

Khoshelham, K. & Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2), 1437–1454.

Knill, D. C. & Richards, W. (1996). *Perception as Bayesian inference*. Cambridge University Press.

Kosiorek, A., Bewley, A., & Posner, I. (2017). Hierarchical attentive recurrent tracking. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 3055–3063).

Koval, M. C., Dogar, M. R., Pollard, N. S., & Srinivasa, S. (2013). Pose estimation for contact manipulation with manifold particle filters. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 4541–4548). Karlsruhe, Germany.

Koval, M. C., Pollard, N. S., & Srinivasa, S. S. (2015). Pose estimation for planar contact manipulation with manifold particle filters. *The International Journal of Robotics Research*, 34(7), 922–945.

Kragic, D., Björkman, M., Christensen, H. I., & Eklundh, J. O. (2005). Vision for robotic object manipulation in domestic settings. *Robotics and Autonomous Systems*, 52(1), 85–100.

Krainin, M., Henry, P., Ren, X., & Fox, D. (2011). Manipulator and object tracking for in-hand 3D object modeling. *The International Journal of Robotics Research*, 17.

Kristjánsson, Á., Eyjólfsdóttir, K. Ó., Jónsdóttir, A., & Arnkelsson, G. (2010). Temporal consistency is currency in shifts of transient visual attention. *PloS one*, 5(10).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 1097–1105).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 1097–1105).

Krüger, N., Geib, C., Piater, J., Petrick, R., Steedman, M., Wörgötter, F., Ude, A., Asfour, T., Kraft, D., Omrčen, D., et al. (2011). Object-action complexes: Grounded abstractions of sensory-motor processes. *Robotics and Autonomous Systems*, 59(10), 740–757.

Kulick, J., Otte, S., & Toussaint, M. (2015). Active exploration of joint dependency structures. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 2598–2604). Seattle, USA.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Machine Learning*, (pp. 282–289).

Le, A., Jung, S.-W., & Won, C. (2014). Directional joint bilateral filter for depth images. *Sensors*, 14(7), 11362–11378.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *In Proceedings of the IEEE*, 86(11), 2278–2324.

Lenz, I., Knepper, R., & Saxena, A. (2015). Deepmpc: Learning deep latent features for model predictive control. In *Proceedings of Robotics: Science and Systems (RSS)*.

Leonard, J. J. & Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics*, 7(3), 376–382.

Lepetit, V. & Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9), 1465–1479.

Levi, P. & Kernbach, S. (2010). Cognitive approach in artificial organisms. *Symbiotic Multi-Robot Organisms*, (pp. 165–228).

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39), 1–40.

Li, W. H. & Kleeman, L. (2011). Segmentation and modeling of visually symmetric objects by robot actions. *The International Journal of Robotics Research*, 30(9), 1124 –1142.

Liu, C. & Tomizuka, M. (2015). Safe exploration: Addressing various uncertainty levels in human robot interactions. In *American Control Conference (ACC), 2015* (pp. 465–470).

Livingstone, M., Hubel, D., et al. (1988). Segregation of form, color, movement, and depth- anatomy, physiology, and perception. *Science*, 240(4853), 740–749.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 91–110.

Lucas, B. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 674–679).

Lynch, K. & Park, F. (2017). *Modern Robotics: Mechanics, Planning and Control*. Cambridge University Press.

Ma, D. & Hollerbach, J. M. (1996). Identifying mass parameters for gravity compensation and automatic torque sensor calibration. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*, volume 1 (pp. 661–666).

Ma, L. & Sibley, G. (2014). Unsupervised dense object discovery, detection, tracking and reconstruction. In *Computer Vision-ECCV 2014* (pp. 80–95). Springer.

Maljkovic, V. & Martini, P. (2005). Implicit short-term memory and event frequency effects in visual search. *Vision Research*, 45(21), 2831–2846.

Maljkovic, V. & Nakayama, K. (1994). Priming of pop-out: I. role of features. *Memory & cognition*, 22(6), 657–672.

Marr, D. (1982). *Vision: A computational approach*. Freeman & Co., San Francisco.

Martín-Martín, R. (2014). Online interactive perception. https://github.com/tu-rbo/omip, Accessed: 2017-09-01.

Martín-Martín, R. (2016). Online multi-modal interactive perception. https://github.com/tu-rbo/omip/tree/omip2, Accessed: 2017-09-01.

Martín-Martín, R. & Brock, O. (2014). Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2494–2501). Chicago, USA.

Martín-Martín, R. & Brock, O. (2017a). Building kinematic and dynamic models of articulated objects with multi-modal interactive perception. In AAAI (Ed.), *AAAI Symposium on Interactive Multi-Sensory Object Perception for Embodied Agents* (pp. 473–476).

Martín-Martín, R. & Brock, O. (2017b). Cross-modal interpretation of multi-modal sensor streams in interactive perception based on coupled recursion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Martín-Martín, R., Höfer, S., & Brock, O. (2016a). An integrated approach to visual perception of articulated objects. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 5091 – 5097).

Martín-Martín, R., Sieverling, A., & Brock, O. (2016b). Estimating the relation of perception and action during interaction. In *International Workshop on Robotics in the 21st century: Challenges and Promises*.

Martinez-Hernandez, U., Dodd, T. J., Evans, M. H., Prescott, T. J., & Lepora, N. F. (2017). Active sensorimotor control for tactile exploration. *Robotics and Autonomous Systems*, 87(Supplement C), 15 – 27.

Mason, M. T. (2001). *Mechanics of robotic manipulation*. MIT press.

Matsuyama, T., Wu, X., Takai, T., & Wada, T. (2004). Real-time dynamic 3D object shape reconstruction and high-fidelity texture mapping for 3D video. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3), 357–369.

Matula, D. W. (1987). Determining edge connectivity in 0 (nm). In *Foundations of Computer Science, 1987., 28th Annual Symposium on* (pp. 249–251).

McCloskey, M., Caramazza, A., & Green, B. (1980). Curvilinear motion in the absence of external forces: Naive beliefs about the motion of objects. *Science*, 210(4474), 1139–1141.

McGurk, H. & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, 264(5588), 746–748.

Michel, D., Zabulis, X., & Argyros, A. A. (2014). Shape from interaction. *Machine Vision and Applications*, 25(4), 1077–1087.

Miksik, O., Munoz, D., Bagnell, J. A., & Hebert, M. (2013). Efficient temporal consistency for streaming video scene analysis. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 133–139).

Miller, A. & Allen, P. (2004). Graspit! a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4), 110–122.

Mishra, A., Aloimonos, Y., & Fermuller, C. (2009). Active segmentation for robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3133–3139).

Motion Analysis (2017). Motion analysis corporation. http://ftp.motionanalysis.com, Accessed: 2017-11-01.

Nagel, H.-H. & Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5), 565–593.

Nayar, S. K. & Nakagawa, Y. (1994). Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 824–831.

Newcombe, R. A., Fox, D., & Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (pp. 343–352).

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., & Fitzgibbon, A. (2011a). KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of the IEEE International Symposium on Mixed and augmented reality (ISMAR)* (pp. 127–136).

Newcombe, R. A., Lovegrove, S., & Davison, A. (2011b). DTAM: Dense tracking and mapping in real-time. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 2320–2327).

Niemi, P. & Näätänen, R. (1981). Foreperiod and simple reaction time. *Psychological bulletin*, 89(1), 133.

Nieuwenhuisen, M., Stückler, J., Berner, A., Klein, R., & Behnke, S. (2012). Shape-primitive based object recognition and grasping. In *Proceedings of ROBOTIK; German Conference on Robotics* (pp. 1–5).: VDE.

Nistér, D., Naroditsky, O., & Bergen, J. R. (2004). Visual odometry. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1 (pp. I–I).

Noë, A. (2006). *Action in Perception*. MIT Press, 2 edition.

Nusseck, M., Lagarde, J., Bardy, B., Fleming, R., & Bülthoff, H. H. (2007). Perception and prediction of simple object interactions. In *Proceedings of the 4th symposium on Applied perception in graphics and visualization* (pp. 27–34).: ACM.

Ochs, P., Malik, J., & Brox, T. (2014). Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6), 1187–1200.

O'Regan, J. K. (2011). *Why Red Doesn't Sound Like a Bell: Understanding the Feel of Consciousness*. Oxford Univ Pr, 1 edition.

Oren, M. & Nayar, S. K. (1994). Generalization of lambert's reflectance model. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (pp. 239–246).: ACM.

Otte, S., Kulick, J., Toussaint, M., & Brock, O. (2014). Entropy-Based strategies for physical exploration of the environment's degrees of freedom. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 615–622). Chicago, USA.

Pahlavan, K., Uhlin, T., & Eklundh, J. O. (1993). Active Vision as a methodology. In *Active Perception*. Psychology Press.

Papazov, C., Haddadin, S., Parusel, S., Krieger, K., & Burschka, D. (2012). Rigid 3D geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, 31(4), 538–553.

Papon, J., Abramov, A., Schoeler, M., & Worgotter, F. (2013a). Voxel cloud connectivity segmentation - Supervoxels for point clouds. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (pp. 2027–2034).

Papon, J., Kulvicius, T., Aksoy, E. E., & Wörgötter, F. (2013b). Point cloud video object segmentation using a persistent supervoxel world-model. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3712–3718).

Park, J. & Kim, K. (2014). Tracking on lie group for robot manipulators. In *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)* (pp. 579–584).

Part, S. (1985). Impedance control: An approach to manipulation. *Journal of dynamic systems, measurement, and control*, 107, 17.

Pauwels, K. & Kragic, D. (2015). Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1300–1307).

Pflueger, M. & Sukhatme, G. S. (2015). Multi-step planning for robotic manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 2496–2501).

Phong, B. T. (1975). Illumination for computer generated pictures. *Communications of the ACM*, 18(6), 311–317.

Pillai, S., Walter, M. R., & Teller, S. (2015). Learning articulated motions from visual demonstration. In *Proceedings of Robotics: Science and Systems (RSS)*.

Poelman, C. J. & Kanade, T. (1997). A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3), 206–218.

Pomerleau, F., Magnenat, S., Colas, F., Liu, M., & Siegwart, R. (2011). Tracking a depth camera: Parameter exploration for fast ICP. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3824–3829).

Prats, M., Sanz, P. J., & del Pobil, A. P. (2007). Task-oriented grasping using hand preshapes and task frames. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 1794–1799).

Proffitt, D. R., Kaiser, M. K., & Whelan, S. M. (1990). Understanding wheel dynamics. *Cognitive psychology*, 22(3), 342–373.

Ramachandran, V. S. (1988). Perception of shape from shading. *Nature*, 331 6152, 163–6.

Rao, R. P. & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1), 79–87.

Ren, C. Y., Prisacariu, V., Murray, D., & Reid, I. (2013). STAR3d: Simultaneous tracking and reconstruction of 3D objects using RGB-D data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 1561–1568).

Rizzi, A., Koditschek, D. E., & others (1996). An active visual estimator for dexterous manipulation. *IEEE Transactions on Robotics*, 12(5), 697–713.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. New York, NY, USA: Springer-Verlag New York, Inc.

Rosenblum, L. D., Miller, R. M., & Sanchez, K. (2007). Lip-read me now, hear me better later. *Psychological Science*, 18(5), 392–396.

Rosenfeld, A., Hummel, R. A., & Zucker, S. W. (1976). Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6), 420–433.

Rosenfeld, A. & Pfaltz, J. L. (1966). Sequential operations in digital picture processing. *Journal of the ACM (JACM)*, 13(4), 471–494.

Ross, D., Tarlow, D., & Zemel, R. (2008). Unsupervised learning of skeletons from motion. In *Proceedings of the European Conference on Computer Vision* (pp. 560–573).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rusinkiewicz, S., Hall-Holt, O., & Levoy, M. (2002). Real-time 3d model acquisition. *ACM Transactions on Graphics (TOG)*, 21(3), 438–446.

Rusu, R., Holzbach, A., Diankov, R., Bradski, G., & Beetz, M. (2009). Perception for mobile manipulation and grasping using active stereo. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)* (pp. 632–638).

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 3857–3867).

Salganicoff, M., Bajcsy, R., & Mitchell, T. (1992). *Learning for Coordination of Vision and Action*. Technical report, University of Pennsylvania.

Schenck, C. & Fox, D. (2016). Detection and tracking of liquids with fully convolutional networks. In *Proceedings of Robotics Science and Systems (RSS) Workshop Are the Skeptics Right? Limits and Potentials of Deep Learning in Robotics*.

Schenck, C. & Fox, D. (2017). Reasoning about liquids via closed-loop simulation. *arXiv preprint arXiv:1703.01656*.

Schiebener, D., Schill, J., & Asfour, T. (2012). Discovery, segmentation and reactive grasping of unknown objects. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)* (pp. 71–77). Osaka, Japan.

Schiebener, D., Ude, A., & Asfour, T. (2014). Physical interaction for segmentation of unknown textured and non-textured rigid objects. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 4959–4966).

Schmidt, T., Newcombe, R., & Fox, D. (2017). Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2), 420–427.

Schmidt, T., Newcombe, R. A., & Fox, D. (2014). Dart: Dense articulated real-time tracking. In *Proceedings of Robotics: Science and Systems (RSS)*.

Schneider, A., Sturm, J., Stachniss, C., Reisert, M., Burkhardt, H., & Burgard, W. (2009). Object identification with tactile sensors using bag-of-features. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 243–248).

Schulman, J., Ho, J., Lee, A. X., Awwal, I., Bradlow, H., & Abbeel, P. (2013a). Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Proceedings of Robotics: Science and Systems (RSS)*, volume 9 (pp. 1–10).

Schulman, J., Lee, A., Ho, J., & Abbeel, P. (2013b). Tracking deformable objects with point clouds. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 1130–1137).

Se, S., Lowe, D., & Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The International Journal of Robotics Research*, 21(8), 735–758.

Seitz, S. & Szeliski, R. (1999). Applications of computer vision to computer graphics. *Computer Graphics*, 33(4), 35–37.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1), 1–114.

Shashua, A. (1995). Multiple-view geometry and photometry. In *Asian Conference on Computer Vision* (pp. 393–404).: Springer.

Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888–905.

Shi, J. & Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (pp. 593–600).

Sigal, L. & Black, M. J. (2006). Predicting 3D people from 2D pictures. *AMDO*, 6, 185–195.

Sinapov, J., Bergquist, T., Schenck, C., Ohiri, U., Griffith, S., & Stoytchev, A. (2011). Interactive object recognition using proprioceptive and auditory feedback. *The International Journal of Robotics Research*, 30(10), 1250–1262.

Sinapov, J., Schenck, C., & Stoytchev, A. (2014). Learning relational object categories using behavioral exploration and multimodal perception. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 5691–5698).

Smallwood, R. & Sondik, E. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21.

Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles* (pp. 167–193). Springer.

Smoljkic, G., Borghesan, G., Reynaerts, D., Schutter, J. D., Sloten, J. V., & Poorten, E. V. (2015). Constraint-based interaction control of robots featuring large compliance and deformation. *IEEE Transactions on Robotics*, 31(5), 1252–1260.

Spelke, E. S., Vishton, P., & Von Hofsten, C. (1995). Object perception, object-directed action, and physical knowledge in infancy. In *The cognitive neurosciences* (pp. 165–179). Cambridge, MA, US: The MIT Press.

Stilman, M. (2007). Task constrained motion planning in robot joint space. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3074–3081).

Stückler, J. & Behnke, S. (2012). Model learning and real-time tracking using multi-resolution surfel maps. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 2081–2087).

Stückler, J. & Behnke, S. (2015). Efficient dense rigid-body motion segmentation and estimation in RGB-D video. *International Journal of Computer Vision*, (pp. 1–13).

Sturm, J., Bylow, E., Kahl, F., & Cremers, D. (2013). Copyme3d: Scanning and printing persons in 3d. In *German Conference on Pattern Recognition* (pp. 405–414).: Springer.

Sturm, J., Jain, A., Stachniss, C., Kemp, C. C., & Burgard, W. (2010a). Operating articulated objects based on experience. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2739–2744).

Sturm, J., Konolige, K., Stachniss, C., & Burgard, W. (2010b). Vision-based detection for learning articulation models of cabinet doors and drawers in household environments. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 362–368).

Sturm, J., Stachniss, C., & Burgard, W. (2011). A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41(2), 477–526.

Sturm, J., Stachniss, C., Pradeep, V., Plagemann, C., Konolige, K., & Burgard, W. (2009). Learning kinematic models for articulated objects. In *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 1851–1856).

Thrun, S., Burgard, W., & Fox, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5(3-4), 253–271.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Tobe, F. (2015). Amazon challenges robotics' hot topic: perception. The Robot Report, https://www.therobotreport.com/amazon-challenges-robotics-hot-topic-perception/, Accessed: 2017-09-01.

Tomasi, C. & Kanade, T. (1991). *Detection and tracking of point features*. Technical report, Carnegie Mellon University.

Tomasi, C. & Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2), 137–154.

Tomasi, C. & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 839–846).

Tovar, N. A. & Suárez, R. (2016). Grasp synthesis of 3d articulated objects with n links. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1–6).

Tresadern, P. & Reid, I. (2005). Articulated structure from motion by factorization. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2 (pp. 1110–1115).

Tsikos, C. J. & Bajcsy, R. K. (1988). *Segmentation via Manipulation*. Technical Report, University of Pennsylvania Department of Computer and Information Science, Pennsylvania.

Tsikos, C. J. & Bajcsy, R. K. (1991). Segmentation via manipulation. *IEEE Transactions on Robotics*, 7(3), 306–319.

Ude, A., Omrcen, D., & Cheng, G. (2008). Making object learning and recognition an active process. *International Journal of Humanoid Robotics*, 5(2), 267–286.

Ullman, S. (1979). The interpretation of structure from motion. *Proceedings of the Royal Society of London B: Biological Sciences*, 203(1153), 405–426.

van Hoof, H., Kroemer, O., Ben Amor, H., & Peters, J. (2012). Maximally informative interaction learning for scene exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5152–5158).

van Hoof, H., Kroemer, O., & Peters, J. (2013). Probabilistic interactive segmentation for anthropomorphic robots in cluttered environments. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)* (pp. 169–176).

van Hoof, H., Kroemer, O., & Peters, J. (2014). Probabilistic segmentation and targeted exploration of objects in cluttered environments. *IEEE Transactions on Robotics*, 30(5), 1198–1209.

Varela, F. J., Thomson, E., & Rosch, E. (1993). *The Embodied Mind: Cognitive Science and Human Experience*. Mit Pr, new edition edition.

Venture, G., Ayusawa, K., & Nakamura, Y. (2009). A numerical method for choosing motions with optimal excitation properties for identification of biped dynamics-an application to human. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 1226–1231).

Wagner, A. (2016). Pre-gibsonian observations on active touch. *History of psychology*, 19(2), 93.

Walsman, A., Schmidt, T., & Fox, D. (2017). Articulated tracking with a dynamic high-resolution surface model. In *Robotics: Science and Systems. Workshop on Articulated Model Tracking*.

Wang, J. & Olson, E. (2016). Apriltag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4193–4198).

Weise, T., Wismer, T., Leibe, B., & Van Gool, L. (2009). In-hand scanning with online loop closure. In *IEEE International Conference on Computer Vision: Workshops (ICCV Workshops)* (pp. 1630–1637).

Weng, S.-K., Kuo, C.-M., & Tu, S.-K. (2006). Video object tracking using adaptive Kalman filter. *Journal of Visual Communication and Image Representation*, 17(6), 1190–1208.

Wertheimer, M. (1912). Experimentelle Studien über das Sehen von Bewegung. *Zeitschrift für Psychologie*, 61, 161–265.

Whitehead, S. D. & Ballard, D. H. (1990). Active Perception and Reinforcement Learning. *Neural Computation*, 2(4).

Willimon, B., Birchfield, S., & Walker, I. (2011). Classification of clothing using interactive perception. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 1862–1868).

Wuthrich, M., Pastor, P., Kalakrishnan, M., Bohg, J., & Schaal, S. (2013). Probabilistic object tracking using a range camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3195–3202).

Xiang, Y. & Fox, D. (2017). DA-RNN: Semantic mapping with data associated recurrent neural networks. In *Proceedings of Robotics: Science and Systems (RSS)*.

Xinjilefu, X., Feng, S., Huang, W., & Atkeson, C. G. (2014). Decoupled state estimation for humanoids using full-body dynamics. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 195–201).

Xu, K., Huang, H., Shi, Y., Li, H., Long, P., Caichen, J., Sun, W., & Chen, B. (2015). Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Transactions on Graphics (TOG)*, 34(6), 177.

Yan, J. & Pollefeys, M. (2006). Automatic kinematic chain building from feature trajectories of articulated objects. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1 (pp. 712–719).

Young, P. C. (2012). *Recursive estimation and time-series analysis: an introduction*. Springer Science & Business Media.

Zhang, L. & Trinkle, J. C. (2012). The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 3805–3812).

Zhou, J., Paolini, R., Bagnell, J. A., & Mason, M. T. (2016). A convex polynomial force-motion model for planar sliding: Identification and application. In *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* (pp. 372–377).