DISS. ETH NO. 19517

Capturing and Synthesizing Hand-Object Interaction

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences (Dr. sc. ETH Zürich)

presented by

Henning Hamer
Dipl.-Inf., University of Ulm
born August 10, 1979
citizen of Konstanz, Germany

accepted on the recommendation of

Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Reinhard Klein, co-examiner
Prof. Dr. Konrad Schindler, co-examiner

2011

TO MY PARENTS.

Abstract

Humans use their hands for interacting with their environment. In particular the ability of hands to manipulate objects dominates our daily life. As a consequence, there are many useful digital applications in the context of hand-object interaction. Realizing such applications requires both knowledge about the manipulating hand and knowledge regarding the manipulated object. The proposed thesis is concerned with the extraction of such information by means of computer vision methods. It also explores interdependencies between object and hand: the manipulation of a specific object implicates appropriate executing hand poses. This ultimately leads to a method to synthesize static grasps as well as dynamic hand motion for object manipulation: the knowledge gained with a camera system is used to ease the generation process of computer animations containing hand-object interaction.

To gather information about the manipulating hand, we first develop a markerless method for tracking the articulated pose of a hand interacting with an object. The scenario implies the challenge of severe self-occlusions of the hand as well as occlusions by the object. We approach the task by belief propagation and use range and color data as input. This data is delivered by a real-time structured-light system.

Next, the manipulated object is not only considered as an occluder but plays an active role. We introduce an object-dependent hand pose prior. The prior represents knowledge regarding the manipulation of a certain object by a certain hand. It can be generalized towards new hands and new objects of the same object class (e.g., from one cup to another). We present two applications of this: improved 3d hand tracking, and grasp synthesis for unobserved hands and objects. Grasp synthesis is highly relevant in the fields of robotics and computer graphics.

Finally, we examine temporal aspects of hand pose generation and extend static grasp synthesis to dynamic hand motion synthesis. The idea is to generate

computer animations containing hand-object interaction solely by animating the object. The corresponding hand information is derived from observation. While animating an object can be easily done by a 3d artist, manually animating a hand is substantially more complicated due to the many degrees of freedom. Our method requires training once for an object of interest. After that, arbitrary manipulation sequences can be realized in 3d modeling software like Autodesk Maya.

Zusammenfassung

Menschen gebrauchen ihre Hände für die Interaktion mit ihrer Umgebung. Besonders die Möglichkeit, mit Händen Objekte zu manipulieren, beherrscht unseren Alltag. Aus dieser Tatsache ergibt sich eine Vielzahl von nützlichen digitalen Anwendungen im Bereich der Hand-Objekt-Interaktion. Die Realisierung solcher Anwendungen erfordert einerseits Wissen über die manipulierende Hand selbst und andererseits Wissen über das manipulierte Objekt. Die vorgelegte Dissertation befasst sich mit der Gewinnung solchen Wissens mittels Computer Vision Methoden. Darüber hinaus werden Abhängigkeiten zwischen dem Objekt und der Hand erforscht: die Manipulation eines speziellen Objekts impliziert entsprechende ausführende Handposen. Dies führt schließlich zu einer Methode zur Erzeugung von sowohl statischen als auch dynamischen Handposen: das Wissen, welches mit einem Kamerasystem gewonnen wird, dient der erleichterten Generierung von Animationen mit Hand-Objekt-Interaktion.

Um Informationen über die manipulierende Hand zu erhalten, entwickeln wir zunächst eine Methode ohne Marker zum Tracken der artikulierten Pose einer Hand, welche mit einem Objekt interagiert. Dieses Szenario bringt Schwierigkeiten mit sich, die durch starke Selbstverdeckungen der Hand und Verdeckungen durch das Objekt bedingt sind. Wir begegnen den Problemen mit dem Belief-Propagation Algorithmus und verwenden sowohl Tiefen- als auch Farbinformationen. Diese Daten werden in Echtzeit von einem Structured-Light-System geliefert.

Danach fungiert das manipulierte Objekt nicht mehr nur als Verdecker sondern nimmt eine aktive Rolle ein. Wir stellen einen objekt-spezifischen Handposenprior vor. Dieser Prior repräsentiert Wissen bezüglich der Manipulation eines bestimmten Objekts durch eine bestimmte Hand, und er ist generalisierbar bezüglich neuer Hände und neuer Objekte der gleichen Objektklasse (zum Beispiel von einer Tasse zu einer anderen). Wir präsentieren zwei Anwendungen hiervon: erstens verbessertes 3d Handtracking und zweitens Griffsynthese

für zuvor nicht beobachtete Hände und Objekte. Die Griffsynthese hat große Relevanz in den Feldern der Robotik und der Computergraphik.

Schließlich untersuchen wir die zeitlichen Aspekte der Handposenerzeugung und erweitern die statische Griffsynthese hin zur Synthese von dynamischen Handbewegungen. Die Grundidee ist die Erzeugung von Computeranimationen, die Hand-Objekt-Interaktion enthalten, allein durch die Animation des Objekts. Die entsprechende Handinformation wird von Beobachtungen abgeleitet. Während es für einen 3d-Animationskünstler relativ einfach ist, ein Objekt zu animieren, ist die direkte Animation einer Hand wegen der vielen Freiheitsgrade deutlich schwieriger. Unsere Methode benötigt eine einmalige Trainingsprozedur für ein gewünschtes Objekt. Danach können beliebige Manipulationssequenzen in 3d Modellierungssoftware, wie zum Beispiel Autodesk Maya, realisiert werden.

Acknowledgements

The research presented in this thesis has been supported by the EC integrated projects CHIRON and 3D-COFORM.

This work would not have been possible without the support and assistance of many others.

First and foremost, I would like to thank my advisor Prof. Dr. Luc Van Gool for the opportunity to work in his research group. I am very grateful for his expert advice and insightful suggestions throughout my research. I would also like to thank my supervisors Dr. Esther Koller-Meier, Prof. Dr. Konrad Schindler, and Dr. Jürgen Gall for excellent guidance. I appreciate the collaboration with Prof. Dr. Ryusuke Sagawa and Prof. Dr. Raquel Urtasun.

Furthermore I thank Prof. Dr. Reinhard Klein and Prof. Dr. Konrad Schindler for reviewing this dissertation.

My appreciation goes to my colleagues at the Computer Vision Laboratory at ETHZ. I particularly thank Dr. Thibaut Weise, Dr. Raphael Höver, Dr. Philipp Fürnstahl, Dr. Michael Van Den Bergh, Alain Lehmann, and Dr. Helmut Grabner for many valuable discussions. A special thank to my office mates Fabian Nater, Severin Stalder, and Tobias Gass for support and a friendly atmosphere in D115. I am also grateful to the secretaries Barbara Widmer, Christina Krüger, and Vreni Vogt for taking care of administrative matters.

Finally, I would like to express my profound gratitude to my family and my friends. My family encouraged me in many ways and believed in me throughout the years.

Contents

List of Figures

1

Introduction

The scope of this work is hand-object interaction. As a starting point, we observe hands manipulating objects and derive information based on computer vision methods. After considering hands and objects in isolation, we focus on the inherent interdependencies. One application of the gained knowledge is the synthesis of interactive hand motion for animated sequences. However, there are many more as we describe below.

The need to study the interaction of human hands and manipulated objects has already been recognized in 1956 by the primatologist, and physician John R. Napier. The driving motivation for this research at that time was the *"evaluation of disability of the hand in connection with industrial and insurance work"* [Napier 1956]. The relevance of such research for the medical application field has significantly increased since then due to the great advances in biomedical engineering. Furthermore, new possibilities for innovative applications in the context of hand-object interaction have also opened up in other fields. The following list introduces some motivating scenarios.

- The topic of hand disability touched upon in [Napier 1956] and rehabilitation has new significance because of the recent development of high tech hand prostheses [Light and Chappell 2000, Warwick *et al.* 2003, Pons *et al.* 2004, Schulz *et al.* 2005, Carrozza *et al.* 2006, Cipriani *et al.* 2009]. Such devices imitate the anatomy of the human hand and some can be linked to the central nervous system of a patient. First prototypes allow for both the control of the device by the brain as well as sensory feedback delivered in the same way.
- In augmented reality, an interaction scenario may be observed. Hand and/or object can then be replaced by virtual surrogates. For example in

[Harders *et al.* 2009], a haptic effector is replaced by a digital scalpel for surgery simulation. In [Seo *et al.* 2008], the manipulated object (e.g., a tea pot) itself is virtual.

- In robotics, artificial end-effectors interact with objects. In many cases the design of such end-effectors is inspired by the human hand. While there are some approaches to let a robot explore objects autonomously, another promising concept is *programming by demonstration* (PbD): real hand-object interaction is observed and either directly transferred to a grasping device or even learned by an intelligent robot.
- In computer graphics, the hands of virtual characters are animated. Even without object interaction, this is not a trivial task because of the complexity of realistic looking hand models. Interdependencies between hand and object during object manipulation are even harder to model. Incorporating the knowledge derived from the observation of real hand-object manipulation has great potential to ease the animation process.

A promising concept with respect to these applications is to use the knowledge derived from observing real hands performing some kind of object manipulation. In general we need two types of information: 1) information about the manipulating hand itself, and 2) information about the manipulated object and the stable grasps it offers.

First we go into **information about the hand**. We are interested in the *articulated* pose of a hand during manipulation, i.e., the posture of every revolute joint within the hand. An X-ray of a human hand is shown in Figure 1.1(a). The *degrees of freedom* (DOFs) between hand segments are constrained by the joints: hinge joints between finger phalanges allow only for bending within a certain range; saddle joints connecting the fingers to the palm additionally allow for spreading the fingers. The wrist has 6 DOFs (translation and rotation). Depending on the exact hand model, at least 26 parameters have to be determined.

One approach to estimate these joint angles is to use a data glove (Figure 1.1(b)). The problem with this technique is its invasiveness, which means the user might be disturbed in her/his natural hand movements. To avoid this issue, we focus on non-invasive vision-based methods in this work.

Visual hand pose estimation has, yet without objects being involved, several important applications. Examples include motion capture, sign language recog-

(a) Human hand *(b)* CyberGlove II©

Figure 1.1*: a) An X-ray of a human hand. b) An invasive data glove used to capture the articulated hand pose of the user.*

nition, intuitive human computer interaction, human behavior and emotion analysis, and safety and process integrity control on the work-floor. Not surprisingly, much research has already gone into computer algorithms for hand pose estimation. A summary of state-of-the-art methods can be found in [Erol *et al.* 2007].

Even when only considering free hands, hand pose estimation is a challenging problem. Common difficulties encountered are the "*The Curse of Dimensionality*", self-occlusions of the hand, and rapid hand motions. In addition, the presence of objects has a significant impact on the complexity and generality of the task. Firstly, the manipulated objects will frequently occlude parts of the hand, and hand poses occurring during the process of grabbing or holding will aggravate the problem of self-occlusion. Secondly, the hand structure itself is less constrained in the presence of objects: when in contact with an object, forces are exerted on the hand, resulting in poses which cannot be achieved with the bare hand (e.g., bending fingers backwards when pressing against a rigid surface). Tracking hands visually under these less favorable conditions is pretty much uncharted terrain.

Next we turn to **information about the object** and how to capture it. The objects discussed in this work are either rigid, or consist of several rigid parts

***Figure 1.2**: Some of the objects considered in this work: a cup, a pair of pliers, a camera, and a clamshell phone.*

that are attached to each other (these objects are from now on called *articulated objects*). See Figure 1.2 for some examples. To be able to relate hand and object to each other at a later stage, we are looking for the exact shape (a 3d model) of the object together with its translation and rotation (6 DOFs). For articulated objects, we seek the shape of the object in its different extreme articulations.

A common approach to obtain the model of an object is to scan the object in 3d. In general, 3d scanning consists of three steps: 1) acquiring a set of 3d range images of the object, 2) aligning the individual surface patches, and 3) integrating these patches into one coherent mesh. In the past, this used to be a time-consuming and costly process. However, an efficient, low cost in-hand modeling solution was recently presented in [Weise 2009]. The user presents all sides of the object to be scanned in front of a structured-light system. The structured-light system delivers dense depth data in real-time. While the object is being reconstructed online, the user can interactively fill in any remaining

holes by offering views of the missing object parts to the scanner. This makes the scanning process fast, intuitive, and ideally suited for our purposes.

Object manipulation is an inherently 3-dimensional phenomenon, whereas 3d pose estimation in monocular video is seriously under-constrained. We therefore choose the structured-light system introduced in [Weise 2009] as our input device, and we derive both information about the hand and the object from this data. The camera-setup delivers not only dense depth but also color information. This is essential for separating hand and object by means of skin color segmentation.

After considering the extraction of hand and object knowledge independently, we now elaborate on **interdependencies**. Most obviously, the hand exerts forces on the object via contact points and vice versa. On the one side, this may result in hand poses which would be unnatural without object contact. On the other side, forces acting on the object can affect the object's 6d pose, i.e the object is translated and/or rotated. In case of an articulated object, forces applied by the hand can cause the transition from one articulated object state to another, for example when a clamshell phone is opened or closed.

Besides physical interaction, there are semantic dependencies. A specific grasp implicitly indicates the handled object (e.g., a hand seems to hold a pen). Vice versa, objects have *affordances* [Gibson 1979], that is they suggest certain hand poses for their manipulation (e.g., a cup is held by its handle). Some effort has been made within the vision community to use these causalities for action and object interpretation [Mann *et al.* 1996, Kjellström *et al.* 2008], but there has hardly been any work exploiting the strong relation between the detailed 3d shape of an object and the corresponding articulated hand poses.

A key point regarding some of the applications suggested earlier is the ability to **generalize interaction knowledge** towards new hands and new objects of the same kind. Humans master this task very well: by watching another person handle a single instance of an unknown object class, they can easily imitate the observed hand poses to manipulate other instances of the same object class. Coming back to the PbD scenario in robotics, it would be useful to demonstrate to a robot how to handle a cup, but it would be even better if the robot could use this knowledge to interact with a variety of different cups. In computer graphics a 3d artist could load the model of a previously unobserved cup into the system. She/He could then generate a grasp for this cup and a hand with desired anatomical proportions.

So far we have neglected the **temporal nature of hand-object interaction**. Although the correct grasps are crucial for object manipulation, it is hand motion that actually changes the state of an object. Temporal considerations bring up new issues, like possible delays between manipulating hand poses and the effect on the object (e.g., the zoom of the camera in Figure 1.2 emerges shortly after a button has been pressed). The application example we adopt in this work is the automatic generation of animated 3d sequences containing hand-object interaction. To this end, we exploit observed object affordances. As stated before, the manual animation of hands manipulating objects is hard. Besides the many DOFs of hands, unnatural looking contacts between hand and object are major problems. Again, as a way out marker-based motion capture systems are used in industry to track the hand and the object. However, it is difficult to alter the performance of an actor retroactively to produce new animations in our context [Kry and Pai 2006]. Innovative solutions to create such animations have high value. For example, we believe there is a great demand for virtual video tutorials demonstrating the usage of tools and devices.

1.1 Contributions

This work contributes to the state of the art regarding the vision-based observation and synthesis of hand-object interaction in three different ways.

First, we present a method for tracking a hand, while it is interacting with an object. To achieve robustness to occlusions, we use an individual local tracker for each segment of the articulated structure. The segments are connected in a pairwise *Markov random field* (MRF), which enforces the anatomical hand structure by putting soft constraints on the joints between neighboring segments. The best hand configuration is found with *belief propagation* (BP). Both range and color data are used as input. Experiments are presented for synthetic data with ground truth and for real data of people manipulating objects.

Next, we turn to the object, and propose an object-dependent hand pose prior that integrates the direct relation between a manipulating hand and a 3d object. Inspired by the ability of humans to learn the handling of an object from a single example, our focus lies on very sparse training data. We express estimated hand poses in local object coordinates, and extract for each individual hand segment the relative position and orientation as well as contact points on the

object. The prior is then modeled as a spatial distribution conditioned on the object. Given a new object of the same object class and new hand dimensions, we can transfer the prior by a procedure involving a geometric warp. In our experiments, we demonstrate that the prior may be used to improve the robustness of the 3d hand tracker and to synthesize a new hand grasping a new object. For this, we integrate the prior into a unified belief propagation framework for tracking and synthesis.

Finally, we introduce a new technique for the synthesis of animated 3d sequences containing hand-object interaction. Objects suggest certain hand poses and motions for their manipulation. This causality between object state and hand motion is first observed and then exploited for a simplified animation procedure: our method takes an animation of an object as input and generates the corresponding hand motion automatically. The approach is data-driven; sequences of hands manipulating objects are captured with the structured-light setup. The training data is then combined with a new animation of the object in order to generate a plausible animation featuring hand-object interaction. Such an animation can optionally be further processed with a commercial animation tool.

1.2 Organization

The remainder of this work is structured as follows.

In **Chapter 2**, we first discuss related literature. The presented work is related not only to computer vision, but also to many other areas of research. Medical studies of human physiology are fundamental. Insights gained from neuroscience with respect to the control of human hands by the brain bear potential to improve technical systems like ours. Furthermore, there are many connections to fields like robotics, artificial intelligence, and action recognition.

Methods that are important for this work are the topic of **Chapter 3**. To begin with, a general description of our input system, a real-time structured-light system, is given. Besides range data, the system also provides color information. We employ skin color segmentation to separate the hand from the object, and concisely describe how this works. Further, an important technique we use to align 3d data obtained by the scanner is the *iterative closest point* (ICP) method. We introduce the main idea of the method and scanner-specific adaptations. On the basis of the above methods, we touch on the concept of in-hand

scanning, a technique to obtain mesh representations of objects demonstrated to the scanner. Thereafter, we present one of our two hand models, namely the synthetic hand often used for the generation of 3d hand data in the course of this work. Lastly, when we infer articulated hand poses, we do so by belief propagation. Hence, we summarize this algorithm.

In **Chapter 4** we develop our method for tracking a hand which is manipulating an object. Special focus lies on robustness to self-occlusions and occlusions caused by the object. After a short overview of the matter, we introduce our second hand model, the local model used for hand pose estimation. Then, the central issue is the tracking method itself, i.e., how we sequentially estimate hand poses by processing recorded manipulation sequences using belief propagation. More precisely, we explain how we compare the hand model to the range data, the way we approach the problem of occlusions, and how we enforce anatomical constraints of the hand. Implementation details are provided for reproducibility. Finally, we present results on both artificial and real data.

In **Chapter 5** we introduce the object-dependent hand pose prior. After motivating statistical priors in general, a formal definition of the proposed prior is given: a spatial distribution over hand poses conditioned on the object. We then show how object and hand information are related to each other in practice. The following sections of that chapter are concerned with the transfer of manipulation knowledge specific to some hand and some object to new hands and new objects of the same object class. In the result section we demonstrate how the prior improves the hand tracking from Chapter 4 and how it can be used to make the synthetic hand (Chapter 3) grasp previously unobserved objects.

Chapter 6 is dedicated to our method to generate 3d animations containing hand-object interaction, based on 1) an animation of the involved object and 2) object affordances derived by observation. It is first explained how the method can alleviate the work of 3d animation artists. We then focus on the training procedure required once for a new object. In more detail, we identify 1) the various states of the object during manipulation, 2) the hand configurations that cause object state transitions, and 3) the spatio-temporal correlations between key hand poses and key object poses. After that, we turn to the testing stage. In this stage, new animations of an observed object can be realized with little effort. Results feature animations containing rigid as well as articulated objects.

Finally, we summarize and discuss our work in **Chapter** 7. First, the individual contributions are reviewed retrospectively, and conceptual differences with regard to other works are highlighted. Then, we consider the presented different building blocks on a more abstract level, and classify our methods in the greater context of hand-object interaction. This naturally leads to a discussion of current limitations that should be addressed, and to an outlook on possible opportunities for the future. We then conclude our work by discussing the meaning and significance of the proposed thesis.

2

Related Work

2.1 Human Physiology

The earliest research related to ours originates from the field of medical science. Most obviously, anatomical studies of the human hand are fundamental since many methods discussed in this work require adequate hand models. Already in 1858, Henry Gray published anatomical studies on hands and other body parts in the first edition of *Gray's Anatomy*, as stated in the newest edition [Standring and Borley 2008]. Since then countless studies have followed. Regarding our work the biomechanical works [Buchholz and Armstrong 1992] and [Hamilton and Dunsmuir 2002] have special significance. They touch on dependencies between individual hand segments, and these dependencies have high relevance for the hand models presented in the course of this work. When modeling the human hand, the thumb is the most difficult part, and many medical studies are concerned with the mechanisms within the thumb [Cooney *et al.* 1981, Buchholz and Armstrong 1992, Hollister *et al.* 1992, McDonald *et al.* 2001]. More information on the structure of hands is provided when we detail our hand models.

A driving force in biomedical engineering for studies of hand-object interaction is the development of progressive *transradial prostheses*, i.e., hand prostheses attached to the forearm. A main purpose of such devices lies in the regained ability of disabled persons to manipulate objects. For this, hand prostheses need to be *dexterous* (allowing an arbitrary change of the location and rotation of a manipulated object) and *prehensile* (suited for grasping). To assess these properties for new prostheses, grasp taxonomies with respect to certain object classes have been suggested. Research in this area ranges from work in the

early twentieth century by Schlesinger [Schlesinger 1919] over the pioneering work of Napier [Napier 1956] to more recent extensions like [Cutkosky and Wright 1986]. [Schlesinger 1919] introduced the basic idea of classifying hand poses with respect to manipulated objects, considering the hand as a tool. Grasps were categorized regarding cylindrical and spherical objects of different sizes. Identified grasps involve palmar prehension, tip prehension, lateral prehension, hook prehension, nippers prehension, as well as certain pinches. In [Napier 1956], the trade-off between power and precision requirements for grasping is addressed. While *power grasps* are appropriate for applying great forces, *precision grasps* are more suitable when small adjustments of posture are needed for better control. [Cutkosky and Wright 1986] later divided power and precision grasps into more sub-categories, again with respect to different manipulated objects. A comprehensive overview of grasp taxonomies can be found in [Iberall 1997].

Just as anatomical research of hands is important for the mechanical design of hand prostheses, studies of the human brain are relevant to derive better control mechanisms for such devices. As an introduction to the topic consider Figure 2.1. The manikin in this figure illustrates how much of the human cortex is used for the motor control of the different body parts. Early work indicating the great relative portion necessary for hand control was already presented in [Penfield and Rasmussen 1950].

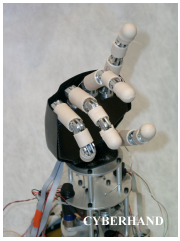
Despite the great relative proportion of the brain dedicated to the control of hands, there is strong evidence that humans usually only use a small subspace of all anatomical possible hand configurations. Grasp taxonomies can be seen as a first attempt to address this issue. Beyond that, researchers try to identify *synergies* in recorded hand data, seeking lower dimensional representations for a set of high dimensional hand poses. This is typically done by applying *principal component analysis* (PCA) to identify linear subspaces termed *eigengrasps* in [Ciocarlie *et al.* 2007]. The various works differ in the selection of the considered hand poses. In the experiment described in [Santello *et al.* 1998] test persons wore a data-glove and were asked to shape their hand appropriately to grasp one of 57 familiar objects. More than 80% of the grasp variance could be covered by the first two principal components. According to [Mason *et al.* 2001], the true dimensionality of hand poses during *reaching* is even lower than the one of grasp poses. [Todorov and Ghahramani 2004] showed that the set of principal components varies depending on the proposed manipulation task, and that more principal components than stated in [Santello *et al.*

***Figure 2.1**: Motor Homunnculus, the size of each body part of the figure reflects the proportion of the cortex dedicated to the control of the body part. Image courtesy of the National History Museum, London.*

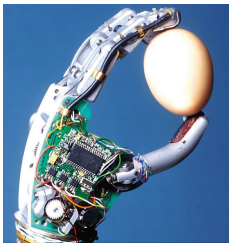
1998] are required to represent complex manipulations like flipping pages or crumbling paper. The authors of [Thakur *et al.* 2008] focus on the even less constrained scenario of object exploration. In that work, the first seven principal components capture more than 90% of the variance in hand posture. In contrast to linear PCA, non-linear manifolds are identified in [Tsoli and Jenkins 2007], which allows for a better separation of task-specific subspaces. A recent discussion on the matter of dimensionality reduction with regard to hands is provided in [Ciocarlie and Allen 2009].

The value of the research motivated by the necessity of artificial limbs is demonstrated by impressive modern hand prostheses [Light and Chappell 2000, Warwick *et al.* 2003, Pons *et al.* 2004, Schulz *et al.* 2005, Carrozza *et al.* 2006, Cipriani *et al.* 2009]. See Figure 2.2 for some examples. Myoelectric prostheses [Light and Chappell 2000, Pons *et al.* 2004, Schulz *et al.* 2005] are controlled by the voltage generated when flexing the large muscles in the forearm. This technique requires no surgery of the patient but permits only limited control of the artificial hand. Another approach is to connect a neural interface to the *central nervous system* (CNS) [Warwick *et al.* 2003, Carrozza *et al.* 2006, Cipriani *et al.* 2009]. This allows more control as well as feedback from sensors of the artificial hand to the human brain. Though neural interfaces open up great possibilities, implants are currently not suited for long-term use.

(a) Southampton hand



(b) CYBERHAND



(c) SNAVE hand

***Figure 2.2**: Some examples of recently developed hand prostheses: the Southampton hand, the CYBERHAND, and the SNAVE hand.*

Besides biomedical engineers, researchers from the robotics community take an active part in studying hand-object interaction. Designing high-tech prostheses and robotic hands are very related tasks. However, before going into the issue of robotic grasping, we will first discuss methods for vision-based hand pose estimation.

2.2 Vision-Based Hand Pose Estimation

The *pose* of a hand is defined in diverse ways in the computer vision literature. In the most basic case, the term refers to the 2d location of the hand in an image (*hand localization*). In *hand pose recognition*, the space of all possible configurations of the hand is discretized into a set of examples or classes, and the pose of a hand expresses its class assignment. Finally, in *model-based hand pose estimation*, the pose of a hand covers all continuous parameters of an articulated 3d hand model (e.g., the position of the wrist and the angles of revolute joints).

Tracking a hand means to estimate its pose in each frame of a continuous sequence. In most works, hand tracking is limited to free hands without object interaction, but we will also describe some exceptions.

2.2.1 Hand Localization

Hand localization is the task of finding the position of a hand in an image. Hand localization in continuous image sequences is crucial for the application area of *gesture recognition*, i.e., the recognition of signs encoded in human hand motion. Consequently a variety of approaches for hand localization have been developed over the last years. Reviews can be found in [Pavlovic *et al.* 1997, Wu and Huang 1999, Wu and Huang 2001].

One fundamental technique is to find hands via their skin color. The tone of skin is quite characteristic and usually lies somewhere between brown and red. Some approaches dynamically adapt skin color segmentation to the current skin and lighting conditions. For instance in [Van Den Bergh *et al.* 2009], the skin model is updated online based on color information from the face, found with a cascaded classifier [Viola and Jones 2004]. Examples of methods for gesture recognition seeking skin colored blobs include [Krahnstoever *et al.* 2002, Argyros and Lourakis 2004, Cooper and Bowden 2007, Farhadi *et al.* 2007, Starner *et al.* 1998].

Another approach explored early is based on shape matching. In these works, typically the silhouette of a hand (e.g., detected by skin color) is compared to some kind of template. One of the first representatives of this approach was the system of Cipolla and Hollinghurs [Cipolla *et al.* 1994, Cipolla and Hollinghurst 1996], which tracks a pointing hand by fitting an active contour as a 2D shape model. [Freeman and Weissman 1995] realized a vision-based television remote control by matching a template via local edge orientation.

Given the success of the method introduced in [Viola and Jones 2004] with respect to face detection, some authors have attempted to detect hands using cascaded classifiers and rectangular 2d haarlets [Kolsch and Turk 2004, Ong and Bowden 2004, Micilotta *et al.* 2005]. However, the appearance of hands is more variable than that of faces. To cope with this, more training examples are commonly added, but this also increases the chances of falsely detecting hands in cluttered background.

Two state-of-the-art systems for gesture recognition are described in [Cooper and Bowden 2007] and [Buehler *et al.* 2008]. In [Cooper and Bowden 2007], a method is presented which avoids tracking for reasons of robustness and instead performs skin-color-based detection in each frame. A set of classifiers is then first used to detect units of activity considering the placement, motion,

and arrangement of the hands. Thereafter, these units are assembled into words via Markov chains. Results are presented for the data-set of Kadir et al. [Kadir *et al.* 2004]. The authors of [Buehler *et al.* 2008] focus on long sequences (more than one hour in length) and try to detect hands robustly by modeling their connection to the human upper body by a pictorial structure.

2.2.2 Hand Pose Recognition

While gesture recognition often focuses on hand location and hand motion, hand pose recognition is explicitly concerned with the different hand configurations that can be adopted by a human hand. Hand localization is often required as a pre-processing step for pose recognition.

Techniques for hand pose recognition are typically example-based. In a *training* stage, a representative set of hand configurations is presented to such a system, so that a discrete representation of the high DOF state space of the hand can be learned. Then, during *testing*, new hand configurations can be classified with respect to the examples encountered during training.

The advantage of the example-based approach is that it often works well on a per-frame basis and therefore can achieve high robustness. On the downside, discretization of the state space means loss of information, i.e., when presenting an arbitrary hand configuration to the system, the output is always constrained to the observed training examples.

Many methods for hand pose recognition are based on eigenspace methods, i.e., on PCA. A pioneering system of this kind was presented in [Starner *et al.* 1998]. To recognize American sign language, skin colored blobs are tracked through sequences. Examples presented to the system are encoded in a 16-element feature vector, characterizing skin blob shape via area, angle of the axis of least inertia defined by the first eigenvector, length of this eigenvector, and eccentricity of a bounding ellipse. Complete sentences are recognized with a *hidden Markov model* (HMM). In [Black and Jepson 1998], images of hands are interpreted as 1d vectors. For a set of 100 images containing four different hand configurations, the first 25 eigenvectors are computed to create a 25d feature space. Using the Euclidean distance within this 25d space, new images can then be recognized, i.e., assigned to one of the four learned configurations. A similar method exploits 3d data and was introduced in [Sato *et al.* 2002]. In that work, voxel data of hands is interpreted as 1d vectors and projected

to a lower-dimensional eigenspace in order to recognize the hand configuration. Recently, another related method was presented in [Van Den Bergh *et al.* 2009]. To reduce the dimensionality of the feature space, 2d training examples are expressed with respect to 2d haarlets instead of eigenvectors. The best set of haarlets is found via *linear discriminant analysis* (LDA). While PCA tries to maximize the variance of training examples, LDA is specifically designed to maximize separability. An issue of all approaches of this kind is normalization: recognition only works well when training and testing examples are comparable. For example, depending on the presence of short or long sleeves, skin color segmentation will deliver different bounding boxes. When normalizing hands with respect to such bounding boxes, the input to the system will be inconsistent. In this case, pose recognition will fail, e.g., when training is performed with long sleeves and testing is not.

A different approach to hand pose recognition is to fit templates associated with some hand configuration to testing examples. The method described in [Triesch and Malsburg 2002] performs such fitting by elastic 2d graph matching. For each of ten different hand postures, a *bunch graph* encodes 1) in *nodes* the variety of local image information at certain positions and 2) in *edges* the distances between nodes. For a new 2d gray-value image, the best fitting graph is found after several seconds. A real-time recognition system is presented in [Lockton and Fitzgibbon 2002]. In that work, 46 different hand poses (including American sign language letters) are recognized by deterministic boosting. For testing, a collection of weak classifiers is used to evaluate the fit of a new image to cluster centers of the training set. There is one weak classifier for each pixel labeled as skin or background. A wrist band is used to register the data.

Recently, some fast techniques have been proposed that aim at the implementation of vision-based remote controls. In contrast to sign language recognition, this application does not require the ability to recognize a great variety of different hand configurations. Instead, focus lies on the speed and robustness necessary for commercial products. To this end, several fast classifiers are used in parallel or in cascades. For example, in [Ike *et al.* 2007] three cascaded classifiers [Viola and Jones 2004] are used in parallel on a multi-core processor. There is one classifier for each hand pose to be detected (*pointing hand*, *fist*, and *open hand*). Similarly, the method presented in [Stenger *et al.* 2010] specializes on the detection of a *fist pose*, an *open hand pose*, and a *thumb up*

pose. In that work, a different state-of-the-art classifier is employed for each hand pose to be recognized.

Many more interesting methods with respect to hand pose recognition have been proposed over the last couple of years. For comprehensive overviews, see [Ong and Ranganath 2005] and [Stenger *et al.* 2010].

2.2.3 Model-Based Hand Pose Estimation

In contrast to example-based hand pose recognition, the idea of model-based hand pose estimation is to approximate an observed hand by a parametric model. The optimal parameters of this model have to be derived from the observation. A common procedure is to initialize the model in the first frame of a sequence manually or by some recognition mechanism. Thereafter, the hand is tracked throughout the sequence, with some temporal model encoding a relation between subsequent frames. The advantage of model-based approaches is that their precision is not limited to a set of examples. On the negative side, hand models have many parameters which have to be estimated. Further, the temporal procedure involves the danger of ”loosing track”, meaning when parameter estimation goes wrong at some point, subsequent frames will seldom be tracked successfully.

An early representative of model-based methods is called *DigitEyes* and was presented in [Rehg and Kanade 1994]. The hand is modeled as a collection of 16 rigid bodies: three phalanges for each of the five fingers, and a palm. A comparison between the modeled phalanges and the image observation is performed as follows. Every phalanx is associated with a cylinder representing skin. The axis of the cylinder is projected into the image. Then, the two enclosing contour lines are found considering the image gradient perpendicular to the projected axis. Based on this comparison, an error function is defined. The best hand configuration is then found by minimizing the error with the Levenburg-Marquardt algorithm [Dennis Jr and Schnabel 1996]. In [Stenger *et al.* 2001], the skin of a skeleton-based model is approximated by truncated quadrics, which allows for more elegant projection techniques. Pose estimation is achieved with an *unscented Kalman filter*, but only 7 of the 27 DOFs are considered. Although skeleton-based models like the one in [Rehg and Kanade 1994] and [Stenger *et al.* 2001] are most commonly used, different types of models have been explored. For example in [Heap and Hogg 2002], a 3d point model of a hand is fitted to 2d data.

A main issue of model-based hand tracking approaches are the many DOFs that have to be estimated. Common ideas to deal with the high dimensionality are 1) to exploit synergies of human hand configurations like for example done in [Wu *et al.* 2001] and 2) to apply efficient sampling strategies to better approximate the state space. With respect to the latter point, *particle filters* [Isard and Blake 1998] have been used to focus on the most relevant portion of the space. In [MacCormick and Isard 2000], an extension has been presented that additionally partitions the state space exploiting the hierarchical nature of structures like hands. These sampling strategies are applied in [Isard and Blake 1998, MacCormick and Isard 2000] to track in real-time the fist, the index finger, and the thumb of a hand using a shape model based on B-splines. In [Bray *et al.* 2004], a fully articulated hand is tracked in 3d data. To be able to process 12.5 frames per second, the authors combine the optimization method *stochastic meta descent* with a particle filter to form *smart particles*.

A different way to deal with the high dimensionality problem is presented in [Sudderth *et al.* 2004b]. In that work, the DOFs are actually increased by introducing a highly redundant representation: each of the 16 hand segment lives in its own 6d space (translation and rotation). However, in this representation hand poses can be estimated by efficient inference algorithms. Sudderth et al. apply *nonparametric belief propagation* (NBP) [Sudderth *et al.* 2003], i.e belief propagation combined with particle filter methods to represent the local 6d spaces. An extension of [Sudderth *et al.* 2004b] is [Sudderth *et al.* 2004a]. That work proposed a method to deal with self-occlusions in the context of NBP.

Given the advantages and disadvantages of both example-based and model-based hand pose estimation, some researches have suggested hybrid systems. For example in [Athitsos and Sclaroff 2003, Athitsos and Sclaroff 2004] an articulated hand model is employed, but the parameters of this model are estimated based on example-based recognition. In a preprocessing step, a database is created that contains a uniform sampling of all possible views of the hand shapes to be recognized. Given a new image, the most similar database views are retrieved, and the parameters of these views determine the parameters for the current frame. In [Stenger *et al.* 2003, Stenger *et al.* 2006] a hierarchical detection method is used for *tree-based filtering*. The hierarchical scheme accelerates detection and helps to focus on the relevant portion of the state space. In a probabilistic framework, the probability of a given hand configuration is

defined by the matching scores of the image with respect to the examples in the detection tree.

A recent summary of state-of-the-art methods for hand pose estimation can be found in [Erol *et al.* 2007].

2.3 Robotic Grasping

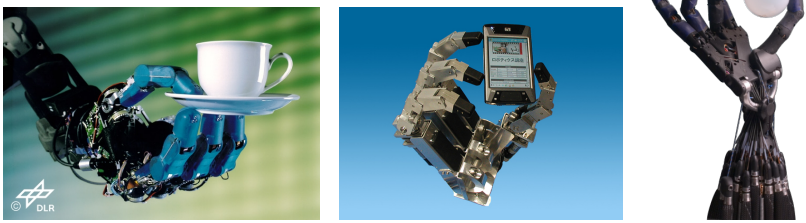
Robots have become an inherent part of our world, and a key feature of many robots is the ability to interact with their environment via robotic arms. Such arms are crucial for tasks at assembly lines, for surgery assistance, and in scenarios where manipulation tasks cannot be performed by humans (e.g., robots explore the deep sea or diffuse bombs). Classical robotic arms are serial links with a jaw-gripper as an end effector. According to [Bicchi and Kumar 2000], these devices have three major draw-backs. 1) They are only suited to grasp planar surfaces. 2) Small reorientations of a grasped object can only be realized via the whole robotic arm. 3) Structural properties of the grasped object cannot be inferred. To solve these issues, researchers have developed multi-fingered robotic hands, imitating the anatomy of human hands. Examples of such hands include the Utah/MIT dextrous hand [Jacobsen *et al.* 2002], the PUMA/RAL hand [Kim *et al.* 1987], the Robonaut hand [Lovchik and Diftler 1999], the Barret hand [Townsend 2000], the DLR-Hand II [Butterfass *et al.* 2006], the Gifu hand II [Kawasaki *et al.* 1999], the Shadow hand [Walker 2004], and the NAIST-Hand [Ishida *et al.* 2005]. See Figure 2.3 for examples.

In the following we discuss two different concepts to determine grasps for the control of such humanoid robotic hands: 1) autonomous grasp synthesis of a robot and 2) the control by a human hand.

2.3.1 Autonomous Grasp Synthesis

For a long time there have been significant efforts to let robots grasp objects autonomously. Two major approaches to reach this goal have evolved in literature. The analytical approach and the biologically motivated approach.

The analytical approach is the classic one. The basic idea is to define grasps by a set of contact points through which the hand applies forces on the object. Grasping, i.e., choosing appropriate contact points and forces, can then be



(a) DLR-Hand II *(b)* NAIST hand *(c)* Shadow hand

***Figure 2.3**: Some examples of modern robotic hands: the DLR-Hand II, the NAIST hand, and the Shadow hand.*

expressed as an optimization problem that takes into account some measures for the quality of a grasp. According to [Shimoga 1996], the commonly used quality measures are essentially dexterity (*"How should grasping fingers be configured?"*), equilibrium (*"How hard to squeeze the grasped object?"*), stability (*"How to remain unaffected by external disturbances?"*), and dynamic behavior (*"How soft a grasp should be for a given task?"*). Desirable properties of resulting grasps are *force closure* (the grasp can resist forces acting on the object) and the more restrictive *form closure* (force closure that holds for arbitrarily slippery objects). A review of methods for force-closure analysis is provided in [Mishra and Silver 1989].

Analytical grasp synthesis has recently been criticized for at least three reasons. 1) It is computationally expensive [Ritter *et al.* 2009], though great speed-ups have been achieved in [Borst *et al.* 2005] by formulating the task as a semi-definite optimization problem and by efficient sampling strategies. Computation time is an issue especially when extending contact points to contact areas to account for the deformable finger tips. 2) Detailed knowledge about the geometry of the manipulated objects is required. This information is not always available. 3) The ideal contact points and forces are often not realizable by real robotic hands [Roethling 2007].

In contrast to the analytical approach to grasp synthesis, the biologically motived approach explicitly considers the structure and proportions of the respective grasping device. An important tool for biological approaches is the open simulation engine *GraspIt!* [Miller and Allen 2004]. Given a full 3d model of

an object and a grasping pose for some hand model, for instance, the quality of the grasping can be evaluated based on pre-computed grasp primitives [Miller *et al.* 2003].

For biological approaches, research regarding the reduced dimensionality of hand poses during object manipulation is highly relevant. For example, [Ciocarlie *et al.* 2007] build on the insight from [Santello *et al.* 1998] that the first two principal components cover more than 80% of the variance in hand posture. They optimize in an 8-dimensional space with simulated annealing to assure that key points on the hand model are close to the object. 6 DOFs correspond to translation/rotation of the hand, the remaining 2 DOFs refer to the principal axes identified in [Santello *et al.* 1998]. Using GraspIt!, the authors evaluate their method with respect to models of a simple gripper, the Barret hand, the DLR Hand, the Robonaut hand, and a human hand model. Because the synergy information from [Santello *et al.* 1998] applies only to human hands, the authors try to transfer those principal axes to the other models manually.

Other non-analytical methods focus on grasping in the absence of a 3d model of the object. In [Saxena *et al.* 2008], the 3d grasp position is estimated from two images where grasp locations are identified. For this, a 2d grasp point detector is trained on synthetic images. The authors of [Morales *et al.* 2006] obtain a top view of the object with a camera, extract curvature and location, and then look for graspable regions based on curvature. An omni-directional camera mounted to the robotic hand is used for the extraction of graspable regions in [Yoshikawa *et al.* 2008].

For more details with respect to autonomous robotic grasping the reader is referred to the surveys in [Shimoga 1996, Bicchi and Kumar 2000, Okamura *et al.* 2000]. Where [Roethling 2007] offers a comparison of analytical and biologically motivated approaches. We did not touch here on the matter of tactile feedback retrieved by robotic hands. A summary of methods concerned with tactile feedback can be found in [Argall and Billard 2010].

2.3.2 Control by a Human Hand

One approach to drive anthropomorphic robotic hands is particularly related to our work: controlling a robotic hand via an observed human hand. Observations of the human hand are typically obtained by a data glove [Fischer *et al.*

1998, Voyles and Khosla 2001, Turner *et al.* 2000] or a vision system [Hueser and Baier 2006]. Such observations can then be used in two different ways. Either 1) directly to control the robotic hand instantly by *motion transfer* or 2) to derive policies in the context of programming by demonstration.

Motion Transfer

During motion transfer, human hand gestures are transferred to a robotic grasping device instantly. The motivation for this technique is that humans can benefit from their experience in object manipulation without using their physical hands/arms. This makes sense when handling hazardous items or when acting in environments that are hostile for humans (e.g., the deep sea or space) via *tele-manipulation*.

A major issue in tele-manipulation is that the anatomy of human hands and anthropomorphic robot hands is still quite different. For example, most robotic hands have three to four fingers. Hence a mapping function is required to transfer hand postures. [Fischer *et al.* 1998] provides a calibration and mapping method for the four-fingered DLR hand. To obtain an accurate mapping from joint angles delivered by a data glove to the positions of the human finger tips, a marker-based vision system is first employed. Thereafter, a second mapping translates the workspace of the human into the workspace of the robot. In [Hu *et al.* 2004] the setup is very similar (the DLR hand and a hand model are calibrated by vision), but a different human-to-robot mapping prevents overlapping workspaces of the fingertips.

In the same context, the two-fingered, planar grasping device *dexter* is controlled by the index finger and the thumb of a human operator wearing a data glove in [Griffin *et al.* 2000]. The operator manipulates a virtual object and the robot acts on a scaled version of this virtual object. To calibrate the system, a new user has to move the index finger and the thumb while preserving rolling contact at the tips. Given this closed kinematic chain and fixed anatomical relationships an *angular mapping* is derived. Human-to-robot mapping then consists of two step. 1) *Point-to-point mapping* maps the 3d positions of the human fingertips to the fingertips of the robot within its planar working space. 2) *Virtual object-based mapping* then transfers the human manipulation performed on the virtual object to the coordinate frame of the scaled version of the virtual object. The setting in [Turner *et al.* 2000] is the same as in [Griffin

et al. 2000], and the same calibration and mapping techniques are used. The difference is that in [Turner *et al.* 2000] the dexter hand manipulates real objects instead of virtual surrogates. The user still acts on a virtual object, but feedback sensed by the robotic hand is transmitted back to the user via the haptic data glove. Surprisingly, the authors state that the addition of force feedback to the user did not improve the speed of task execution.

Programming by Demonstration

In programming by demonstration (PbD), human performance is not necessarily mapped instantly to a robotic device. Instead, a systems *learns* from demonstration. For example, in [Hueser and Baier 2006] a service robot is taught grasping skills by a human instructor who demonstrates a grasping action in front of a camera. Several objects are placed on a table. These objects are trained for recognition in advance and offline. For demonstration the teacher says "start", performs some grasping skill with the hand, and then says "stop". The robot observes the performed action several times to collect sufficient data. Thereafter, it attempts to grasp the objects on the table, imitating the human performance.

The general goal of PbD is to derive policies that encode the relationship between the state of the world and possible actions [Argall *et al.* 2009]. Special techniques have been developed to derive the set of possible actions for a robotic hand from human demonstration. In [Jo *et al.* 1998], manipulative hand gestures are visually recognized with the aid of a state transition diagram. The diagram encapsulates task knowledge, e.g., an object has to be grasped before it can be moved. Visual feature extraction is based on thresholding the hue value, so that the person has to wear special gloves. Gestures are only simulated, without a real object being involved. In [Ekvall and Kragíc 2005], grasps are recognized with respect to the taxonomy defined in [Cutkosky and Wright 1986]. Hidden Markov models are used to model the hand posture sequence during grasping. As a result it is possible to recognize a sequence even before it is completed. Real objects are handled, a data glove provides the hand pose. In [Kjellström *et al.* 2008], the correlation between a manipulating hand and the manipulated object is exploited for both better hand pose and object recognition. The action-object correlation over time is modeled using conditional random fields. An example-based method for hand pose recognition in the context of hand-object interaction is presented in [Romero *et al.* 2010]. Observed hand poses are compared to a large database containing hands manipulating objects.

In the field of PbD, a variety of methods has been suggested to learn and encode dependencies between the perceived state of the world and suitable actions. To give just one example, in [Chella *et al.* 2004] the authors model several cognitive hierarchies. In this way they can teach a robotic hand the *rock-paper-scissors* game. A comprehensive review of PbD techniques can be found in [Krüger *et al.* 2007].

PbD methods are strongly interwoven with works from the research areas of *scene interpretation* and *action recognition*, so to complete this section we touch on some of these works. In [Mann *et al.* 1996], video sequences are processed with the goal of a computational perception of scene dynamics. The sequences contain some object (e.g., a coke can) and a human hand manipulating the object. The 2d position and rotation of hand and object are tracked using optical flow and 2d template matching. For each frame of the sequence, multiple hypotheses are then evaluated to explain the scene. To this end, possible relationships between hand and object (like an attachment or a contact) are considered.

In action recognition, a common task is to divide observed sequences into action units. In [Rao *et al.* 2002], sequences containing hand-object interaction are analyzed. The hand is tracked in 2d by skin color detection and with the mean-shift technique [Comaniciu *et al.* 2000]. Action units are then identified with respect to the velocity changes of the hand. For task analysis, the authors of [Sato *et al.* 2002] consider not only the speed of a hand, but also its speed in relation to the velocity of a manipulated object. For this, hand and object are tracked in 3d data delivered by a range sensor. For a general review of methods for action recognition, please refer to [Krüger *et al.* 2007].

3

Materials and Methods

3.1 Structured-Light System

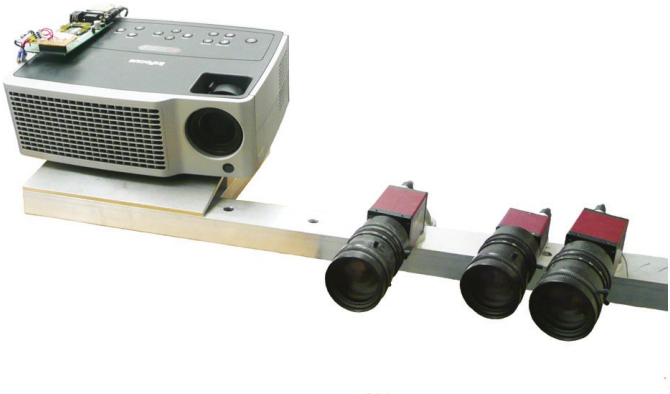


***Figure 3.1**: The structured-light setup we use to capture hand-object interaction data. The system consists of two monochrome cameras, one color camera, one DLP projector, and one micro-controller.*

In this work we make use of range data. One well-known approach to obtain such data is the passive stereo method: given the images of two calibrated cameras and correspondence pairs between these two images, depth information can be inferred by triangulation. However, the depth accuracy of real-time systems of this type is very limited, in particular for homogeneous regions of the scene.

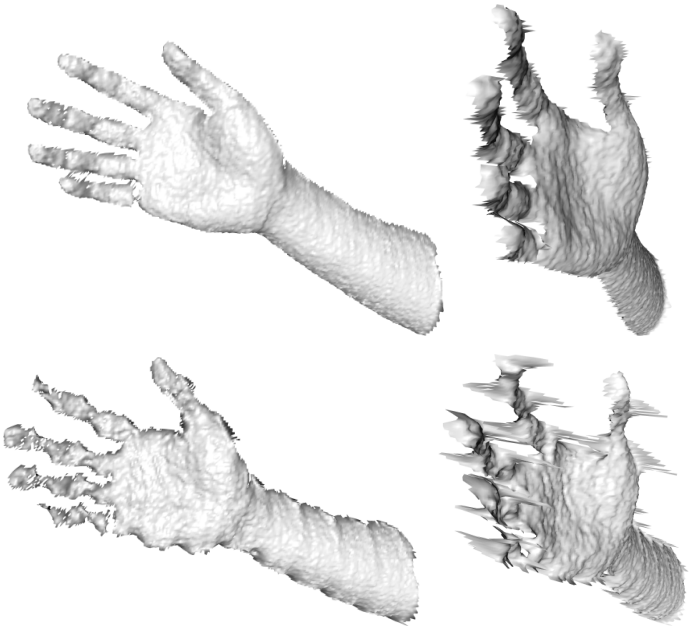


***Figure 3.2**: Depth data of a hand delivered by the structured-light system. The top row illustrates the case when the hand is located ≈ 1.5 meters away from the projector and does not move (left: front view, right: side view). The bottom row shows artifacts caused by fast lateral movement of the fingers.*

In contrast to passive stereo systems, structured-light systems replace one of the two cameras by an active light source, projecting some kind of patterns. Those patterns can be detected with the remaining camera to again perform triangulation. A popular method to encode information in the patterns is the phase-shift method: for a high accuracy of the delivered range data, three or more phase-shifted cosine patterns are projected into the scene in fast succession. The main problem with this technique is its inability to handle dynamic scenes.

Our system is a combination of the two approaches: the temporal phase-shift technique is combined with stereo-based decoding of the patterns. The method was developed in [Weise 2009] and a picture of the hardware setup can be seen in Figure 3.1. The components are two high-speed monochrome cameras (*Al-*

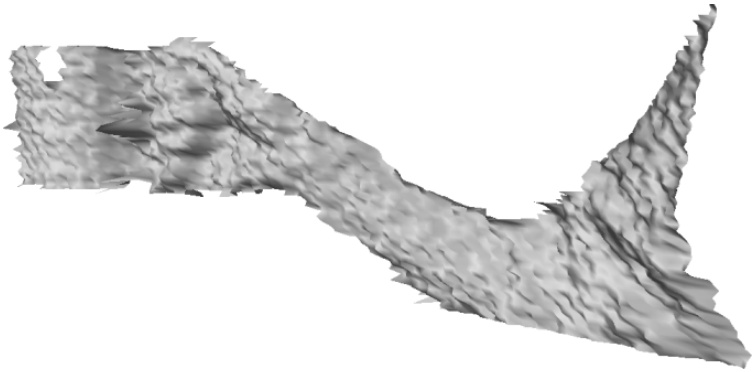


***Figure 3.3**: Range data captured with a Swissranger SR4000 time-of-flight camera. The image resolution is* 176×144*, and depth precision lies within a few centimeters. We included the arm for better interpretability of the data.*

lied Vision Tec PIKE F-032B, 640×480), one color camera (*Allied Vision Tec STINGRAY F-046C*, 780×580), a standard DLP projector (*InFocus IM38*, 1024×768), and a micro-controller for synchronization (*Microchip PICDEM HPC Explorer Board*). To produce the phase-shifted signal, the color wheel of the projector has been modified. The scanning system captures dense high-quality depth maps at 30 fps and a resolution of 640×480 pixels, achieving sub-millimeter accuracy for a working volume of $60^3 cm^3$.

The average quality of our data is illustrated in the top row of Figure 3.2. The hand in this example is located ≈ 1.5 meters in front of the cameras. Quality improves as the hand approaches the setup. Also, we usually smooth the data with a Gaussian kernel. Because the backside of the scene is missing, such 3d data is also referred to as 2.5d data. One issue of the system is demonstrated by the bottom row of Figure 3.2. Fast lateral movement of fine structures causes artifacts. In this example, the edges of the fingers are strongly corrupted.

A competing technique to capture 2.5d data is to use a *time-of-flight* camera. With this technique, impulses of light are sent into the observed scene, and a camera measures for each pixel the time until the impulses return due to reflection. For an impression of such data, we captured a hand holding a cup with a state-of-the-art Swissranger SR4000 time-of-flight camera (see Figure 3.3). The image resolution is only 176×144, and depth precision lies in the

range of a few centimeters. Hence, the quality of such data is inferior to ours and not sufficient for our purposes.

Most of our data has been published as the Hand-Object Interaction (HOI) data-set 2010. The data-set features 10 different objects and 9 different subjects. Range and color information is available at http://www.vision.ee.ethz.ch/~hhamer/hand_object_interaction_10/ . We hope that this data encourages further research with respect to hand-object interaction.

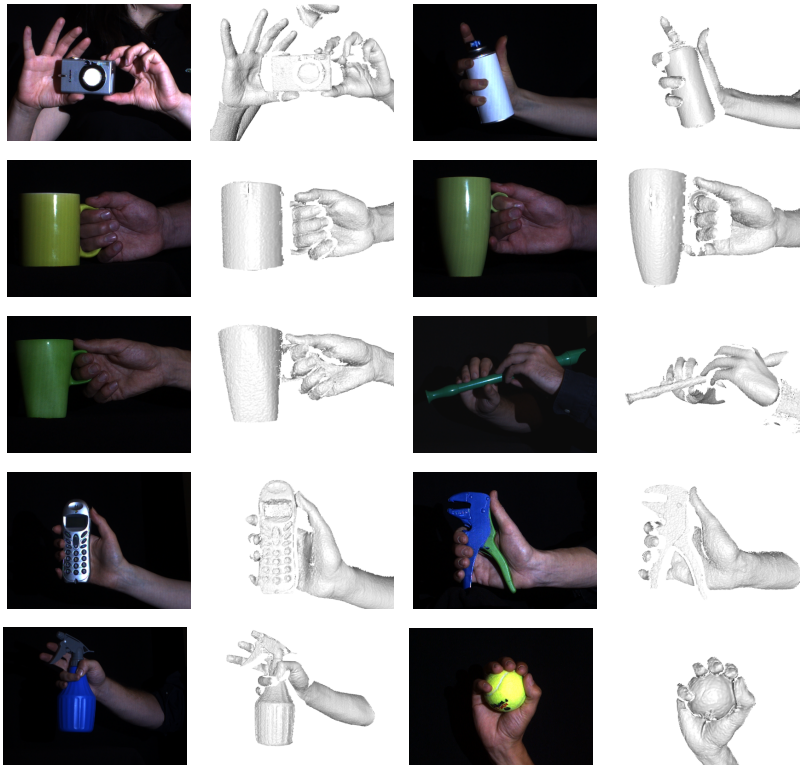


Figure 3.4: *Some frames from the HOI data-set 2010, featuring 10 different objects and 9 different subjects performing object manipulation. Both range and color information is available.*

3.2 Skin Color Segmentation

The image from the color camera of the structured-light setup is mapped to the range data, assigning an RGB color value to each depth point. To separate the depth data of a manipulating hand from that of the manipulated object we apply skin color detection to each of the depth points. The detection mechanism is based on the method proposed in [Jones and Rehg 2002] and will be discussed next.

Given a large amount of images containing skin-colored pixels and a manual labeling, two *Gaussian mixture models* (GMMs) were learned by *expectation maximization* (EM) [Jones and Rehg 2002]. These two models express the probability of some color c to occur in a skin region $p(c|skin)$ or non-skin region $p(c|\neg skin)$. Using Bayes' rule and ignoring the prior $p(skin)$, we can compute

$$p(skin|c) = \frac{p(c|skin)}{p(c|skin) + p(c|\neg skin)} \tag{3.1}$$

Depending on wether $p(skin|c)$ is greater or smaller than a threshold θ_{skin} that lies between 0 and 1, the respective depth point is classified as skin or non-skin. For better robustness to changes in illumination, color c is defined in the rg color space, i.e.,

$$r = \frac{R}{R+G+B} \tag{3.2}$$

$$g = \frac{G}{R+G+B} \tag{3.3}$$

where R, G and B define c in RGB space. Since there is only a limited amount of possible combinations of r and g, $p(skin|c)$ can be stored in a lookup table for fast access. After labeling each depth point as skin or non-skin, a binary median filter is applied across the labels for a more robust final segmentation.

In practice, colors between red and brown are typically detected as skin. For a better intuition of the detector, we computed *receiver operating characteristic* (ROC) curves for different homogeneously colored objects, namely a red candle, a green candle, a blue candle, and a brownish pot (see Figure 3.5). A

lower threshold θ_{skin} causes a higher *true positive rate*, i.e., more pixels of the hand are detected as skin. However, a lower threshold also causes more pixels of the objects to be classified as skin (*false positives*). This is a problem especially with respect to the brown and the red object. Figure 3.5 also shows the segmentation result for a scene in which a hand holds the red candle.

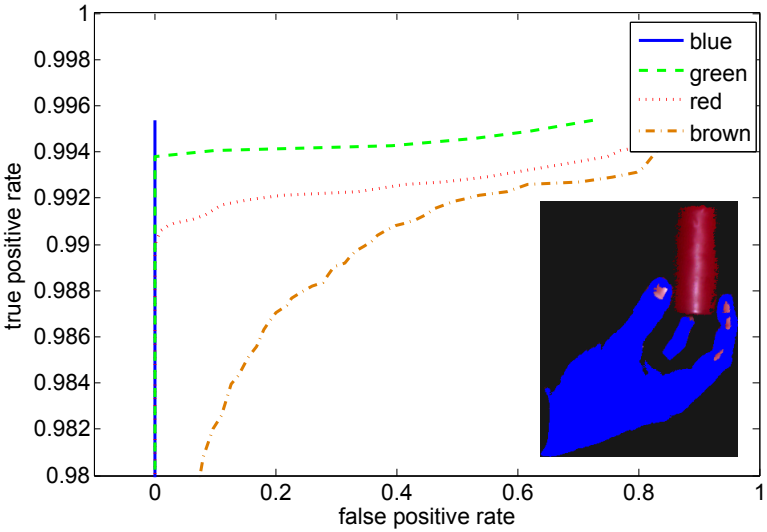


***Figure 3.5**: ROC curves for objects with different colors: a red candle, a green candle, a blue candle, and a brown pot. The y-axis shows the true positive rate (percentage of skin pixels classified as skin), and the x-axis provides the false positive rate (percentage of object pixels classified as skin). To obtain positive and negative samples, the hand and the individual objects were captured in isolation. The threshold for skin detection is decreased from bottom-left to top-right. The smaller the threshold θ_{skin}, the more pixels of the hand and the object are detected as skin (higher true/false positive rate). Some (almost black) skin pixels are never classified as skin because their $p(skin|c) = 0$. A segmentation result for the red candle with favor of a low false-positive rate is shown in the bottom-right of the figure: pixels detected as skin are drawn in blue.*

3.3 Iterative Closest Point Method

The *iterative closest point* (ICP) method introduced in [Chen and Medioni 1992, Besl and McKay 1992] is a popular technique for the alignment of two 3d surfaces. We use it at several occasions for the alignment of observed hands and objects, respectively. The iterative algorithm assumes an initial coarse registration of the surfaces and performs two fundamental steps in each iteration.

1. In the first step, closest-point correspondence pairs between the vertices p_i of the first surface and the vertices q_i of the second surface are found. The closest-point search has a complexity of O(log n).

2. Given the correspondence pairs, the rigid transformation that minimizes an error metric with respect to the corresponding vertices is estimated and applied.

The algorithm terminates after a defined number of iterations or as soon as some convergence criterion is reached. As an error metric, [Besl and McKay 1992] suggested minimizing

$$\sum_{i=1}^{N} ||w_i(T_{pq}^{k+1}p_i - q_i')||^2, \tag{3.4}$$

where w_i is a correspondence specific weighting term, q_i' is the vertex on the second surface that is closest to p_i, and T_{pq}^{k+1} is the transformation minimizing the error metric after $k+1$ iterations using the closed-form solution by [Horn 1987]. For better movement tangential to the surface, in [Chen and Medioni 1992] the term

$$\sum_{i=1}^{N} ||w_i n(q'i)(T_{pq}^{k+1}p_i - q_i')||^2, \tag{3.5}$$

is minimized. $n(q_i')$ is the normal at q_i'. No closed-form solution exists for this, but the expression can be linearized. A number of efficient variants of the ICP methods have been suggested. A comparison is provided in [Rusinkiewicz and Levoy 2001]. We use the method presented in [Jaeggli and Van Gool 2003].

Correspondence pairs are found along the line of sight of the projector with a constant complexity of O(1), and the error metric in Equation 3.5 is minimized. Also, we sample correspondence pairs uniformly for better performance. The correspondence weight w_i is used to discard invalid correspondence pairs: if the distance between two points is too large, or if their normals are not consistent w_i is changed from 1 to 0.

3.4 In-hand Scanning

Some of the methods we propose in this work are based on 3d models of the manipulated objects. We acquire such models by *in-hand scanning* as described in [Weise *et al.* 2008]. During in-hand scanning, a human demonstrator rotates an object by hand in front of a system like our structured-light setup (see Section 3.1). The object is reconstructed in real-time and the user can control the process online. This allows the user to fill in holes in the preliminary model in a selective and very intuitive way.

In [Weise *et al.* 2008], skin-colored depth points are detected (see Section 3.2) and removed in each frame. The subsequent range scans of the object are then registered with ICP (see Section 3.3). To initialize the ICP algorithm, texture information is exploited: in two subsequent frames, interest points are detected using Harris corners [Harris and Stephens 1988] and their SURF descriptors [Bay *et al.* 2008] are computed. Correspondences between these features in the first and the second frame are used to derive an initial coarse registration. By aligning the individual range scans, inconsistent depth points can be removed, and one complete model is obtained.

In practice there are some limitations to the in-hand scanner. 1) When an object is skin-colored, the demonstrator has to wear black gloves to allow for the separation of hand and object. 2) Highly specular surfaces cannot be reconstructed, because the structured-light setup fails to provide depth data. 3) If an object is symmetric and has uniform color, registration fails and obtaining a complete model is difficult if not impossible.

For our purposes, partial object models often suffice. In Figure 3.6, examples of such partial models are shown.

(a) Camera *(b)* Clamshell phone *(c)* Cup

***Figure 3.6**: Partial meshes of a camera with a zoom, a clamshell phone, and a cup.*

3.5 Synthetic Hand

We use two different hand models in the course of this work. The first one is based on a 3d hand scan which was bound to a forward-kinematics skeleton in Maya. We call this model the *synthetic hand*. The second model (called *tracking model*) consists of one geometric primitive per hand segment. Since the second model is used for hand tracking in Chapter 4, it will be discussed in detail there. We now describe the anatomy of human hands in general and then explain how this anatomy is approximated by the synthetic hand.

3.5.1 Hand Anatomy

A human hand is shown in Figure 3.7. The main components are the wrist, the palm, the thumb (*1st*), and the four fingers: index (*2nd*), middle (*3rd*), ring (*4th*), pinky/little finger (*5th*). There are 27 bones in total. Eight of these are located within the wrist (*carpals*), four make up the palm, the others belong to the fingers and the thumb. Each finger has three *phalanges*: a *proximal phalanx* (PP), an *intermediary phalanx* (IP), and a *distal phalanx* (DP).

[1] University of Washington, http://uwmsk.org/RadAnat

***Figure 3.7**: An X-ray of a human hand. The main components are the wrist, the palm, the thumb, and the four fingers: index, middle, ring, and pinky/little finger. Image courtesy of M. L. Richardson*[1].

The palm is formed by the four *meta carpals* (MCs) of the fingers (2nd-5th). It is actually not a completely rigid body but allows for some deformation. In particular, the 4th and the 5th MC contribute to this deformation [McDonald *et al.* 2001].

Articulation of the fingers is possible due to *revolute joints*. *Metacarpal phalangeal joints* (MCP joints) connect MCs to PPs, *proximal interphalangeal joints* (PIP joints) connect PPs to IPs, and *distal interphalangeal joints* (DIP joints) connect IPs to DPs. There are different types of joints. PIP and DIP joints are *hinge joints* and permit only one DOF (*flexion, extension*). MCP joints are of the *saddle/condyloid* kind. They also allow for spreading of the finger (*adduction, abduction*), so they have 2 mayor DOFs. Additionally, [Cailliet and Davis 1972] show that spreading automatically causes a few degrees of twist around the bone axis within the MCP joints.

The anatomy of the thumb is more complex. The carpal supporting the thumb is called *trapezium*. The joint connecting the trapezium to the 1st MC is the

trapeziometacarpal joint (TM joint). Like in the case of the four fingers, the 1st MCP joint connects the MC to the PP, but the 1st IP joint connects the PP directly to the DP - there is no 1st IP. The motion range of the 1st MCP joint is rather small compared to that of the other four. The great variability of the thumb arises from the TM joint, which has been studied extensively [Cooney *et al.* 1981, Buchholz and Armstrong 1992, Hollister *et al.* 1992, McDonald *et al.* 2001]. According to [Hollister *et al.* 1992], this joint has two non-orthogonal and non-intersecting rotation axes. Additionally, [Buchholz and Armstrong 1992, McDonald *et al.* 2001] argue that there is a third DOF within the TM joint: a minor *twist*, possible due to the loose connection of trapezium and 1st MC.

Besides limitations on hand motion resulting from the individual joints there are also constraining dependencies between different joints, both within the same finger (*intra-finger constraints*) as well as between fingers (*inter-finger constraints*). Such causalities arise from the tendons in the hand that cause hand motion. One intra-finger dependency commonly assumed is the "2/3 rule": the flex angle of the DIP joint equals roughly $2/3$ of the angle of the corresponding PIP joint [Rijpkema and Girard 1991]. An example of an inter-finger dependency can be observed when the little finger of a hand flexes: the ring finger will usually also bend a little. However, such dependencies between joints are no longer guaranteed when the hand is in contact with an object.

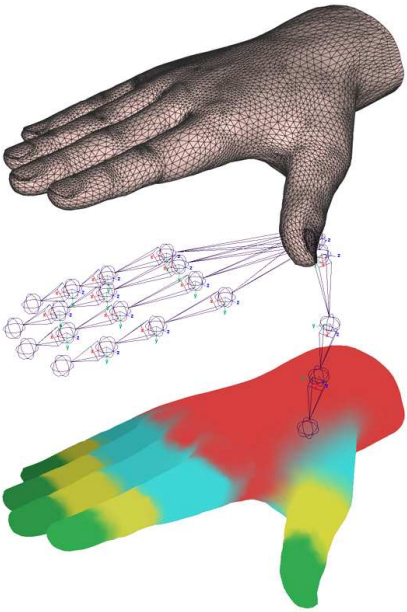


***Figure 3.8**: Synthetic hand: a 3d scan of a hand (top) bound to a forward-kinematics skeleton (middle) in Maya. The weights (bottom) define the binding of the skin to the bones. Image taken from [Bray 2004].*

3.5.2 Model of the Synthetic Hand

The synthetic hand described in [Bray 2004] is illustrated in Figure 3.8. Essentially, this is a 3d scan of a hand (top of the figure) bound to a forward-kinematics skeleton (middle of the figure) in Maya. The weights shown at the bottom of the figure define the binding of the skin to the phalanges.

The skeleton of this model contains 16 joints. In total there are 30 DOFs. The wrist (the *root joint*) has 6 DOFs (3 for translation and 3 for rotation). The MCP joints of the fingers permit 3 DOFs to account for flexion/extension, adduction/abduction, and also for a small amount of possible twist. PIP and DIP joints of the fingers are modeled with 1 DOF. Regarding the thumb, the TM joint is approximated with only 2 DOFs because the potential twist is considered to be negligible. The 1st MCP joint as well as the 1st IP joint both have one DOF. To reduce the dimensionality from 30 to 26, the DIP joints of the

fingers are defined by the respective PIP joints in [Bray 2004], exploiting the "2/3 rule".

An important work with respect to the dimensions of the bones within the synthetic hand is [Buchholz *et al.* 1992]. In their anthropometric studies the authors identified various dependencies between the size of individual hand components. For example, it is stated that there is a relationship between the total length of a hand and the length of the different phalanges. These ratios where respected when designing the kinematic structure.

The advantage of the synthetic hand lies in its realistic looks. In [Bray 2004], this model was used for tracking a hand in 3d data. This might be problematic when the tracked hand is not the one that was originally scanned: adapting the polygonal mesh to a hand with different anatomical dimensions is not trivial. Another problem we encountered is that the control of the thumb in Maya is rather unintuitive: when trying to reach a certain thumb pose, it is not always clear how to set joint angles appropriately.

While we do not track with this model, we use it for the visualization of estimated hand poses and for the generation of artificial depth data with ground truth. To generate artificial data, we employ a C++ version of the model that can be rendered using OpenGL. OpenGL commands usually draw 3d objects to a frame buffer which is then displayed on the screen. However, it is also possible to render into a depth texture on the GPU. By mimicking the external and internal camera parameters of the structured-light setup described in Section 3.1, we obtain range data of the synthetic hand. Intuitively, the synthetic hand is observed by our camera setup. For an example of such artificial data see Figure 3.9.

3.6 Belief Propagation

In Chapter 4 we estimate hand poses by belief propagation (BP), as suggested in [Sudderth *et al.* 2004b, Sudderth *et al.* 2004a]. While the application of BP to hand tracking will be addressed in that Chapter, we now provide a general introduction to the algorithm first presented in [Pearl 1982].

BP is a message passing algorithm used for inference on graphical models, i.e., on probabilistic models for which a graph denotes the conditional independence structure between random variables. Many interesting problems can

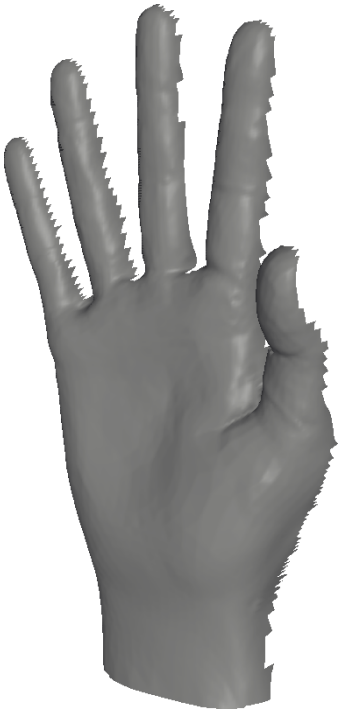


***Figure 3.9**: Simulated range data produced by rendering the synthetic hand on the GPU. The used external and internal camera parameters correspond to the parameters of the structured-light system.*

be expressed in the form of such a graphical model, e.g., medical diagnosis on the basis of a number of given symptoms, stereo reconstruction in computer vision, and error-correcting codes in signal processing.

There are different types of graphical models, for example *Bayesian networks*, *Markov random fields* (MRFs), and *factor graphs*. However, it is shown in [Yedidia *et al.* 2003] that conversion between each of these models is possible, thus it is sufficient to formulate BP with respect to one model type. We follow here [Yedidia *et al.* 2003] and consider the case of MRFs.

A MRF has a set of nodes, each of which corresponds to a variable or a group of variables, as well as a set of edges each of which connects a pair of nodes. Nodes can be either *observed* or *unobserved*. In the case of an observed node,

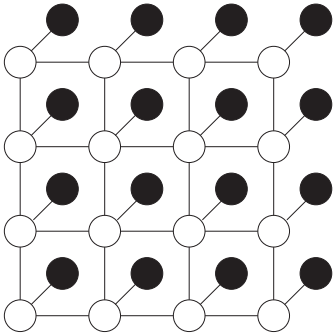


***Figure 3.10**: A square lattice MRF. The filled-in circles represent observed nodes, while empty circles stand for hidden nodes. The hidden nodes are pairwise connected in a grid.*

the associated random variable is set to a specific observed value. In contrast, no observation is available for unobserved or *hidden* nodes. An example of a MRF is given in Figure 3.10.

BP calculates the marginal distribution for each unobserved node x_s, conditional on any observed nodes y_s. In computer vision, the index s often represents pixel positions, for example when inferring the depth of each pixel for stereo-reconstruction. It is assumed that there is a statistical dependency between x_s and y_s at each position s, expressed by a *data* term $\phi_s(x_s, y_s)$. In addition, a *compatibility* term $\psi_{st}(x_s, x_t)$ defines the dependency between neighboring nodes. Considering the observed nodes to be fixed we can write $\phi_s(x_s)$ as a short-hand for $\phi_s(x_s, y_s)$. The joint probability distribution for the unknown variables $\{x\}$ is then given by

$$p(\{x\}) = \frac{1}{Z} \prod_{st} \psi_{st}(x_s, x_t) \prod_{s} \phi_s(x_s), \tag{3.6}$$

where Z is a normalizing constant.

BP estimates marginals with the aid of messages. A message from node t to neighboring node s can intuitively be understood as the opinion of node t about

what state node s should be in. According to the BP algorithm, the belief at node s is proportional to the product of the local observation and all messages coming in to node s. The distribution of x_s is hence defined as

$$p(x_s) = \phi(x_s) \prod_{t \in \mathcal{N}(s)} m_{t \to s}(x_s) \; . \tag{3.7}$$

Messages are determined by the *message update rule*. According to this rule, a message from node s to node t assigns a probability to each realization x_t^i of x_t as follows:

$$m_{s \to t}(x_t^i) = \sum_{x_s} \phi(x_s) \cdot \psi(x_s, x_t^i) \prod_{t^* \in \mathcal{N}(s) \setminus t} m_{t^* \to s}(x_s) \; . \tag{3.8}$$

$\mathcal{N}(s) \setminus t$ contains all neighboring nodes of s except t. The product combines the incoming messages, the sum marginalizes over x_s.

The basic procedure of the algorithm is that a node s sends a message to neighbor t as soon as the messages from all neighbors (except t) have arrived. In this way, the available information is distributed throughout the complete graph. The algorithm terminates when some convergence criterion with respect to the estimated marginals has been reached.

The example in Figure 3.10 is actually not trivial in terms of inference since the MRF contains *loops*. In the case of loops the BP algorithm is not guaranteed to converge, although good results have been reported in practice. However, in the case of tree structured graphs without loops the algorithm converges quickly to the exact marginals: messages are ideally first sent from the leaves towards the root, and then back from the root to all the leaves. For a more detailed description of BP and possible generalizations of the algorithm, please refer to [Yedidia *et al.* 2003].

4

Occlusion Robust Hand Tracking

In this chapter we present a method to track a hand manipulating an object. Special focus lies on robustness to self-occlusions and occlusions caused by the object. Experiments are presented for synthetic data with ground truth and for real data of people manipulating objects. The work in this chapter was also presented at the *IEEE International Conference on Computer Vision 2009* [Hamer *et al.* 2009].

4.1 Overview

Visual hand tracking has several important applications, including sign language recognition, intuitive human-computer interaction, human behavior and emotion analysis, safety and process integrity control on the work-floor, rehabilitation, and motion capture. Not surprisingly, much research has already gone into computer algorithms for hand tracking. Yet, the majority of contributions have only considered free hands, whereas in many applications the hands will actually be manipulating objects. In this chapter, we present a system which can track the articulated 3d pose of a hand, while the hand interacts with an object (such as depicted in Figure 4.1).

The presence of objects has a significant impact on the complexity and generality of the task. First, the manipulated objects will frequently occlude parts of the hand, and hand poses occurring during the process of grabbing or holding will aggravate the problem of self-occlusion (e.g., in Figure 4.1 parts of the fingers are partially or even fully occluded). Second, the hand structure itself is less constrained in the presence of objects: parameter ranges have to be reconsidered and some simplifying assumptions derived from human anatomy

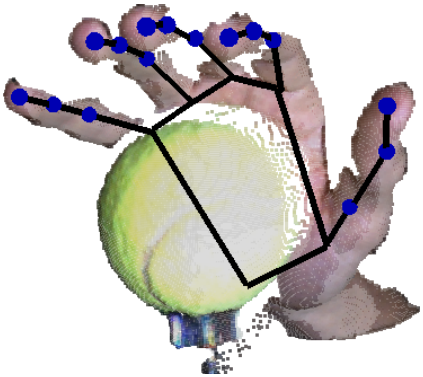


***Figure 4.1**: The goal of our hand tracker: recovering the articulated 3d structure of a hand during object manipulation.*

no longer hold. When in contact with an object, forces are exerted on the hand, resulting in poses which cannot be achieved with the bare hand (e.g., bending fingers backwards when pressing against a rigid surface, breaching the "2/3-rule" between the joints of a finger when pushing a button, etc.). Tracking hands under these less favorable conditions is the topic of this chapter. To the best of our knowledge, visual hand tracking in the presence of objects is uncharted terrain.

As argued in Chapter 1, object manipulation is an inherently 3-dimensional phenomenon, where 3d pose estimation in monocular video is seriously under-constrained. We therefore track hands using not only color information but also range data, delivered by the structured-light system introduced in Section 3.1.

Our approach was motivated by an established trend in object recognition and detection. Occlusion is a frequent and not reliably solved problem in these applications. Models are split into local parts, and each part separately contributes evidence about the complete model. In this way robustness to partial occlusion is achieved and the estimation relies only on observable parts, e.g., [Lowe 1999, Leibe *et al.* 2004, Felzenszwalb and Huttenlocher 2005]. The underlying global configuration can then be used to infer information regarding the occluded parts. In much the same way, we intentionally refrain from employing a single high dimensional model, but use local 6 DOF trackers for individual hand segments. We then exploit anatomic constraints between adjacent seg-

ments to enforce the hand structure. The constraints are represented by a first order Markov random field (MRF). Each of the local trackers corresponds to one rigid hand segment, and independently recovers a *pdf* over the segment pose from local evidence. Then, a valid hand structure is enforced by belief propagation (see Section 3.6) on the hand graph. An explicit occlusion model makes sure that missing local observations do not corrupt the estimation.

The tracker we present has also been inspired by [Sudderth *et al.* 2004b], which introduced the idea of belief propagation on a graph consisting of local hand parts. Although the approach has been extended in [Sudderth *et al.* 2004a] to incorporate self-occlusion between hand-parts, their method targets only bare hands. In our work, we consider not only self-occlusion but also occlusion by an object, which is handled explicitly with an occlusion model.

In [Sudderth *et al.* 2004b, Sudderth *et al.* 2004a] sampling is performed along the kinematic chain. For example, given the pose of the proximal phalanges, samples of the intermediary phalanges are generated within assumed joint limits. We refrain from this for two reasons: firstly, the observation of the palm is important in such an approach, but the palm is often occluded during object handling; secondly, such a sampling imposes hard constrains on the joints, since samples even slightly violating anatomical assumptions are never drawn. However, some anatomical constraints no longer hold strictly when contact with an object is involved. As a remedy one could try to widen angle intervals, but this would in turn question the guidance provided by this sampling strategy.

Instead of sampling along the kinematic chain, we focus on the independence of the local trackers and sample from local proposal functions. To cover the state space appropriately we proceed hierarchically. To avoid impossible configuration, we impose soft constraints by penalizing a sample's deviation from a valid hand shape.

Finally, in contrast to [Sudderth *et al.* 2004b, Sudderth *et al.* 2004a] we model the hand as a collection of surface patches, rather then only silhouette edges, which allows for a richer representation. The observation likelihood of these patches is measured using a modified 3d distance transform.

4.2 Hand Model

[1] University of Washington, http://uwmsk.org/RadAnat

Figure 4.2: *An X-ray picture of a human hand shows the 27 bones. Image courtesy of M. L. Richardson*[1].

We now introduce the hand model used for tracking. Human hand anatomy (see Figure 4.2 for an X-ray) is discussed in detail in Section 3.5.1. For tractability our model makes some assumptions with regard to this anatomy.

- The palm is assumed to be a rigid body. While this is not entirely true for real hands [McDonald *et al.* 2001] the amount of achievable deformation is quite small. We therefore choose to neglect it, and represent the palm as a single segment in our model, following [Wu and Huang 2001]. The geometry of the palm is bounded by the center of the wrist, the trapezium connecting wrist and thumb, and the 4 MCP joints of the fingers.

- As the wrist itself is irrelevant in terms of object grasping [Cutkosky and Wright 1986], we do not model it explicitly. The wrist is however represented implicitly as part of the palm.

- Referring to [Hamilton and Dunsmuir 2002] we assume the following relationships between phalanges. For all the fingers, there is a ratio of 1 for the distance between the MCP and PIP joint and the distance between

the PIP joint and the finger tip. The ratio for the distance between the PIP and DIP joints and the distance between the DIP joint and the tip approximates to 1.3 for the index, middle and ring fingers and to 1.0 for the little finger.

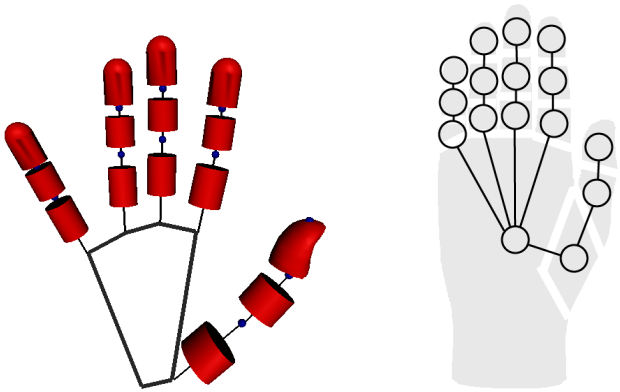
- The thumb is structurally treated like a fifth finger, connected to the trapezium by the TM joint. Though it is difficult to model the complicated anatomy of the thumb precisely [Cooney *et al.* 1981, Buchholz and Armstrong 1992, Hollister *et al.* 1992, McDonald *et al.* 2001] experiments show that our simple model achieves good results.

- The PIP and DIP joints of the fingers have one DOF (flexion/extension). MCP joints have two (flexion/extension and adduction/abduction). Finally, we allow for a limited amount of twist for all joints. Regarding PIP and DIP joints this twist accounts for a small rotation around the bone axis possible due to a little slack within the joints. In the case of MCP joints, spreading the fingers automatically implies a few degrees of twist [Cailliet and Davis 1972]. The following joint limits are assumed by our model:

$$
\begin{aligned}
0^\circ \leq\ & \Theta^{FLX}_{TM} && \leq 50^\circ \\
-20^\circ \leq\ & \Theta^{ABD}_{TM} && \leq 20^\circ \\
-5^\circ \leq\ & \Theta^{TWS}_{TM} && \leq 5^\circ \\
0^\circ \leq\ & \Theta^{FLX}_{MCP} && \leq 70^\circ \\
-15^\circ \leq\ & \Theta^{ABD}_{MCP} && \leq 15^\circ \\
-5^\circ \leq\ & \Theta^{TWS}_{MCP} && \leq 5^\circ \\
0^\circ \leq\ & \Theta^{FLX}_{PIP,DIP} && \leq 70^\circ \\
0^\circ \leq\ & \Theta^{ABD}_{PIP,DIP} && \leq 0^\circ \\
-5^\circ \leq\ & \Theta^{TWS}_{PIP,DIP} && \leq 5^\circ.
\end{aligned}
$$

FLX stands for flexion/extension, *ABD* for abduction/adduction, and *TWS* for twist. For example, the FLX angles of a straight finger are 0.

In contrast to the model of the synthetic hand discussed in Section 3.5.2 we do not describe the hand as a whole but split it up into local segments. Consequently, we consider $3 \times 5 = 15$ phalanges for the fingers and the thumb, plus the palm.



(a) Hand model *(b)* Hand graph

***Figure 4.3**: (a) The complete hand model consisting of a skeleton and ruled surfaces for the skin. Each hand segment contributes a part of the stick-model forming the skeleton. A sphere indicates the end point of each phalanx. (b) The graph encoding the structure of a hand.*

4.2.1 Local Hand Segments

In our model, each hand segment (either phalanx or palm) has its own six dimensional state space, with three dimensions corresponding to the position of the segment and the other three to its orientation ($16 \times 6 = 96$ DOFs for the whole hand). The state of a segment is represented by a local coordinate system aligned with the segment. To complete the model, we associate every phalanx with a mesh approximating the skin. Each mesh is a composition of shape primitives like cylinders and spheres, with the exception of the more detailed thumb tip.

The local hand model is visualized in Figure 4.3(a). Each hand segments contributes a part of the stick-model forming the skeleton. A sphere indicates the end point of each phalanx. Meshes corresponding to individual segments are drawn in red.

4.2.2 Adaptability to Different Anatomies

One advantage of the presented model compared to the synthetic hand (Section 3.5.2) with respect to tracking is its adaptability to new hand anatomies.

- The meshes representing skin are modeled in Maya. As the example of the thumb tip demonstrates there is no restriction to a set of shape primitives. Any arbitrary shape can be created in a 3d modeling software to approximate the respective hand segment better.
- The length of the phalanges can be adapted to fit new hand anatomies. We currently define the pose of a hand to be tracked manually in the first frame of the sequence for initialization. The user performs some mouse clicks at the depth data to indicate the position of the center of the wrist, the trapezium, the joints, and the finger tips. Depending on the normal of a selected depth point and the diameter of the respective segment, the wrist/trapezium/joint/tip is placed shortly behind the observed skin in 3d. This provides us not only with an initial hand pose but also with a measure of the length of each phalanx. We rectify the manual input with respect to the anthropometric data of [Hamilton and Dunsmuir 2002] and elongate or shorten each phalanx and the associated mesh.
- The scale of a mesh includes not only its elongation but also its thickness (from front to back) and its broadness (from side to side). These additional parameters can be used to fit a given hand anatomy even better. However, in practice the same settings worked well for many different hands.

4.2.3 Anatomical Constraints

The individual hand segments are connected within a graph representing the structure of a hand. The graph is shown in Figure 4.3(b). Since each segment has its own pose, constraints are required to ensure that neighboring segments stay connected at the joints, and that their respective orientations result in a valid configuration.

Note that in the chosen parameterization the constraints obey the first order Markov property (i.e., they apply only to adjacent segments), and that the hand

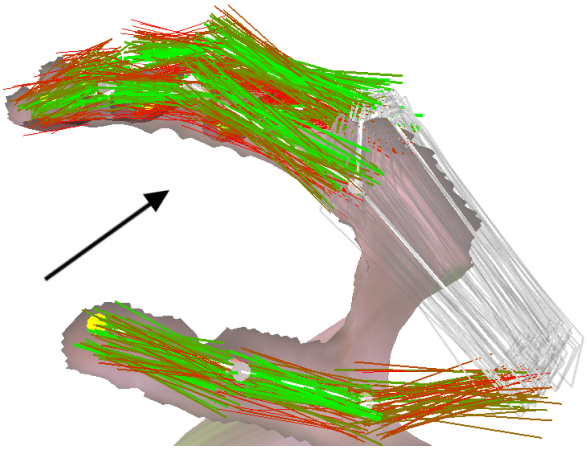


***Figure 4.4**: Hand segment samples. Color encodes relative observation likelihood: green is highest, red is lowest. The palm has no model, hence uniform observation likelihood. The arrow indicates the viewing direction of the camera.*

graph is a tree with the root at the palm and leaves at the fingertips. The constraints can therefore be efficiently optimized with belief propagation. A general introduction to belief propagation can be found in Section 3.6.

We use soft constraints: to make sure hand segments stay (nearly) connected, we employ *proximity constraints*, meaning that we penalize configurations of neighboring segments proportionally to the distance between their endpoints. To ensure valid joint angles, we use *angle constraints*. As already argued, the traditional anatomical limits for the free hand are no longer valid in contact with objects, so soft constraints are well suited. Details about the penalty function and how it exploits the constraints are given in Section 4.3.3.

4.3 Tracking Method

Every segment of the hand model has its own local tracker. In each computation step, the local tracker draws a number of samples from a local proposal function. The sample space is 6d – three parameters for the position of the

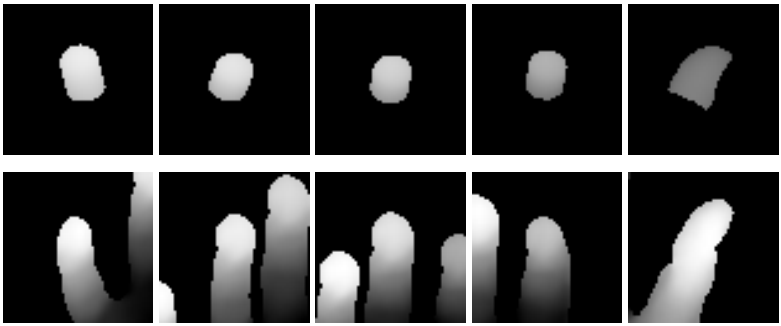


***Figure 4.5**: Top row: examples of model patches – one rendered sample for each finger tip (from left to right: little, ring, middle, index, thumb). Brighter pixels are closer to the camera than darker ones. Bottom row: the corresponding data patches. Often, parts of more than one finger are contained. Black areas represent background (unknown/infinite depth).*

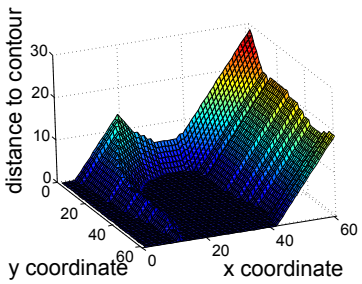
segment, and three rotation angles. Figure 4.4 shows examples of hand segment samples. We sample each parameter uniformly, within a different range to account for the kinematics of the human hand (e.g., it is easy to check that we can bend our fingers faster than we can spread them). For each sample, the likelihood is computed by comparing it locally to the observation (Section 4.3.1), taking into account occlusion information (Section 4.3.2). Then, belief propagation (see Section 3.6) is applied to combine the evidence of the local trackers (Section 4.3.3). The resulting weights, together with the corresponding samples of a hand segment, are a discrete representation of the posterior *pdf* over the segment's pose. The posterior *pdf* is then transformed to the next time step with a dynamic prediction to yield the new proposal function.

4.3.1 Observation Model

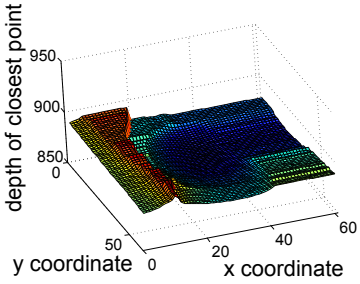
The input for our method is the range data provided by the structured-light system introduced in Section 3.1. The color information delivered by the system is exploited in a preprocessing step to locate the hand and to detect object occlusion (see Section 4.3.2) via skin color segmentation.

The local observation of a phalanx consists of a rectangular patch of range data D (the "data patch") around the predicted position. We do not consider here

(a) Depth data



(b) Distance Transform



(c) Depth of Closest Observation

Figure 4.6*: Extended model-to-data distance. (a) Depth observation of the thumb tip. (b) Distance transform showing for each pixel the 2d-distance (ignoring depth) to the nearest skin point in (a). (c) Extended distance transform visualizing for each location the depth of the nearest skin point.*

the observation of the palm, which is often completely occluded during object handling.

To compare a pose sample for a given phalanx to the data, its local surface mesh is rendered into a depth image, using the known camera calibration. This projection yields a depth image M (the "model patch"), which is in point-wise correspondence with the range data for that segment, so that the two patches are directly comparable. Figure 4.5 provides examples of model and data patches. The patches $\{M\}$ for the entire set of samples can be computed efficiently by rendering all samples as one big texture on the GPU.

To evaluate the likelihood of a sample x, we compare its rendered depth image M to the corresponding data patch D with a simple distance measure d_u, computed for all pixels u_M of the hand surface in the model patch. If a pixel $u_M = (u_{1,M}, u_{2,M}, u_{3,M})$ belongs to the surface in both the model *and* the data patch, we directly use the depth difference, $d_u = \|u_{3,M} - u_{3,D}\|$. If the

hand pixel in the model at location u_M does not belong to the hand according to the data, then we use a distance to the nearest hand pixel $\bar{u}_D$:

$$d_u = \sqrt{(u_{1,M} - \bar{u}_{1,D})^2 + (u_{2,M} - \bar{u}_{2,D})^2 + (u_{3,M} - \bar{u}_{3,D})^2} \,. \tag{4.1}$$

The distance between u_M and $\bar{u}_D$ can be computed efficiently using an extended 3d distance transform. The extended distance transform is illustrated in Figures 4.6(a)–4.6(c).

In comparing the data with the model, we do not consider the situation where there is a hand pixel in D, but not in M (unexplained observation): it cannot be decided locally, whether the data is observed by another hand segment, since the local tracker has no information from other parts of the hand. The likelihood of a sample x with patch M is hence defined as

$$L(D|x) = \frac{1}{Z}\, e^{-\left(\frac{\hat{d}}{\sigma_{obs}}\right)^2} \,, \tag{4.2}$$

where σ_{obs} is a user parameter which specifies the accuracy of the range data. Z is a normalization factor, which assures probability distributions integrate to 1, and will from now on be omitted for brevity. $\widehat{d}$ is the mean value over all T considered distances, $\widehat{d} = \frac{1}{T} \sum_{u_M \in M} d_u$.

We prefer to use the average error for the sample, rather than computing an individual likelihood $L(D|u_M)$ for each pixel and multiplying them together. This choice is motivated by the nature of range scanners: the point density of such systems depends on the surface orientation. Furthermore, there are occasionally missing depth values, and these tend to be clustered, forming holes in the observed surface. These two properties may cause a heavy bias in error measures that depend on the number of observed pixels.

4.3.2 Occlusion Model

Given the large amount of occlusion during object manipulation, an explicit occlusion model is required to achieve robustness.

After obtaining the observation patch of a hand segment, we label self-occlusion within the patch. Accurate detection of self-occluded pixels requires the global

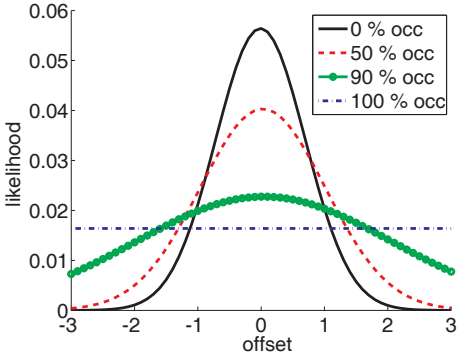


***Figure 4.7**: Illustration of the occlusion model. If the model is moved away from its correct position (offset 0), its likelihood decreases. However, with increasing occlusion the difference in likelihood becomes smaller, as less and less evidence supports it.*

hand configuration. Locally, the only way to detect self-occlusions is to find regions where the data is substantially closer to the camera than predicted by the model. We do this by applying a distance threshold of 10 mm (approximately the diameter of a finger). Object points (depth observations inconsistent with the skin color model) are also regarded as occluders, if they are closer to the camera than the skin mesh of a sample.

When computing the mean depth error $\widehat{d}$ as explained in Section 4.3.1, we do not count the occluded model pixels as part of the model, which is equivalent to assigning the average error of the visible pixels to the occluded ones. Such a definition does not penalize hand samples for moving into occlusion, but attracts them to the data, as soon as they move out of the occlusion. This behavior has proved to be desirable in our experiments.

As explained above, we do not take into account the number of observed points on a hand segment, because it would bias the estimation from range data. However, in the presence of occlusion the amount of visible surface *does* matter: if a hand segment is completely occluded, the *pdf* from the observation model should be uniform, since there is no information available about its pose. Moreover, if it is *partially* occluded, there are observations about the pose of the segment, but they should not carry the same weight as for an unoccluded seg-

ment, both due to the smaller number of observations and to the narrower field of view. We introduce a smooth dependence on the amount of occlusion by introducing an additional factor α in the exponent of Equation 4.2 as follows:

$$L(D|x) = e^{-\alpha \cdot \left(\frac{\hat{d}}{\sigma_{obs}}\right)^2}, \tag{4.3}$$

where α is the fraction of unoccluded pixels, estimated from the predicted state. Intuitively, this definition produces peakier distributions for unoccluded segments, and gradually flatter distributions as the degree of occlusion increases. In the extreme case of total occlusion, the exponent in (4.3) vanishes. All samples are assigned an equal likelihood, and the pose is entirely determined by the structural constraints. Figure 4.7 graphically illustrates this definition.

4.3.3 Enforcing Constraints

As already discussed, the (soft) constraints modeling the structure of the hand can be divided into two categories, those acting on the position of neighboring segments (*proximity constraints*), and those acting on their orientation (*angle constraints*). The constraint network is a tree obeying the first-order Markov property, hence constraint enforcement by belief propagation will yield a globally optimal configuration.

A general introduction to belief propagation can be found in Section 3.6. In that section, the joint probability distribution of a set of random variables $\{x\}$ connected in a MRF is defined as

$$p(\{x\}) = \frac{1}{Z} \prod_{st} \psi_{st}(x_s, x_t) \prod_{s} \phi_s(x_s), \tag{4.4}$$

where Z is a normalizing constant. This definition is now applied to the hand graph in Figure 4.3(b). The data term is defined by the likelihood term described in Section 4.3.1 and Section 4.3.2: $\phi_s(\boldsymbol{x}_s) = L(x_s|D) = L(D|x_s)$, assuming for now a uniform prior of x_s and D, although we will define an object-dependent prior for x_s in Chapter 5.

We now focus on the compatibility term $\psi_{st}(x_s, x_t)$ that defines the dependency between two nodes s and t connected by an edge. Sending a message

from node s to node t (in this case hand segments), requires a $N \times N$ constraint matrix, N being the amount of samples at each node. The matrix is computed considering all possible combinations of samples of the two nodes s and t, and defined by

$$\psi(x_s, x_t) = p_{prox}(x_s, x_t) \cdot p_{ang}(x_s, x_t) \ , \tag{4.5}$$

with p_{prox}, p_{ang} the two types of constraints.

Proximity constraints make sure the hand segments stay connected. We define the proximity error ϵ_{prox}—the degree to which adjacent segments violate the constraint—as the Euclidean distance between the corresponding endpoints,

$$p_{prox} = e^{-\left(\frac{\epsilon_{prox}}{\sigma_{prox}}\right)^2} \ . \tag{4.6}$$

The parameter σ_{prox} specifies the importance of the observed error, and also the relative weight of this error against those of the angle constraint errors to be defined next. We set σ_{prox} to 5 mm.

Angle constraints are defined in a similar way. In analogy to the segments of the hand model, each sample has a local coordinate system. Consider the angles (*flexion*, *abduction*, *twist*) rotating the local coordinate system of a sample of node s into the local coordinate system of a sample of node t. We compare these angles to anatomically valid angles for the connecting joint and compute error values ϵ_{flex}, ϵ_{abd} and ϵ_{twist} such that

$$p_{ang} = e^{-\left(\frac{\epsilon_{flex}}{\sigma_{flex}}\right)^2} \cdot e^{-\left(\frac{\epsilon_{abd}}{\sigma_{abd}}\right)^2} \cdot e^{-\left(\frac{\epsilon_{twist}}{\sigma_{twist}}\right)^2} \ . \tag{4.7}$$

Again, σ_{flex}, σ_{abd} and σ_{twist} encode the relative importance of the different constraints, and their importance compared to the other observed errors. In our case, $\sigma_{flex} = \sigma_{abd} = \sigma_{twist} = 10$ degrees. Figure 4.8 gives an example for the first term of the equation.

Once all samples have been generated and their likelihoods with respect to the observations have been computed, belief propagation is used to propagate the local probabilities through the graphical model.

An obvious constraint, which we have not used so far, is that fingers cannot intersect. The reason for not using it is that in our experiments, we have seldom

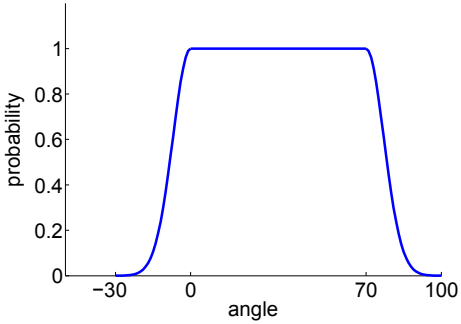


***Figure 4.8**: Soft constraints on the flexion of a joint. These probabilities are assigned when assuming a motion range from* 0 *to* 70 *degrees and a* σ_{flex} *of* 10 *degrees. Angles outside the assumed range are less probable but still possible.*

observed such problems. This testifies to the robustness of the proposed technique, but it also prevents the graphical model from including loops (see Figure 4.3(b)). This is important as loop-free belief propagation delivers an exact solution – all available information is distributed throughout the entire graph – and is guaranteed to yield the optimum. Moreover, computational efficiency is better than with loopy belief propagation, which would be the fall-back strategy in case loops need to be included [Sudderth *et al.* 2004b]. Hence, including non-intersection of fingers is feasible without major alterations to the system.

4.3.4 Hierarchical Computation

The local trackers discretize their state space by sampling. The computational cost of evaluating the compatibility functions within the belief propagation scales quadratically with the amount of samples drawn by the local trackers (see Section 4.3.3). To guarantee a sufficiently fine discretization of the state space and an acceptable computation time, we proceed hierarchically. For each frame of the input data, pose sampling, observation evaluation and belief propagation (i.e., one complete computation step) are performed several times. At first we sample in a large region of the state space in order to cover the required portion of the space. The S' samples with the highest weights are selected (in our implementation $S' = 5$), and uniform kernels are placed at their positions, as new local proposal functions for the next step. We have experimentally

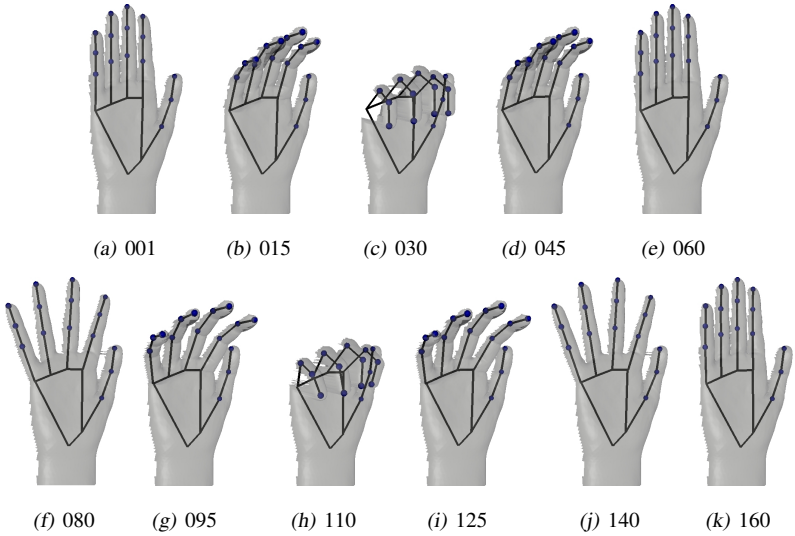


***Figure 4.9**: An artificial sequence of 160 frames demonstrating the hand tracker's robustness to strong self-occlusions. The hand forms a fist and opens up again twice. The first time all fingers are straight, the second time fingers are spread. Tracking results are indicated by the stick model skeleton. Little blue spheres represent end points of phalanges. The number below each image identifies the respective frame.*

found that the number of modes in the state space is usually ≤ 3, so that no important samples are lost by the intermediate hard decisions.

We use 10 hierarchy levels, i.e., we sample, observe, and infer 10 times per frame. When the last level is completed and the transition to the next frame occurs, it has proved beneficial in practice to include a simple dynamic model in the proposal function. In our implementation we use ICP (see Section 3.3) to predict large global hand motion, and a linear (constant velocity) prediction for the motion of individual hand segments. With the prediction step, sampling can focus on deviations from the dynamic model.

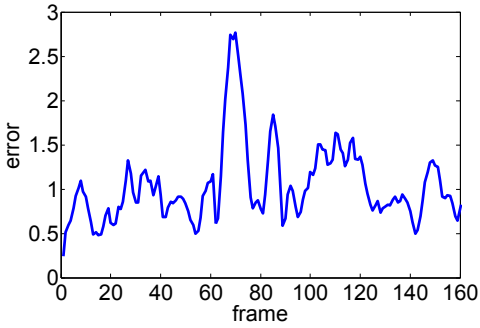


Figure 4.10: *Tracking error of the sequence with strong self-occlusion.*

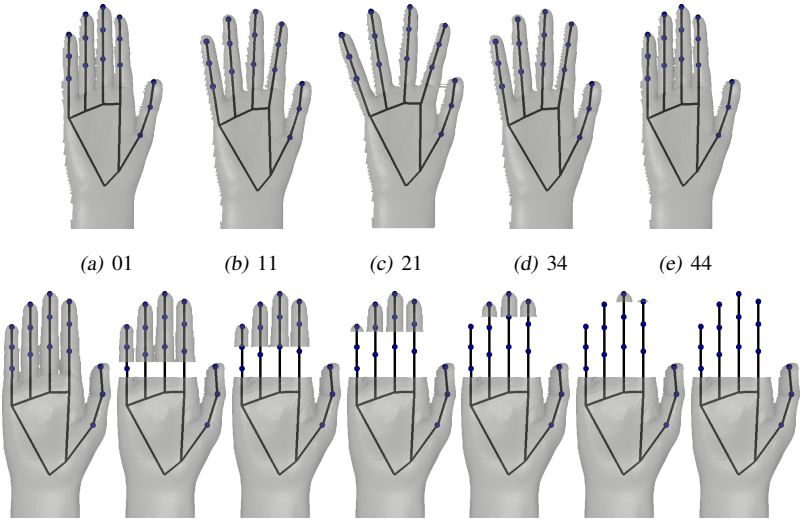
4.4 Results

We have conducted experiments with both synthetic and real data. Synthetic data serves for quantitative evaluation, while the real data confirms that the proposed method is applicable to the input delivered by our structured-light setup. Computation time of our C++/Cg implementation is $\approx$ 6.2 sec/frame, with a 3GHz CPU and a GeForce 8800 Ultra.

4.4.1 Artificial Data

Artificial data was generated by rendering the synthetic hand into a depth texture on the GPU as described in Section 3.5.2. The advantage of this kind of data is that we have *ground truth*, i.e., we know the hand poses that produced the data. This allows for a quantitative evaluation. Two experiments are presented with respect to artificial data.

Strong self-occlusion. The artificial hand has been tracked over a period of 160 frames, taken at normal video rate. The hand forms a fist twice, producing extreme self-occlusion, once starting with joined fingers and once after spreading them. The sequence and tracking results are illustrated in Figure 4.9. The anatomy of the hand model for tracking (Section 4.2) was adapted to fit the anatomy of the synthetic hand. For initialization the hand pose of the first frame was taken directly from the ground truth. As an error measure, we define the error of a phalanx as the mean distance of its two endpoints from those



(a) 01 (b) 11 (c) 21 (d) 34 (e) 44

Figure 4.11: *(top) Artificial sequence used to evaluate the robustness to object occlusion. Fingers are spread, then joined again. The number below each image identifies the respective frame. (bottom) Different levels of occlusion, ranging from 0% to 100% occlusion of the fingers.*

of the ground truth given by the kinematic hand model, and the *frame error* as the mean over all segments in that frame. Frame errors for the whole sequence are plotted in Figure 4.10 and lie in the range [0.24. . . 2.77] mm (mean 1.04, median 0.92). For comparison, the distance from the base of the palm to the tip of the extended middle finger is 230 mm.

Occlusion by an object. To verify the robustness of our method in the presence of an occluding object, we have introduced artificial occluders into a sequence of 45 frames. The hand first spreads, rests for 3 frames, and then returns to its initial pose (see the top row of Figure 4.11 for illustration). Figure 4.11 (bottom) demonstrates the seven tested degrees of occlusion, ranging from no occlusion to full occlusion. The error over all hand segments in the different occlusion scenarios is plotted in Figure 4.12. Up to occlusion level four there is almost no increase of the error. At higher levels fingers are fully occluded so their state has to be hallucinated, based only on the anatomic constraints.

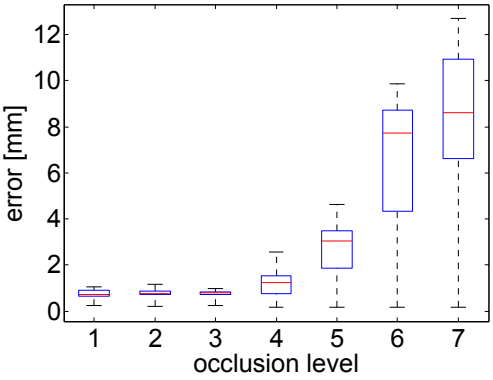


***Figure 4.12**: Seven occlusion levels. Occlusion ranges from 0% to 100% occlusion of the fingers. For each level, the median error (red), the lower/upper quartile (box) and the entire error range (whiskers) are displayed.*

4.4.2 Real Data

To assess the validity of our hand tracking method with respect to real object manipulation observed by an actual 2.5d camera, we now focus on the manipulation of three different everyday objects recorded with our real-time structured-light system. All three sequences contain severe occlusions. In order to represent different aspects of object handling, each of the sequences features a different grasp type with respect to the grasp taxonomy defined in [Cutkosky and Wright 1986]. The three objects considered are a cup/mug, a tennis ball, and a pair of pliers (see Figure 4.13). In the following we discuss the tracking results visualized in Figure 4.15, Figure 4.16, and Figure 4.17. In these figures, estimated hand poses are again visualized by a stick model skeleton. Little blue spheres represent distal endpoints of the phalanges. There may be a small gap between segments due to our soft constraints. The mean gap of all three sequences is 1.0042 mm (deviation:0.6049 mm).

In all experiments, the initial state of the hand was determined by manual initialization in the first frame. Initialization, while not the topic of this work, could be automated by using a standardized pose, comparable to the "T-pose" in commercial motion capture systems. It may even be possible to initialize the hand pose on the fly, while the hand is not in contact with an object.

***Figure 4.13**: We present results regarding the tracking of a hand manipulating three different everyday objects: a cup, a tennis ball, and a pair of pliers.*

The first sequence (Figure 4.15) shows a hand which approaches a cup, grasps it by the handle (*precision grasp*), lifts it up, then places it back on the table and releases the grip. Of particular interest are the moments at which parts of the index and middle fingers are occluded and disoccluded as they grasp the handle. The finger tips smoothly move into the occlusion, since occluded model pixels do not decrease the likelihood of a pose (Section 4.3.2). While occluded, the fingertips continue to move with the rest of the hand, as their local prediction and the proximity constraints at the joints push them forward. As soon as the skin is observed again inside the handle, the model is pulled towards the new observation by the extended distance function d_u (Section 4.3.1), because of the increased penalty for samples far away from the observed skin pixels.

The second sequence (Figure 4.16) shows a hand manipulating a tennis ball. The ball is gripped from behind with a *power grasp*, then lifted up, lowered, and released. A critical moment is shown in frame 030. The palm is largely occluded by the ball, and the intermediary phalanges of the fingers are occluded by the finger tips and are about to reappear above them. These segments now have to reattach to the skin. The sequence illustrates a limitation of our current tracker. Figure 4.14 shows that the reattachment of the ring finger lags behind several frames. The finger thus continues in an anatomically valid, but inaccurate position, until enough evidence is available for it to recover.

The third sequence (Figure 4.17) is the most complex one. The handled object is a pair of pliers, which is not only lifted with a *hook grasp*, but also pressed together. Note how the hand constraints ensure correct tracking of the fingers in spite of long occlusions and scarce ambiguous evidence (e.g., the little finger in the fourth image).

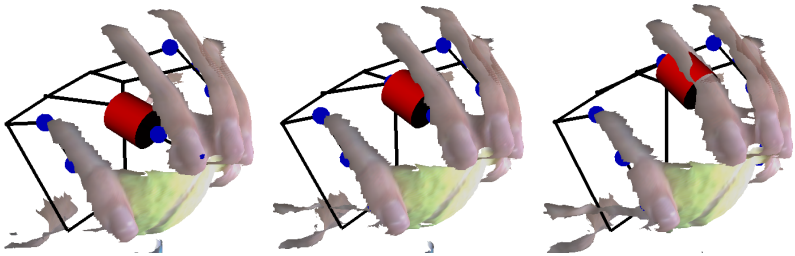


Figure 4.14: *Failure and recovery of a local tracker. The structured-light system looks at the scene from the right. The rendered side-views show how the ring finger initially fails to reattach to the back of the finger after reappearing from behind the tip, but as soon as enough evidence is available, the local tracker recovers.*

4.5 Conclusion

In this chapter we have presented a method to track the articulated pose of a hand in 2.5d range data. In particular, the method handles self-occlusions and occlusions by an object within the hand. Therefore it can be applied successfully in hand-object interaction scenarios.

In more detail, we first introduced the hand model used for tracking. The model consists of 16 local segments. Each hand segment has a six dimensional state space, and every phalanx is associated with a mesh representing skin. The hand model can be adapted to suit new hand anatomies. To enforce soft kinematic constraints (proximity and angle constraints) the local hand segments are connected in a MRF, which encodes the structure of a human hand. Due to the Markov property the graph can be optimized efficiently by belief propagation.

Next, the actual tracking method was presented. Each hand segment has its own local tracker that draws samples from a local 6d proposal function. The samples are then evaluated with regard to our observation model. According to this model the observation of a segment corresponds to a local depth data patch around the predicted position. This patch is compared to a rendered 2.5d version of the respective segment's surface model. The basic idea for this comparison is an extended, fast-to-compute distance transform.

Unexplained depth data is disregarded in the observation model because of the local nature of the model. However, in the presence of occlusion the amount of

visible surface has an important impact within the presented occlusion model. The less data supports the likelihood of a segment, the less impact has the respective segment on the global configuration/ the global hand pose.

After elaborating on the occlusion model we discussed the issue of enforcing kinematic constraints. Before a global hand configuration can be found by belief propagation it is necessary to consider proximity and angle constraints by means of compatibility matrices. We explained how to compute such matrices based on products of negative log-likelihoods, admitting slight anatomical violations possibly caused by hand-object interaction.

For reproducibility of our method we further gave some implementation details. In particular, we went into our hierarchical computation scheme. This scheme involves performing belief propagation several times per frame to cover the hand's state space with a sufficiently fine granularity.

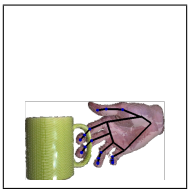
Finally, we presented results on artificial and real data. While the results on artificial data served for quantitative evaluation, those on real data show that we can successfully track the articulated pose of a hand manipulating objects. This achievement contributes to the state of the art in the field of marker-less hand tracking.

Our system offers the information a classical data glove provides. Therefore, previous work based on the output of data gloves can be applied on top of our method - e.g., different grasp types like precision and power grasps can be recognized [Ekvall and Kragíc 2005] and associated with the grasped object.

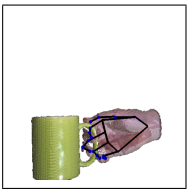
There are limitations to our method. When the articulation of a hand changes too rapidly hand segments might *detach* from their range data, despite our hierarchical computation scheme. The same might occur when the hand observation is too corrupted by the occluding object, e.g., when the object fully occludes the hand. Figure 4.14 shows a case in which the system recovers and the affected phalanx *reattaches* to the observation. However, this is not always possible and the system sometimes *looses track*. Loosing track is an inherent problem of model-based hand tracking approaches like ours, as opposed to the (less precise) discriminative systems that work on a per frame basis. One remedy might be to (re-)initialize our hand tracking system with a discriminative approach like the one recently proposed in [Romero *et al.* 2010] after failure. [Romero *et al.* 2010] is also concerned with hand-object interaction, so the method might be a perfect fit.

We emphasize that the method introduced in this chapter does not depend on knowledge about the manipulated objects. However, when the object geometry is known (e.g., from CAD-models or range scans), then this delivers valuable additional constraints. To give just one example: hand segments cannot penetrate the object. Therefore in the remainder of this work we do not focus on isolated hand tracking but explore possible benefits of object knowledge in our setting.

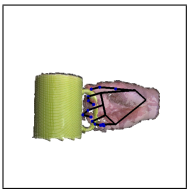
(a) frame 003



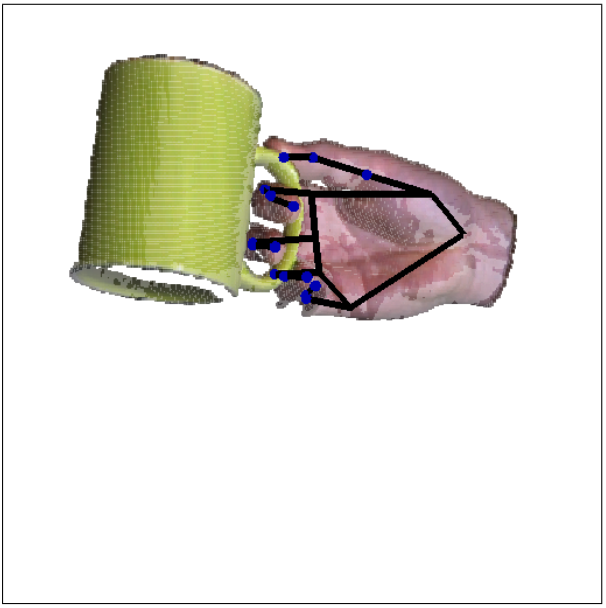
(b) frame 021



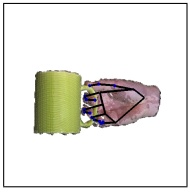
(c) frame 041



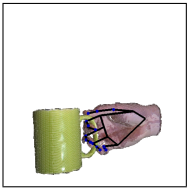
(d) frame 048



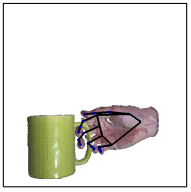
(e) frame 055



(f) frame 063



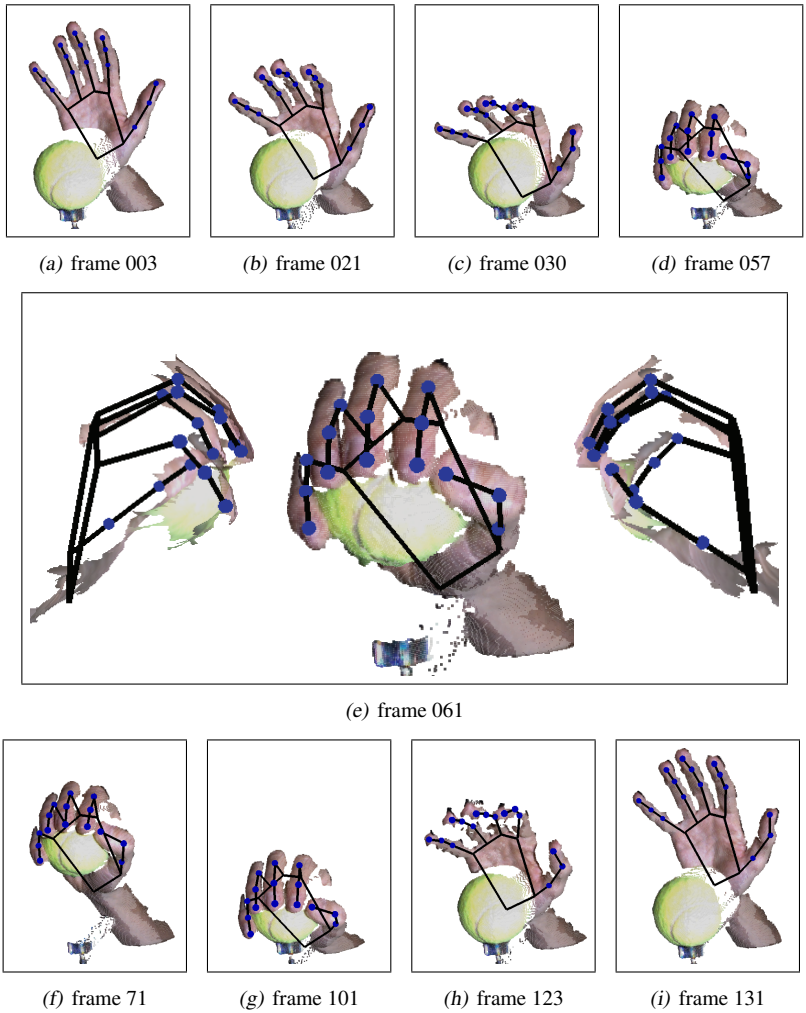
(g) frame 071



(h) frame 086

(i) frame 101

Figure 4.15*: In this sequence a hand approaches a cup, grasps it by the handle, lifts it up, then places it back on the table and releases the grip. The hand is tracked successfully throughout the sequence. Tracking results are indicated by the stick model skeleton. Little blue spheres represent end points of phalanges.*

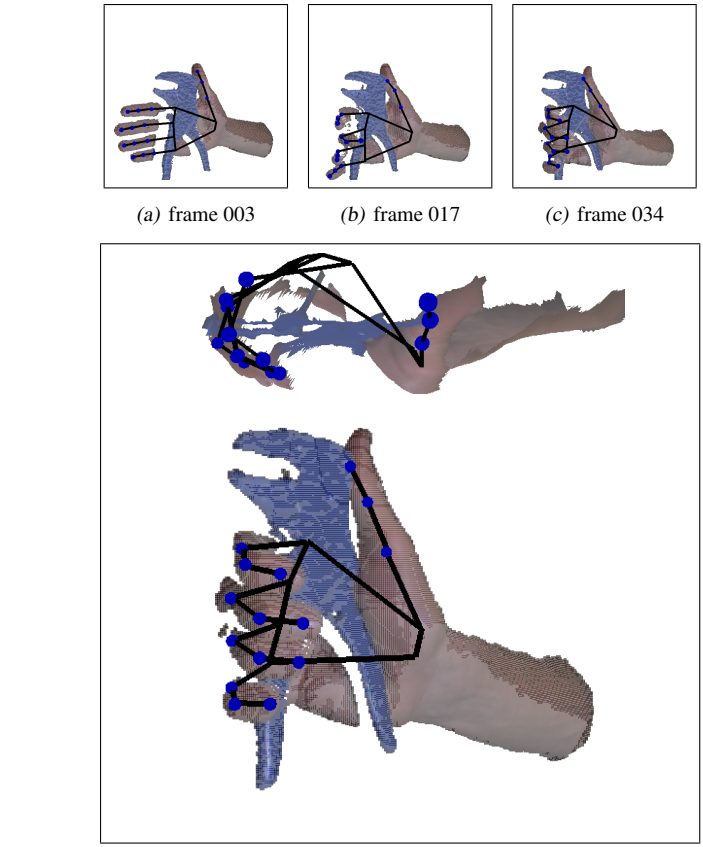


(a) frame 003 *(b)* frame 021 *(c)* frame 030 *(d)* frame 057

(e) frame 061

(f) frame 71 *(g)* frame 101 *(h)* frame 123 *(i)* frame 131

***Figure 4.16**: A tennis ball is gripped from behind, then lifted up, lowered, and released. To give a better impression of the three dimensionality of data and estimated hand poses, frame* 061 *is illustrated with three different views: a frontal view (center), looking at the scene from the left (left), and looking at the scene from the right (right).*



(a) frame 003 *(b)* frame 017 *(c)* frame 034

(d) frame 045

(e) frame 062 *(f)* frame 081 *(g)* frame 114 *(h)* frame 154

***Figure 4.17**: A pair of pliers is seized, lifted, pinched together, opened, and put down again. Two different views visualize frame 045: a top view (top) and a frontal view (bottom).*

5

Hand-Object Interdependency

In Chapter 4 we discussed the tracking of a hand manipulating an object. However, the object plays only a passive role as an occluder. The hand is treated more or less in isolation and the inherent dependencies between hand and object are not accounted for.

In this chapter we go one step further and explicitly consider such dependencies during hand-object interaction. More precisely we encode the influence of a manipulated object on the manipulating hand in an *object-dependent hand pose prior*, which is derived from sparse training data. Intuitively, this prior represents information of how a certain object is to be grasped by a certain hand.

Starting point for the techniques proposed in this chapter is the acquisition of knowledge about the manipulated object, i.e., detailed information regarding the shape and the pose of a (rigid) object. For this purpose we use the structured-light scanning setup.

A key feature of the prior is that it can be generalized towards new hands and new objects of the same object class. This opens up new possibilities for a variety of interesting applications. We use the prior to improve the hand tracker introduced in Chapter 4, and to let the synthetic hand (Section 3.5) grasp previously unobserved objects. The work in this chapter has also been presented at the *IEEE Conference on Computer Vision and Pattern Recognition 2010* [Hamer *et al.* 2010].

5.1 Overview

Humans are accustomed to using their hands for the manipulation of objects and their eyes for observing such manipulation. By watching another person handle a single instance of an unknown object class, humans can easily imitate the observed hand poses to manipulate other instances of the same class. Although the strong correlation between the nature/shape of an object and the hand poses for its manipulation is obvious, only little work has been done so far to exploit this information for vision-based hand pose acquisition.

Hand tracking approaches either focus on freely moving hands for gesture recognition or regard a grasped object only as an occluder [Rehg and Kanade 1994, MacCormick and Isard 2000, Wu *et al.* 2001, Stenger *et al.* 2001, Athitsos and Sclaroff 2003, Sudderth *et al.* 2004b, Stenger *et al.* 2006, Hamer *et al.* 2009]. For object handling, however, the degree of occlusion can be so large, that occlusion robustness alone is not sufficient. Due to missing observations, many spatial ambiguities for the phalanges occur that cannot be resolved without additional knowledge.

In the context of marker-less human motion capture, this issue has been addressed by introducing priors on motion patterns [Sidenbladh *et al.* 2002, Moon and Pavlovic 2006, Urtasun *et al.* 2006, Baak *et al.* 2009] that are learned from a motion database. Regarding the interaction of the tracked human bodies with their environment, basic constraints like contact with the ground plane have been used in [Rosenhahn *et al.* 2008, Vondrak *et al.* 2008].

For hands, the only priors that have been used so far in this context rely on the static or dynamic space of hand poses. To obtain these priors a large data-set of hand poses has been acquired by synthesizing hands or with data gloves [Athitsos and Sclaroff 2003, Stenger *et al.* 2006]. Since these works focus on gesture recognition, they consider only freely moving hands. None of these priors capture the interaction with objects.

Here a prior is proposed which integrates such relation. We proceed in three steps:

1. A specific hand is tracked in 3d while it manipulates a specific object of a certain object class. We map the captured poses, i.e., the 3d position and rotation of each hand segment (like a phalanx), into the local coordinate

system of the object. Then, contact points on the object are detected. For illustration see Figure 5.1.

2. The knowledge coming from several observed manipulations - performed by different hands on different class members - forms the prior, i.e., a spatial distribution of the pose samples.

3. The prior is generalized towards expected manipulations of new objects from that class, possibly by new hands, based on a geometric warp.

The adapted, object-dependent prior can be used both to improve hand tracking and to synthesize grasps. Both tasks are embedded in the same belief propagation framework as described in the course of this chapter. A hand pose's probability can then be defined with respect to the prior, contact points, object intersection constraints, anatomical constraints of the hand, and data likelihood.

The synthesis of grasps and manipulating hand motion has been addressed in the field of computer graphics. Most similar to our approach are data-driven approaches like [Pollard and Zordan 2005, Kry and Pai 2006, Li *et al.* 2007]. [Pollard and Zordan 2005] and [Kry and Pai 2006] both target physically-based grasp synthesis and recorded object manipulation with a marker-based Vicon system to train grasp controllers. New interactions are synthesized by running physically-based simulations, taking into account the knowledge encoded in the controllers and contact forces estimated during collisions. The controller in [Pollard and Zordan 2005] compensates for arm movement. The authors of [Kry and Pai 2006] also recorded real contact forces to integrate the stiffness of hands into their controller. In [Li *et al.* 2007], grasp candidates are found in a database for a given object by matching contact points and surface normals. The most stable grasp is then selected considering gravity.

Grasps have also been studied in robotics [Bicchi and Kumar 2000]. Given a full 3d model and a grasp pose, for instance, the quality of the grasping can be evaluated based on pre-computed grasp primitives [Miller *et al.* 2003]. In [Saxena *et al.* 2008], the 3d grasp position is estimated from two images where grasp locations are identified. For this, a 2d grasp point detector is trained on synthetic images. Other approaches are based on learning by demonstration and imitate human behavior. For instance in [Hueser and Baier 2006], a very small set of task relevant hand poses is selected and used to build a low dimensional hand model for grasp pose detection.

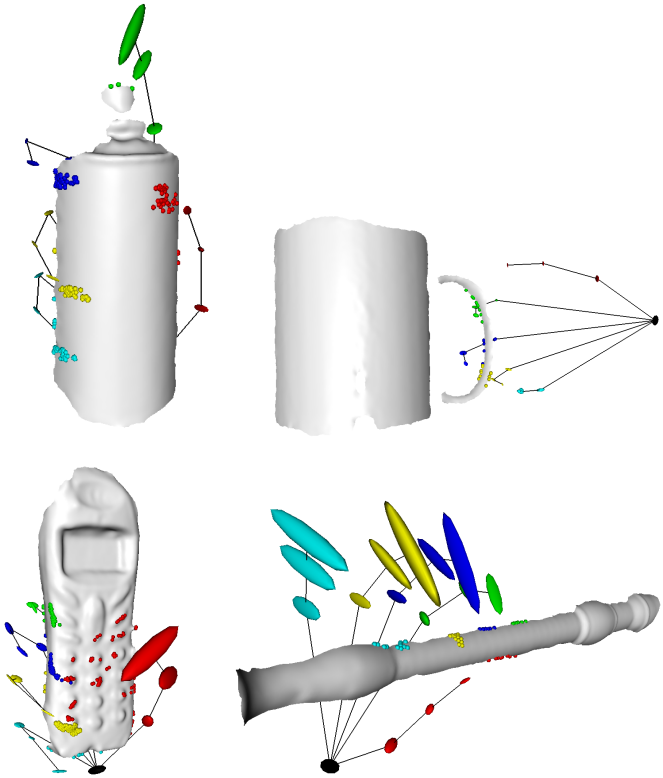
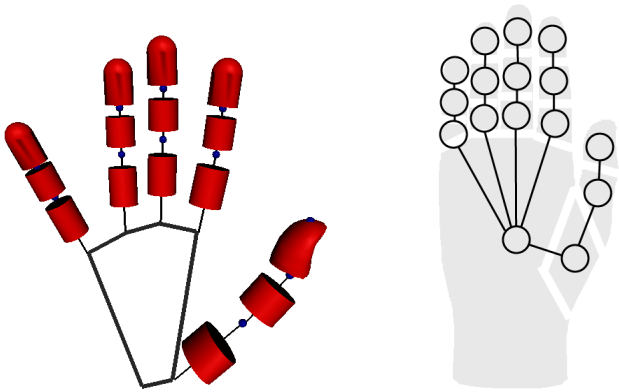


Figure 5.1: *Captured object manipulation in a simplified illustration. An ellipse stands for a set of center points observed for some hand segment. Each finger has its own color. On the meshes obtained by in-hand scanning, contact points of the individual hand segments are visualized as little dots. Intuitively, the images can be considered as automatically generated instruction manuals.*

Inspired by the ability of humans to learn the interaction with an object from a single example, we focus on sparse training data, i.e., we can already build the prior by seeing only one instance of an object class being manipulated by one person.



(a) Hand model *(b)* Graph

***Figure 5.2**: (a) Hand model with a skeleton and ruled surfaces for the skin. (b) Graphical model for inference.*

5.2 Prior

Statistical priors on hand poses are useful since they constrain the search space for tracking and allow for the prediction of hand poses when combined with additional constraints. The first property is important to overcome ambiguities due to missing data or occlusions and thus to improve tracking. The second property is important as well. In robotics, unseen instances of an object class need to be grasped. In computer graphics, the hand of an animated character should snap to a virtual object automatically. Since in both scenarios hand poses occurring during object interaction are the most interesting ones, we aim to model a prior for the hand that depends on the object, i.e., we model the probability of a hand pose $\mathcal{P}$ conditioned on an instance $\mathcal{O}$ of a known object class and a hand size $\mathcal{H}$: $p_{prior}\left(\mathcal{P} | \, \mathcal{O}, \mathcal{H}\right)$.

We introduced our hand model consisting of local parts (Figure 5.2(a)) in Section 4.2. Hand pose $\mathcal{P}$ is here defined with respect to the state space of that model. Because of the direct relation of the hand model and the nature of the proposed prior, we shortly repeat the most important properties of this model. Afterwards we describe the prior in detail.

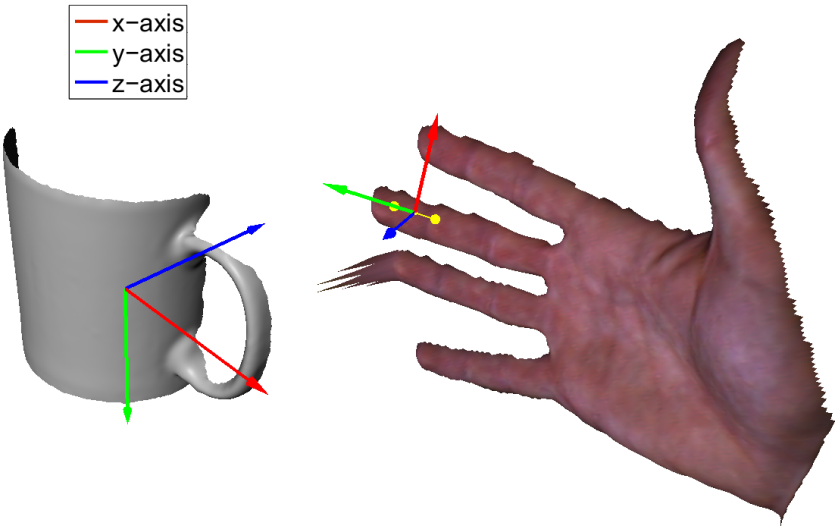


***Figure 5.3**: Hand segment sample in object coordinates. The local coordinate system of the depicted DP sample of the middle finger is now expressed in the coordinate system of the manipulated object. The origin of the object is defined by its geometric center, the axes were manually aligned.*

5.2.1 Hand Model

Each hand segment has its own 6-dimensional state space: three dimensions correspond to the position of the segment, three to its orientation (hand pose $\mathcal{P}$ has $16 \cdot 6 = 96$ DOFs). The state of a segment $x_s \in \mathbb{R}^6$ is represented by a local coordinate system, which is now defined with respect to the coordinate system of the manipulated object (see Figure 5.3). In addition, every phalanx is associated with a mesh approximating its skin for tracking. Each mesh is a composition of shape primitives like cylinders and spheres, with the exception of the more detailed thumb tip.

When modeling the hand by a set of individual segments, the likelihood of each segment s can be estimated independently with respect to simple local terms. The connections between segments (Figure 5.2(b)) are then used to enforce anatomical correctness.

5.2.2 Prior Model

Consistent with the hand model, $p_{prior}\left(\mathcal{P}|\,\mathcal{O},\mathcal{H}\right)$ is defined by a product of local 6d hand segment distributions:

$$p_{prior}\left(\mathcal{P}|\,\mathcal{O},\mathcal{H}\right)=\prod_{s}p_{prior}\left(x_s|\,\mathcal{O},\mathcal{H}\right). \tag{5.1}$$

We learn the hand segment distributions from a finite set of hand segment samples x_s^i, observed for instances $\mathcal{O}_k$ of the object class manipulated by the hands $\mathcal{H}_l$. For density estimation, we use a Parzen-Rosenblatt estimator with a 6d Gaussian kernel, defining $p_{prior}\left(x_s|\,\mathcal{O},\mathcal{H}\right)$ by

$$\frac{1}{(2\,\pi\,\sigma^2)^{6/2}\,N}\sum_{i=1}^{N}\exp\left(-\frac{\left\|x_s-f_{(\mathcal{O},\mathcal{H})}(x_s^i)\right\|^2}{2\,\sigma^2}\right), \tag{5.2}$$

where N denotes the number of training samples x_s^i. In our experiments we first computed σ with the maximum nearest neighbor heuristic, but as this value turned out to be too restrictive we applied a fixed scaling factor of 10. Since we estimate the probability of x_s conditioned on $(\mathcal{O},\mathcal{H})$, we have to map the samples x_s^i, observed conditioned on $(\mathcal{O}_k,\mathcal{H}_l)$, to hand $\mathcal{H}$ (*retargeting*) and into the coordinate system of object $\mathcal{O}$ by a mapping function $f_{(\mathcal{O},\mathcal{H})}$. When retargeting we preserve finger tip positions. Mapping to the coordinate system of a new object involves a geometric warp. Details on $f_{(\mathcal{O},\mathcal{H})}$ are given in Section 5.4.

In analogy to Chapter 4, we model the overall probability of hand pose $\mathcal{P}$ as

$$p(\mathcal{P})=\frac{1}{Z}\prod_{st}\psi_{st}(x_s,x_t)\prod_{s}\phi_s(x_s), \tag{5.3}$$

where the compatibility term $\psi_{st}(x_s,x_t)$ enforces anatomical constraints between adjacent hand segments, $\phi_s(x_s)$ contains the data term with respect to the observation, and Z is a normalizing constant. The integration of the prior into $\phi_s(x_s)$, and our unified framework for tracking and synthesis are the topic of Section 5.5.

5.3 Data Acquisition

As stated before, we require information about both the hand (the articulated hand pose) as well as information about the object (object shape and pose) for

our purposes. Chapter 4 explained how to gather hand knowledge from the data delivered by the structured-light setup. The great advantage of that setup is that it also provides the necessary raw data to derive object knowledge.

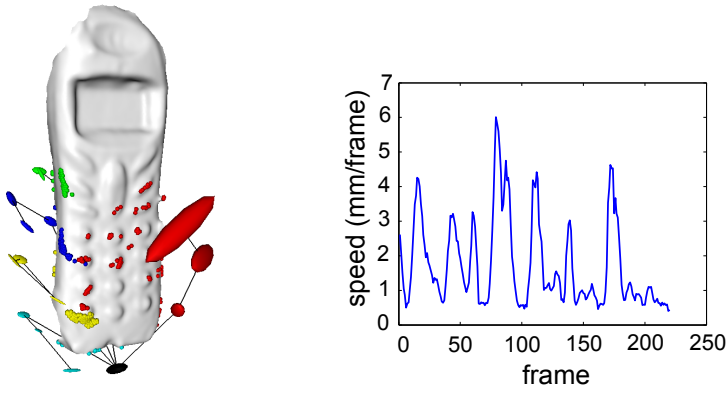
Hand Pose In the learning phase of the method presented in this chapter we track the hands in training sequences with the principles of Chapter 4. As described in the limitation paragraph at the end of Chapter 4, the tracker sometimes fails when the data is too corrupted. As a way out, we label the position of the finger tips in some key frames, making the training process semi-automatic. In those frames, finger tips are attracted by the labels instead of the local data. This said, one motivation of the proposed prior is to improve tracking and to eliminate the need for such manual intervention.

Object Shape and Pose As range scans of the object are captured continuously, we register these scans online and build up a coherent mesh of the already observed parts of the surface by in-hand scanning. In-hand scanning is the topic of Section 3.4. Examples of meshes obtained by this procedure are shown in Figure 5.1. With the mesh of the object available, we determine in an offline process the object's 6d pose (translation and orientation) for each frame of a training sequence containing the object and some manipulation. This is done by fitting the mesh to the observation with ICP (see Section 3.3).

Temporal Segmentation At this point, temporal segmentation is required to select the frames of interest. In action recognition, [Rao *et al.* 2002] identify action components with respect to global velocity changes of a manipulating hand. Inspired by [Sato *et al.* 2002], we instead consider the hand's velocity in relation to the object's velocity. Another interesting approach could be to use the velocity of individual hand segments. Consider Figure 5.4 for a motivation.

5.4 Mapping to New Hands and Objects

After having observed some hands manipulate some objects of a class, we can transfer the prior to another hand grasping another object of that class. We first retarget the acquired training examples x_s^i to the new hand size $\mathcal{H}$, and then warp them into the coordinate system of our newly observed object. The



(a) Manipulation

(b) Speed of the thumb tip

***Figure 5.4**: (a) Captured hand segment centers and contact points. The thumb clearly is the most active finger and touches the dialing area at various positions. Considering the speed of the thumb tip plotted in (b), one could recognize the dialed phone number: when the thumb rests it is most likely to press on a digit.*

adapted prior (Eq. (5.1)) can be evaluated efficiently. We now describe the two steps of the mapping process.

5.4.1 Hand Retargeting $\mathcal{H}$

Hand retargeting maps the samples to a new hand anatomy, i.e., adapts the length of the phalanges and the proportions of the palm. We preserve the positions of the finger tips in space and elongate or shorten the finger segments, from farthest to closest to the palm, respecting joint angles (see Figure 5.5). After this, the proportions of the palm (i.e., the relative positions of the attachment points of the four fingers and the thumb) are set. Finger and palm adaptation may create gaps between the fingers and the palm. We therefore apply the rigid motion to the palm that minimizes these gaps.

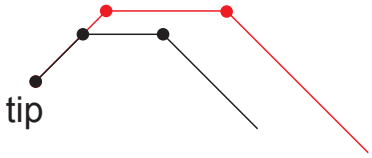


***Figure 5.5**: Retargeting of a finger. The smaller (black) finger stands for the original one and the bigger (red) one is a scaled version. The position of the finger tip and the joint angles are preserved but the translation of the segments changes.*

5.4.2 Object Warping $\mathcal{O}$

To map the prior from one object to a different one, we first warp the geometry of the corresponding meshes. For this, we use the method proposed in [Zhang *et al.* 2004]. Initially, corresponding points on the geometry have to be determined (see Figure 5.6 for an example). We currently do this manually but a fully-automatic mechanism based on 3d features like spin images [Johnson 1997] could be realized. The output of mesh warping is an offset for each vertex of the original mesh that yields a point on the target mesh.

To apply the geometry warp to a hand segment sample, we need to map three points in 3d space (e.g., for a phalanx sample, the center, one end point, and one point fixing rotation around the roll-axis). This is necessary to fully define the warped 6d sample with respect to the coordinate system of $\mathcal{O}$. One approach to map a point in 3d space (outside the mesh) is to find the closest vertex of the original object's mesh and to choose the offset assigned to that vertex. While this might work in some cases, the accuracy was insufficient in our experiments. Instead, we use radial basis functions (RBFs) [Botsch and Kobbelt 2005] to extrapolate the warp field outside the mesh and move the hand segments with respect to this warp field.

5.5 Framework for Synthesis and Tracking

Since we have modeled our hand pose prior as a product of hand segment distributions (Eq. (5.1)), it is consistent with the hand tracking framework described

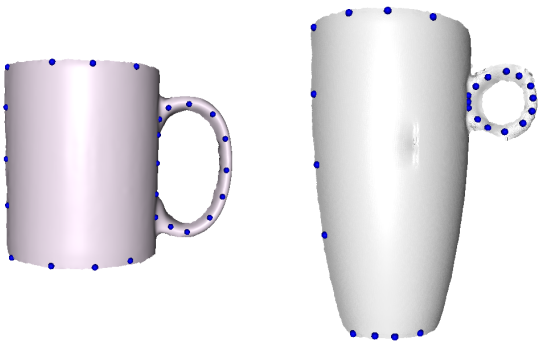


***Figure 5.6**: Correspondence pairs serving as a starting point for object warping. We currently define the pairs manually.*

in Chapter 4. This simplifies the integration of the prior to improve tracking. We use the same belief propagation framework not only for tracking but also for synthesis. Eq. (5.3) specifies the probability of hand pose $\mathcal{P}$. We define $\phi(x_s) = p(x_s|I, \mathcal{O}, \mathcal{H})$, where I is a depth image, $\mathcal{O}$ an instance of an object with a pose and observed contact points, and $\mathcal{H}$ stands for the hand anatomy. According to Bayes' theorem,

$$p(x_s|I, \mathcal{O}, \mathcal{H}) = \frac{p(I|x_s, \mathcal{O}, \mathcal{H}) \cdot p(x_s|\mathcal{O}, \mathcal{H})}{p(I|\mathcal{O}, \mathcal{H})}. \tag{5.4}$$

The denominator can be considered a normalization factor as it does not contain x_s. $p(x_s|\mathcal{O}, \mathcal{H})$ is defined by the object-dependent prior and augmented with two additional factors that enforce contact point attraction and intersection constraints:

$$p_{prior}(x_s|\mathcal{O}, \mathcal{H}) \cdot p_{contact}(x_s|\mathcal{O}, \mathcal{H}) \cdot p_{inter}(x_s|\mathcal{O}, \mathcal{H}).$$

Because the likelihood with respect to depth data is modeled as an exponential, we write

$$\phi(x_s) = \frac{1}{Z} \exp\left(-\sum_{f=1}^{4} V_f(x_s)\right), \tag{5.5}$$

For a detailed description of the data likelihood $exp(-V_1(x))$, we refer to Chapter 4. The other terms are described next.

Hand Pose Prior The hand pose prior can be integrated in a straight-forward manner by taking the negative log probability of a sample with respect to the prior:

$$V_2\left(x_s\right) = -\log\left(p_{prior}\left(x_s \middle| \mathcal{O}, \mathcal{H}\right)\right). \tag{5.6}$$

Since the training samples are acquired from sequences of varying length, we weight the samples within the Parzen estimate (Eq. (5.2)) such that the sequences contribute equally to the prior.

Contact Point Attraction Although RBFs yield good results regarding warp extrapolation, small inaccuracies still occur when warping hand segment samples. Because of this, finger tips in contact with the original mesh do not always touch the mesh after warping. To yield stable grasps, we use contact points c_s^i observed for segment s on the original mesh. Since these contact points lie on the mesh they can be warped accurately without extrapolation. After warping, we proceed with 3d contact points as we did with 6d hand segment samples above and build a kernel estimate. The likelihood term $V_3\left(x_s\right)$ of the distal phalanges with respect to the N_c contact points is then given by

$$-\log\left(\frac{1}{\left(2\,\pi\,\sigma_c^2\right)^{\frac{3}{2}}\,N_c}\sum_{i=1}^{N_c}\exp\left(-\frac{\left\|x_s - f_{\mathcal{O}}(c_s^i)\right\|^2}{2\,\sigma_c^2}\right)\right), \tag{5.7}$$

where $f_{\mathcal{O}}$ is the geometric warp. We again compute σ_c based on the maximum nearest neighbor distance between training samples.

Intersection Constraints Intersection constraints concern hand segment samples that penetrate the mesh of the object after warping. We compute the (unsigned) distances between each sample and all vertices of the mesh using the Hessian normal form. Because of the computational complexity, this is done for all samples in parallel on the GPU. For each sample, the smallest distance d_{min} is selected. Then, the degree of intersection of a sample is given by $d_{inter} = max(0, -(d_{min} - d))$, where d corresponds to the diameter of the respective hand segment. We define

$$V_4\left(x_s\right) = -\log\left(\frac{1}{Z}\exp\left(-\frac{{d_{inter}}^2}{{\sigma_{inter}}^2}\right)\right). \tag{5.8}$$

σ_{inter} is a user parameter, in our case set to 0.5. Note that there is no need to compute the normalizing constant Z, since it has no effect on belief propagation.

Synthesis and Tracking In our experiments, we demonstrate that the prior can be used within the proposed framework to improve tracking and to synthesize hand poses for a given object. Tracking is performed as described in Chapter 4 but with the additional term of the pose prior. Grasp synthesis is realized within the same framework. For initialization, we consider all warped samples and choose one sample for each hand segment by belief propagation. Warped samples do not necessarily result in anatomically valid hand configurations. Because of this, we then perform local sampling and belief propagation several times. Within this belief propagation, anatomical constraints are enforced in the same way as during tracking.

5.6 Results

To evaluate our method, we have tracked the hand of seven different persons (one female) grasping, lifting, and putting down three different kinds of cups (Figure 5.7). Based on the criterion regarding the temporal correlation of hand and object velocity (see Section 5.3 and Figure 5.8), we selected those frames from the 21 sequences in which the hand firmly grasps the handle of the respective cup.

(a) Cup 1 *(b)* Cup 2 *(c)* Cup 3

***Figure 5.7**: Partial meshes of three different cups. The meshes were created by in-hand scanning.*

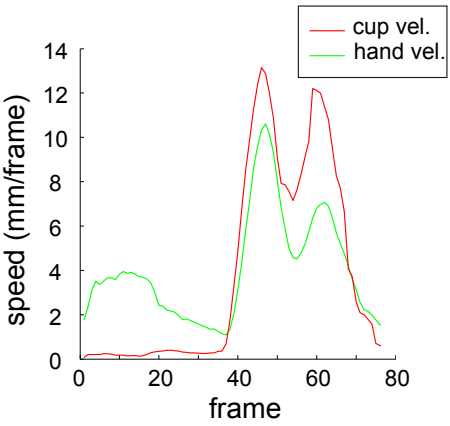


***Figure 5.8**: Speed of a cup and the manipulating hand. For temporal segmentation, we select the frames from about 40 to 80.*

In the following we present two types of results to address the two application examples of our method: improved hand tracking and grasp synthesis.

5.6.1 Improved Hand Tracking

We first elaborate on results showing that a prior observed for one hand and one cup can improve the tracking of a different hand manipulating a different cup. Figure 5.9 contains four frames of one of the 21 sequences. The handled object is cup 1 and the hand is the one of test person 2 (with a rather large hand). The prior we used in this experiment was observed on cup 3 with the manipulating hand belonging to test person 5 (an average sized hand). Without the labeling described in Section 5.3, tracking of the sequence fails due to ambiguity of the observation (see Figure 5.10(b)). The pose of the distal phalanx of the middle finger significantly differs from the labeled ground truth and the data. The red curve in Figure 5.11 documents this. When we introduce the prior (Fig 5.10(a)) the middle finger remains in place (Figure 5.10(c)) and we can track the sequence without any labels. The reduced error curve is also plotted in Figure 5.11. Tracking with the prior and the hardware indicated above takes less than 10 seconds per frame.

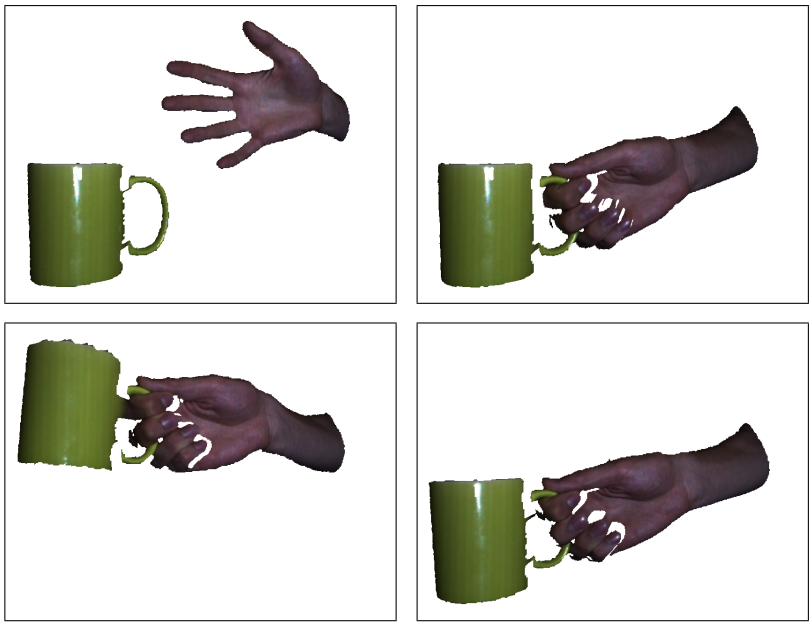
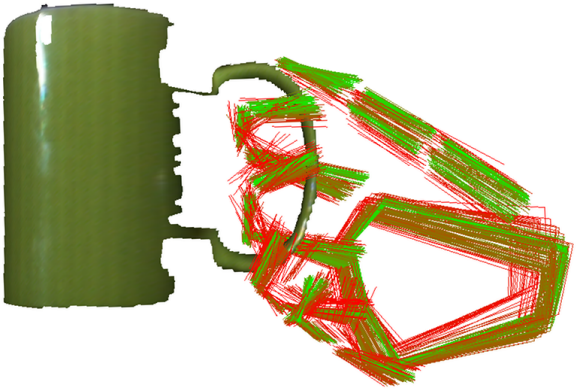
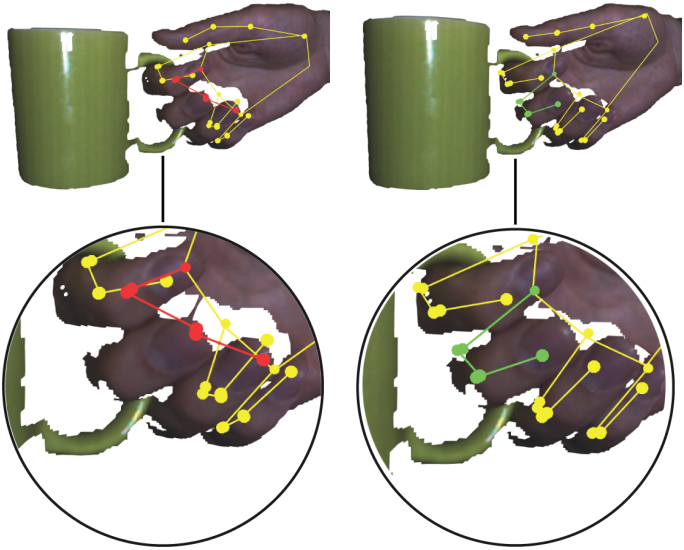


Figure 5.9: *Four frames of the sequence showing person 2 grasping and lifting cup 1.*



(a) prior



(b) failure *(c)* correct

Figure 5.10: *(a) The prior obtained from cup 3 and person 5. (b) While the cup is lifted, tracking fails due to ambiguities in the observation: the distal phalanx of the middle finger looses track. (c) The same frame as shown in (b), successfully tracked due to the prior, which stabilizes the tips within the handle.*

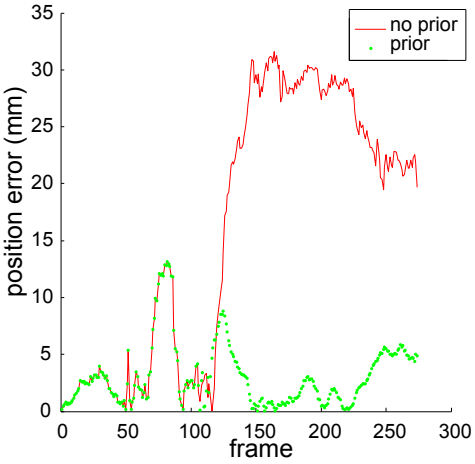
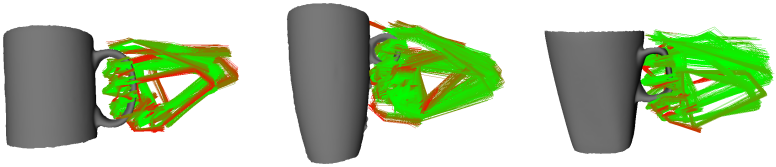


Figure 5.11: *Tracking error of the distal phalanx of the middle finger. Without the prior obtained on the basis of a different cup and a different anatomy the error is significant. With the prior, the hand segment remains in place.*

5.6.2 Grasp Synthesis

In Section 5.5 we introduced three factors influencing the probability of a sample: consistency with the prior, contact point attraction, and intersection constraints. The prior is visualized in Figure 5.12 (a)-(c) for the different cup types. In these figures, the information of all seven tracked sequences is contained. With a 3GHz CPU and a GeForce 8800 Ultra, it takes $\approx$ 35 seconds to obtain each prior: 25 seconds to load in the database and to adapt hand anatomy, 10 seconds to transfer the prior to the target cup by warping.

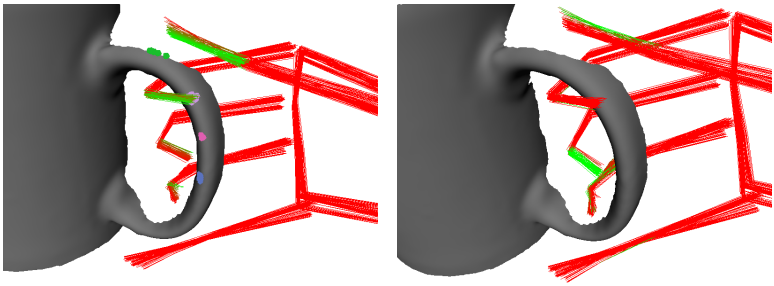


(a) Cup 1 *(b)* Cup 2 *(c)* Cup 3

***Figure 5.12**: Prior for the three cup types. All seven test persons contribute to each prior. The variety of grasping is largest for cup 1 (two or three fingers in the handle), less for cup 3 (mostly two fingers in the handle) and least for cup 2 (anatomically, only one finger fits into the handle). The color of each sample encodes the probability with respect to the density defined by the prior itself (Eq. (5.6)). Red stands for a low and green for a high probability.*

Figure 5.13 gives examples with regard to the contact and intersection probabilities. In Figure 5.14 and Figure 5.15 we show the process of synthesizing a grasp for a given cup and for a hand with a given anatomy (in this case the anatomy of the synthetic hand introduced in Section 3.5.2). The first image in each figure shows the cup on which a grasp was actually observed. Contact points of the individual fingers are indicated as colored dots on the mesh. The rest of the figures illustrate the collected frames (adapted to the required hand anatomy), the transformed prior, and the synthesized grasp (once rendered to visualize the 6d hand segment space and once using the synthetic hand to allow for a better intuition of the results). Grasp synthesis based on a prior requires $\approx$ 30 seconds.

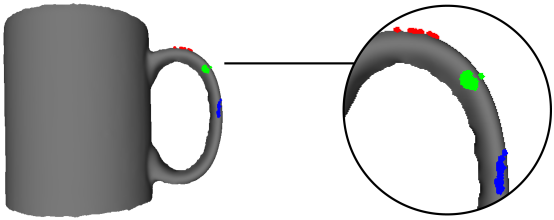
Figure 5.16 shows the warp of a prior consisting of only six of the seven sequences from cup 3 to cup 2. The figure demonstrates well the nature of our



(a) Contact point attraction *(b)* Intersection avoidance

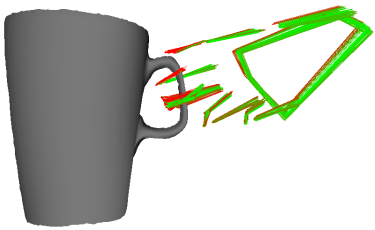
***Figure 5.13**: Probability terms favoring contact and avoiding object intersection. Colors are normalized from red (low probability) to green (high probability), therefore all samples of hand segments without contact/intersection are drawn in red.*

data-driven system: in all six sequences the test persons grasped the handle with two fingers. As a result, the hand synthesized for cup 3 exposes strong self-intersection of two fingers in the tiny handle (violating intersection constraints). However, the situation is resolved correctly by the system as soon as the seventh sequence is added to the prior (Figure 5.17).

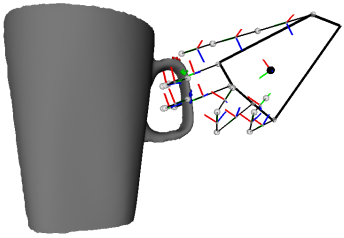


(a) Contact

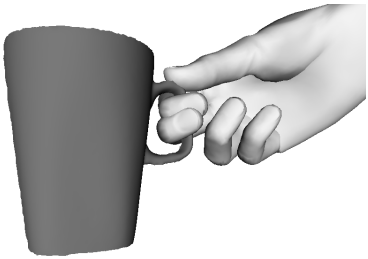
(b) Prior



(c) Warped prior

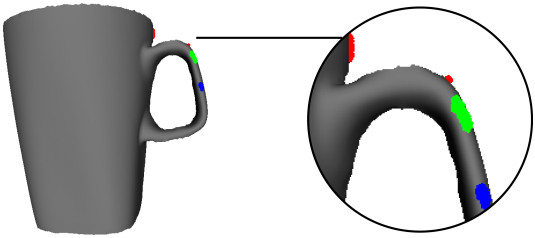


(d) Synthesized grasp

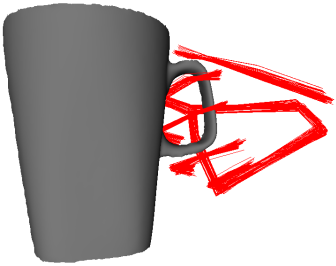


(e) Synthetic hand

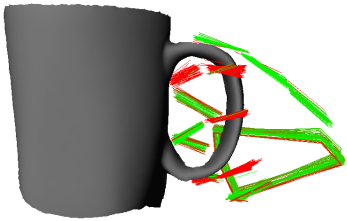
Figure 5.14*: Synthesized grasp, derived on the basis of the observation of only one person grasping a different cup. Grasp of cup 3, using the observation of person 2 grasping cup 1. (a) Originally observed contact points. (b) The derived prior. (c) The transferred prior. (d) The selected grasp (visualizing the 6d hand segment space). (e) The selected grasp (rendered using the synthetic hand).*



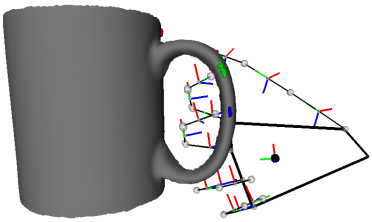
(a) Contact



(b) Prior



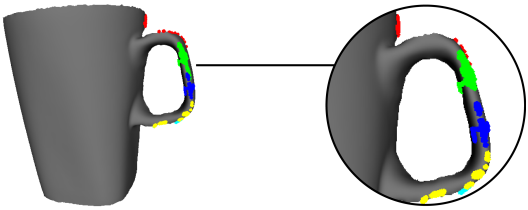
(c) Warped prior



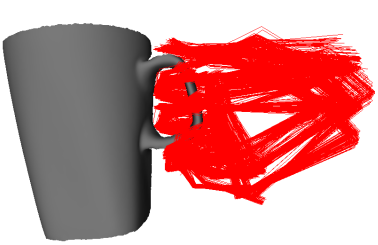
(d) Synthesized grasp

(e) Synthetic hand

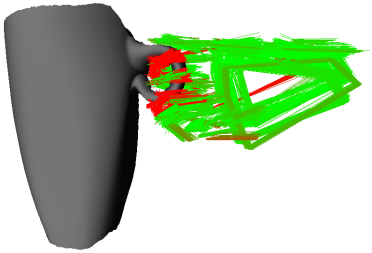
Figure 5.15*: Synthesized grasp, derived on the basis of the observation of only one person grasping a different cup. Grasp of cup 1, using the observation of person 1 grasping cup 3. (a) Originally observed contact points. (b) The derived prior. (c) The transferred prior. (d) The selected grasp (visualizing the 6d hand segment space). (e) The selected grasp (rendered using the synthetic hand).*



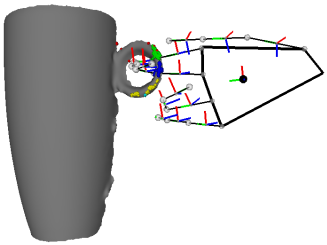
(a) Contact



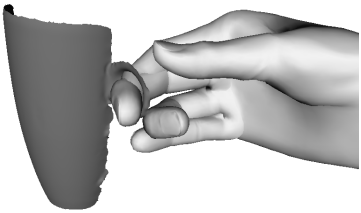
(b) Prior



(c) Warped prior

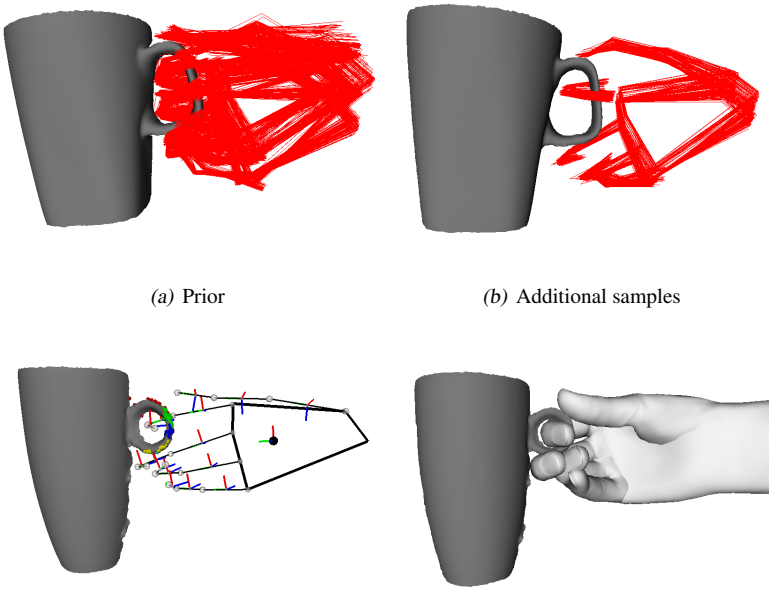


(d) Synthesized grasp



(e) Bad grasp

Figure 5.16*: Erroneous grasp synthesized for cup 2 using only six of the seven sequences observed on cup 3. (a) Originally observed contact points. (b) A prior considering only six of the seven sequences. All six test persons grasped the handle with two fingers. (c) The warped prior. (d,e) The selected grasp. Intersection constraints are strongly violated since there is not enough room for two fingers in the small handle.*



(a) Prior *(b)* Additional samples

(c) Final grasp *(d)* Result

Figure 5.17*: Grasp synthesized for cup 2 using all seven sequences observed on cup 3. (a) The prior considering only six of the seven test persons. All six persons put two fingers into the handle. (b) Hand poses from the seventh sequence used to augment the prior. The observed person has a rather large hand and hence used only one finger to grasp the handle. (c,d) The selected grasp.*

5.7 Conclusion

In this chapter we have presented an object-dependent hand pose prior, which is useful even when derived from sparse training data. A key feature of the prior is that it can be generalized towards new hands and new objects. Therefore it has high relevance for many interesting applications.

After repeating some key properties of the employed hand model we first introduced the prior model. In analogy to the hand model, the prior is defined by a product of local hand segment distributions. These distributions are learned from training sequences and with a Parzen-Rosenblatt estimator.

Next we discussed the raw data that is necessary to produce a prior and how we acquire this data with our structured-light system. Required information includes hand poses, object shape, object poses, and temporal segmentation.

Focus then lay on the transfer of the prior from observed hands and objects to hands with unseen anatomies and to novel objects of the same class. The transfer involves 1) hand retargeting, which maps observed hand segment samples to new hand anatomies, and 2) object warping, mapping the samples in accordance with the respective geometry warp from one cup to another.

The next section was dedicated to the integration of the prior into the belief propagation framework presented in Chapter 4. The prior is introduced via the data term of each node in the hand graph, together with contact point attraction and intersection constraints. As a result we obtain a unified framework for both hand tracking and grasp synthesis.

The result section finally provided experiments with respect to the two applications we targeted: improved hand tracking and grasp synthesis. 21 training sequences containing 7 different hands manipulating 3 different cups served as input for qualitative and quantitative evaluation.

There are some limitations with respect to the geometry warping. Firstly, we stated that we currently define correspondences between the initial and the target mesh manually, although there may be methods to do this automatically. Another issue is the following: if the geometry of the two meshes is too different (like the big handle of cup 1 and the tiny one of cup 3) the warp field has strong discontinuities. This is reflected in the extrapolated warp field and warped hands strongly violating anatomical constraints.

In contrast to Chapter 4, we have dealt here with true hand-object interaction by modeling implications of a manipulated object for the manipulating hand. However, regarding synthesis we have somehow neglected the inherent temporal nature of object manipulation. While grasps are an essential intermediary step, it is the change of hand states that causes changes in the object state. In the next chapter we make an effort to synthesize computer animations containing not only static grasps but also manipulations of non-rigid objects.

6

Data-Driven Animation of Hand-Object Interactions

Like Chapter 5, this Chapter is concerned with the synthesis of manipulating hand poses. However, in Chapter 5 temporal considerations were limited to the segmentation of observed actions. In contrast, we now introduce the temporal component into synthesis, and extend static grasp synthesis to dynamic hand motion synthesis. In particular, we also account for possible temporal offsets between manipulating hand poses and the effect on the object. Another extension to the previously presented methods is that we treat not only rigid objects but also *articulated objects*, i.e., objects consisting of more than one connected rigid parts.

Our goal is to generate animated sequences containing hand-object interaction, solely by animating the involved object. Animating the many DOFs of a manipulating hand manually is the more difficult part in the context of hand-object interaction. However, animating an object can easily be done by a 3d artist with the aid of 3d modeling software, if there are only a few DOFs to control or if the DOFs are largely independent. Based on such an object animation and causalities derived from observations of real object manipulations, the corresponding hand motion can be synthesized automatically. The described method was accepted for publication at the *IEEE International Conference on Automatic Face and Gesture Recognition 2011* [Hamer *et al.* in press].

6.1 Overview

When humans interact with objects, hand and object motions are strongly correlated. Moreover, a hand manipulates an object usually with a purpose, chang-

***Figure 6.1**: Clamshell phone. The phone has to be opened before a number can be dialed.*

ing the state of the object. Vice versa, an object has certain affordances [Gibson 1979], i.e., it suggests a certain functionality. Consider the clamshell phone in Figure 6.1 as an introductory example. Physical forces are applied to pick up such a phone and to open it. Once the phone is opened, the keys with the digits suggest dialing a number.

The affordances of an object have the potential to ease hand animation in the context of hand-object interaction, e.g., given the clamshell phone and a number to dial, the necessary hand motions to make a call can be synthesized. This is particularly interesting when the object has fewer degrees of freedom (DOFs) than the hand (e.g., opening the phone requires just a one-dimensional rotation) or when the DOFs are largely independent (like in the case of the separate digits of the phone). Animating such an object is easier for an artist than animating the hand or both. Ideally, simple scripting of object state changes infers a complete hand animation to carry out these changes.

Inspired by these considerations, we present a method to animate a manipulating hand conditioned on an animation of the manipulated object. The approach is data-driven, so we require that the object has previously been observed during manipulation. This training phase involves a semi-automatic acquisition of hand poses and object poses from the structured-light data. The pose of an object always comprises its translation and rotation. In case of articulated objects, the object's pose also includes information regarding the arrangement of its parts. Based on the tracked hand and the tracked object, we infer 1) the various states of the object during manipulation, 2) the hand configurations that cause object state transitions, and 3) the spatio-temporal correlations between key hand poses and key object poses. For instance, the state of the clamshell

phone can be either closed or open and a specific temporal hand movement is required for opening and closing. Data acquisition and training is required only once for a new object.

For animation, the object pose and contact points optionally created by the artist are used to generate hand poses for key frames. The hand pose transitions that have been observed during training then form the basis for hand pose interpolation to obtain a plausible hand-object animation. With this technique an artist can quickly produce a great variety of different animations without the need of acquiring new data.

Many approaches in computer graphics are concerned with realistic hand animations. For example, in [Albrecht *et al.* 2003] an anatomically-based model is animated by means of muscle contraction. However, there has been less work with respect to hand-object interaction. Some approaches address the synthesis of realistic static grasps on objects [Li *et al.* 2007] or grasp-related hand motion [Pollard and Zordan 2005, Kry and Pai 2006, Liu 2008, Liu 2009]. Li et al. [Li *et al.* 2007] treat grasp synthesis as a 3d shape matching problem: grasp candidates are selected from a large database by matching contact points and surface normals of hands and objects. Pollard and Zordan [Pollard and Zordan 2005] propose a grasp controller for a physically-based simulation system. To obtain realistic behavior, the parameters of the controller are estimated from motion sequences captured with markers. A similar method is used by Kry and Pai [Kry and Pai 2006] where hand motion and contact forces are captured to estimate joint compliances. Recently, Liu [Liu 2008, Liu 2009] formulated the synthesis of hand manipulations as an optimization problem where an initial grasping pose and the motion of the object are given. Besides grasping motions, hand motions for musical instruments have also been modeled [Kim *et al.* 2000, ElKoura and Singh 2003]. In these works, a hand plays a specific musical instrument, e.g., violin or guitar.

We now classify our approach and at the same time point out differences to the other works.

1. Our approach is data-driven as we exploit observations of real manipulations to ease the synthesis of new animations. This is a common strategy with regard to the animation of manipulating hand motion, since manual modeling of hand-object interaction does not achieve realistic results. However, in contrast to our method most data-driven systems use in-

vasive techniques like markers or gloves [Li *et al.* 2007, Pollard and Zordan 2005, Kry and Pai 2006].

2. We consider not only grasping but also manipulations where contact points change dramatically during hand-object interaction. Works like [Kim *et al.* 2000, ElKoura and Singh 2003] in which musical instruments are played are other notable exceptions.

3. The hand is controlled by the state of the manipulated object. In [Liu 2008, Liu 2009] a hand is also controlled by means of the manipulated object, but their objects are not articulated and typically only grasped. Moreover, an initial grasp has to be defined which is not necessary with our method. In [Kim *et al.* 2000, ElKoura and Singh 2003], a hand plays violin or guitar. The hand is somehow controlled by the object (a certain musical score is requested), but in those works the object state does not involve a significant geometric deformation of the object. [Pollard and Zordan 2005] also do not deal with articulated objects, and the hand state is determined by a learned grasp controller and not by a manipulated object.

6.2 Learning by Human Demonstration

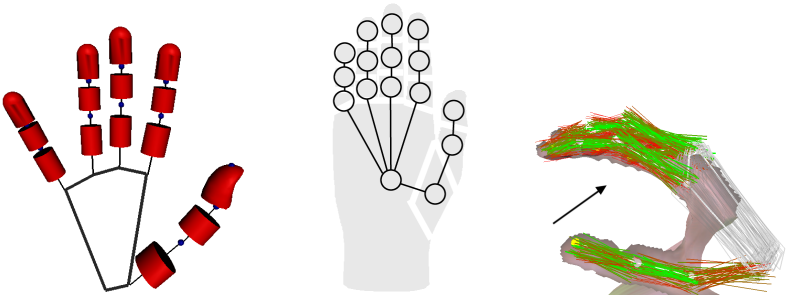
Our goal is to generate animations of a hand manipulating an object by animating the object only. To this end, we fuse several types of information. On the one side, there is the object animation created for example in Maya. On the other side, we use information regarding the manipulation of the respective object (e.g., hand poses in relation to the object, possible articulations of the object, and timing information). The latter is obtained from human demonstration.

6.2.1 Capturing Object Manipulation

Again, all our observations are retrieved by the structured-light setup. We observe the manipulation of a specific object by a human hand and gather information regarding a) the fully articulated hand pose and b) the object's surface geometry and the object pose. The pose of a rigid object is 6 dimensional (translation + rotation). In case of an articulated object there are additional

DOFs. For example, the clamshell phone has one extra DOF representing the angle between the main body and the display.

Hand poses are captured as described before (see Chapter 4) with the hand model consisting of local parts (Figure 6.2(a)) connected in a hand graph (Figure 6.2(b)). Samples are drawn (Figure 6.2(c)) and belief propagation is performed to find the best estimate.

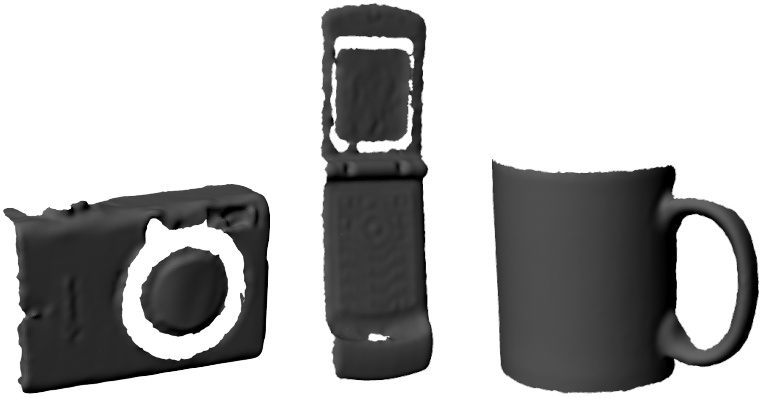


(a) Hand model *(b)* Graph *(c)* Local samples

***Figure 6.2**: Hand tracking. (a) Hand model with a skeleton and ruled surfaces for the skin. (b) Graphical model for inference. (c) Depth data and hand segment samples. Color encodes relative observation likelihood: green is highest, red is lowest. The palm has uniform observation likelihood. An arrow indicates the viewing direction of the camera.*

In analogy with Chapter 5 we perform in-hand scanning (Section 3.4) to obtain partial meshes of objects (see examples in Figure 6.3). With the partial mesh of an object available, we determine in an offline process the object's 6d pose (translation and orientation) for each frame of a sequence containing the object and some manipulation. For this, we again use ICP to fit the respective mesh to the observation.

For articulated objects we produce a separate mesh for each extreme articulation. In the example of the clamshell phone one mesh represents the *phone closed* state and a second one the *phone open* state. We then fit the respective mesh to the data, depending on the object's state. However, this leaves us without a registration during object state transitions from one state to the other.



(a) Camera *(b)* Clamshell phone *(c)* Cup

***Figure 6.3**: Partial meshes of a camera with a zoom, a clamshell phone, and a cup. The meshes were created by integrating several range scans.*

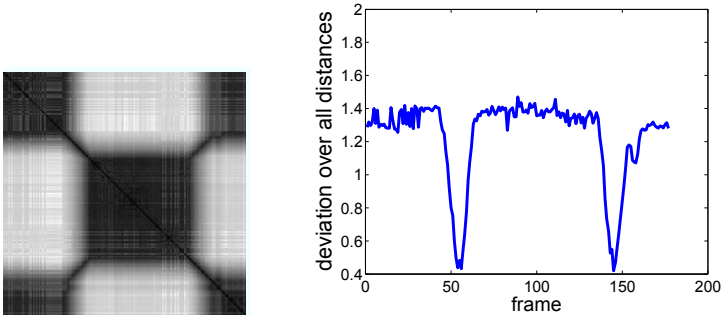
6.2.2 Identifying Articulated Object States

There is a strong dependency between the state of an articulated object and its usage. We will consider the camera and clamshell phone examples of Figure 6.3. A closed clamshell phone is treated differently than an open one. Identifying the articulated states of an object manipulated in front of the structured-light setup is key to extracting manipulation knowledge. We approach the issue with a distance matrix for all frames of an observed sequence. To measure the distance between two range scans S_1 and S_2, we first remove all 3d points that have skin color. For each remaining point p of scan S_1, the closest point q_p in S_2 is found after global ICP alignment. To obtain a symmetric measure, we compute the smallest distances in both directions and take the sum as the distance:

$$d(S_1, S_2) = \sum_{p \in S_1} \|p - q_p\| + \sum_{q \in S_2} \|q - p_q\|. \tag{6.1}$$

Figure 6.4(a) shows the distance matrix for a sequence of 177 frames in which the camera is manipulated. The lens of the camera first emerges and then returns to the original position. The two different states - lens completely moved in or out - are visible. To obtain a significant measure for frame segmentation, we compute the standard deviation for each column of the distance matrix (Fig-

ure 6.4(b)). High values indicate frames in which the object is in one of the two binary states.



(a) Distance matrix *(b)* Standard deviation

Figure 6.4*: Detecting object states in observed data. (a) Distance matrix for a sequence of 177 frames in which the camera is manipulated. Dark means similar. (b) Standard deviation of the columns of the distance matrix.*

6.2.3 Transition Intervals of Object and Hand

A manipulating hand is typically most active when it causes the object to pass from one state to another (object state transition). In order to find the hand poses that produce a certain object transition, we look for corresponding hand transition intervals. In the easiest case, hand transition intervals are temporally identical with object transition intervals. This is usually the case when the object is physically forced into the new state, e.g., the clamshell phone is opened by a push. However, hand transition intervals can also differ temporally from the object transitions.

Figure 6.5 shows three frames of the camera sequence analyzed in Section 6.2.2. The tracked hand pushes an activation button on the camera, and thereby causes the first object state transition visible in Figure 6.4(b). All three frames are relevant and should be reflected in the animation. The camera has a time delay, and by the time the lens state changes the finger already starts to move upwards again.

More generally speaking, hand motion performed for object manipulation can be approximated by a sequence of characteristic key poses, each with some

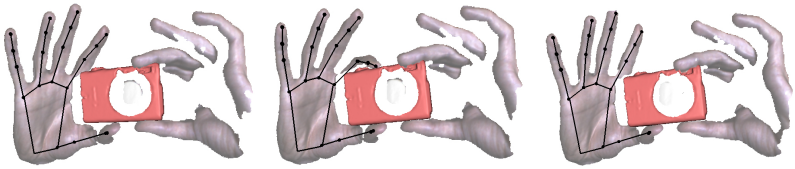


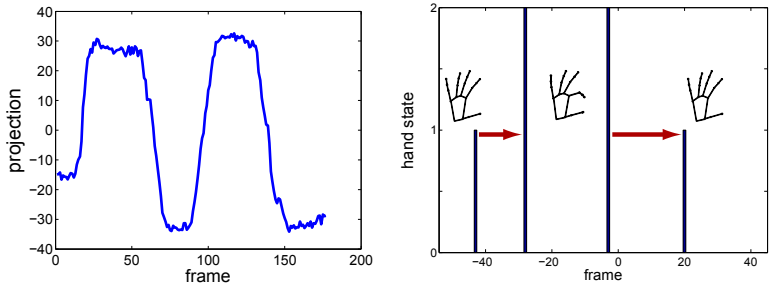
***Figure 6.5**: Three frames showing an observed hand that pushes an activation button on the camera. The black stick-model skeleton illustrates the estimated hand pose. The registered mesh of the camera is drawn in red. In this case, we excluded the lens so that the same mesh can be registered throughout the complete sequence.*

temporal offset with respect to the object state transition. We assume that significant hand poses are those surrounding intervals of rapid change in hand state space (excluding wrist translation and rotation). To reduce noise from this high dimensional state space, we apply principal component analysis.

Figure 6.6(a) shows the projection of the hand poses of the camera sequence to the first principal component. The two relevant hand states are visible at -30 and 30. The figure can be interpreted as follows: the index finger of the manipulating (right) hand is extended in the beginning of the sequence. It then approaches the activation button of the camera, presses the button, and then raises again. This causes the lens of the camera to emerge (zoom). This hand motion is shortly after repeated, this time with the purpose to make the lens go back. Figure 6.6(b) focuses on frames 0 to 100 of the sequence and the first object state transition. The beginning and end of each transition interval of the hand are expressed relative to the middle of the object state transition, i.e., the lens is in the middle of emersion (Figure 6.4(b)). Finally, the tracked sequence is divided into a series of hand transition intervals indicated by the arrows in Figure 6.6(b).

6.3 Animation Framework

Figure 6.7 gives an overview of our method. The previous section shows how to acquire and process training examples (left of Figure 6.7 - training). We now describe how to create a new animation. First, the artist chooses a hand to be



(a) Hand PCA *(b)* Hand Transitions

***Figure 6.6**: (a) The two states of the hand are indicated by the values* -30 *(index finger extended) and* 30 *(index finger flexed). The sequence starts with the extended index finger (frame 0). Around frame 20, the finger flexes to press the activation button on the camera, causing the lens to emerge. After frame 50, the index finger begins to extend again. The same hand motion is repeated, starting near frame 90, to make the lens go back again. (b) The beginning and end of each transition interval of the hand are expressed relative to the middle of the object state transition, i.e., the lens is in the middle of emersion. Red arrows indicate the transition from extended to flexed index finger and vice versa.*

animated, and *hand retargeting* is performed. Then the artist defines an object animation (right of Figure 6.7 - animation). Finally, the training information and the artist's input are combined to generate a new animation (bottom of Figure 6.7).

6.3.1 Hand Retargeting

All hand poses estimated from real data exhibit the anatomical dimensions of the demonstrating hand. Since we seek an animation of the synthetic hand (Section 3.5) we adapt the hand anatomy as described in Section 5.4.1, preserving finger tip positions. Further we convert the resulting hand poses from the state space of the tracking model to the state space of the synthetic hand. On the one side, we discard some information (e.g., little gaps between fingers possible due to soft constraints). On the other side we sometimes violate the

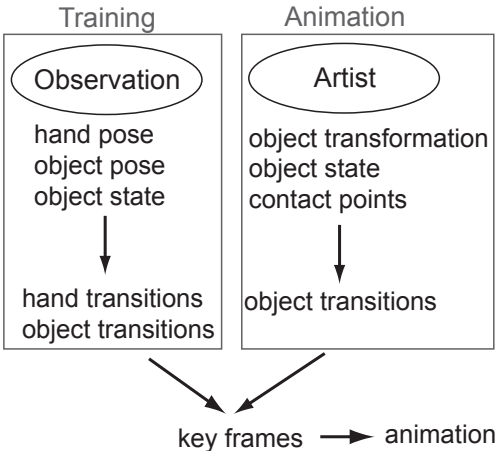


***Figure 6.7**: Animation procedure. The observations are processed only once for training (left). A new object animation can be created in Maya where the object transitions are automatically extracted (right) to obtain key frames. The training data is then used to generate a corresponding hand animation.*

anatomical constraints of the synthetic hand to allow for a wider range of hand poses.

6.3.2 Object Animation

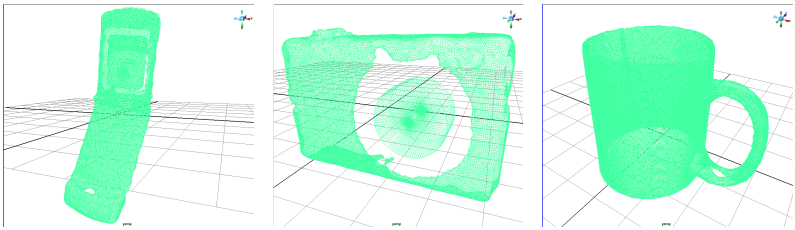


***Figure 6.8**: Rough object models created in Maya on the basis of the partial meshes. The clamshell phone contains a joint controlling the angle between main body and display. For the camera a cylinder was added to represent the lens. The mesh of the cup was created by mirroring and is almost closed.*

Based on partial meshes created by integrating several range scans (Figure 6.3), we created three Maya models (Figure 6.8). In the case of the clamshell phone, a joint was added to enable the animation of the opening and closing process. For the camera, a polygonal cylinder represents the lens. As input to our system, the artist creates an animation of the object, varying translation, rotation, and the object's articulation over time. Articulation is represented by continuous parameters, e.g., the translation of the lens of the camera or the angle of the joint of the clamshell phone. In addition, the artist can optionally specify contact points between the hand and the model in desired key frames, e.g., when the animated hand should dial a specific digit.

6.3.3 Combining Information

At this point, the information from the training data and the artist can be combined. Contact points defined by the artist are used to compute hand key poses. These key poses are derived taking into consideration the desired contact points, as well as all hand poses observed for a certain articulated pose of the object. Figure 6.9 shows all hand poses of a training sequence observed while the clamshell phone is open.

In more detail, we seek the hand pose which is 1) close to the observed hand poses, 2) realizes the contact best, and 3) does not intersect with the object's geometry. We perform inference by running belief propagation on the hand graph. Note that this is the very procedure described in Section 5.5, i.e., the observed hand poses form an object-dependent hand pose prior.

Other key frames result from the defined object state transitions (Section 6.3.2). Their midpoints determine the timing of the corresponding hand pose transitions observed in Section 6.2.3. Hand pose interpolation between key frames of the hand is performed as follows:

- If the animator wants to pause in a certain object state this leads to a simple freeze.
- Between key frames specified via contact points, a linear interpolation regarding the joint angles of the animated hand is applied. The time warping is non-linear and reflects the assumption that human hands at first approach their targets fast but slow down in the end [Rao *et al.* 2002]. We transfer this observation to individual hand segments. The

***Figure 6.9**: Hand poses observed in a training sequence while the clamshell phone is opened. All poses are expressed in local object coordinates.*

duration of the transition is normalized to $t = [0, 1]$. The angle vector θ contains three angles with respect to the rotation of a certain joint and is defined by

$$\theta_t = \theta_{t=0} + \sqrt{t} \cdot (\theta_{t=1} - \theta_{t=0}). \tag{6.2}$$

The square root of t causes a decrease of the speed as t approaches 1.

- For hand transitions between key frames caused by object state transitions, we follow a two-stage procedure. Initially, we temporally scale the observed hand transition, to synchronize it with the artist's prescription. However, this is more or less a replay and does not suffice. Observed transitions are characterized by a key frame at their start and end. An ending key frame and the subsequent starting key frame may be quite different, hence, the final hand animation has to blend out such abrupt changes. We formulate this as an optimization problem that strikes a bal-

ances between staying close to the observed transitions, while producing good blends between their boundaries:

$$\underset{d\Theta_t}{\operatorname{argmin}} \sum_t \|d\Theta_t - d\tilde{\Theta}_t\|^2 + \alpha \cdot \|\Theta_0 + \sum_t d\Theta_t - \Theta_1\|^2.$$

A transition is split into increments $d\Theta_t$, and $d\tilde{\Theta}_t$ represents the corresponding increments of the stretched replay. Hence, the first term enforces compliance with the replay. The second term ensures the blending. Θ_0 and Θ_1 are the joint angles at the start of two subsequent transitions. α is a user parameter and controls the trade-off between compliance with the stretched replay and smooth blending. In our experiments we set α to 10.

6.4 Results

We now present results of the proposed method with respect to the three objects introduced earlier: the camera, the cup, and the clamshell phone. We also discuss the additional example of a mortar and the appendant pestle. Tracking is required only once for training. The artist can then create animated sequences by only defining the (articulated) state of the object. Our models are quite rough, but they suffice for illustration and could be replaced by high quality ones.

The example of the mortar and the pestle is the most basic one, but illustrates well how animated sequences can clarify the intended usage of tools. The animation depicted in Figure 6.10 (top) is based on a single observed frame showing a hand holding the pestle (see Figure 6.10 (bottom,left)). The estimated hand pose in that frame is expressed in the coordinate system of the pestle, and the crushing movement of the pestle was defined in Maya. The mortar itself plays only a passive role.

The example of the camera (Figure 6.11) is more advanced because the lens can be in or out, and temporal dependencies have to be considered: the index finger approaches the button and starts to flex again *before* the lens comes out. In the tracked sequence (top row, left), the demonstrator presses a button on the camera twice, causing the lens of the camera to emerge and then to retract again. In the object animation created in Maya, the zoom emerges and retracts

twice, triggering the respective hand movements to create the final animation (two cycles of the bottom row).

The case of the cup is a little different. Since the cup consists of a single rigid body, the artist can only animate its translation and rotation in Maya. However, to model the grasping process, we augment the cup's state space with a binary flag indicating whether the animated cup is moving or not. When it does move, a firm grasp of the hand on the handle must already be established, to suggest it is the hand moving the cup. Consequently, the process of grasping must be initiated *before* the artist wants to change the position of the cup. This temporal offset, the key hand poses, and the hand pose transitions between key poses are again obtained from the observation. Figure 6.12 is dedicated to the cup example. In the tracked sequence (top row), the cup is grasped, lifted, put down, and released. In contrast, in the animation (middle row), the cup is not only lifted but also poured out. Two close-ups (bottom rows) illustrate this difference. The cup model was created by mirroring the corresponding mesh and only has a few holes.

Finally, we come to the clamshell phone. The artist controls its translation and rotation, as well as the articulated state (phone closed or open). In addition, object contact can be enforced in desired frames in order to let the animated hand dial an arbitrary number. The tracked sequence is shown in the top row of Figure 6.13. To track the object, we registered the respective mesh (phone closed or open) with the data. The tracked hand initially holds the closed phone. The phone is then opened and the digits from one to nine are dialed in order. Thereafter the phone is closed again. In the animation (middle row), the phone is first picked up. This results from a simple rigid transformation of the phone in its closed state. Then, the phone is swung open. In this case the timing of the animation is different than that of the observed demonstration, so the observed hand pose transition has to be stretched. While the phone is open, the animated hand dials a number defined by the artist. Finally, the phone is closed again, and a rigid transformation is applied to lay the phone down. Some texture information was added to the model in Maya. Close-ups are provided in the bottom row.

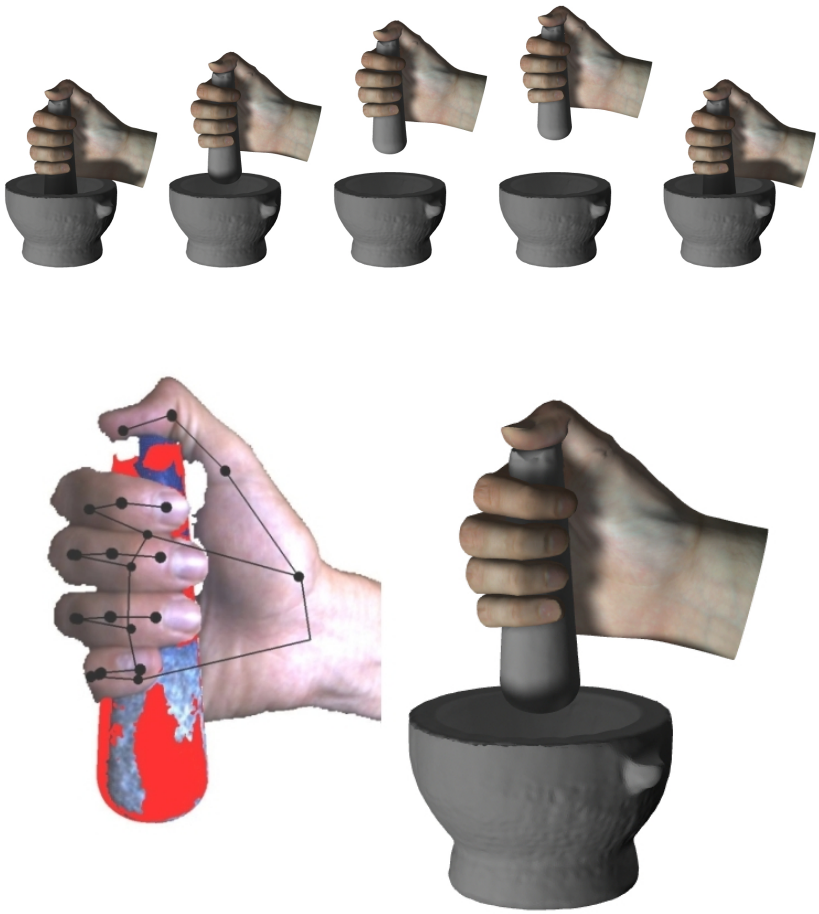


Figure 6.10: *Generating a sequence with a mortar and a pestle used for crushing. The animation (top) is based on a single observed frame showing a hand holding the pestle (bottom,left). The estimated hand pose in that frame is expressed in the coordinate system of the pestle, and the crushing movement of the pestle was defined in Maya. (bottom,right) Close-up of one of the animated frames.*

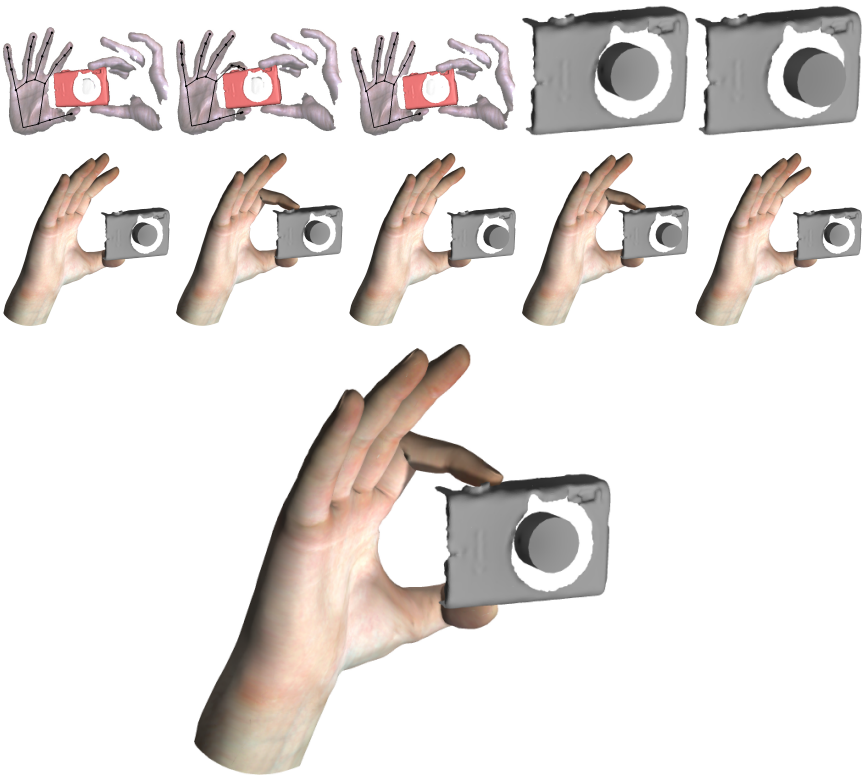


Figure 6.11: *Generating a sequence involving manipulation of the camera. (top,left) Three frames of an observed sequence in which the hand and the camera was tracked. The estimated hand pose is indicated by the black stick-model skeleton, the partial mesh of the camera registered with the data is drawn in red. In the observed sequence, the lens of the camera emerges and goes back once. (top,right) Close-up of the rendered model of the camera, once with retracted lens and once with emerged lens. (middle) Frames of the animated sequence. In the complete sequence, the zoom emerges and retracts twice, triggering the respective hand motions with the temporal offsets observed in real data. (bottom) Close-up of one of the animated frames.*

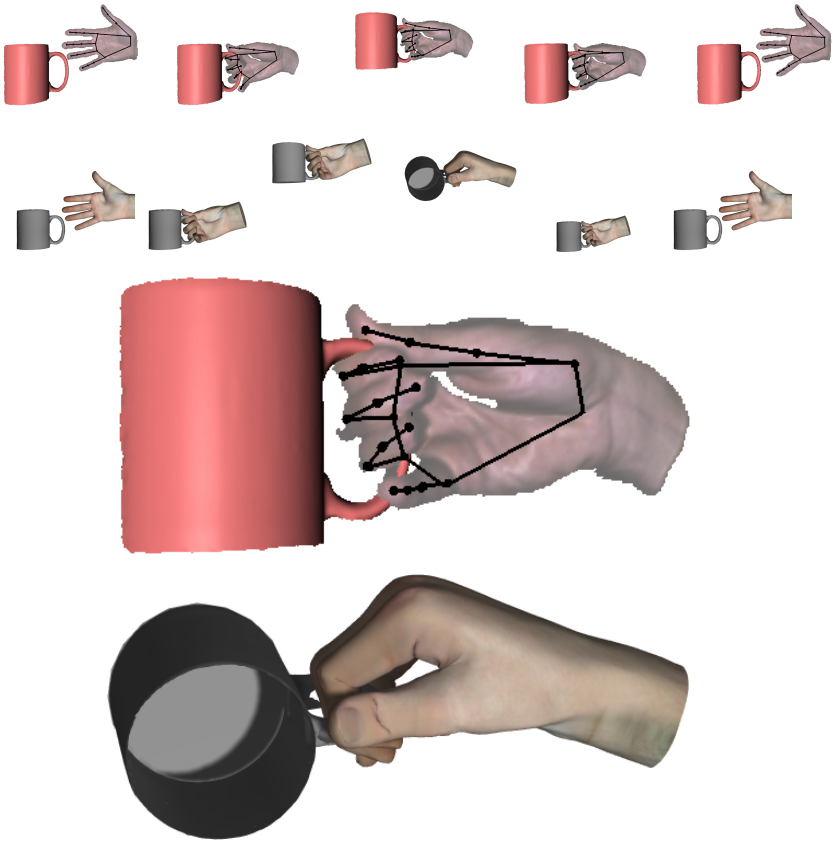


Figure 6.12: *Generating a sequence involving manipulation of the cup. (top) The tracked sequence. Hand poses are drawn in black, the registered mesh of the cup in red. The cup is grasped, lifted up, put down, and released. No pouring is demonstrated. (middle) An animated sequence in which the cup is not only lifted but also poured. The movement of the cup and the pouring together with the corresponding hand motion results from the object animation in Maya. (third and fourth row) Close-up of one tracked and one animated frame.*

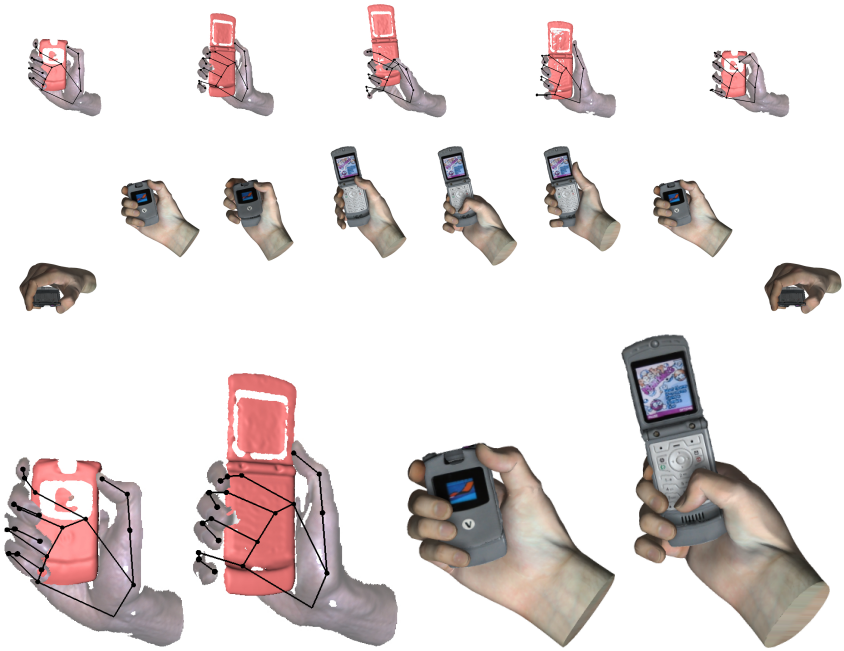


Figure 6.13*: Generating a sequence involving the clamshell phone. (top) The tracked sequence. Hand poses are drawn in black, the registered mesh of the phone in red. The phone is opened, the digit from 1..9,0 are dialed in order, and the phone is closed again. (middle) In the animated sequence the phone is first picked up (which was never observed) and then opened. The thumb movement during opening is interpolated based on the observation, resulting in a kind of flicking motion. After opening the phone, the animation artist can dial an arbitrary number via the definition of contact points. The interpolation between dialing poses is fast in the beginning and slower in the end, to create a realistic impression. Finally, the phone is closed and put down. (bottom) Close-up of some frames.*

6.5 Conclusion

We presented in this Chapter a method to automatically synthesize the motion of a hand manipulating an object, given 1) observations of a human hand interacting with the object and 2) an animation of the manipulated object. Animating the hand directly is very hard because of the many DOFs. Instead, the artist has full control of the object for creating an initial object animation.

First, we introduced the concept of combining an object animation created in a 3d modeling software with manipulation knowledge derived from observation. On the observation part we referred to the mechanisms for hand and object tracking introduced earlier in this work. With respect to object tracking we discussed the extensions to support articulated objects.

Then we described how to identify discrete object states of observed, articulated objects. In more detail, a distance matrix referring to the individual frames of a manipulation sequence was introduced. Based on this matrix, a metric for frame segmentation with respect to object articulation was proposed.

The discrete object states lead naturally to object state transition intervals, i.e., intervals in which the object changes from one articulated state to another. We then show how we find associated hand transition intervals and explain that object and hand transition intervals are not always in direct temporal correspondence.

We further demonstrated how combining the 3d artist's object animation with observed hand and object state transition intervals delivers key frames for the hand in the output animation. Our data-driven approach to interpolate between those key frames includes an optimization step and we gave details on that matter.

In the result section we presented four animated sequences containing hand-object interaction that were produced by our method. The manipulation performed by the synthetic hand on each of the four objects differs significantly from the observed manipulation, which illustrates the freedom of the 3d artist.

Summarizing, we contributed to the state of the art in this field by a method that eases the generation process of animated sequences containing hand-object interaction. We can encode temporal aspect like the delay between a manipulating hand motion and the effect on the object. The involved objects may

optionally exhibit articulation. In addition to animating the object's pose, an animation artist can enforce contact points in desired key frames.

The most significant limitation of our system is that the observed and the animated objects are currently the same. In the future we imagine a system which allows the artist to load an arbitrary, unobserved object mesh. Based on object recognition techniques (maybe exploiting 3d features like spin images), a similar object for which interaction was observed, could be loaded from a data base. Regarding the transfer of manipulation knowledge from the observed object to the new one, we consider the warping techniques proposed in Chapter 5 as a first step.

7

Discussion

The content of this work shall now be recapitulated and discussed. We begin with a short summary of our contributions, and for each contribution we point out the difference to previous methods. Our work will then be considered on a more abstract level, and we will arrange it with respect to the diagram in Figure 7.1. Thereafter, we elaborate on issues not addressed in this work and highlight future opportunities. This will then lead us to the conclusion of this work.

7.1 Contributions

First, we introduced a marker-less method to track a hand which interacts with an object. For better robustness, each hand segment has its own local tracker. The local trackers are connected in a pairwise Markov random field which encodes the structure of the manipulating hand. In each frame, the best hand pose estimate is found by belief propagation. For tracking, we use not only color information but also range data delivered by a structured-light system. To this end we developed a special observation model, comparing local surface patches rendered on the GPU to the depth data. In contrast to previous hand pose estimation techniques, we handle explicitly not only severe self-occlusions but also occlusion of the hand by a manipulated object. The method was validated on artificial data with ground truth as well as on real data obtained by the structured-light setup.

Second, we encoded observed manipulation knowledge in an object-dependent hand pose prior. After tracking hands and objects, the estimated hand poses are expressed in local object coordinates. The prior is then modeled as a spatial distribution conditioned on the object. The probability of a hand pose can then

be computed with respect to the prior, contact point attraction, and intersection constraints. A key feature of the prior is that it can be generalized, i.e., transferred to other objects of the same object class and to new hands. We presented two applications of this: 1) improved hand tracking and 2) grasp synthesis for unobserved hands and objects. Compared to other approaches analyzing interdependencies during hand-object interaction, we address the relationships between the hand and the detailed shape of the manipulated object.

Third, a data-driven method to animate hand-object interaction was presented. The method takes as input an animation of an (articulated) object and generates the corresponding hand motion automatically. For this, training data of real object manipulation performed in front of the structured-light system is exploited: for a new object, we infer from training data 1) the various states of the object during manipulation, 2) the hand configurations that cause object state transitions, and 3) the spatio-temporal correlations between key object poses and key hand poses. Training is only required once. After that, an artist can quickly produce a great variety of different animations without the need of acquiring new data. Other approaches to animate hands conditioned on a manipulated object do not support articulated objects, significantly changing contact points, and temporal dependencies.

Having summarized the individual building blocks of this work, we will now consider them in the greater context of modeling hand-object interaction. Creating simplifying models for the subject of interest is a common procedure in science. Modeling hands and objects bears many challenges. However, the task is somehow concrete for such physical matters. In contrast, the correlations and interdependencies between a manipulating hand and the manipulated object are less clearly defined: a great variety of different factors have an impact on the scenario. Examples of such factors include the anatomy and strength of the hand, personal preferences of the acting human, shape and surface properties of the object, and physical quantities like gravity.

Despite the complexity of the setting we have suggested methods that address certain aspects of hand-object interaction. We will now arrange these methods with respect to the diagram in Figure 7.1. In the center of the diagram stands the model encoding hand-object interaction. Hand information is divided into the actual state of the hand (*ground truth*) and the hand observation. Object information is split likewise. In the following text we refer to the different arrows within the diagram by their number (1-8).

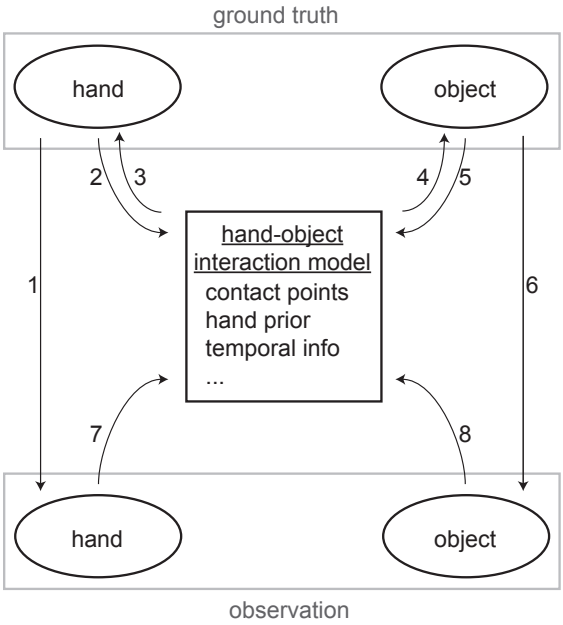


***Figure 7.1**: A diagram illustrating the interplay of our hand-object interaction model and the ground truth/observation of hands/objects. Numbers at the arrows serve for reference and imply no temporal sequence.*

The ground truth state of a hand in front of a 3d acquisition setup is reflected in the observation, i.e., the range data of the hand (1). Vice versa, the true hand pose can be inferred from observation by hand pose estimation (7,3). For this purpose, we have developed our hand tracking method in Chapter 4. Hand tracking delivers estimates of the ground truth hand poses, and these contribute to our model (2).

We track not only the hand but also the object. Like in case of the hand, the ground truth object generates the object observation (6), and tracking the object (8,4) augments our model of hand-object interaction (5). Since we obtain not only the pose of the object but also detailed shape information, it is now possible to compute hand-object contact points and to evaluate intersection constraints. To go one step further, we developed the object-dependent hand pose prior in Chapter 5 and presented two different applications. First, the prior

can improve hand tracking (3). Secondly, we use the prior to synthesize grasps for novel objects. i.e., based on a novel object (5) and hand-object interaction knowledge encoded in the model, ground truth hand poses are defined (3). For visualization, observations of the novel object and the synthesized grasp can be obtained by rendering (1,6), for example in Maya.

Grasp synthesis was extended to motion synthesis in Chapter 6. For training, we again acquired the poses of hands (1,7,3) and objects (6,8,4), but this time an object's pose could also comprise its articulation. Based on the estimated poses, additional types of information were introduced (2,5) into the model of hand-object-interaction: temporal correlations between key hand poses and key object poses. An object animation (5) of an artist can then be used to synthesize the motion of a hand (3). Again, rendering (1,6) is necessary to visualize the results.

7.2 Limitations and Future Work

While we contributed to the state of the art with respect to the analysis of hand-object interaction, there is still room for improvement, and we see many opportunities. In some cases, direct extensions of the developed methods suggest themselves. In other cases, our work led us to completely new ideas.

The development of the hand tracker described in Chapter 4 was an essential step for our research as it lay the foundation for all results presented in this work. Regarding visual hand pose estimation, we have the following suggestions for interesting enhancements.

- *Extension of the hand model with respect to the wrist and the palm.* Initially, our model consisting of local surface patches (see Figure 4.3(a)) ignored both the wrist and the palm. Later we added a simple sphere to represent the wrist. This already increased robustness. However, a sphere is a rather poor approximation of the highly non-rigid wrist. Better representations of the wrist and the also non-rigid palm could be very beneficial.

- *Integration of global synergies into the local model.* The local nature of our hand-tracking method has many advantages. However, the integration of global hand pose priors is not straight forward, i.e., it is not clear

how to enforce constraints on the hand as a whole. We state in Chapter 2 that many researchers have identified strong synergies in hand motion [Santello *et al.* 1998, Mason *et al.* 2001, Todorov and Ghahramani 2004, Ciocarlie *et al.* 2007, Tsoli and Jenkins 2007, Thakur *et al.* 2008, Ciocarlie and Allen 2009]. These synergies can improve hand tracking, like for examples demonstrated in [Wu *et al.* 2001]. Experimenting with a combined method which estimates likelihood terms locally but also considers global synergies would be interesting.

- *Combination with example-based hand pose recognition*. Our approach to hand tracking is model-based. On the positive side, model-based approaches are not constrained to a set of examples (see Chapter 2). On the down-side, model-based trackers require initialization and run the risk of loosing track. Both issues could be approached by combining our method with an example-based recognition technique. Attempts in this direction have been made in [Athitsos and Sclaroff 2003, Stenger *et al.* 2003]. A recognition approach with regard to manipulating hand motion was recently introduced in [Romero *et al.* 2010]. We think that combining such a system and ours has high potential.

The object-dependent hand pose prior and the related techniques presented in Chapter 5 also provide opportunities for improvement. Again we present and discuss some of our ideas.

- *Finding a good trade-off between the prior and the observation.* We have demonstrated that the prior observed for some hand and some object can ease the tracking of a sequence with a different hand and a different object. However, we did not dedicate much time to tuning the trade-off between conformity with the prior and compliance with the depth data. This trade-off is a standard issue with respect to priors [Willoughby 1979, Poggio and Smale 2005] and should be addressed.

- *Automatic detection of correspondence pairs for warping.* To generalize the prior from one object to another we apply a warp which is initialized by a set of correspondence pairs. These correspondence pairs relate the two object meshes to each other and are at the moment determined manually. While we did not explore this matter, automatic correspondence determination seems possible. 3d image features like *spin images* [Johnson 1997] might serve as a starting point.

- *Generalizing towards objects of other classes.* In Chapter 5, we generalized the prior within the same object class, i.e., we chose the example of cups. While this is an important first step, it would be interesting to generalize across different object classes. Staying with the examples in this work, imagine how to dial a number not on a phone but on a cup. One could picture a dialing thumb and fingers performing a typical grasp on the cup's handle. While dialing on a cup is senseless, it is often the extreme examples that provide new insights.

Finally, we touch on unaddressed issues and future work with respect to the animation method presented in Chapter 6.

- *Registration of subsequent range scans during object state transitions.* We express the observed hand poses for the manipulation of an object in the coordinate system of the object. During object state transitions of articulated objects, this coordinate system is currently undefined because we cannot register a mesh. One remedy could be to compute the increments between frames by registering subsequent range scans. The coordinate system can then be defined at any time, considering the coordinate system before the transition and a number of incremental offsets.

- *Incorporating several demonstrations of the same manipulation.* We currently create animations based on single demonstrations of the respective object manipulation. We consider it as an advantage of our method that such sparse data is sufficient to produce good results. However, observing several persons manipulate the same object, and maybe offering different manipulation *styles* to the animation artist could provide a valuable extension of our system.

- *Animating unobserved objects.* We demonstrated that our method can produce animations for objects that have been previously observed during manipulation. We did not attempt to animate an object based on the manipulation knowledge observed on a different object. But we did show how to generalize the object-dependent hand pose prior in Chapter 5. In addition to that, great progress has recently been made in the field of object recognition. We envision a system in which one can load an arbitrary object model. Based on object recognition techniques, similar objects could be found in a database, and the associated manipulation

knowledge would be transferred to the object of interest. Such a system would be highly relevant not only for animating hand-object interaction but also for a variety of other tasks, for example in the field of robotics.

7.3 Conclusion

In this work we have been concerned with the topic of hand-object interaction. Analyzing such interaction is a complex matter, but we contributed to the state of the art by focusing on certain aspects. In particular, it was demonstrated that manipulation knowledge can be 1) extracted from depth data, 2) encoded in an object-dependent hand pose prior, 3) transferred to unobserved objects, and 4) used to ease the animation process of sequences containing hand-object interaction.

More concretely, we presented a method to track a hand manipulating an object, an object-dependent hand pose prior, and a method to animate hands conditioned on a manipulated object. Each of these three components introduced new ideas and concepts. The significance of 3d techniques like ours will continue to increase in the future due to the rapid development of new 3d sensing devices, like the recently released Microsoft Kinect®.

While developing our methods, a non-invasive end-to-end pipeline was created. We started off by capturing hands with a structured-light system, and concluded with a useful method for the animation of hand-object interaction, which has the potential to ease the work of 3d animation artist. This application is highly relevant, e.g., for the production of 3d movies or digital video tutorials demonstrating the usage of tools. In addition, other applications like the control of robotic hands suggest themselves.

To conclude this work, we hope that the methods and results collected in the proposed thesis encourage other researchers across fields to advance work related to hand-object interaction.

Bibliography

[Albrecht *et al.* 2003] I. Albrecht, J. Haber, and H.-P. Seidel. Construction and animation of anatomically based human hand models. In *ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 98–109, 2003. 6.1

[Argall and Billard 2010] B. D. Argall and A. G. Billard. A survey of Tactile HumanRobot Interactions. *Robotics and Autonomous Systems*, 58(10):1159–1176, October 2010. 2.3.1

[Argall *et al.* 2009] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009. 2.3.2

[Argyros and Lourakis 2004] A. Argyros and M. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision (ECCV)*, pages 368–379, 2004. 2.2.1

[Athitsos and Sclaroff 2003] V. Athitsos and S. Sclaroff. Estimating 3D Hand Pose from a Cluttered Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 432–439, 2003. 2.2.3, 5.1, 7.2

[Athitsos and Sclaroff 2004] V. Athitsos and S. Sclaroff. Database indexing methods for 3D hand pose estimation. In *Gesture Workshop*, pages 288–299, 2004. 2.2.3

[Baak *et al.* 2009] A. Baak, B. Rosenhahn, M. Müller, and H.-P. Seidel. Stabilizing Motion Tracking Using Retrieved Motion Priors. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1428 – 1435, 2009. 5.1

[Bay *et al.* 2008] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008. 3.4

[Besl and McKay 1992] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, February 1992. 3.3, 3.3

[Bicchi and Kumar 2000] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 348 – 353, 2000. 2.3, 2.3.1, 5.1

[Black and Jepson 1998] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision (IJCV)*, 26(1):63 – 84, 1998. 2.2.2

[Borst *et al.* 2005] C. Borst, M. Fischer, and G. Hirzinger. Efficient and Precise Grasp Planning for Real World Objects. *Interaction with Real and Virtual Objects*, 18(6):91–111, 2005. 2.3.1

[Botsch and Kobbelt 2005] M. Botsch and L. Kobbelt. Real-Time Shape Editing using Radial Basis Functions. *Computer Graphics Forum (CGF)*, 24(3):611–621, 2005. 5.4.2

[Bray *et al.* 2004] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for 3D hand tracking. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 675–680, June 2004. 2.2.3

[Bray 2004] M. Bray. *Stochastic Meta-Descent Optimization for High-Dimensional Tracking*. PhD thesis, ETH Zurich, 2004. 3.5.2

[Buchholz and Armstrong 1992] B. Buchholz and T. J. Armstrong. A KINEMATIC MODEL OF THE HUMAN HAND TO EVALUATE ITS PREHENSILE CAPABILITIES. *Journal of Biomechanics*, 25(2):149–162, 1992. 2.1, 3.5.1, 4.2

[Buchholz *et al.* 1992] B. Buchholz, T. J. Armstrong, and S. Goldstein. Anthropometric data for describing the kinematics of the human hand. *Ergonomics*, 35(3):261–273, March 1992. 3.5.2

[Buehler *et al.* 2008] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *British Machine Vision Conference (BMVC)*, pages 1105–1114, 2008. 2.2.1

[Butterfass *et al.* 2006] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger. DLR-Hand II: Next generation of a dextrous robot

hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 109–114, 2006. 2.3

[Cailliet and Davis 1972] R. Cailliet and F. Davis. Hand pain and impairment. *Plastic and Reconstructive Surgery*, 49(4):452, 1972. 3.5.1, 4.2

[Carrozza *et al.* 2006] M. C. Carrozza, G. Cappiello, S. Micera, B. B. Edin, L. Beccai, and C. Cipriani. Design of a cybernetic hand for perception and action. *Biological cybernetics*, 95(6):629–44, December 2006. 1, 2.1

[Chella *et al.* 2004] A. Chella, H. Dzindo, I. Infantino, and I. Macaluso. A posture sequence learning system for an anthropomorphic robotic hand. *Robotics and Autonomous Systems*, 47(2-3):143–152, June 2004. 2.3.2

[Chen and Medioni 1992] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, April 1992. 3.3, 3.3

[Ciocarlie and Allen 2009] M. T. Ciocarlie and P. K. Allen. Hand posture subspaces for dexterous robotic grasping. *International Journal of Robotic Research (IJRR)*, 28(7):851–867, June 2009. 2.1, 7.2

[Ciocarlie *et al.* 2007] M. T. Ciocarlie, C. Goldfeder, and P. K. Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3270–3275, 2007. 2.1, 2.3.1, 7.2

[Cipolla and Hollinghurst 1996] R. Cipolla and N. Hollinghurst. Human-robot interface by pointing with uncalibrated stereo vision. *Image and Vision Computing*, 14(3):171–178, April 1996. 2.2.1

[Cipolla *et al.* 1994] R. Cipolla, P. Hadfield, and N. Hollinghurst. Uncalibrated stereo vision with pointing for a man-machine interface. In *IAPR Workshop on Machine Vision Applications*, pages 163–166, 1994. 2.2.1

[Cipriani *et al.* 2009] C. Cipriani, M. Controzzi, and M. C. Carrozza. Progress towards the development of the SmartHand transradial prosthesis. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 682–687, 2009. 1, 2.1

[Comaniciu *et al.* 2000] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 142 – 149, July 2000. 2.3.2

[Cooney *et al.* 1981] W. P. Cooney, M. J. Lucca, E. Y. Chao, and R. L. Linscheid. The kinesiology of the thumb trapeziometacarpal joint. *The Journal of bone and joint surgery. American volume*, 63(9):1371–81, December 1981. 2.1, 3.5.1, 4.2

[Cooper and Bowden 2007] H. Cooper and R. Bowden. Large lexicon detection of sign language. In *International Conference on Human-Computer Interaction*, pages 88–97, 2007. 2.2.1

[Cutkosky and Wright 1986] M. Cutkosky and P. Wright. Modeling manufacturing grips and correlations with the design of robotic hands. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1533–1539, 1986. 2.1, 2.3.2, 4.2, 4.4.2

[Dennis Jr and Schnabel 1996] J. Dennis Jr and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16)*. 1996. 2.2.3

[Ekvall and Kragíc 2005] S. Ekvall and D. Kragíc. Grasp recognition for programming by demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 748 – 753, 2005. 2.3.2, 4.5

[ElKoura and Singh 2003] G. ElKoura and K. Singh. Handrix: animating the human hand. In *ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 110–119, 2003. 6.1, 2, 3

[Erol *et al.* 2007] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):52–73, October 2007. 1, 2.2.3

[Farhadi *et al.* 2007] A. Farhadi, D. Forsyth, and R. White. Transfer Learning in Sign language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2.2.1

[Felzenszwalb and Huttenlocher 2005] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, January 2005. 4.1

[Fischer *et al.* 1998] M. Fischer, P. V. D. Smagt, and G. Hirzinger. Learning Techniques in a Dataglove Based Telemanipulation System for the DLR Hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1603–1608, 1998. 2.3.2, 2.3.2

[Freeman and Weissman 1995] W. Freeman and C. Weissman. Television control by hand gestures. In *International Workshop on Automatic Face and Gesture Recognition*, pages 179–183, 1995. 2.2.1

[Gibson 1979] J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Miffin, Boston, 1979. 1, 6.1

[Griffin *et al.* 2000] W. Griffin, R. Findley, and M. Turner. Calibration and mapping of a human hand for dexterous telemanipulation. In *Haptic Interfaces for Virtual Environments and Teleoperator Systems Symposium*, pages 1145–1152, 2000. 2.3.2

[Hamer *et al.* 2009] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1475–1482, 2009. 4, 5.1

[Hamer *et al.* 2010] H. Hamer, J. Gall, T. Weise, and L. Van Gool. An Object-Dependent Hand Pose Prior from Sparse Training Data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 671 – 678, 2010. 5

[Hamer *et al.* in press] H. Hamer, J. Gall, R. Urtasun, and L. Van Gool. Data-Driven Animation of Hand-Object Interactions. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, in-press. 6

[Hamilton and Dunsmuir 2002] R. Hamilton and R. Dunsmuir. Radiographic assessment of the relative lengths of the bones of the fingers of the human hand. *The Journal of Hand Surgery: Journal of the British Society for Surgery of the Hand*, 27(6):546–548, 2002. 2.1, 4.2, 4.2.2

[Harders *et al.* 2009] M. Harders, G. Bianchi, B. Knoerlein, and G. Székely. Calibration, registration, and synchronization for high precision augmented reality haptics. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 15(1):138–49, 2009. 1

[Harris and Stephens 1988] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, pages 147–152, 1988. 3.4

[Heap and Hogg 2002] T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 140–145, 2002. 2.2.3

[Hollister *et al.* 1992] A. Hollister, W. Buford, L. Myers, D. Giurintano, and A. Novick. The axes of rotation of the thumb carpometacarpal joint. *Journal of Orthopaedic Research*, 10(3):454–460, May 1992. 2.1, 3.5.1, 4.2

[Horn 1987] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629, April 1987. 3.3

[Hu *et al.* 2004] H. Hu, X. Gao, J. Li, and J. Wang. Calibrating human hand for teleoperating the hit/dlr hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4571–4576, 2004. 2.3.2

[Hueser and Baier 2006] M. Hueser and T. Baier. Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2795–2800, 2006. 2.3.2, 2.3.2, 5.1

[Iberall 1997] T. Iberall. Human prehension and dexterous robot hands. *International Journal of Robotic Research (IJRR)*, 16(3):285, June 1997. 2.1

[Ike *et al.* 2007] T. Ike, N. Kishikawa, and B. Stenger. A real-time hand gesture interface implemented on a multi-core processor. In *Machine Vision Applications (MVA)*, pages 9–12, 2007. 2.2.2

[Isard and Blake 1998] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In *European Conference on Computer Vision (ECCV)*, page 893, 1998. 2.2.3

[Ishida *et al.* 2005] Y. Ishida, M. Kondo, and T. Ogasawara. Development of the NAIST-Hand with Vision-based Tactile Fingertip Sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2332–2337, 2005. 2.3

[Jacobsen *et al.* 2002] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers. Design of the Utah/MIT dextrous hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 1520–1532, 2002. 2.3

[Jaeggli and Van Gool 2003] T. Jaeggli and L. Van Gool. Online 3D Acquisition and Model Integration. In *IEEE International Workshop on Projector-Camera Systems, ICCV03*, 2003. 3.3

[Jo *et al.* 1998] K. H. Jo, Y. Kuno, and Y. Shirai. Manipulative hand gesture recognition using task knowledge for human computer interaction. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 468–473, 1998. 2.3.2

[Johnson 1997] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997. 5.4.2, 7.2

[Jones and Rehg 2002] M. Jones and J. M. Rehg. Statistical Color Models with Application to Skin Detection. *International Journal of Computer Vision (IJCV)*, 46(1):81–96, 2002. 3.2

[Kadir *et al.* 2004] T. Kadir, R. Bowden, E. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *British Machine Vision Conference (BMVC)*, volume 2, pages 939–948, 2004. 2.2.1

[Kawasaki *et al.* 1999] H. Kawasaki, T. Komatsu, K. Uchiyama, and T. Kurimoto. Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 782–787, September 1999. 2.3

[Kim *et al.* 1987] J. Kim, D. Blythe, D. Penny, and A. Goldenberg. Computer architecture and low level control of the PUMA/RAL hand system: Work in progress. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1590–1594, 1987. 2.3

[Kim *et al.* 2000] J. Kim, F. Cordier, and N. Magnenat-Thalmann. Neural network-based violinist's hand animation. In *Computer Graphics International (CGI)*, pages 37 – 41, 2000. 6.1, 2, 3

[Kjellström *et al.* 2008] H. Kjellström, J. Romero, D. Martínez, and D. Kragíc. Simultaneous visual recognition of manipulation actions and manipulated objects. In *European Conference on Computer Vision (ECCV)*, pages 336–349, 2008. 1, 2.3.2

[Kolsch and Turk 2004] M. Kolsch and M. Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, page 158, 2004. 2.2.1

[Krahnstoever *et al.* 2002] N. Krahnstoever, E. Schapira, S. Kettebekov, and R. Sharma. Multimodal human-computer interaction for crisis management systems. In *Workshop on Applications of Computer Vision (WACV)*, pages 203–207, 2002. 2.2.1

[Krüger *et al.* 2007] V. Krüger, D. Kragíc, A. Ude, and C. Geib. The Meaning of Action: A Review on action recognition and mapping. *Advanced Robotics*, 21(13):1473–1501, 2007. 2.3.2

[Kry and Pai 2006] P. Kry and D. Pai. Interaction capture and synthesis. *ACM Transactions on Graphics (TOG)*, 25(3):872–880, July 2006. 1, 5.1, 6.1, 1

[Leibe *et al.* 2004] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *European Conference on Computer Vision Workshop (ECCVW)*, 2004. 4.1

[Li *et al.* 2007] Y. Li, J. L. Fu, and N. S. Pollard. Data-Driven Grasp Synthesis Using Shape Matching and Task-Based Pruning. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13(4):732–747, August 2007. 5.1, 6.1, 1

[Light and Chappell 2000] C. M. Light and P. H. Chappell. Development of a lightweight and adaptable multiple-axis hand prosthesis. *Medical engineering & physics*, 22(10):679–84, December 2000. 1, 2.1

[Liu 2008] C. K. Liu. Synthesis of interactive hand manipulation. In *ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 163–171, 2008. 6.1, 3

[Liu 2009] C. K. Liu. Dextrous manipulation from a grasping pose. *ACM Transactions on Graphics (TOG)*, 28(3):1–6, August 2009. 6.1, 3

[Lockton and Fitzgibbon 2002] R. Lockton and A. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *British Machine Vision Conference (BMVC)*, pages 817–826, 2002. 2.2.2

[Lovchik and Diftler 1999] C. Lovchik and M. Diftler. The Robonaut hand: a dexterous robot hand for space. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 907–912, 1999. 2.3

[Lowe 1999] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150 – 1157, 1999. 4.1

[MacCormick and Isard 2000] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2000. 2.2.3, 5.1

[Mann *et al.* 1996] R. Mann, A. D. Jepson, and J. M. Siskind. The Computational Perception of Scene Dynamics. In *European Conference on Computer Vision (ECCV)*, pages 528–539, 1996. 1, 2.3.2

[Mason *et al.* 2001] C. Mason, J. Gomez, and T. Ebner. Hand synergies during reach-to-grasp. *Journal of Neurophysiology*, 86(6):2896, 2001. 2.1, 7.2

[McDonald *et al.* 2001] J. McDonald, J. Toro, K. Alkoby, A. Berthiaume, R. Carter, P. Chomwong, J. Christopher, M. J. Davidson, J. Furst, B. Konie, G. Lancaster, L. Roychoudhuri, E. Sedgwick, N. Tomuro, and R. Wolfe. An improved articulated model of the human hand. *The Visual Computer*, 17(3):158–166, May 2001. 2.1, 3.5.1, 4.2

[Micilotta *et al.* 2005] A. Micilotta, E. Ong, and R. Bowden. Detection and tracking of humans by probabilistic body part assembly. In *British Machine Vision Conference (BMVC)*, pages 429–438, 2005. 2.2.1

[Miller and Allen 2004] A. Miller and P. K. Allen. GraspIt!: A Versatile Simulator for Grasp Analysis. *Robotics & Automation Magazine*, 11(4):110–122, December 2004. 2.3.1

[Miller *et al.* 2003] A. Miller, S. Knoop, and H. Christensen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1824–1829, 2003. 2.3.1, 5.1

[Mishra and Silver 1989] B. Mishra and N. Silver. Some discussion of static gripping and its stability. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4):783–796, 1989. 2.3.1

[Moon and Pavlovic 2006] K. Moon and V. Pavlovic. Impact of Dynamics on Subspace Embedding and Tracking of Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 198–205, 2006. 5.1

[Morales *et al.* 2006] a. Morales, P. Sanz, A. del Pobil, and A. Fagg. Vision-based three-finger grasp synthesis constrained by hand geometry. *Robotics and Autonomous Systems*, 54(6):496–512, June 2006. 2.3.1

[Napier 1956] J. Napier. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, 38(4):902–913, 1956. 1, 2.1

[Okamura *et al.* 2000] A. M. Okamura, N. Smaby, and M. Cutkosky. An overview of dexterous manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 255–262, 2000. 2.3.1

[Ong and Bowden 2004] E. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 889–894, 2004. 2.2.1

[Ong and Ranganath 2005] S. C. W. Ong and S. Ranganath. Automatic sign language analysis: a survey and the future beyond lexical meaning.

IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27(6):873–91, June 2005. 2.2.2

[Pavlovic *et al.* 1997] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):677–695, July 1997. 2.2.1

[Pearl 1982] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *National Conference on Artificial Intelligence*, pages 133–136, 1982. 3.6

[Penfield and Rasmussen 1950] W. Penfield and T. Rasmussen. *The Cerebral Cortex of Man: A Clinical Study of Localization of Function*. Macmillan, 1950. 2.1

[Poggio and Smale 2005] T. Poggio and S. Smale. The Mathematics of Learning: Dealing with Data. *Foundations and Advances in Data Mining*, 50(5):537–544, 2005. 7.2

[Pollard and Zordan 2005] N. S. Pollard and V. Zordan. Physically based grasping control from example. In *ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 311–318, 2005. 5.1, 6.1, 1, 3

[Pons *et al.* 2004] J. Pons, E. Rocon, R. Ceres, D. Reynaerts, B. Saro, S. Levin, and W. Van Moorleghem. The MANUS-HAND Dextrous Robotics Upper Limb Prosthesis: Mechanical and Manipulation Aspects. *Autonomous Robots*, 16(2):143–163, March 2004. 1, 2.1

[Rao *et al.* 2002] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *International Journal of Computer Vision (IJCV)*, 50(2):203–226, November 2002. 2.3.2, 5.3, 6.3.3

[Rehg and Kanade 1994] J. M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European Conference on Computer Vision (ECCV)*, pages 35–46, 1994. 2.2.3, 5.1

[Rijpkema and Girard 1991] H. Rijpkema and M. Girard. Computer Animation of Knowledge-Based Human Grasping. *ACM SIGGRAPH Computer Graphics*, 25(M):339–348, 1991. 3.5.1

[Ritter *et al.* 2009] H. Ritter, R. Haschke, and J. Steil. *Trying to Grasp a Sketch of a Brain for Grasping*, pages 1–21. Springer, 2009. 2.3.1

[Roethling 2007] F. Roethling. *Real Robot Hand Grasping using Simulation-Based Optimisation of Portable Strategies*. PhD thesis, Bielefeld Univesity, 2007. 2.3.1

[Romero *et al.* 2010] J. Romero, H. Kjellström, and D. Kragíc. Hands in action: Real-time 3D reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 458 – 463, 2010. 2.3.2, 4.5, 7.2

[Rosenhahn *et al.* 2008] B. Rosenhahn, C. Schmaltz, T. Brox, J. Weickert, D. Cremers, and H.-P. Seidel. Markerless motion capture of man-machine interaction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1063–6919, 2008. 5.1

[Rusinkiewicz and Levoy 2001] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001. 3.3

[Santello *et al.* 1998] M. Santello, M. Flanders, and J. F. Soechting. Postural hand synergies for tool use. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 18(23):10105–10115, December 1998. 2.1, 2.3.1, 7.2

[Sato *et al.* 2002] Y. Sato, K. Bernardin, H. Kimura, and K. Ikeuchi. Task analysis based on observing hands and objects by vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1208–1213, 2002. 2.2.2, 2.3.2, 5.3

[Saxena *et al.* 2008] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics (IJRR)*, 27(2):157–173, February 2008. 2.3.1, 5.1

[Schlesinger 1919] G. Schlesinger. *Der mechanische Aufbau kuenstlicher Glieder*, page 21. Springer, 1919. 2.1

[Schulz *et al.* 2005] S. Schulz, C. Pylatiuk, M. Reischl, J. Martin, R. Mikut, and G. Bretthauer. A hydraulically driven multifunctional prosthetic hand. *Robotica*, 23(3):293–299, May 2005. 1, 2.1

[Seo *et al.* 2008] B.-K. Seo, J. Choi, J.-H. Han, H. Park, and J.-I. Park. One-handed interaction with augmented virtual objects on mobile devices. In *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, page 1, 2008. 1

[Shimoga 1996] K. Shimoga. Robot grasp synthesis algorithms: A survey. *International Journal of Robotic Research (IJRR)*, 15(3):230, June 1996. 2.3.1

[Sidenbladh *et al.* 2002] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision (ECCV)*, pages 784 – 800, 2002. 5.1

[Standring and Borley 2008] S. Standring and N. Borley. *Gray's Anatomy: The Anatomical Basis of Clinical Practice*. Churchill Livingstone, 2008. 2.1

[Starner *et al.* 1998] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(12):1371 – 1375, 1998. 2.2.1, 2.2.2

[Stenger *et al.* 2001] B. Stenger, P. Mendonca, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 310–315 vol.2, 2001. 2.2.3, 5.1

[Stenger *et al.* 2003] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1063–1070, 2003. 2.2.3, 7.2

[Stenger *et al.* 2006] B. Stenger, A. Thayananthan, and P. Torr. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1372–1384, 2006. 2.2.3, 5.1

[Stenger *et al.* 2010] B. Stenger, T. Woodley, and R. Cipolla. *A Vision-Based Remote Control*, pages 233–262. Springer, 2010. 2.2.2

[Sudderth *et al.* 2003] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–612 vol.1, 2003. 2.2.3

[Sudderth *et al.* 2004a] E. Sudderth, M. Mandel, W. Freeman, and A. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *Neural Information Processing Systems (NIPS)*, pages 1369–1376, 2004. 2.2.3, 3.6, 4.1

[Sudderth *et al.* 2004b] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Visual Hand Tracking Using Nonparametric Belief Propagation. In *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 189–189, 2004. 2.2.3, 3.6, 4.1, 4.3.3, 5.1

[Thakur *et al.* 2008] P. H. Thakur, A. J. Bastian, and S. S. Hsiao. Multidigit movement synergies of the human hand in an unconstrained haptic exploration task. *The Journal of neuroscience*, 28(6):1271–81, February 2008. 2.1, 7.2

[Todorov and Ghahramani 2004] E. Todorov and Z. Ghahramani. Analysis of the synergies underlying complex hand manipulation. In *Engineering in Medicine and Biology Society (EMBC)*, pages 4637–4640 vol.6, 2004. 2.1, 7.2

[Townsend 2000] W. Townsend. The BarrettHand grasper programmably flexible part handling and assembly. *Industrial Robot: An International Journal*, 27(3):181–188, 2000. 2.3

[Triesch and Malsburg 2002] J. Triesch and C. V. D. Malsburg. Classification of hand postures against complex backgrounds using elastic graph matching. *Image and Vision Computing*, 20(13-14):937–943, 2002. 2.2.2

[Tsoli and Jenkins 2007] A. Tsoli and O. C. Jenkins. 2d subspaces for user-driven robot grasping. In *Robotics, Science and Systems Conference: Workshop on Robot Manipulation*, 2007. 2.1, 7.2

[Turner *et al.* 2000] M. Turner, R. Findley, W. Griffin, M. Cutkosky, and D. Gomez. Development and testing of a telemanipulation system with arm and hand motion. In *ASME Dynamic Systems and Control Division*, 2000. 2.3.2, 2.3.2

[Urtasun *et al.* 2006] R. Urtasun, D. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 238–245, 2006. 5.1

[Van Den Bergh *et al.* 2009] M. Van Den Bergh, E. Koller-Meier, F. Bosche, and L. Van Gool. Haarlet-based hand gesture recognition for 3D interaction. In *Workshop on Applications of Computer Vision (WACV)*, pages 1–8, December 2009. 2.2.1, 2.2.2

[Viola and Jones 2004] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision (IJCV)*, 57:137–154, 2004. 2.2.1, 2.2.2

[Vondrak *et al.* 2008] M. Vondrak, L. Sigal, and O. C. Jenkins. Physical Simulation for Probabilistic Motion Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1 – 8, 2008. 5.1

[Voyles and Khosla 2001] R. Voyles and P. Khosla. A multi-agent system for programming robots by human demonstration. In *AI and Manufacturing Research*, pages 59–67, 2001. 2.3.2

[Walker 2004] R. Walker. Developments in dextrous hands for advanced robotic applications. In *World Automation Congress*, pages 123–128, 2004. 2.3

[Warwick *et al.* 2003] K. Warwick, M. Gasson, B. Hutt, I. Goodhew, P. Kyberd, B. Andrews, P. Teddy, and A. Shad. The application of implant technology for cybernetic systems. *Archives of neurology*, 60(10):1369–73, October 2003. 1, 2.1

[Weise *et al.* 2008] T. Weise, B. Leibe, and L. Van Gool. Accurate and Robust Registration for In-hand Modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1 – 8, 2008. 3.4

[Weise 2009] T. Weise. *Real-time 3D Scanning*. PhD thesis, Computer Vision Laboratory, ETH Zurich, 2009. 1, 3.1

[Willoughby 1979] R. A. Willoughby. Solutions of Ill-Posed Problems (A. N. Tikhonov and V. Y. Arsenin). *SIAM Review*, 21(2):266–267, 1979. 7.2

[Wu and Huang 1999] Y. Wu and T. Huang. Vision-based gesture recognition: A review. *Gesture-Based Communication in Human-Computer*, 1739:103–115, 1999. 2.2.1

[Wu and Huang 2001] Y. Wu and T. Huang. Hand modeling, analysis and recognition. *IEEE Signal Processing Magazine*, 18(3):51–60, 2001. 2.2.1, 4.2

[Wu *et al.* 2001] Y. Wu, J. Lin, and T. Huang. Capturing natural hand articulation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 426 – 432, 2001. 2.2.3, 5.1, 7.2

[Yedidia *et al.* 2003] J. Yedidia, W. Freeman, and Y. Weiss. *Understanding belief propagation and its generalizations*, pages 239–269. Morgan Kaufmann Publishers Inc., 2003. 3.6, 3.6

[Yoshikawa *et al.* 2008] T. Yoshikawa, M. Koeda, and H. Fujimoto. Shape recognition and grasping by robotic hands with soft fingers and omnidirec-

tional camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 299–304, 2008. 2.3.1

[Zhang *et al.* 2004] L. Zhang, N. Snavely, B. Curless, and S. Seitz. Spacetime Faces: High-Resolution Capture for Modeling and Animation. In *ACM International Conference on Computer Graphics and Interactive Techniques*, pages 548–558, 2004. 5.4.2