# UNIVERSITÀ DEGLI STUDI DI PARMA

*Dottorato di Ricerca in Tecnologie dell'Informazione*

*XXV Ciclo*

# TOWARD EFFECTIVE PHYSICAL HUMAN-ROBOT INTERACTION

Coordinator:

*Chiar.mo Prof. Marco Locatelli*

Tutor:

*Chiar.mo Prof. Stefano Caselli*

Dottorando: *Vincenzo Micelli*

Gennaio 2013

*“Law I / A robot may not injure a human or, through inaction,*
*allow a human being to come to harm.*

*Law II / A robot must obey orders given to it by human beings*
*except where such orders would conflict with the first law.*

*Law III / A robot must protect its own existence as long as such*
*protection does not conflict with the first or second law.”*

*Isaac Asimov*

# Table of contents

# Introduction

Today robots are mostly employed in factories. These robots are mainly programmable industrial machines that offer modest challenges in human-robot interaction. Now, advances in computer technology, artificial intelligence, speech simulation and understanding, and remote controls have led to breakthroughs in robotic technology offering new opportunities to employ robots. Research laboratories are building new autonomous mobile robots that can identify and track user's position, respond to spoken questions, display text or spatial information, and travel in unstructured environment while avoiding obstacles. These robots will soon assist in a range of tasks that are unpleasant, unsafe, taxing, or boring to people. For example, nurses making shifts in assisted living facilities spend much of their time sorting and administering medications. A robotic assistant could do some of this work, as well as chores that are difficult for elderly people such as fetching newspapers and mail, getting up and down stairs, getting things out of high or low cabinets, and carrying laundry, thereby enabling elderly people to remain independent for a longer time. Robotic assistants in the future might act as guards, help fight fires, deliver materials on construction sites and in mines, and distribute goods or help consumers in retail stores. Robots might even provide high-interaction services such as taking blood and coloring hair.

In these contexts robots will inevitably interact with people. The emerging field of human-robot interaction, brings together research and application of methodology from robotics, human factors, human-computer interaction, interaction design, cognitive psychology, education and other fields to enable robots to have more effective and more seamless interactions with humans throughout their spheres of functioning.

Among the various kinds of possible interactions, in this thesis I am particularly interested in physical human-robot interaction. Physical Human-Robot Interaction (pHRI) represents one of the most motivating, challenging and ambitious research topics in robotics. Many of the future and emerging applications of robotics, be they in service, care and assistance, rehabilitation, or in more traditional working contexts, will indeed require robots to work in close proximity if not in direct contact with humans. The first issue that must be addressed when designing a robot for pHRI applications is safety. Once the safety issue has been addressed it is important that the interaction is effective, seamless, and most of all accepted by humans. To achieve this goal it is important for robots to have a strong perception of the human partner they are interacting with.

In order to study how a robot can successfully engage in physical interaction with people and which factors are crucial during this kind of interaction, I investigated how humans and robots can hand over objects to each other. While this task addresses only a portion of what we need to know in order to successfully create robots that can effectively physically interact with humans, it begins to give us an idea of what aspects we can influence when building sociable robots in the near future. This specific highly collaborative task is very common in every day life and is executed without difficulties by humans. However such an easy task for humans is hard for robots which, in order to accomplish the task, need to recognize the intent of the human to perform the handover, negotiate with the human the timing and the location for the handover, execute suitable arm trajectories and perform a stable grasp of the object. All these sub-tasks are complex areas of research and still all must be accomplished together in order to perform seamless handovers.

To study this specific interactive task I developed two robotic systems. The first system enables robots to receive objects from humans. The second system enables robots to hand object to humans. Although various aspects of human-robot handovers have been deeply investigated in the state of the art, during my studies I focused on three issues that have been rarely investigated so far:

- Human presence and motion analysis during the interaction in order to infer non-verbal communication cues and to synchronize the robot actions with the human motion;

- Development and evaluation of human-aware pro-active robot behaviors that enable robots to behave actively in the proximity of the human body in order to negotiate the handover location and to perform the transfer of the object;
- Consideration of objects grasp affordances during the handover in order to make the interaction more comfortable for the human.

Pro-activity and human-awareness are two fundamental capabilities that robot must have to perform seamless handovers and seamless physical human-robot interaction in general. These capabilities are tightly related to the ability of the robot to perceive people. Human detection and motion analysis are challenging and complex issues. In this thesis I will show how the recent advancement in this field can lead to very intuitive, comfortable and effective human-robot interactions. This thesis is outlined as follows. In chapter 1 I discuss the state of the art of human-robot interaction with particular attention to the interactions that occur during human-robot handovers. In chapter 2 I describe the robotics applications developed to investigate human-robot handovers and present the main findings of my research. In chapter 3 I merge the findings of my research work with the state of the art and I suggest a holistic handover structure that robots should follow in order to achieve seamless human-robot handovers. In chapter 4 I offer the conclusions and discuss what future work will follow this research. In Addition, In Appendix A I discuss the main tools and sensors used to perform gesture recognition. A key point during the interaction is the communication between the partners. Gestures are often used to convey communication cues during human-human interactions. Therefore robots must be able to recognize and interpret human gestures to interact with humans in an intuitive and effective manner.

# Chapter 1

# Toward Social Robots

## 1.1 Human-Robot Interaction

With the fast advancement of technology, modern robots are continuously upgrading their role in the society. While traditionally robots have been successfully employed in industrial settings to improve productivity and perform dangerous tasks, in recent years, robotics technology has significantly matured and produced robots that are able to successfully operate in unstructured environments. As a result of this ongoing process, the application domains of robots have slowly expanded into domestic environments, offices, hospitals and other human-inhabited locations. These new robots are usually referred to as Personal Robots (or Social Robots). An exhaustive survey of Personal Robots is beyond the scope of this thesis, however, I will mention a few recent efforts (Fig. 1.1). The Intel HERB mobile manipulation platform [1] has demonstrated impressive capabilities in indoor environments ranging from being able to pick and place objects [2] to push-based manipulation on tabletop environments [3]. The ARMAR-III robots have been used for tasks in a prototype kitchen setting demonstrating impressive capabilities including combined grasp and motion planning [4]. The Kaspar robot [5] has demonstrated to be effective in therapy for children with autism. Other examples include the PR2 robot [6], Kismet [7] and RIBA [8]. In the near future, due to this recent remarkable improvements in robotic intelligence and

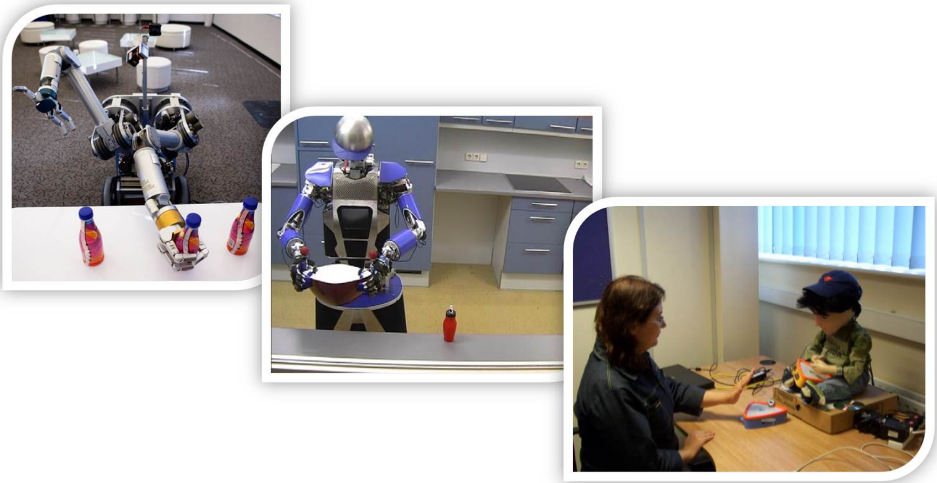


Figure 1.1: Left: HERB 2.0: A bimanual mobile manipulator developed at the Personal Robotics Lab at Carnegie Mellon University; Center: the humanoid robot ARMAR-III, developed at the Karlsruhe Institute of Technology; Right: KASPAR, child-sized humanoid robot developed by the Adaptive Systems Research Group at the University of Hertfordshire.

technology, it is expected that robots will coexist with humans to assist or cooperate with them [9]. In this context, the interaction and cooperation between humans and robots has become an increasingly important and, at the same time, challenging aspect of robot development. Robots must be able to interact with humans in a safe and user-friendly manner while performing cooperative tasks. The research area that aims to achieve this ambitious objective is a multidisciplinary field with contributions from Robotics, Artificial Intelligence, Human-Computer Interaction, and Cognitive Psychology. This entirely new field, referred to as Human-Robot Interaction (HRI), has the goal to develop principles, techniques and algorithms to allow for direct, safe and effective interaction between humans and robots. Moreover, a person working with a robot should not be required to learn a new form of interaction. Thus, we need to develop computational models of social intelligence for these robots that will al-

low them to have interactions that are natural and intuitive for a human partner.
Defining a general taxonomy of HRI is a complex task. Human-robot interaction is a vast field and HRI applications are usually cross-disciplinary and different in many attributes. In [10] Yanco at al. propose an extended taxonomy for the field of Human Robot Interaction. This thesis focuses on physical interaction between a human and a robot which share the same workspace. Following Yanco's classification the *physical proximity* between the human and the robot in the applications that are relevant to this thesis can have values *approaching* and *touching*. Physical interaction between a couple of human or robotic agents considers situations where the agents are in physical contact with each other and exchange mechanical energy. The contact can be either direct, i.e. part of the human body is in contact with part of the robot links, or indirect, i.e it is established through collaborative manipulation of a common object.

The presence of physical interaction is a key factor in the development of a HRI system. Some tasks such as transferring an object from a human to a robot [11] or lifting a human for nursing-care purposes [12] require physical contact. Other tasks such as conveying/detecting emotion [13] can be achieved without engaging in physical contact. Although all the systems in which robots and humans share the workspace need to deeply address the issue of safety, coping with this issue is even more important in systems that involve physical contact between humans and robots. The more humans get close to robots, the more issues like how they feel safe during the interaction [14] and how safe the interaction is become prominent. The issue of safety has been vastly investigated in literature. Significant approaches include the introduction of compliance at the mechanical design level [15], control methods that identify when safety is threatened and generate real-time motion trajectories to move the robot to a safe location during a potential collision event [16], collision detection and reaction [17], and inherent safety [18]. In addition, for successful physical interaction and cooperation, the robot must have the ability to adapt its behavior to the human counterpart. Therefore, the robot must be strongly aware of the physical presence of the person it is interacting with. This aspect is particularly relevant when the robot has a proactive behavior and takes the initiative during the interaction.

Another key factor in a HRI system is the manner in which information is exchanged between the human and the robot. Measures of the efficiency of an interaction include the *interaction time* required for intent and/or instructions to be communicated to the robot [19], the *cognitive* or *mental workload* of an interaction [20], the amount of *situation awareness* produced by the interaction [21] (or reduced because of interruptions from the robot), and the amount of shared understanding or common ground between humans and robots [22]. There are two primary dimensions that determine the way information is exchanged between a human and a robot: the communications medium and the format of the communications. The primary media are delineated by three of the five senses: sight, hearing, and touch. The format of communication exploited in HRI varies across applications and can be summarized as follows:

- visual displays, typically presented as graphical user interfaces or augmented reality interfaces [23, 24, 25],
- gestures, including body postures and motion, hand gestures, facial expressions and gaze direction [13, 26, 27, 28, 29, 30],
- speech and natural language, which include both auditory speech and text-based responses, and which frequently emphasize dialog and mixed-initiative interaction [31, 32],
- non-speech audio, frequently used in alerting [33],
- haptics, used either remotely in augmented reality or in teleoperation to invoke a sense of presence especially in telemanipulation tasks [34], or in proximity to promote emotional and social exchanges and to provide feedbacks during interaction tasks [35, 36, 37].

Recently, the attention has been focused on building multimodal interfaces [38, 39], partly motivated by a quest to reduce workload in accordance to Wickens' multiple resource theory [40] and partly motivated by a desire to make interactions more natural and easier to learn [41, 42, 43].

In this thesis I am particularly interested in gestures. Gestures are an important feature of social interaction, frequently used by human speakers to illustrate what speech alone cannot provide, e.g. to convey referential, spatial or iconic information. Accordingly, robots that are intended to engage in natural human-robot interaction should recognize and interpret gestures to enable easy and intuitive interaction. There are many facets of the modeling and recognition of human gesture: gestures can be expressed through hands, faces, or the entire body. Gesture recognition is especially valuable in applications involving Human-Robot interaction for several reasons. First, it provides a redundant form of communication between the user and the robot. For example, the user may say “Stop” at the same time that he is giving a stopping gesture. The robot needs to recognize one of the two commands, and gestures are crucial in situations where speech may be garbled or drowned out (e.g., in space, underwater, on the battlefield). Second, gestures are an easy way to give geometric information to the robot. Rather than give coordinates to where the robot should move, the user can simply point to a spot on the floor. In addition, in some situation humans use gestures as unique format of communication to convey an intention. For example when two humans are handing over objects to each other often a partner reaches out his hand waiting for the other partner to grasp the object, without using verbal communication. In appendix A an overview of the main tools and techniques used for gesture recognition is reported. In the following, the most relevant research works in the field of physical human-robot interaction are discussed.

## 1.2 Physical Human-Robot Interaction

Close physical interaction between robots and humans is a particularly challenging aspect of robot development. For successful interaction and cooperation, the robot must have the ability to adapt its behavior to the human counterpart. This particular kind of interaction is necessary for many tasks and has been vastly investigated in literature. A key feature in physical human robot interaction is whether the partners have an active or passive behavior. In some tasks only one of the partners operates actively, while other tasks require both partners to act during the interaction. An example of tasks where the robot has a passive behavior is given by Learning from Demonstration (LfD) tasks. In [37] Balasubramanian at al. present a novel and simple exper-

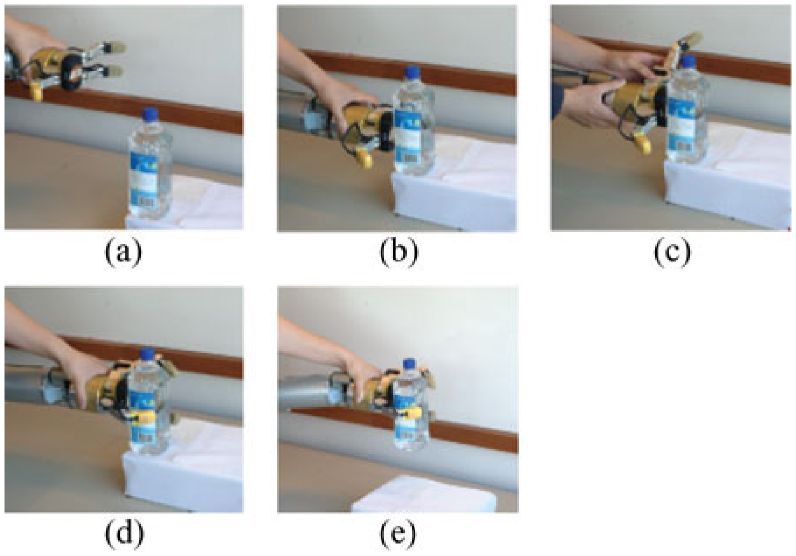


Figure 1.2: Experimental procedure of a human subject guiding the robot to grasp an object: (a) and (b) Approach the object, (c) adjust wrist orientation and finger spread, (d) fingers close in on the object, and (e) lift object. Note that the subject was free to move around the workspace to view the physical interaction from multiple angles (from [37]).

imental method called physical human interactive guidance to study human-planned grasping. Instead of studying how the human uses his/her own biological hand or how a human teleoperates a robot hand in a grasping task, the method involves a human interacting physically with a robot arm and hand, carefully moving and guiding the robot into the grasping pose, while the robot's configuration is recorded (Fig. 1.2). Analysis of the grasps from this simple method has produced two interesting results. First, the grasps produced by this method perform better than grasps generated through a state-of-the-art automated grasp planner. Second, this method when combined with a detailed statistical analysis using a variety of grasp measures (physics-based heuristics considered critical for a good grasp) offered insights into how the human grasping method is similar or different from automated grasping synthesis

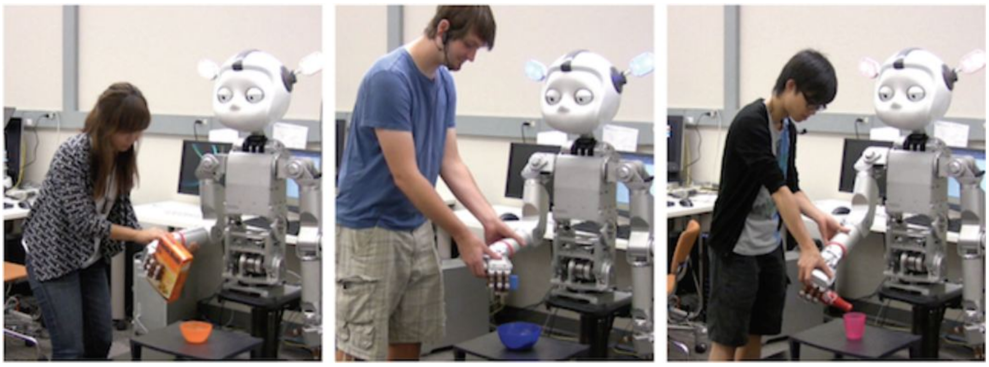


Figure 1.3: Subjects demonstrating three different skills to a robot (from [44]).

techniques. Specifically, data from the physical human interactive guidance method showed that the human-planned grasping method provides grasps that are similar to grasps from a state-of-the-art automated grasp planner, but differed in one key aspect. The robot wrists were aligned with the object's principal axes in the human-planned grasps (termed low skewness in the paper), while the automated grasps used arbitrary wrist orientation. Preliminary tests show that grasps with low skewness were significantly more robust than grasps with high skewness (77-93%). Another example of LfD that involves physical interaction is given in [44] (Fig. 1.3). Here the robot does not act physically but performs active learning in the sense that it asks question during the interaction.

On the other hand, an example of physical interaction where the robot has an active behavior while the human does not act is given in [12]. In [12] Mukai at al. present a prototype nursing-care assistant robot named RIBA (Robot for Interactive Body Assistance) designed to conduct physically taxing tasks while in contact with a human as the manipulated object. RIBA has succeeded in transferring a human between a bed and a wheelchair, using human-type arms (Fig. 1.4). It has sufficient power to lift up a human weighing over 60 kg. It interacts with the human as the object through distributed surface contact with a finite area on its outer shell. Information on the contacts is obtained by tactile sensors mounted on a wide area of its arms.

Lastly, there are tasks where the two partners have to collaborate in order to achieve

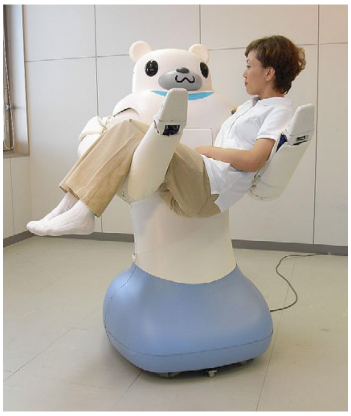


Figure 1.4: RIBA lifting a human in its arms (from [12]).

a common goal. For example a robot and a human can collaborate to carry an object [45]. A common task that requires collaboration between the interacting partners is the handover of an object. In this thesis I am particularly interested in this important and highly-collaborative task which enables a vast number of human-robot interactions. The next section discusses the state of the art of this specific interactive task.

### 1.2.1 Human-Robot object hand over

In human-human interaction "handing over" is the act of passing an object to another person. Personal robots that will assist humans in different environments such as homes, offices or hospitals, will inevitably face tasks that require handing over objects to humans. Robots can fetch desired objects for the elderly living in their homes or hand tools to a worker in a factory. Recently, different aspects of this particular kind of physical human-robot interaction have received a lot of attention in robotics.

#### Human-inspired Robots

A common approach in literature is to take inspiration from human-human interactions to shape the behavior and the motion of the robot during robot-human handover. In [46] Kajikawa at al. propose a method which generates the motion for a receiver

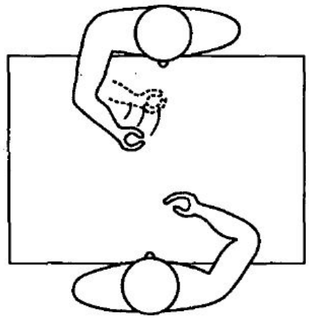


Figure 1.5: Handover operation (from [46]).

robot in a handover operation between a human and a robot. In this work it is assumed that the handover motion is performed in the horizontal plane as shown in Fig.1.5. In designing the controller, they aim at the generation of human-like motion, because that motion is supposed to be smooth, easily predictable and more natural for the human. First they analyze trajectories and velocity patterns of a handover motion performed by two humans. As the results, they notice that the receiver motion can be explained by some characteristics, summarized as follows:

1. The receiver starts his motion after he notices the beginning of a deliverer motion. So his start tends to be delayed. The delay is within $1s$.

2. At the start of his motion, the receiver approaches the deliverer with a straight and rapid trajectory, and without accurately determining the direction of his hand.

3. The receiver then adjusts the direction of his hand by generating a rotational trajectory.

4. Finally, the receiver sufficiently decreases the relative velocity to match that of the deliverer.

The authors assume that the receiver's motion can be divided into two different types:

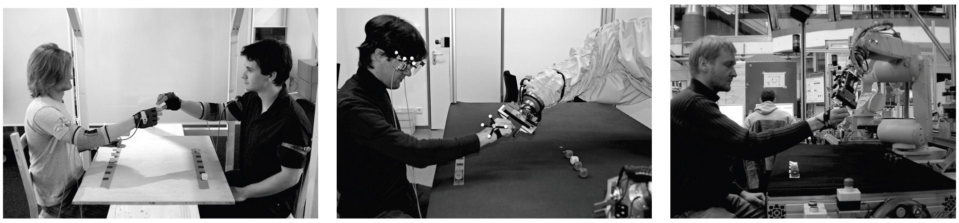


Figure 1.6: Handover experiments (from [47]): left: human-human (Experiments 1, 2, and 3), middle: human-humanoid robot (Experiments 4a and 4b), right side: human-industrial robot (Experiments 5a and 5b).

- Mode1 motion, which is more apparent in the first half of a receiver motion and includes characteristics 2 and 3.

- Mode2 motion, which appears markedly at the end of the receiver's motion, considers soft catching with reduction of the relative velocity as described by characteristic 4.

Next, the authors plan the robot motion based on this hypothesis. First, the Mode1 motion is produced using a potential field method. Then, the Mode2 motion is generated by formulating some kinds of boundary conditions. Finally, a smooth transfer between the two motions is considered. An off-line experiment performed using a measured data of a deliverer's motion has shown that the proposed method can produce a motion which contains similar characteristics to the trajectory and velocity generated by a human and thus emulates the human motion. However human-robot interaction experiments would be useful to confirm the effectiveness of the approach.

Another approach based on human-human interactions is reported in [47]. In [47] Glasauer et al. investigated how handover is executed by humans and how it can be transferred to robotic systems. First they investigated how a handover task between two humans is achieved (Fig. 1.6 left). Data analysis focused on reaction times and spatial handover positions. Reaction time was defined as the duration from lifting the object until the receiving subjects reacted by lifting their hand to grasp the object. While the spatial position of the handover remained almost constant, the reaction time

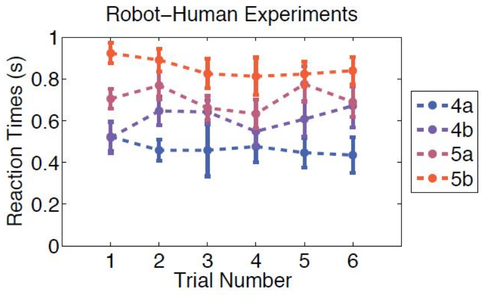


Figure 1.7: Reaction time in the human-robot handover experiments conducted in [47]). 4: humanoid robot, 5: industrial robot. 4a, 5a: minimum jerk profile; 4b, 5b: trapezoidal joint velocity profile.

significantly decreased over the course of 6 trials. To explain this result the authors suggest that a person observing human goal-directed movements, such as the delivery hand movement, is able to predict the endpoint of the movement. Therefore, to enable effective human robot interaction, the robot movement should be human-like in order to be predictable. To test the efficacy of such biological motion in a human-robot handover task (Fig. 1.6 middle), a humanoid robot was equipped with two different velocity profiles, a typical robotic motion profile (trapezoidal joint velocity [48]) and a human-like minimum jerk profile [49]. The comparison showed significantly shorter reaction times for minimum jerk profiles than for trapezoidal profiles (Fig. 1.7, experiment 4a and 4b). Adaptation over trials was only obvious for minimum jerk profiles. Finally, the human-robot handover was repeated with an industrial robot system (Fig. 1.6 right) to investigate the importance of the robot's appearance. As before, reaction times were shorter for minimum-jerk. However, average reaction time for the industrial robot was significantly longer than for the humanoid robot, suggesting that the robot's appearance plays a role in efficient joint action ( Fig. 1.7, experiment 5a and 5b).

Other research works that take inspiration from handovers between two humans include [50] and [51].

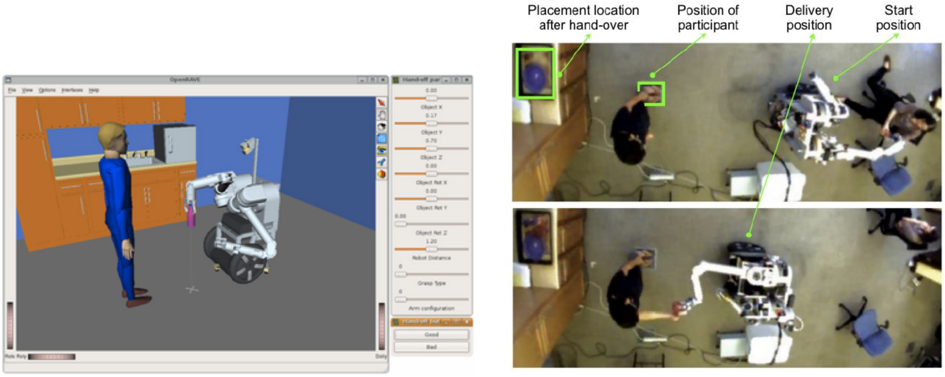
### Human preferences on robot behaviors

A different point of view is proposed in [11], where instead of taking inspiration from human-human interactions, it is evaluated how humans would prefer being handed an object by a robot. The paper presents a user study which consists of two parts. In the first part data on human preferences about handover configurations are collected and used to learn preferable handover configurations. In the second part the learned handover configurations are compared with configurations planned using a kinematic model of the human. A handover configuration is specified by three variables $C^r_{handover} = (P^r_{grasp}, C^r_{arm}, P^r_{base})$ where $P^r_{grasp}$ denotes the grasp pose of the robot's hand relative to the object, $C^r_{arm}$ denotes the robot's arm configuration and $P^r_{base}$ denotes the robot's position relative to receiver. In order to get input on how the robot should handover different objects the researchers carried out a user study where the participants were asked to give good and bad examples of handover configurations trough a graphical user interface (Fig. 1.8a). The interface provides sliders to change each degree of freedom of the handover configuration variables. The configurations provided by the users are used to choose the configuration for the hand over. Each configuration is evaluated based on how similar it is to good examples and how different it is from bad examples given by users. The value function is written as:

$$f = \frac{\frac{1}{|S_{good}|} \sum_{C_j \in S_{good}} d\left(C^r_{handover}, C_j\right)}{\frac{1}{|S_{bad}|} \sum_{C_i \in S_{bad}} d\left(C^r_{handover}, C_i\right)}$$

Here $S_{bad}$ and $S_{good}$ are the set of collected good and bad examples, and $d(C1, C2)$ is a similarity function defined between two configurations. It takes a maximum value of 1.0 when the two configurations are exactly the same and goes to zero as the configurations become dissimilar. Among available handover configurations the robot picks the one that maximizes this functions. Analyzing the good and bad examples configured by participants the authors make the following observations:

- the good examples given across participants are concentrated around few values of each variable. The distribution of examples is unimodal and has a small



(a) User interface for collecting good and bad examples of handover configurations.

(b) Setup for the human-robot interaction study for comparing two approaches.

Figure 1.8: User study conducted in [11]

variance. This indicates that preferences for the object configuration in the handover is similar across different people.

- The positions of the object in the good examples given by participants are often well chosen in terms of their reachability for the human model. Orientations, on the other hand, do not overlap as much. It turned out that for some objects, such as the shaker and the notebook, reachability of good examples is rather low. This points towards the necessity of a planning approach which makes sure an object is reachable to the human. In addition, bad examples given by participants are much less reachable than good examples. This shows that it is important for the robot to present an object in a reachable configuration for it to be considered a good handover.

- In good examples, the orientation of the object is the one in which an object is viewed most frequently in everyday environments (default orientation ). The percentage of default-orientation examples is much higher in good examples than in bad examples. This shows that participants generated bad examples by altering this property which they thought was important.

Moreover, in [11] a human-robot interaction experiment (Fig. 1.8b) was conducted to evaluate handover configurations obtained with the approach described by comparing them with configurations planned using a kinematic model of the human. In the experiment the robot delivers five objects to a user. Two instances of each object are delivered one after the other using the configurations generated with the two approaches. During the user study the robot was not able to perceive the user, who was supposed to stand in a specific place close to the robot. The task was completed by 10 right-handed participants (6 male, 4 female between the ages of 20-32). After the two deliveries, participants were asked to compare the two handover configurations by answering four questions:

- Liking: Which one did you prefer?
- Naturalness: Which one looked more natural?
- Practicality: Which one was easier to take?
- Appropriateness: Which one was more appropriate?

The results from the survey comparing the two approaches are summarized in Table 1.1. It turned out that the handover configurations learned from user examples are preferred over the configurations produced with planning in all dimensions. The difference in preferences is most significant for naturalness, which shows that humans' notion of a good handover configuration includes naturalness and the planning approach does not spontaneously produce natural-looking configurations. While a preference was not found for configurations produced with planning in terms of practicality, this was the dimension that was least in favor of learned configurations. However analysis of the videos showed better reachability for objects presented with planned configurations.

Human preferences in robot-human handover have been also investigated in [52] and [53]. In [52] the authors analyze human preferences on the robot's handover behaviors in terms of the approach direction as well as height and distance of the object. In [53] Jindai et al. investigated human preferences concerning the movement velocity of their partners, yielding very low values for preferred peak velocity.

| Criteria | Bottle | Mug | N.book | Plate | Shaker |
|---|---|---|---|---|---|
| Liking | 7 | 8 | 4 | 6 | 6 |
| Naturalness | 6 | 8 | 6 | 7 | 5 |
| Practicality | 5 | 7 | 4 | 5 | 6 |
| Appropriateness | 8 | 7 | 4 | 6 | 6 |

Table 1.1: Comparison of two approaches on survey questions for individual objects conducted in [11]. Number of participants out of 10 who preferred learned configurations are given.

Another common approach in studying robot human handovers is to use a kinematic model of a human. Different aspects of handover interactions have been studied with this approach, including motion control [54, 55, 56], grasp planning [57, 58] and grip forces to be applied during handover [59].

### Handover scenario

Besides the study of how a handover is performed another important aspect is how it can be exploited. In [60] Edsinger at al. present a robotic application that relies on this form of human-robot interaction. Domo, the robotic platform used for the experiments, detects when an object has been placed in its hand, attempts to grasp the object, and then detects whether or not the grasp has been successful. Domo also detects when the user attempts to acquire an object from its grip. In addition Domo is able to to find a person in the room and to compute his location. The authors first demonstrate that subjects without explicit instructions or robotics expertise can successfully hand objects to a robot and take objects from a robot in response to reaching gestures. Moreover they found out that, when handing an object to the robot, subjects control the object's position and orientation to match the configuration of the robot's hand, thereby simplifying robotic grasping and offering opportunities to simplify the manipulation task. Then an example scenario for cooperative manipulation is proposed to illustrate the utility of the robot behaviors:

1. Domo is positioned at a table cluttered with objects and near a shelf. Domo first physically verifies the location of the shelf.
2. A person asks for help in preparing a drink. He hands Domo a cup and bottle of juice. Domo pours the juice into the cup.
3. Domo hands the bottle of juice back to the person.
4. The person now hands Domo a spoon. Domo inserts the spoon into the cup and "stirs" the drink.
5. Domo hands the spoon back to the person and then places the prepared drink on the shelf.
6. Next, the person asks for help in putting away groceries. He hands Domo a box of crackers. Domo passes the box to the other hand and puts them upright on the shelf.
7. The person hands Domo a paper bag of coffee and Domo places it on the shelf as well.
8. Now, the person asks for help in clearing off the table. He hands Domo a box and Domo grasps it with both hands.
9. Domo keeps the box near the person as he goes about clearing the table into it.
10. Finally, the task is done and Domo lowers the box onto the table.

As shown in Fig. 1.9, a very similar scenario was realized by Domo and the author as one consecutive task, punctuated by vocal requests for the robot, over the course of 5 minutes. Of course, other scenarios are possible using this approach. For example, Domo could assist a person working on an assembly line by holding a tool tray for the person, putting tools away, holding a tool and then handing it back when the person is ready, and performing the insertion of two parts during assembly.

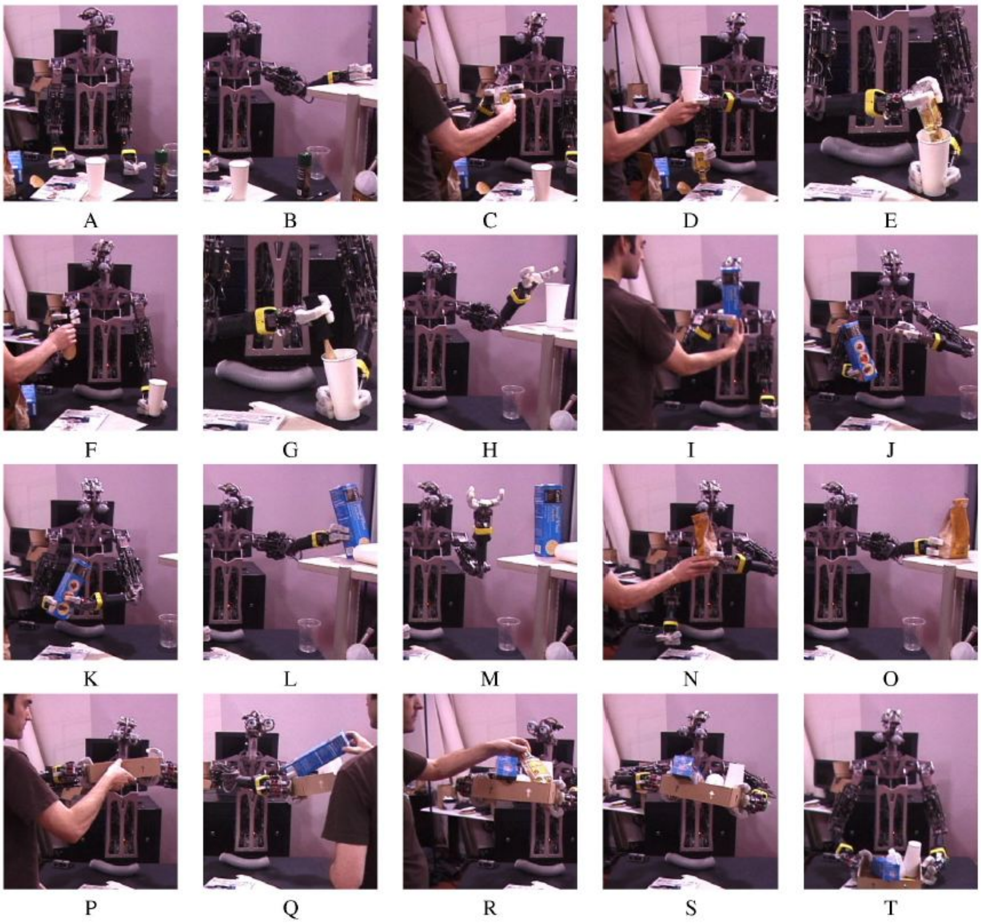
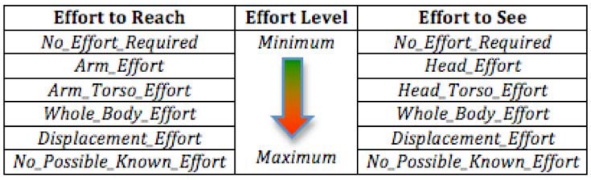


Figure 1.9: In this sequence, Domo assists in a variety of manual tasks. (A) Domo begins at a cluttered table. (B) A shelf appears and Domo verifies its location. (C-D) A juice bottle and cup are handed to Domo. (E) Domo visually guides the bottle into the cup. (F-G) Now, Domo is handed a spoon and it “stirs” the drink. (H) Domo puts the finished drink on the shelf. (I-L) A box of crackers is handed to Domo’s right hand. It transfers them to the left hand and places them on the shelf. (N-0) A bag of coffee beans is handed to Domo. It then puts the bag on the shelf. (P) Domo grasps on a box. (Q-R) Domo keeps the box near the person as they clean up the table and put items in the box. (S-T) Finally, Domo lowers the box onto the table (from [60]).

### Human-Aware Robots

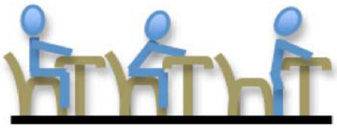
Although all the works cited above analyze important aspects of human-robot handover and provide valuable insights for the development of socially interactive robots, they rarely present systems that are exploitable in real environments. In real environments the robot should be able to accurately perceive the human partner in order to approach him and synchronize its motion with the human motion. Human detection and human motion tracking are complex tasks. Traditionally these tasks have been achieved in a reliable way using intrusive trackers (such as magnetic trackers or special gloves ), while camera-based solutions have not guaranteed high reliability. Most of the systems presented above do not perceive the human partner in real time, use intrusive tracking systems to perceive the human, or have a weak perception of the human. Systems that have a weak perception of the human can work only if the majority of the handover burden is put on the human partner. In this case, as shown in [60] humans take the object from the robot and position and orient the object into the robot's hand, while the robot only needs to show that it is ready to interact. This strategy works well because it is an easy task for humans, but if humans have impairments, or the hand over occurs while they are focusing their attention on a secondary task, the robot needs to participate actively to complete the handoff. The recent advancement in computer vision can enable robotic systems to better perceive humans therefore enhancing their role in human robot interaction.

In [61] Pandey at al. present a robotic system for human-robot interaction that exploits the data provided by a perception module to build a model of the human partner and uses this model to behave proactively. The framework predicts *where* the human can perform a particular task and compute how the robot could support it. The paper shows how such proactive behaviors reduce the human's confusion and effort as well as how the robot seems to be more aware about the task and the human. The example scenario used to show the potential of the system includes a hand over task where the robot asks to a human to hand it a toy dog. The robot tries to proactively support the task minimizing the human's effort. The efforts have been categorized as shown in Fig. 1.10a. The classification is motivated from the studies of human movement and behavioral psychology [62], where different types of reach actions of the human have



| Effort to Reach | Effort Level | Effort to See |
|---|---|---|
| *No_Effort_Required* | *Minimum* | *No_Effort_Required* |
| *Arm_Effort* | | *Head_Effort* |
| *Arm_Torso_Effort* | | *Head_Torso_Effort* |
| *Whole_Body_Effort* | | *Whole_Body_Effort* |
| *Displacement_Effort* | | *Displacement_Effort* |
| *No_Possible_Known_Effort* | *Maximum* | *No_Possible_Known_Effort* |

(a) Effort classes for visuo-spatial abilities.



(b) Taxonomy of reach actions:(a) arm-shoulder reach, (b) arm-torso reach, (c) standing reach.

Figure 1.10: Human effort Classification for reach actions presented in [61].

been identified and analyzed, as shown in fig. 1.10b.The classification includes reach involving simple arm extension (arm-only reach), shoulder extension (arm-shoulder reach), leaning forward (arm-torso reach) and standing reach. The robot aims to compute the object exchange point that is reachable for both the human and the robot and minimize the effort of the human. Starting from the minimum effort level for the human, the robot executes the following steps:

1. Computes all the points that the human can reach with his current effort level (Figures 1.11 and 1.11b).

2. Finds the points in the candidate points obtained for the human for his current effort level, which are also reachable and visible by robot (Fig. 1.11c).

3. Then it assigns weights to the resultant candidate points. The weights are assigned based on the closeness to the target-object position, with the hypothesis that a human needs to put less effort in giving the object to the robot if he has to carry the object for a shorter distance (Fig. 1.11d).

4. Then, starting from the highest weight candidate point, the robot finds a possible collision free placement configuration of the object at that point.

5. For that placement of the object, the robot finds a grasp for that object. Since it is a task of handing over an object, it ensures that the selected grasp allows the human to grasp the object simultaneously.

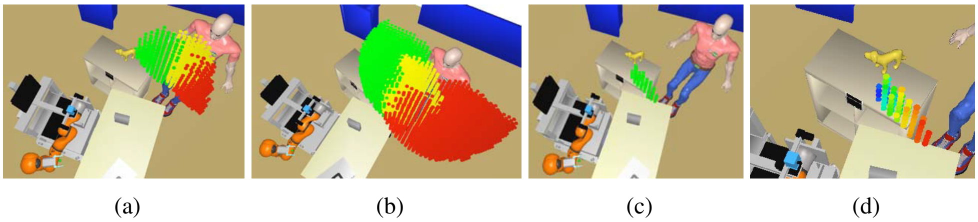


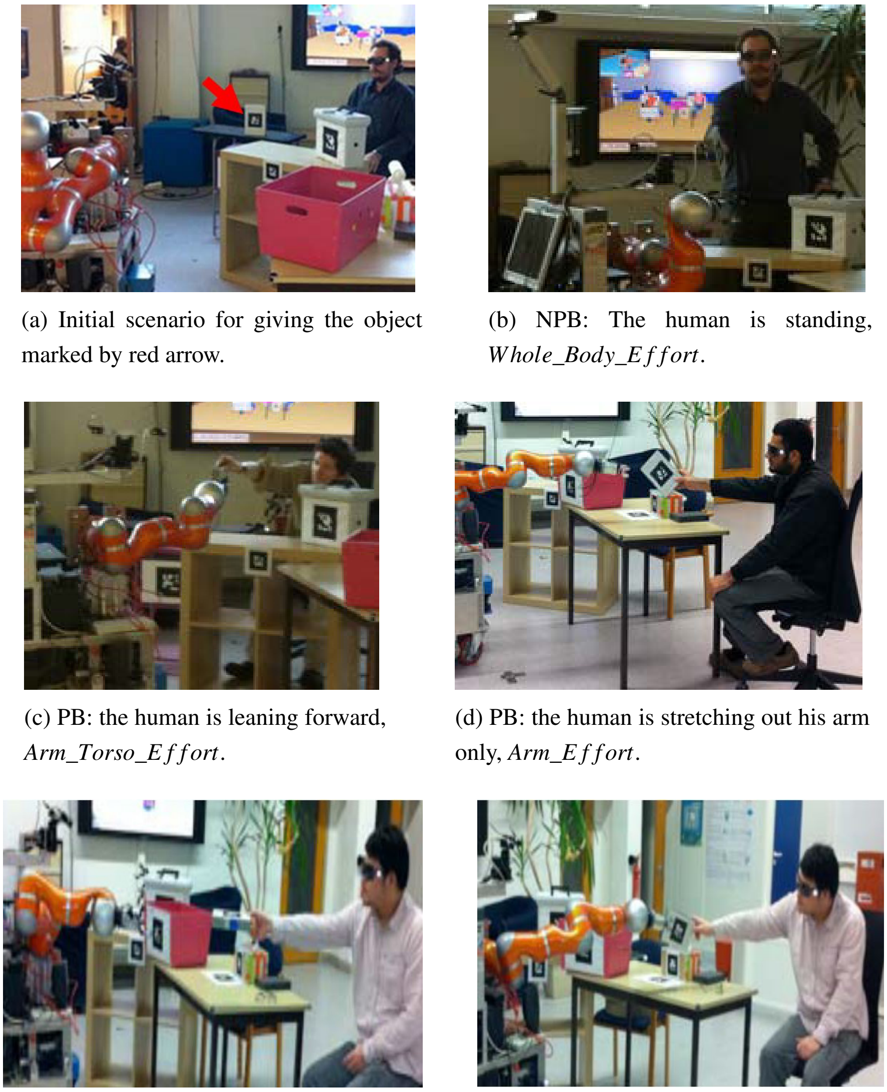
Figure 1.11: From [61]. Candidate points for giving an object by the human (green: by right hand, red: by left hand, yellow: by both hands) (a) from his current position, (b) if the human will make effort to move (lean forward or turn) his torso while remain seated. (c) Candidate points from where the robot can take the object for the effort level of (b) for the human, (d) weight assignment on the candidate points based on the closeness to the target-object, toy dog.

6. Then depending upon the environment, collision, visibility constraints, etc., the robot just filters out the unwanted placements and grasps by putting the object at a particular point in a particular configuration.

7. The robot then tries to find a collision free path for reaching to that point.

8. If a particular candidate point passed all the feasibility tests for a particular task, then that particular point is considered to show the proactive behavior and the smooth trajectory is generated for execution. Otherwise the level of effort for the human is increased and the algorithm starts again from 2 , considering the new effort level.

The environment is actively monitored during execution and if the human's intention or attention has changed or the human is carrying the object away from the current feasible point, the exchange point and the trajectory of the robot arm can be re-planned. The object handing-over movement is identified by the robot through its touch and force sensors associated with the gripper, which triggers to close the gripper for taking the object. The robotic system has been tested through a user study. The robot uses an integrated planning and visualization platform and through its various



(a) Initial scenario for giving the object marked by red arrow.

(b) NPB: The human is standing, $Whole\_Body\_Effort$.

(c) PB: the human is leaning forward, $Arm\_Torso\_Effort$.

(d) PB: the human is stretching out his arm only, $Arm\_Effort$.

(e) NPB: this particolar user is holding the object and waiting for the robot to take.

(f) PB: the same user in (e) is putting effort to give.

Figure 1.12: Experiments conducted in [61].

sensors maintains and updates the 3D world state. For object identification and localization it uses a tag-based stereovision system. For localizing human it uses data from Kinect motion sensor mounted on it. In the user study each user has been exposed to two different behaviors of the robot:

- Non Proactive Behavior (NPB): Robot just asks to the user “Please give me the <object name>” and waits in its current state;

- Proactive Behavior (PB): robot asks the same but also starts moving its arm along the trajectory returned by the proactive planner.

The order to exhibit PB or NPB to a particular user was random and there were a total of 12 participants. After being demonstrated to both behaviors, each user was requested to fill a questionnaire. The overall response was that with proactive behavior the human was in less confusing states and also the human effort compared to non-proactive behavior was reduced (Fig. 1.12). Also users have reported that the robot seems to be more aware about the users’ capabilities in the cases it behaved proactively.

Apart from the observations from the direct responses from users, the authors found the following interesting observations:

- For the cases where proactive behavior of robot has been demonstrated first, users seem to be biased towards expecting similar behavior for the later demonstration in which non-proactive behavior has been demonstrated. In such cases users’ responses were: “I thought that experiment had failed, since the robot didn’t move”, “I was waiting for the robot to take it from me” (Fig. 1.12e).

- For the cases where non-proactive behavior has been shown first, even if the robot has asked to give the object by name, some users have been found ‘searching’ for the object to give, if the table-top environment was somewhat cluttered. This suggests that such proactive behaviors also help in fetching the human’s attention to the object of interest.

However further user studies are absolutely required to validate these hypotheses. Another work that takes into account the human presence in order to enhance human-robot interaction is [63]. The planner proposed, which is applied into “robot handing over an object” scenarios, breaks the human centric interaction that depends mostly on human effort and allows the robot to take initiative by computing automatically where the interaction takes place, thus decreasing the cognitive load of interaction on human side. To achieve this objective a three-stage approach is adopted. The approach consists in the following steps:

1. Choosing Object Transfer Point (OTP): The planner finds a safe and comfortable place for the human to receive the object.

2. Calculating Object Path: From the robot hand current position to OTP, a path for the object is found as it is a free flying body.

3. Generating Robot Path:With the object path obtained, the planner finalized the process by generating robot motion that will follow this path. The robot can take the initiative reaching out to OTP.

In this human-aware manipulation planner, three different interaction properties, which are called “safety”, “visibility” and “human arm comfort” are represented as grids with their corresponding cost functions and are used to determine the OTP. The cost of a point in the safety grid represents the measure of safety for the object placed in that particular point. The farther the object is placed from human, the safer the interaction is. The safety cost function $f_{Safety}(H,i,j,k)$ is a decreasing function according to the distance between the human $H$ and object coordinates $(i,j,k)$ in the grid (Fig. 1.13). The visibility property is represented by a visibility cost function $f_{Visibility}(H,i,j,k)$. This function alone represents the effort required by the human head and body to get the object in his field of view. If the object is placed directly in front of the human, as the object is completely visible and no effort is required, the resulting cost of objects placement will be null. On the contrary, when placed behind the human, as in order to see that object the human needs to turn his head and his body, the effort is higher, and thus results in a higher cost (Fig. 1.14a). The last property of the placement of the object is the comfort of human’s arm configuration when he/she tries to reach to the object. It is a key notion to take into account for a comfortable handing over motion. The robot should reason about human’s accessibility

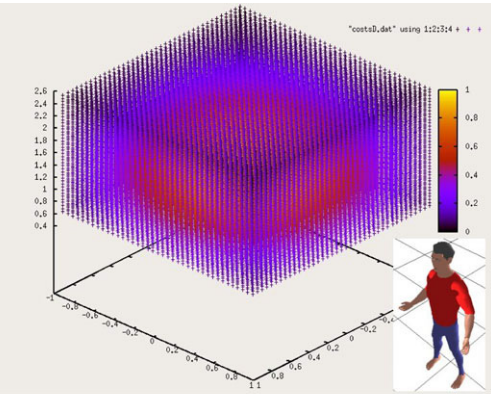


Figure 1.13: Costs of safety function mapped around the human. The human is placed at the center of this grid but illustrated at the lower corner for the clarity of the figure. This function creates a protective bubble around the human where costs increase when approaching the person (from [63]).

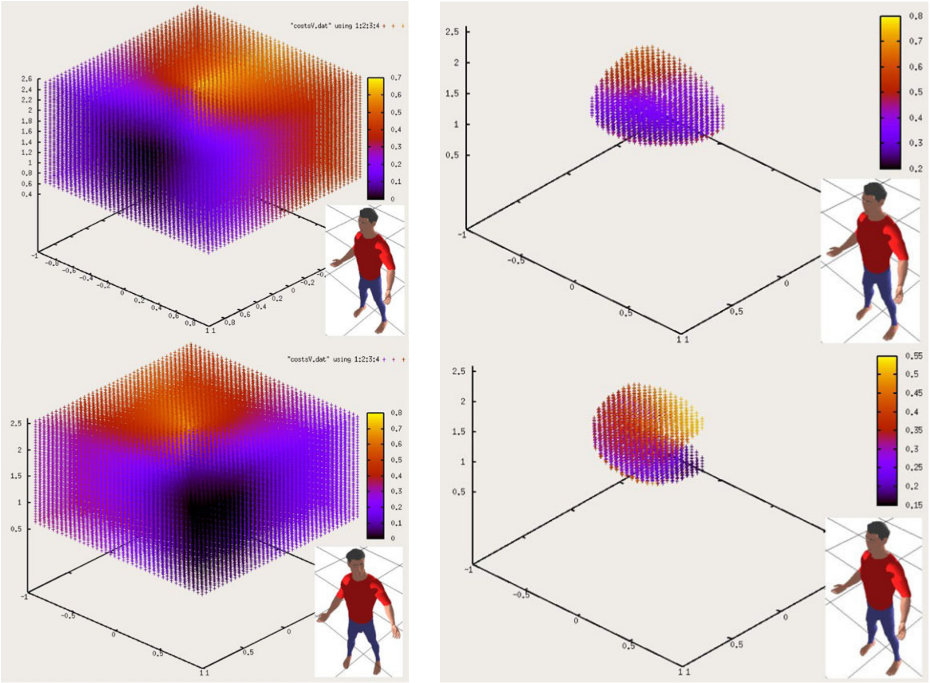
and his kinematics to find an OTP which is not only reachable by the human but is comfortable to reach as well. This property is represented by a function $f_{AC}(H,i,j,k)$ that takes into account the angular change in human arm's degrees of freedom (DOF) with respect to the rest position, and the potential energy of the arm when performing a reaching motion (Fig. 1.14b).

In order to find the OTP, the cost functions mentioned previously are combined to form a single cost function

$$\begin{aligned} f_{OT}(H,i,j,k) =& w_{Safety} f_{Safety}(H,i,j,k) \\ &+ w_{Visibility} f_{Visibility}(H,i,j,k) \\ &+ w_{ArmComfort} f_{AC}(H,i,j,k). \end{aligned}$$

With the weighted sum of all three functions, the costs in the final function obtain a balance between safety, visibility, and human arm comfort. In order to find the OTP, $f_{OT}$ is mapped around the human to form the object transfer grid, $G_{OT}$. The cells in this grid are scanned and the cell with the minimum cost is assigned to be the OTP

$$OTP = ((i,j,k) | \min_{i,j,k}(f_{OT}(H,i,j,k))).$$



(a) Costs of visibility function distributed around the human. The human is placed at the center of this grid but illustrated at the lower corner for the clarity of the figure. Points where the human has difficulty to see have higher costs.

(b) Arm comfort function for a left-handed person. Although the shapes of left and right arm functions are the same, a penalty is applied to the right arm, thus increasing its costs. Note that only the accessible and more comfortable points are illustrated. Other points around the human have the highest costs in this grid.

Figure 1.14: Visibility and arm comfort cost functions (from [63]).

Note that the function $f_{OT}$ depends on the OTP and the human representation $H$ which contains the kinematic structure of the human, his configuration, as well as his states (e.g., sitting/standing). $H$ is obtained exploiting the robot perception system. As the target position of the object is found in the previous stage, in the second step, a path connecting the object's actual position (robot's hand) and final position (OTP) will be found. The object is considered as a free flying body. In order to compute the path, the object path grid $G_{ObjectPath}$ is built by combining $f_{Safety}$ and $f_{Visibility}$ cost functions. As the human will not reach the object during its motion, the function $f_{AC}$ is not considered in the process. With this definition, the object path grid represents a combination of visibility and safety grids. After its construction, a $3D\ A^*$ search with diagonal distance heuristic is used to find a minimum cost path that will be safe and visible at the same time.

Even though a path for the object (and robot's hand) has been found, it is not enough to produce an acceptable robot motion in HRI context where the motion should be safe, comfortable, and "legible". With this motion, the robot must make clear its intention. The third and final stage of planner consists of finding a path for the robot that will follow the object's path. The object's path is computed as if it was a free flying object. However, in reality, it is the robot who holds the object and who will make the object follow its path. The algorithm that computes the robot arm path tests the robot posture against collision with the obstacles, with itself and with the whole human body. In addition it takes into account the robot's gaze to increase the legibility of robot's motion by expressing explicitly its intention by looking at the object. At the end of this stage, a path is obtained for the robot which is safe, visible, and comfortable to the human. The planner generates and sends the path in the form of successive configurations to the execution modules. Once the robot reaches its final position, it waits for the human to pull the object. In addition, a supervision system detects if the human is moving his hand toward the robot. If that is the case, the robot overwrites its calculated OTP and moves toward the person's hand.

The authors of the paper report that the planning is performed in about 6 $s$ on a Pentium 4 3.2-GHz computer. However they believe that planning algorithms can be drastically optimized to allow shorter planning times.

A user study has been conducted in order to set up an objective evaluation of robot's motion behavior synthesized by the human-aware manipulation planner proposed. In this study, three different ways of handing over an object are evaluated according to electromyogram data from 12 users. The subjects have been asked to sit on a chair placed in front of the robot (Fig. 1.15). The study begun with the robot holding a bottle in its resting position with the arm folded. Then it performs a handover motion.

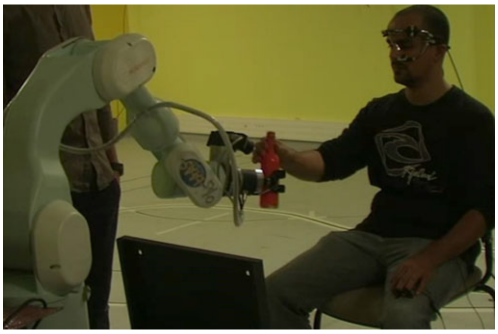


Figure 1.15: Experimentation setting (from [63]).

The three different motions evaluated are:

1. a motion planned by the human-aware planner with moderate velocity and force detection;

2. a faster no human-aware path motion (the robot performs a reaching gesture without taking into account the actual human position and configuration);

3. a motion planned by the human-aware planner with slow velocity and without force detection during the execution of the trajectory.

Motion 1, which is the human-aware motion, has appeared to be the one that requires the minimum amount of effort of human arm (electromyogram results). In the same experiment, a questionnaire-based survey has been conducted where subjects were asked to evaluate each motion's predictability and safety along with the overall physical effort. The findings showed that motion 1 has been distinguished as more legible, safe, and comfortable than the two others. On the contrary, motion 2 was subjectively assessed as the most unsafe. Motion 3 was ranked as the least physically comfortable and the least legible for the subjects since its low velocity and the inhibition of the force sensor led the participants to struggle prematurely with the robot to get the bottle.

As shown in this section various aspects of human-robot handover have been studied in literature. In recent years the advancement of technology has led to robots that are more safe and human-aware. These characteristics can enable future robots to actually interact with people. The researches conducted in [61] and [63] are preliminary studies and address only part of the many challenges that need to be addressed in order to have robots that behave actively during human-robot interaction and are accepted by humans. However they show how having an accurate perception of the human partner can enable the design of robotic systems that take into account human presence and behavior, therefore enhancing the abilities of the robot.

## 1.3 Hiring Social Robots

While human interactive robots have to evolve more to populate our homes and offices, they already packed their luggages and left research labs to apply for jobs in factories. This is the case of Baxter [64] (fig. 1.16). Whereas traditional industrial robots perform one specific task with superhuman speed and precision, Baxter is neither particularly fast nor particularly precise. But it is able to perform any job that involves picking stuff up and putting it down somewhere else while simultaneously adapting to changes in its environment, like a misplaced part or a conveyor belt that suddenly changes speed. In addition, Baxter is designed to be inherently safe. With their fast, powerful motors and hefty limbs, industrial robots are typically kept fenced off from people. Baxter is limited speed and lower weight (about 75 kilograms or as much as an average adult man) meaning that it can operate right alongside human workers. There are two other major barriers to the adoption of industrial robots that Baxter's creators want to overcome: ease of use and cost. As for the first, Baxter does not rely on custom programming to perform new tasks. Once it is wheeled into place and plugged into an ordinary power outlet, a person with no robotics experience can program a new task simply by moving Baxter's arms around and following prompts on its user-friendly interface (which doubles as the robot's face). Instead of actually programming the robot, it is simply shown what to do. To show Baxter how to take an object out of a box and put it on a conveyor belt, you start by grabbing the robot by the wrist to get its attention. Baxter will stop whatever it is doing and look at you with the calm, confident eyes displayed on its LCD. You then move the arm over to the box and use buttons and a knob on the arm to navigate a series of menus on the LCD, telling the robot to use its vision to find the object. Finally, you move the arm over to the conveyor and push some more buttons to let Baxter know that this is where you want the object dropped off. Baxter even nods its head, as if to say, “I get it”. Pressing the play button makes the robot execute the task on its own.

Furthermore, while a traditional two-armed robot will typically cost hundreds of thousands of dollars (including sensors and programming), Baxter costs just $22000. To achieve that, the robot was designed from scratch. Underneath Baxter’s plastic exterior lie thousands of ingeniously engineered parts and materials that enable the robot to do what it does for the cost of a midsize car. One of Baxter’s key features is compliance. A robot is said to be compliant when it is not completely rigid and when it can sense and control the forces it applies to things. Through compliance Baxter can get feedback during the execution of tasks and be safe around humans.

Baxter is not the only human interactive robot designed to work in manufacture. Swiss-Swedish giant ABB has developed a dual-arm prototype, reportedly for assembly applications. Japanese firm Kawada Industries has a similar robot named Nextage. They might not cost as little as Baxter, but they will likely be able to perform high-precision tasks that Baxter can not do, like assembling electronics boards, another potentially huge market for robotic automation. Other start-ups are also looking to enter the low-cost robotics market. Robots are ready to work with us and are looking forward to living with us.

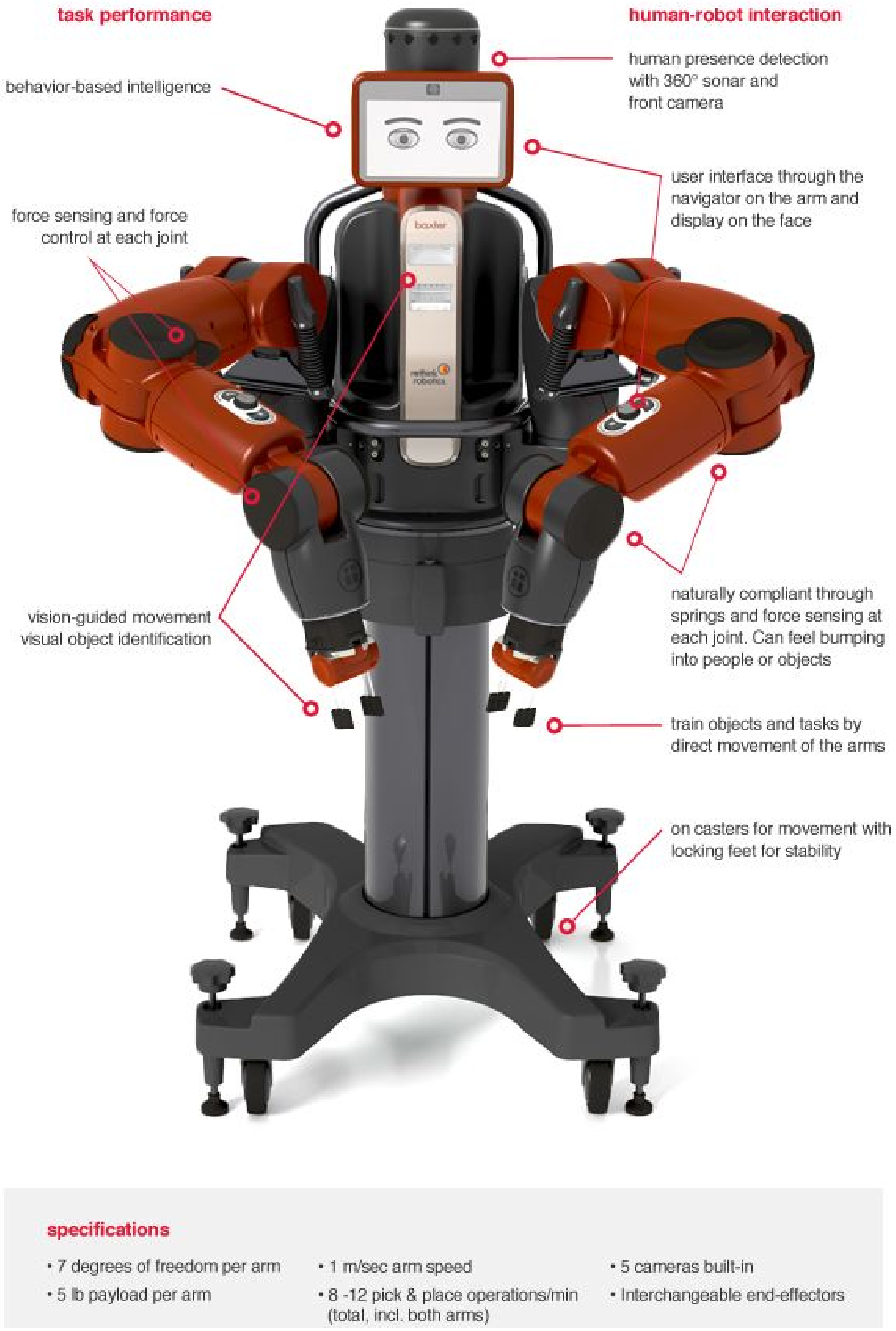

task performance
behavior-based intelligence
force sensing and force control at each joint
vision-guided movement visual object identification
human-robot interaction
human presence detection with 360° sonar and front camera
user interface through the navigator on the arm and display on the face
naturally compliant through springs and force sensing at each joint. Can feel bumping into people or objects
train objects and tasks by direct movement of the arms
on casters for movement with locking feet for stability
specifications
• 7 degrees of freedom per arm
• 5 lb payload per arm
• 1 m/sec arm speed
• 8 -12 pick & place operations/min (total, incl. both arms)
• 5 cameras built-in
• Interchangeable end-effectors

Figure 1.16: Baxter main functionalities.

# Chapter 2

# Development of Human-Robot Physical Interaction Tasks

The research activities described in this dissertation have been focused on the field of physical human-robot interaction. In order to investigate issues arising in the physical interaction between a human and a robot I developed robot prototypes that were able to hand over objects to humans and receive object from humans. Handing over objects to a partner is a very common task which is necessary for diverse interactive and collaborative activities. Furthermore, the approaches used to enable human-robot handover and the findings from user studies can be used as a baseline to develop effective human-robot interaction systems.

## 2.1 Handoff of an object between human and robot

By handoff (or handover) we mean the transfer of an object from a giver to a receiver. Handoffs are an important function in everyday life. For humans, handoffs are an easy task and are usually routine rather than deliberative. Even if humans perform several handoffs on a daily basis with different types of objects and in different situations and roles, they usually cannot remember how exactly they performed the handoff. This fact is an indication of how routinary and innate handoff tasks are for humans. In addition, motions of the giver and the receiver are often synchronized and they work together until the transfer of the object is complete [51]. It turns out that such an easy task for humans is hard for robots. In order to complete the handoff a robot should perform several challenging subtasks, such as detecting when to give or receive the object (e.g. performing gesture recognition), identifying the object and synchronizing its movement with the partner without hurting the partner itself. Depending on the shape of the object and the movements of the partner the robot needs to compute sophisticated trajectory plans to complete the handoff. Each of these subtasks is itself an active area of research, and yet all must be accomplished simultaneously to produce a good handoff system.

I address this task because it is a common and highly collaborative task and I want to investigate how the recent availability of technologies that provide advanced human motion perception, such as Microsoft Kinect, enable the development of new robotic systems for effective human-robot collaboration. The Kinect is a low cost RGB-D camera that records two images, a RGB image and a depth image. These images can be used by the software bundled with the Kinect to detect and track humans in a fast and reliable way. This new technology enables the active participation of robots in human-robot handoffs, previously unfeasible, by improving the perception capabilities of robots. However, Kinect-based human tracking is not perfect and is sensitive to fast human movement, non-frontal views of humans, and situations where the human is near other objects. Thus, while the Kinect provides improved perception to the robot, it is far from providing the same high-quality perception available to human.

In addition to study the technical solution that enable advanced human-robot interaction I aim to investigate how people feel when they physically interact with a robot that is able to perceive their presence and their motion and uses this information to behave proactively.

In this chapter two *human-robot handoff systems* developed during this thesis work are presented. The first system enables *human to robot* handoffs, the second one enables *robot to human* handoffs. In Both systems robots are *human aware*.

## 2.2 Robots receiving objects from humans

Humans perform handoffs on a daily basis without any difficulty. What happens when one of the humans is replaced by a robot? Due to the difficulty to reliably detect humans and track their motion, until few years ago it was hard for robots to actively participate in handoffs. In order to complete the task, humans had to take the most of the burden completing the handoff while the robot waited for the human to accomplish the operation. When receiving object from a human, robots performed reaching gestures and let the human push the object into the robot's stationary hand fixing the pose of the object to match the configuration of the robot's hand [60]. Although this strategy works well because it is an easy task for humans, it needs the human partner to be focused on the handoff and is not effective in situations where the human needs help from the robot to achieve the task. This could be the case of a worker handing an object to the robot while performing a job on a ladder or a human with arm impairment that is handing a glass to the robot in order to put it in the dishwasher. Furthermore the exchange of an object is a joint interaction and needs the partners to work together in order to be effective.

Recently, thanks to advancements in computer vision, robots have started to take into account the human presence and motion. In the case of human to robot handover, robots compute the object transfer point that minimizes the human effort and reach out there [61]. However, these robots never take the object but, once the transfer location is reached, they always wait for the human to push the object into their hand. This behavior is effective and appropriate in many situation, but sometime it could be useful if the robot took the initiative and the responsibility to take the object from the human hand. I developed a handover system where the robot actively takes the object from the human partner. This behavior is enabled by a reliable and accurate perception system. I addressed a handover scenario where the human initiates the interaction, therefore the robot has to infer the human intention to hand off an object before starting the reaching motion. In addition, I carried out a user study to assess the effectiveness of the system and to evaluate how humans feel when interacting with a robot with such behavior. Moving the robot arm close to the human up to the

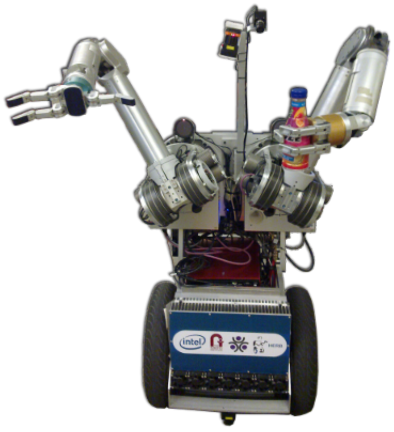


Figure 2.1: HERB, the Home-Exploring Robotic Butler.

contact between the two, can raise on the human side reactions that need to be studied in order to develop robotic behaviors that are accepted by humans.
This section presents my progress towards a working human to robot handoff algorithm, where the robot actively takes the object from the human [65]. Section 2.2.2, describes how the robot senses what the human is doing and computes where to reach to take the object. This includes compensating for noise from the human tracking software, detecting handoff intent, and detecting objects in the human's hand. In Section 2.2.3 I describe two different robot control methods. In Section 2.2.4 I outline an informal study that was carried out to test the algorithm. Finally, in Sections 2.2.5 I discuss the results of the user-study and what I learned from this work about human-robot handoffs.

### 2.2.1 The Framework

For this work I used HERB, the Home-Exploring Robotic Butler (Fig 2.1) [66, 1]. HERB is a robot from the Intel Personal Robotics Lab and a research platform at Carnegie Mellon's Quality of Life Technology Research Center. HERB, in the current version (Herb 2.0), has two Barrett WAM arms mounted on a Segway base that enable it to move around an environment and perform advanced manipulation tasks.

HERB has a suite of onboard sensors to help it perceive the world, including a spinning laser scanner for building 3D world models, a vision system for object recognition and pose estimation [67], and a commercial system for indoor localization. In addition to these sensors the robot has a pair of low-power onboard computers. Onboard components communicate over a wireless network with offboard off-the-shelf PCs.

A key challenge for robot systems in the home is to produce safe goal-driven behavior in a changing, uncertain and dynamic environment. A complex system like HERB, that has a host of sensors, algorithms, and actuators, must address issues ranging from software robustness (sensors or communication failing, processes dying) to problems that emerge from inaccurate or unknown models of the physical world (collisions, phantom objects, sensor uncertainty). To address this challenge, HERB uses a software architecture loosely based on the sense-plan-act model for providing safe and rich interactions with humans and the world. Figure 2.2 shows the interaction between the different components of the robot architecture: perception, decision and execution components.

The robot exploits its sensors to gather information about the world in the form of fixed and dynamic objects, agents (humans and other robots), and semantics (e.g. HERB's location in the home).

HERB has three classes of components that can make decisions: safety, agent, and teleoperation components. Safety components ensure that the robot does not harm humans, the environment, or itself. Some examples of safety components include the limitation of forces that the arm is allowed to exert, or the limitation of joint velocities. Safety limits cannot be overridden by other components. Agent components try to accomplish goals based on the perception of the world, including manipulating objects, expressing intent via gestures, and physically interacting with humans. The teleoperation components enable users to explicitly command the robot, both at a low level (e.g. moving single joints) or at a high level (e.g. grasping a particular object). These components can override agents, but cannot override safety components.

Finally, the execution components perform actions to accomplish the tasks commanded by the decision components. This includes planning and executing arm and

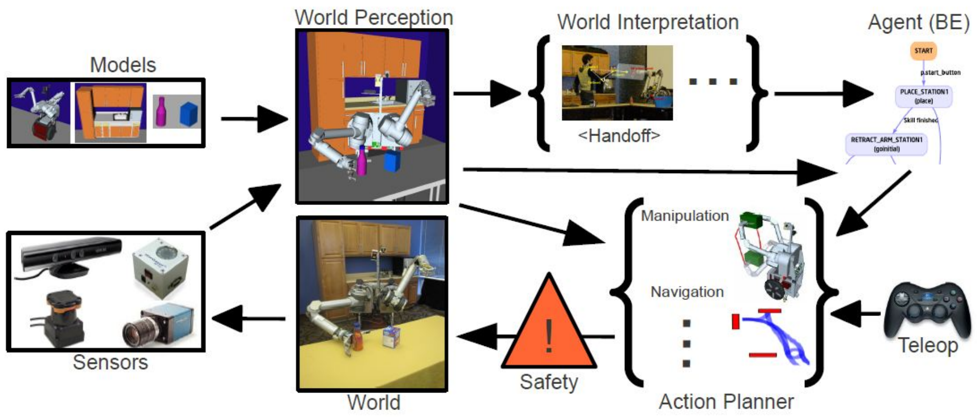


Figure 2.2: HERB 2.0 system architecture.

base trajectories, making sounds, and interacting with other computer systems.

HERB works across many computers and employs several distributed robotics packages to accomplish its tasks. The entire HERB system consists of a group of separate processes that communicate with the others through the network (Fig 2.3). The *Robotic Operating System* (ROS) package [68] is used for the communication infrastructure and process management. ROS allows us to easily transfer processes onto different computers as necessary. When deciding where and how each algorithm should be computed, as much computation as possible is moved to dedicated computers off the robot. The onboard computational power is always limited due to weight and power constraints, so it should be used for real-time tight-feedback processes only. The design space for computation is tricky because the onboard and offboard computation are separated by a wireless network and bandwidth/latency become an issue. In HERB, the execution layer lies on-board the robot because the arm movement and Segway navigation require tight feedback loops at rates greater than 10 Hz. The sensing component is divided between onboard real-time obstacle avoidance and offboard perception. The onboard camera data is compressed and streamed offboard to construct a snapshot of the environment. The manipulation planning algorithms producing global plans are strictly offboard since each planner returns a new

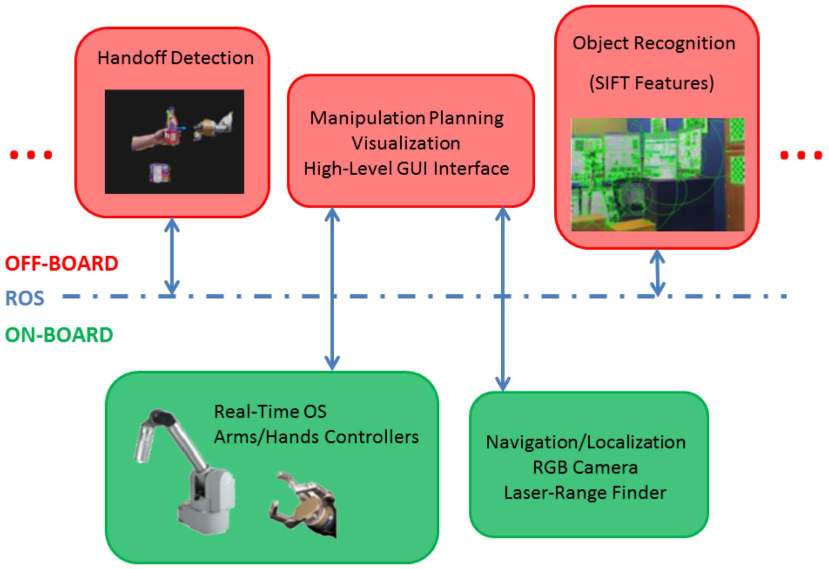


Figure 2.3: Distribution of computing resources. Each box represents a dedicated computer.

trajectory on the order of seconds.

### Scenario

HERB typically works in a domestic kitchen environment, as shown in Fig. 2.4. In this work, HERB is positioned next to and facing a table which has two types of items on it: Pop-Tarts boxes and Fuze bottles. HERB has already demonstrated its ability to recognize and manipulate these objects autonomously [66], while avoiding humans safely. In this work, the goal is to explore how HERB can actively collaborate with a human to accomplish the handoff task.

A Microsoft Kinect camera is located $2.5m$ from the table in order to have a complete view of the scene. Images from the camera provide two types of data at $30Hz$: raw depth and color, and human tracking data. The human tracking data is a 14-point skeleton outlining a human's pose from head to feet (Fig. 2.7).

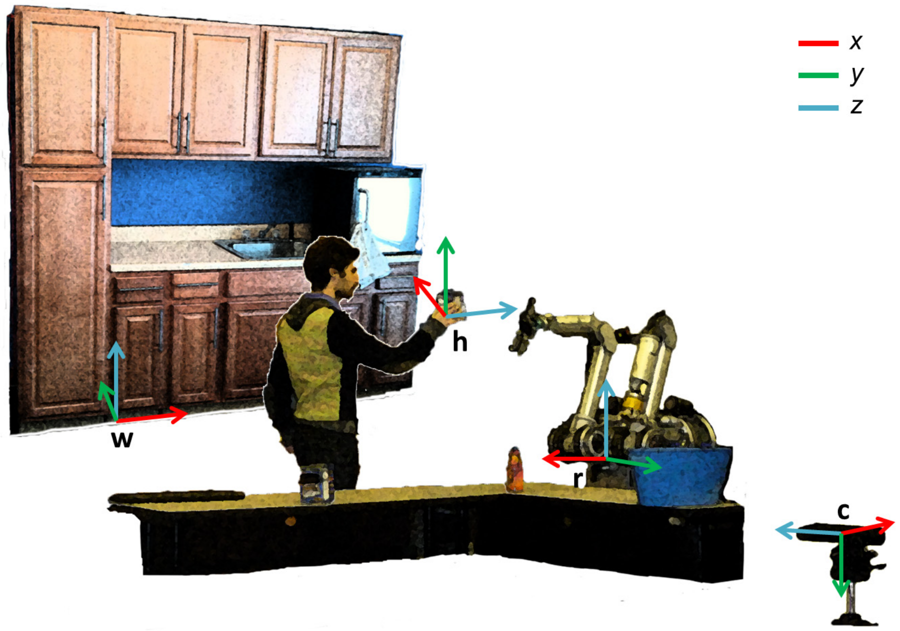


Figure 2.4: Scenario: The human stands in front of the robot and hands off objects to it. The robot processes data from 4 different frames; world frame $w$, hand frame $h$, robot frame $r$, camera frame $s$. Thanks to a localization system on the robot and the calibration system of the camera, the transformations between the frames are known.

Information from four different coordinate frames are combined (Fig. 2.4): the fixed world frame $w$ located at the bottom of the kitchen cabinets, the fixed camera frame $c$ located on the Kinect camera, the robot frame $r$ located on HERB's base, and the moving human hand frame $h$ located on the hand.

HERB's localization system is used to obtain its transform $r$. The extrinsics of the camera $c$ are calibrated with a calibration procedure where salient point correspondences on the kitchen cabinets and on HERB are matched. When a human is detected, a hand frame $h$ is computed online (Fig. 2.5). The hand frame is centered at the hand point detected by the human tracking system with axes defined as follows. Let $X_h$

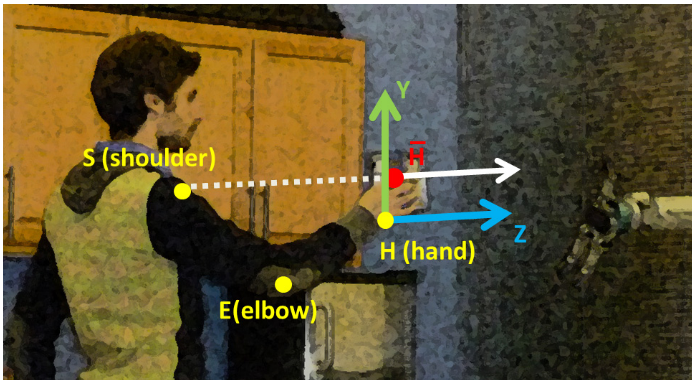


Figure 2.5: Hand frame.

and $X_{sh}$ be the 3D positions of the human skeleton hand and shoulder, respectively, as detected by the human tracking software and with respect to the world frame. Let $\overline{X}_h$ be the modified position of the hand in the world frame where the z-component has been matched to the height of the shoulder.

$$\overline{X}_h = \langle X_{h,x}, X_{h,y}, X_{sh,z} \rangle$$

Then the axes of the hand frame $(\vec{x}_h, \vec{y}_h, \vec{z}_h)$ with respect to the world frame are given by

$$\vec{z}_h = \frac{\overline{X}_h - X_{sh}}{|\overline{X}_h - X_{sh}|} \quad \vec{y}_h = \langle 0, 0, 1 \rangle \quad \vec{x}_h = \vec{y}_h \times \vec{z}_h$$

This frame is used as a target for the robot's planning and control. The Z-axis of the hand frame lies on the world X-Y plane and points along the shoulder-hand direction, and the Y-axis points upwards opposing gravity.

### System Description

The system developed is mainly divided into two subsystems: a *Perception system* and a *Robot Control system* ( Fig 2.6 ). The *Perception system* consists of two modules: a *data acquisition* module reads raw data from the Kinect and sends them to the *handoff detector* that processes the data in order to infer handoffs. When the

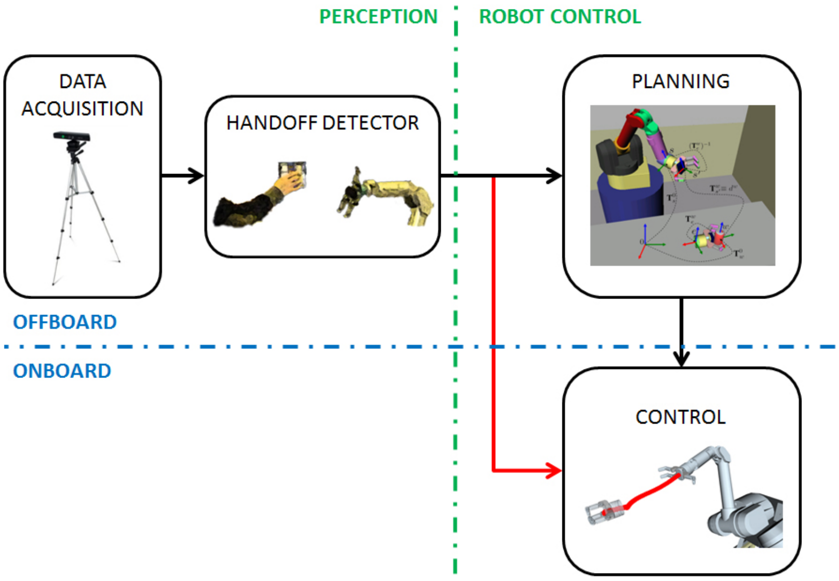
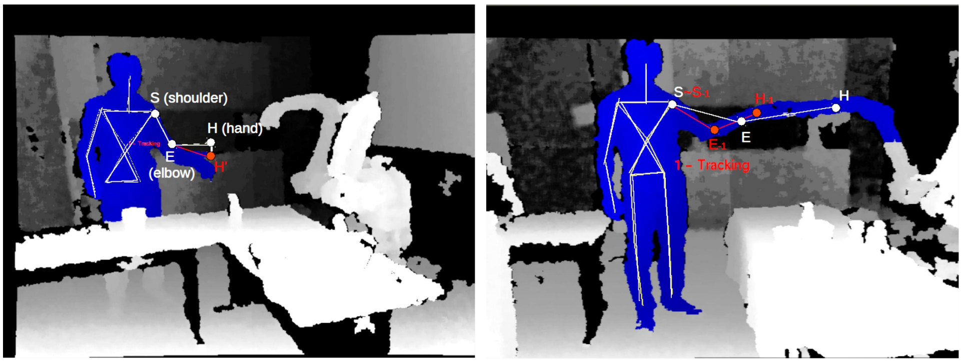


Figure 2.6: Architecture of the system.

*Perception system* detects a handoff the *Robot Control system* moves the robot arm to perform the handoff. Two different control methods are available. A first control method consists of two modules: a *planning module* that computes a trajectory and a *control module* that executes the planned trajectory. A second method bypasses the planning phase and uses only the *control module* to move the robot arm towards the object. These components are described in the next paragraphs.

### 2.2.2 Perception

As already mentioned, the sensor used for the perception system is the Microsoft Kinect. This sensor provides raw depth and color images. The OpenNI Framework is used to process the raw data and compute a 14-point skeleton for each human in the camera view. Although the OpenNI software provides useful human tracking data, I had to address various perception issues to make it suitable for handoffs. The post-processing included refining the skeleton tracking for better wrist positions, par-



(a) Skeleton computed when the human is moving fast his hand (b) Skeleton computed when the robot and human arm point clouds have merged

Figure 2.7: Skeleton tracking. The skeleton tracker compute a 14-joints skeleton. White lines: OpenNI skeleton tracker output. Red lines: corrected positions

ticularly when the human is close to or partially occluded by HERB, detecting the intention of a human to handoff an object, learning to identify if an object is held and recognizing the object type, and calculating where HERB should put his hand for the handoff.

### Correcting Hand Pose

Although the Kinect human tracking system is reliable, the tracked position of the hand has been observed to have errors. These errors occur when the human is moving fast and when the human is near or touching other objects.

In the former case, the human tracker often lags the true human motion. This issue is solved with a correction that moves the hand to the closest 3D point in a small box around the estimate (Fig 2.7a) . Let $P_s$ be the 3D points in the scene and $X_h$ be the original hand position as reported by the tracking system and $P_b$ be the points within a box of size $k$ around $X_h$. The new hand position $X_{h'}$ is the point in $P_b$ closest to $X_h$ as given by

$$X_{h'} = \{X \in P_b : |X - X_h| = \min_i |X_i - X_h|,\ i = 0, \ldots, n\}$$

where

$$P_b = \{X \in P_s : |X - Xh| < [k, k, k]\}$$

This correction is fast, and is naturally robust: if the hand tracking is originally good, the correction does not change the estimate.

In the latter case, when HERB is close to the human, the two are often partially merged into the human point cloud. This is of particular concern to us since the human and the robot need to be close to each other in the final part of the handoff (Fig 2.7b). We take advantage of the fact that we can query the joint configuration of the robot. When the robot hand and the human hand are closer than $15cm$ and the robot hand position is embedded within the human point cloud, the new data is filtered out and the last reliable data is used. Since the human does not move much when very close to the robot, this approximation is far better than incorrect tracking.

### Handoff Detection

After correcting the hand pose, we infer impending handoff based on two features: 1) the human is in a handoff pose and 2) the human is holding an object. The former is detected by analyzing the human skeleton, and the latter using a support vector machine (SVM) to classify an image patch around the detected hand.

### Handoff Pose Detection

Humans use several cues to signal a handoff including speech, gaze, motion, and posture. In this work, I focus on postural cues inferred from human tracking. The following cues (Fig. 2.8), motivated by studying human-human handoffs, were used to infer the handoff signal:

1. The human is near HERB (the distance between them is less then a threshold).

2. The vector from the shoulder to the hand points towards HERB's hand or upper body, as shown in Fig. 2.8.

3. The human elbow is bent at an angle $\alpha \leq \alpha_{\max} \equiv 150^o$.

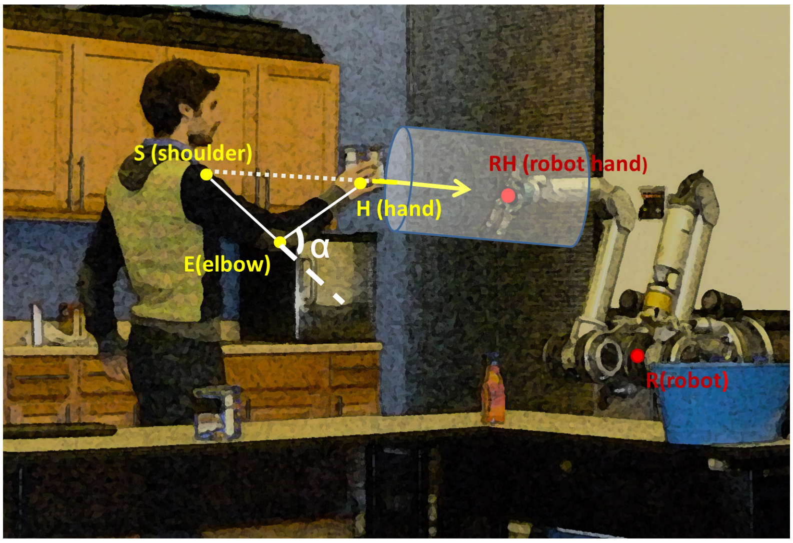


Figure 2.8: Cues used to infer a Handoff pose.

To avoid false positives when the human is moving and momentarily assumes a handoff pose, a detection is triggered only if the human is in the handoff pose for 5 consecutive frames (which at 5fps is $1sec$). The handoff pose detector is implemented by a Moore FSM. The FSM is described by:

- The input alphabet $I = \{a,b\}$, where

$$a = \begin{cases} 1 & \text{if all the cues described above are observed} \\ 0 & \text{otherwise} \end{cases}$$

$$b = \begin{cases} 1 & \text{if the cues have been observed for at least 5 following frames} \\ 0 & \text{otherwise} \end{cases}$$

- The finite set of states $S = \{S_0, S_1, S_2\}$;

- The initial state $S_0$;

- The state-transition function $f : S \times I \to S$ represented by the state diagram in figure 2.9.

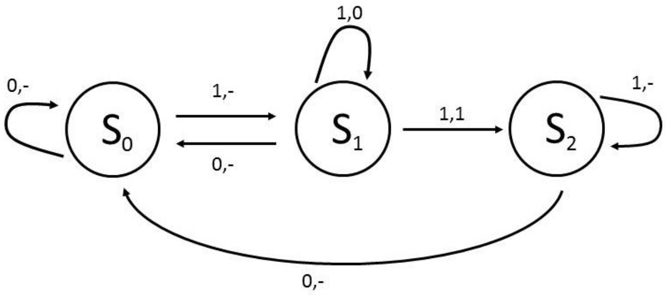


Figure 2.9: FSM for Handoff pose detection.

The system is in the state $S_0$ when a human is not in a hand off pose; in $S_1$ when a human has been observed to be in a handoff pose for less then 5 following frames; in $S_2$ when a handoff gesture is inferred.

### Object detection

People often wave their arms and assume postures that are akin to handoffs. To reliably detect a true handoff, I found it critical to detect if the human was actually holding an object before starting a handoff response in HERB. To enable this, I developed an efficient and reliable object detection system that detects if the human is holding an object, and identifies the held object.

The system is composed of two main components: an algorithm for extracting the bounding box around the human hand, and a support vector machine (SVM) [69] that classifies the bounding box. The algorithm for extracting the bounding box consists in the following steps (Fig. 2.10):

1. Obtain a depth image in the camera frame $c$ (Fig. 2.10a).
2. Transform the depth image into the hand frame $h$. Crop to an axis-aligned bounding box (Fig. 2.10b).
3. Decimate the depth into clusters of contiguous points, removing stray outliers. The hand is often detected as a single cluster (Fig 2.10c). However, sometimes

the hand and the robot are detected together as one cluster (Fig 2.10d). This happen in the final part of the handoff when they are close to each other.

4. Since the transformed depth image is centered at the hand, the cluster closest to the origin is labeled as the hand cluster.

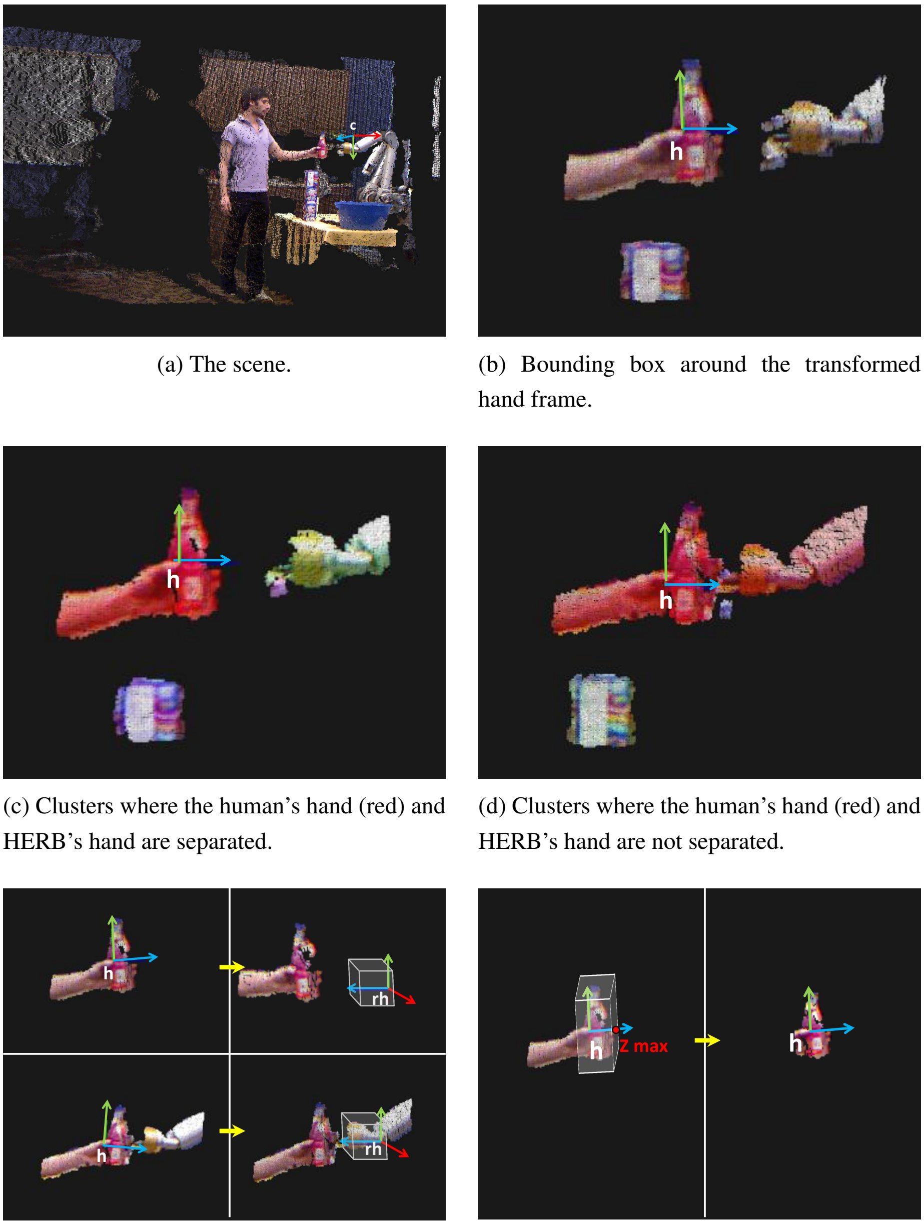
5. To detect if HERB's hand is clustered with the human hand, we use forward kinematics to determine the pose of HERB's hand and check if it lies within the cluster (Fig 2.10e).

    (a) If HERB's hand is within the hand cluster, bypass object detection and trigger the final phase of handoff.

    (b) If not, we crop the human hand. We assume that the object is held farthest from the hand and fit a smaller box tightly around the points. (Fig. 2.10f). If the human is not holding anything, the smaller box contains just the hand.

Once the final box is available, color and geometry features are computed from the points in the box, and a SVM model [69] predicts the class of the object. I created several SVM models that were able to predict up to six classes: Pop-Tart boxes, Fuze bottles, green jars, juice cartons, tea carton and empty hands (Fig 2.11). SVMs have proven to be quite useful for data classification. We use color histograms (16 bins for RGB, with a total of 48 features) and the height of the bounding box (1 feature) and an RBF kernel.

Despite their simplicity, color histograms have demonstrated good results in practice [70, 71]. We found that although the color features are often able to predict the object correctly, the height feature allows us to distinguish better between objects with similar color but different height, like for instance a Fuze bottle and a pink empty hand. Table 2.1 reports the results achieved with the object recognition algorithm. In the table five SVM models are reported. Each model is different in terms of number of features and number of classes. The two models with 3 classes ( Pop-Tart boxes, Fuze bottles, and empty hands ) consider the same testing data with and without the height feature.



(a) The scene.

(b) Bounding box around the transformed hand frame.

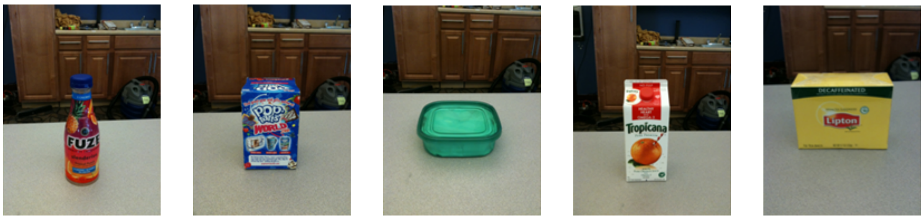
(c) Clusters where the human's hand (red) and HERB's hand are separated.

(d) Clusters where the human's hand (red) and HERB's hand are not separated.

(e) Extracting the hand cluster.

(f) Refining the bounding box.

Figure 2.10: Stages of the bounding box extraction algorithm.



(a) Fuze bottle. (b) Pop-Tart box. (c) Green jar. (d) Juice carton. (e) Tea carton.

Figure 2.11: Objects detected using SVM.

Because the object is small and far from the camera, we found local descriptors like NARF [72] to be far less useful when compared with global descriptors like color histograms. For the same reason, we found the depth information in the bounding box to be far too corrupted by noise and quantization to be useful beyond a global descriptor like the height of the object.

The SVM model was trained with indoor lighting. Most of the errors occur when the Fuze bottle is misclassified as a hand. This happens because, as shown in fig. 2.12, the histograms of the hand and the Fuze bottle are very similar. The model is not robust to strong light variations. Fig. 2.12 shows that the R,G,B histograms change significantly when the room is illuminated with daylight coming from a big window. In that case often the Fuze bottle and the hand are misclassified as a PopTarts box. Neither the Hue histogram is able to give the same hue information of an object when the light conditions change. Other methods for object recognition like ferns [73] (for images) or NARF (for point clouds) [72] are robust to light variations. These methods rely on feature descriptors for extracted keypoints, but since the object that we are trying to detect is small and far from the camera neither the shape nor the texture of the object are accurate enough to extract reliable keypoints, while is possible to use global features like RGB histograms or the height of the object to train an SVM model.

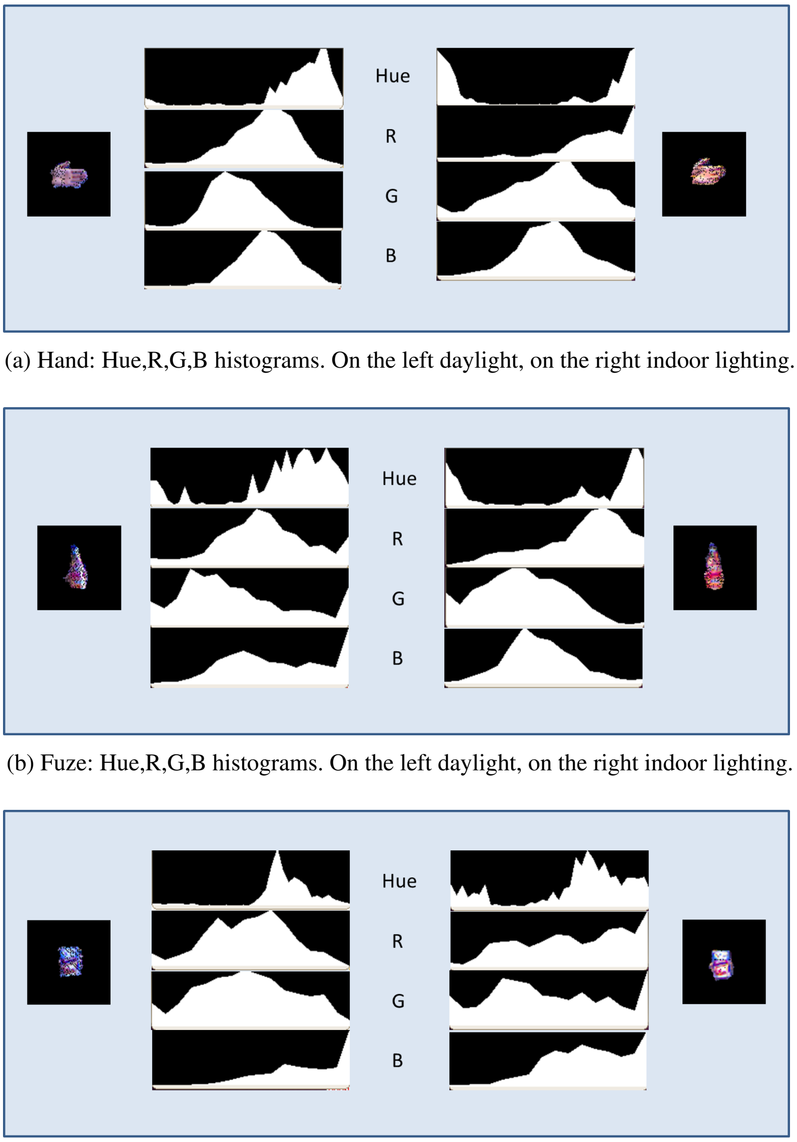
| Features | Classes | Training Set | Test Set | Cross Validation | Prediction Accuracy |
|---|---|---|---|---|---|
| **RGB (48)** | 3 | 4766 | 1623 | 99.96% | 93.16% |
| **RGB (48) + Height (1)** | 3 | 4827 | 1623 | 99.96% | **96.61%** |
| **RGB (48) + Height (1)** | 4 | 6048 | 3417 | 100% | 99.53% |
| **RGB (48) + Height (1)** | 5 | 7197 | 4298 | 100% | 99.28% |
| **RGB (48) + Height (1)** | 6 | 8392 | 5149 | 99.96% | 98.78% |

Table 2.1: SVM performance.

### 2.2.3 Robot Control

As soon as the perception system triggers an impending handoff, HERB moves to take the object. We compared two autonomous motion strategies, a planner and a controller. Both strategies receive the same input: a target object pose from the perception system at 5fps. Both strategies control the arm via joint velocities (converted internally by the arm driver into joint torques) and have access to the 6 axis force-torque sensor on the wrist for detecting forces.

The execution of these two strategies differs in the temporal position of the planning phase with respect to the execution loop as shown in Figure 2.13. The controller plans in the near future during each iteration of the execution loop, while the planner plans for the entire execution before the execution loop begins. The effects of this are that the controller begins execution immediately but cannot predict problems in the future like joint limits and collisions, while the planner takes several ($\approx 5$) seconds to plan the entire trajectory that avoids joint limits and collisions. Nevertheless the controller updates the goal position every loop cycle and is able to recover from changes in the object position adapting the trajectory of the robot arm to the movements of the human, while the planner fails if the position of the object changes significantly after the handoff is triggered. The details of each control method are presented in the following two sections.



(a) Hand: Hue,R,G,B histograms. On the left daylight, on the right indoor lighting.

(b) Fuze: Hue,R,G,B histograms. On the left daylight, on the right indoor lighting.

(c) PopTarts: Hue,R,G,B histograms. On the left daylight, on the right indoor lighting.

Figure 2.12: Hue, R, G, B Histograms of a hand, a Fuze and a PopTarts Box

```
procedure CONTROLLER( )
    while CLOSEHAND( ) == False do
        $\theta \leftarrow$ GETCURRENTJOINTANGLES()
        $goal \leftarrow$ GETCURRENTGOAL()
        $d\theta \leftarrow$ CALCULATEJOINTINCREMENT($\theta, goal$)
        GOTOCONFIG($\theta + d\theta$)
    end while
end procedure
```

```
procedure PLANNER( )
    $\theta \leftarrow$ GETCURRENTJOINTANGLES()
    $goal \leftarrow$ GETCURRENTGOAL()
    $path \leftarrow$ CALCULATEPATH($\theta, goal$)
    while CLOSEHAND( ) == False do
        $t =$ GETCURRENTTIME()
        GOTOCONFIG($path(t)$)
    end while
end procedure
```

Figure 2.13: Controller vs Planner Algorithms.

### Planner

The planner takes the very first reported human hand pose and plans to get to it. For this work I used a randomized planner (developed by the Personal Robotics Lab at Carnegie Mellon University) that efficiently explores high-dimensional constraint manifolds [74]. The planner is tasked with producing a feasible, collision-free, reasonably smooth path as quickly as possible. Any updates to the human hand pose are ignored during planning. As soon as a path is returned, the robot executes it.

In our case, the planner takes as input a starting arm configuration and a goal end-effector pose, consisting of the end-effector position and orientation. With the goal pose, the planner finds several candidate goal Inverse Kinematics (IK) solutions. Then, the planner expands random trees from each of the start and candidate goal configurations until a path is found from start to goal that is collision-free and within the joint limits. This path is often very jagged, so the final step is to run the path through a trajectory smoother to remove backtracking and corners.

Randomized planners are probabilistically complete: guarantee to find a feasible path if one exists. However, more search time is required in situations with more constrained configuration spaces. Also, while the output of the planner is smoothed, there is no guarantee of global or local optimality.

### Controller

The controller is an implementation of inverse Jacobian control with constraints [75]. The Jacobian relates angular velocities of the arm joints to hand velocities. For the controller, we use two Jacobian operators, one relating joint angular velocities to end-effector velocity, $J_X$, and the second relating joint angular velocities to end-effector twist via the quaternion velocities, $J_q$. Combining these two Jacobians into one gives $J$ as

$$J = \begin{bmatrix} J_X \\ J_q \end{bmatrix}$$

Let the Moore-Penrose pseudo-inverse of $J$ be $J^+$. Then, for a given change in the end-effector position $\delta X$ and orientation $\delta q$, the approximate change in joint angles required to accomplish this change, $\delta\theta_{pose}$, can be calculated as

$$\delta\theta_{pose} = J^+ \begin{bmatrix} \delta X \\ \delta q \end{bmatrix}$$

HERB has a redundant DOF meaning that he has more arm freedom than is required to achieve a 6 DOF pose of its end-effector. HERB has 7 joints to position the end-effector in 6 DOF, so in the case where its arm is not in a singularity, HERB can move about in the null-space of his Jacobian without changing its end-effector pose. We can move around in the null space to accomplish several different goals. For instance, avoiding joint limits, avoiding singularities, minimizing joint velocity, and minimizing joint torques. For this controller, the null space has 1 DOF and is used to avoid joint limits. The change in joint angles to minimize the difference between the current joint angles $\theta$ and desired joint angles $\theta_{des}$ is

$$\delta\theta_{limits} = N_J N_J^T \cdot (\theta - \theta_{des})$$

where $N_J$ is the null space of the Jacobian.

Finally, the output joint angular velocity $\dot{\theta}$ is a combination of the two joint differentials with some scaling values $\alpha$ and $\beta$.

$$\dot{\theta} = \alpha \cdot \delta\theta_{pose} + \beta \cdot \delta\theta_{limits}$$

Up to this point the controller has no concept of collisions or exceeding joint limits. Therefore, before commanding the desired joint velocities, collision checks and joint limit checks must be performed. If there is a collision or joint limit problem, then the joint velocities are commanded to zero and the controller is stuck until the desired joint velocities produce a path that is collision-free and within the joint limits. To force movement in these situations, a planner like the one described in the last section can be used to move away from the problem.

### Completing the handoff

Both strategies use the same sensors for detecting when to close the hand to complete the handoff: a force sensor and an RGB-D camera. The force is calculated using a 6 axis force-torque sensor located at the wrist of the arm. If the force into the palm is greater than $5N$, then the arm stops moving and a close hand command is initiated. The force-torque sensor can sense forces smaller than $5N$, but the mass of the hand creates a non-zero measurement when the arm moves. This behavior is valid for both control methods. The way the camera is used depends on the control method. With the reactive controller the close hand command is triggered when the robot hand is stationary at the given object location for at least 0.5 second. With the planner, since the position of the object is not updated during the execution of the trajectory, the robot reaches the supposed object point and close its hand. The combination of the two sensors results in the following behaviors:

- $Reactive\ Controller$ : if the robot senses the contact with the object it starts the transfer immediately, otherwise it relies on its perception system and if the object stays still near the robot hand the transfer of the object is started. We found that both triggers are required for a reliable handoff. If only the force trigger is used, then the robot sometimes hovers around the object and the users do not intuitively know push the object into the robot's hand. On the other hand, if only the timeout is used, then the robot will not respond to contact with the human which makes the robot seem very aggressive to the user.
- $Planner$ : if the contact with the object is sensed, the robot starts the transfer

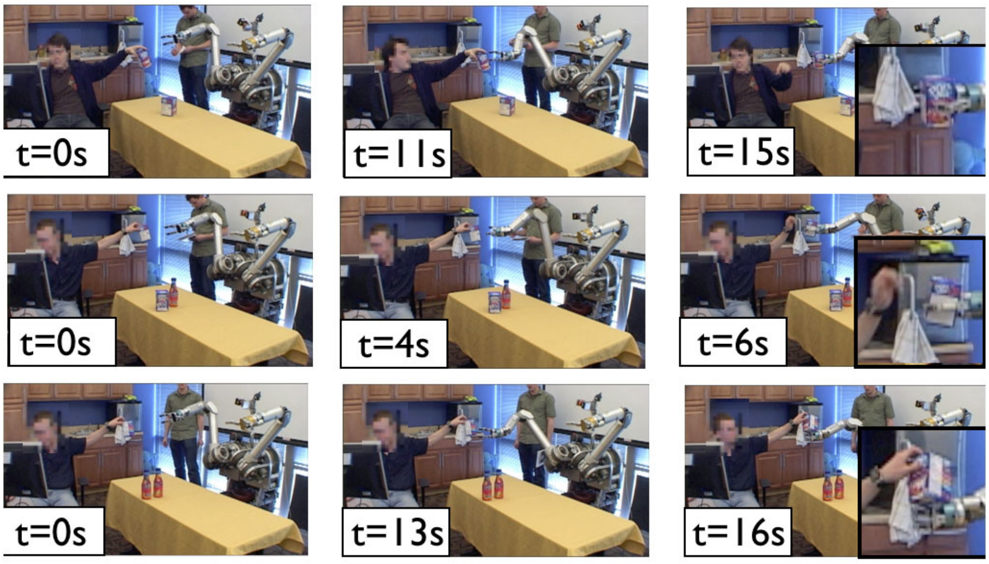


Figure 2.14: Results: Top: Planner success, Middle: Controller success, Bottom: Planner failure

otherwise the robot reaches the supposed position of the object and tries to perform the transfer of the object.

If the hand is closed due to force or the vision system, then success is returned based on whether or not the fingers close all the way into a fist. If the hand closes into a fist, then the handoff is reported as failed, otherwise, the handoff is reported as successful.

### 2.2.4 User Study

We wanted to test HERB's ability to take objects from a human during a handoff while minimizing the effort contributed by the human. To this end, we created a user study where the subjects were told to perform two tasks at once, a computer task and a handoff. The subjects were told to focus their attention on the computer task, thereby letting HERB take the object from the human while getting little or no help from the subject.

The user study investigated 5 subjects, tested individually. The subjects were seated at the table with a monitor, mouse, and 7 objects in front of them. HERB was positioned to the subject's left where it could be seen in the subject's peripheral vision and visible if the subject looked away from the monitor. The subjects were instructed to play a computer task and, when prompted by HERB, to handoff one of the objects to HERB. The subjects used their right hand to move the mouse and used their left hand to perform handoffs to HERB (Fig. 2.14).

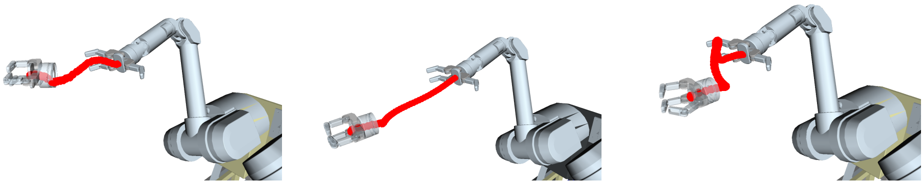
Two control algorithms were tested in sequence. The subjects performed 7 handoffs with the controller, had a short break of 30 seconds, then performed 7 handoffs with the planner. Failed handoffs were recorded and if the subject still had the object, the object was taken away by an investigator.

The computer task was a slightly modified version of the PEBL Continuous Performance Task. The computer task presented the subject with a small target on the monitor where the subject was to move the mouse. Once the target was reached, the subject was presented with another target in another location. In the original computer task, the subject clicked in between targets to signal his readiness. In our version there is no delay between targets. To make the computer task more challenging, the mouse input was summed with a random error which made the mouse quiver by a small amount. This addition of noise required more attention from the subject, leaving less attention for the handoff.

After completing the handoffs with both control algorithms, each subject was asked to compare the two controllers as well as state any comments they had. Subjects were asked to choose which controller they preferred in 5 areas and state why. The 5 areas were preference, natural-looking, easier, safer, and human-like.

### 2.2.5 Results from the user study

We had 5 subjects participate in our informal study. The results are summarized in table 2.2 and figure 2.16. Overall, the combination of perception with the Kinect data and a controller with a take attribute resulted in a handoff success rate of 83% for the 70 handoff attempts. The majority of these attempts were with the users completely distracted, just holding the object up and waiting for the robot to take it while they



(a) Typical controller trajectory (b) Good planner trajectory (c) Bad planner trajectory

Figure 2.15: Hand trajectories for three handoffs, one with the controller and two with the planner. The bad planner trajectory is chosen to illustrate the random nature of the planned paths.

continued with the computer task.

The two control algorithms were compared by number of successes and total handoff time. The controller was faster than the planner by a factor of nearly 2, taking an average time of 8.46 seconds from detection to the grasp finishing while the planner took an average time of 14.23 seconds. These handoff times include the time it takes to close the hand, usually around 2 seconds. The two control methods are similar in timing, except the planner has an additional $5sec$ planning phase tacked on at the beginning.The paths that the two controllers executed were sometimes quite different. The controller always takes a predictable curved path to the goal, whereas the planner can have different unpredictable trajectories even with the same input, as shown in Figure 2.15. For the human, working with an unpredictable robot can be disconcerting.

The planner had more successes (91.43%) than the controller (74.29%) as shown in table 2.2. Failures with the controller were caused by three problems: the controller hit the arm joint limits and became stuck, the object detector failed to detect the object for the duration of the handoff attempt which caused the robot hand to retreat unexpectedly, and the object fell during the transfer. Failures with the planner were only caused by the object falling during the transfer since the planner avoids joint limits during its planning phase and only the first handoff detection is used to trigger the handoff attempt. In principle, the reactive controller can be made more reliable using

| | planner | controller |
|---|---|---|
| **average time (s)** | 14.23 | 8,46 |
| **standard deviation (s)** | 2.08 | 2.32 |
| **% success rate (handoff completed)** | 91.43 | 74.29 |
| **% success rate (svm- object detected correctly)** | 90.63 | 96.15 |

Table 2.2: User study results.

planned trajectories to get unstuck from joint limits and some filtering of the SVM output. If we exclude the handoffs with the controller that failed due to erroneous SVM object detection or joint limits, then the handoff success rate for the controller jumps to 90%, similar to the planner. The object detection algorithm recognized correctly 94% of the grasped objects.

After the test, subjects were asked about what they liked or disliked in the two control algorithms. Subjects found the planner aggressive while they liked that the controller was more "gentle". This happened because the velocity of the controller is directly proportional to the distance from the object. Moreover, sometimes the planner executed strange trajectories that made users feel less safe since they didn't understand what the robot was doing. Although the users found the interaction with the controller more natural compared to the planner, they found the interaction to be not human-like with both control systems since it was slow compared to human-human handoff. However, they found the handoffs performed by HERB appropriate for a robot. Both control methods were found easy to interact with.

The interesting insights that came out of the subject's questionnaire and comments can be summarized in four features that the subjects found crucial during the interaction:

- $Forcefulness$ : The force applied by robot to the object was very useful in the final part of the reaching and during the transfer of the object. Indeed pre-grasp touching signals to the human the robot *readiness* to the handover. Some of the subjects liked pre-grasp touching since it provided feedback, others did not

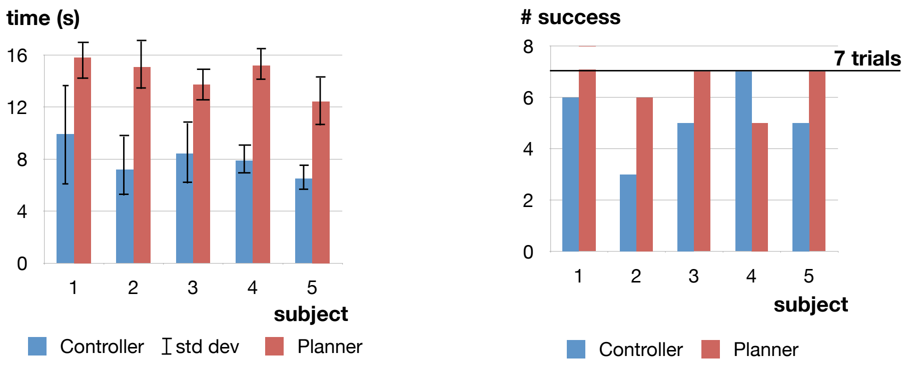


Figure 2.16: User study results: (Left) Average time to handoff, (Right) Number of successful handoffs

like that the robot pushed against their hand. The force applied by robot to the object is useful for both the robot and the human to get feedback during the exchange of the object, but if the force is too strong the human partner feels uncomfortable and unsafe. In order to better tune the forcefulness we found sensor fusion very useful. The robot used both touch and vision to sense the proximity to the object. With the reactive controller, the robot touched the object just a bit without being too intrusive. If the robot did not sense the touch, instead of pushing against the object, it relied on the vision system. If the observed position of the object was stationary and close to its hand, the robot tried to perform the grasp. The planner was generally more intrusive because it did not update the position of the object during the execution of the trajectory and sometimes reached out a bit too far. This happened because sometimes the subjects slightly changed the position of the object during the reaching of the robot. The change in position was not enough to make the planner fail but, when the object was moved toward the robot, the planner pushed it while the reactive controller was able to apply the appropriate force to the object since it monitored the position of the object during the reaching phase.

- *Aggressiveness* : subjects felt that the robot was aggressive when it approached the object quickly. While the velocity of the planner was constant during the execution of the trajectory, the velocity of the reactive controller was directly proportional to the distance from the object. Subjects felt safer and more comfortable when the velocity of the robot hand decreased in the proximity of the human hand. In fact, also in human-human handover, the receiver usually quickly gets close to the hand of the giver and then slow down to accurately grasp the object [46].

- *Predictability* : The paths that the two behaviors executed when reaching out were sometimes quite different. The reactive controller always takes a predictable path that heads straight to the goal, whereas the planner could have different unpredictable trajectories. Predictability makes the humans more comfortable around HERB because they can plan into the future and know that HERB won't do anything strange.

- *Timing* : subjects pointed out that human-human handoffs are faster. The reactive controller was faster than the planner by a factor of nearly 2, taking an average time of 8.46 seconds from detection to grasp finishing while, the planner took an average time of 14.23 seconds. The difference is mainly due to the initial planning phase of the planner. Even if also the reactive controller was slow compared with human human handoffs, it can be made faster with more aggressive gains. Tuning the gain of the controller without making the robot too aggressive is challenging but could lead to effective human-robot handoffs. Another way to make the handoff faster could be detecting the intent to handoff an object before that the human arm motion begins, as shown in [76].

Although these features are always relevant in human-robot handovers, they are even more prominent when the robot behaves actively until the exchange of the object.
The posture of the subjects during the handoffs varied from relaxed to fully extended arms. The relaxed posture forced HERB to go and take the object, whereas the fully extended arms positioned the handoff object as close to the robot's hand as possible. The subjects rarely moved the object once stopped, meaning that the controller and

planner would go to the same place. If the subjects had moved after a handoff was detected, only the controller would have successfully followed the motion.

## 2.3 Robots handing objects to humans

After investigating how a robot should receive an object from a human, I focused my research activity on the issue of robot to human object handover. Direct delivery of an object from a service robot to a person is a valuable behavior that enables co-operation through physical interaction [77]. In addition to sensory-motor skills, the task of handing over an object involves social skills such as the ability to deliver the object at the convenience of the receiver. Indeed, when offering an object to someone we, as humans, usually orient the object taking into account etiquette factors, object usage or habits so that the receiving person can easily grasp the object. For example, sharp objects or tools are usually delivered so that their handle is oriented towards the receiver.

User comfort in object delivery tasks has been rarely investigated in previous works. Some approaches compute the best location where the the object should be delivered taking into account visibility, safety and comfort parameters [78, 79, 55]. Others pay particular attention to object shape and appropriate grasping regions [58]. However there are no approaches that take into account the choice of comfortable object orientation in order to make the grasp as easy as possible to the human. The novelty of the approach I used during my work is the proposal of a human-aware robot system for object handover that considers user comfort. Here comfort is taken into account by delivering the object so that one of its constituent parts (e.g. a handle) is oriented towards the user. Indeed, it is assumed that objects can be segmented into meaningful parts and that there is a most appropriate part, known a priori to the robot system, to be served to the user. This idea is supported by the observation that most ordinary objects (such as tools) have grasp affordances, i.e. preferential and comfortable ways of grasping an object to achieve a particular function. Moreover, by the fact that grasp affordances can be discovered by segmenting an object into its constituent parts.

This section presents a complete robot system for object handover that not only supports social awareness, as stated above, but it also features sensory-motor skills such as object perception and recognition, people detection, robot grasping and motion planning. I first present an initial version of the system that enables the handover of a single object and a related user study. Then I present how the initial system has been expanded by generalizing the handover task to environments with multiple objects. Moreover, the system lets the user choose which object has to be delivered by natural interaction through voice recognition.

The approach proposed in this work for comfortable and natural handover consists of three phases [80], which are described in the following sections. Section 2.3.1 describes the robot setup. Section 2.3.2 describes the initial phase of the task where the robot, after scanning the environment using an eye-in-hand laser scanner, detects the object and performs 3D reconstruction and part segmentation. Section 2.3.3 illustrates the second phase of the task, where the human approaches the robot and the robot motion for the handover task is planned. Section 2.3.4 describes the final phase of the task, where the system recognizes the intention of the user to grasp the object being offered and then the gripper is opened to release the object. In section 2.3.5 experimental results from a user study are reported. Section 2.3.6 presents an extended approach for the handover task that handles the case of multiple objects in the environment. The extended approach supports object recognition and allows the user to select the object to be delivered by voice interaction.

### 2.3.1 The Framework

#### Scenario

Figure 2.17 shows the experimental setup for the evaluation of the proposed method. The system includes a six degree of freedom robot manipulator (Comau SMART SiX) that is equipped with a two-finger parallel gripper (Schunk PG-70). The robot is located behind a table. The object to be served to the user lays on a support plate. In this work it is assumed that there is only one object in the environment. The table also serves as a barrier that keeps the user at a safe distance from the robot. A laser scanner (SICK LMS400), mounted on the robot end effector (eye-in-hand configuration), is

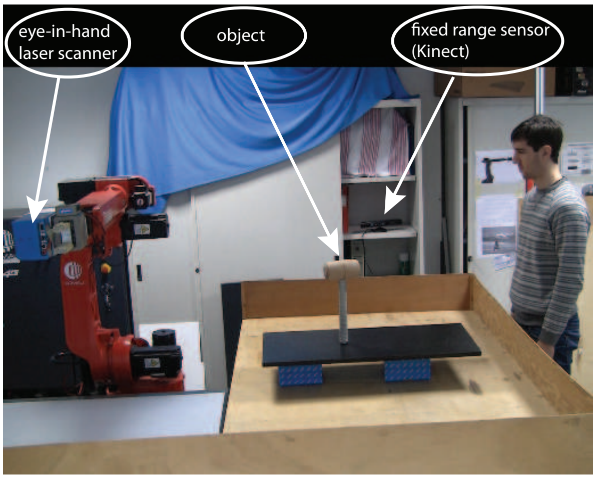


Figure 2.17: Experimental setup.

used to acquire range data of the environment for detecting the object. Range data from the LMS400 eye-in-hand sensor are collected by moving the robot arm along a pre-computed path that allows the sampling of most of the surface of the object. The laser scanner is calibrated with respect to the robot base reference frame and is used with a field of view of 70 degrees (140 beams) and a scanning frequency of $190Hz$. The laser sensor has a minimum measurement range of $0.7m$. The statistical error is about $1.5cm$ depending on the remission of the object material and on the angle of incidence. The system also includes a Microsoft Kinect depth sensor that is located in a fixed configuration as shown in figure 2.17. The Kinect sensor, calibrated with respect to the robot base reference frame, is used for human detection and body tracking.

### System Description

Figure 2.18 shows the architecture of the system. The system is divided in four modules:

- $Object\ modeling$: this module is responsible for generating a model of the object to hand to the human. The robot uses a laser scanner to acquire range

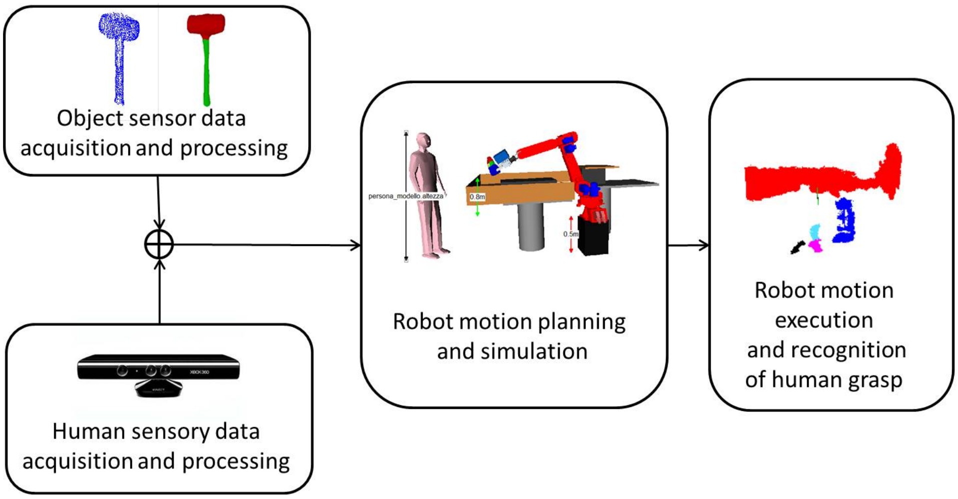

Object sensor data acquisition and processing
Human sensory data acquisition and processing
persona_modello altezza
0.8m
0.5m
Robot motion planning and simulation
Robot motion execution and recognition of human grasp

Figure 2.18: Architecture of the system.

data of the object, and after some processing steps a segmented mesh of the object is obtained.

- *Human detection*: this module performs human detection and computes the position and the orientation of the person detected. This information is used to create a 3D model of the human that represents the real pose of the person.

- *Motion planning and simulation*: This module performs the robot motion planning. After the 3D models of the object and the person are available, they are inserted in a 3D environment that includes also the robot model. The robot to human handover is then planned in the simulation environment.

- *Human grasp detection*: After the planned robot motion has been executed, the system aims to recognize the intention of the user to grasp the object being offered. When the human grasp is inferred the robot opens the gripper to release the object.

The steps performed by these modules are described in detail in the next sections.

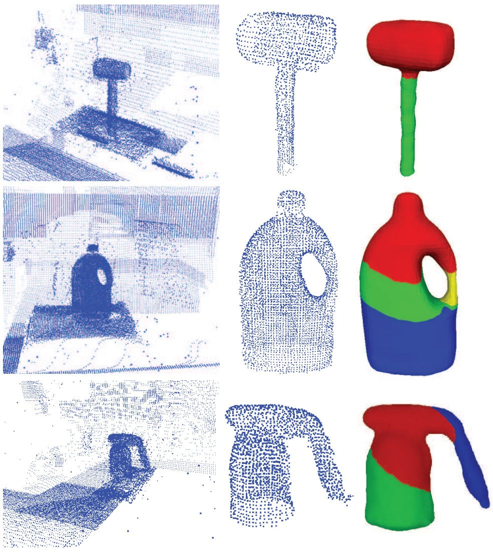


Figure 2.19: Raw point clouds (left) obtained from the eye-in-hand laser scanner and the extracted clusters (middle) of three objects used in the experiments: a hammer ($1.5K$ points), a jug ($4K$ points) and a blow torch ($2K$ points). Reeb graph segmentation (right) of the three objects.

### 2.3.2 Object modeling from range data

This section describes the first phase of the handover task for scenarios with a single object in the scene. The first phase includes range scan of the environment as well as 3D modeling and segmentation of the object. This phase, being time consuming, is run in absence of the user (i.e. when the user has not yet triggered the handover task by standing in the proximity of the robot). Once the robot terminates the scanning process, range data (collected from the eye-in-hand laser) are stored in a point cloud as a set of 3D points in the robot reference frame. Range data are then processed using the approach illustrated below, which has been introduced in previous works [81, 82], in the context of object categorization and manipulation planning.

Initially, the point cloud is filtered by removing sparse noisy data (statistical outliers) as well as points that are outside a fixed box around the support plate. The point cloud is then downsampled. After downsampling, the points belonging to the dominant plane (i.e. the support plate) are detected and removed through the sample consensus algorithm. The remaining points constitute the cluster that corresponds to the target object in the scene. Point cloud pre-processing is based on the *Point Cloud Library* (PCL) [83]. Figure 2.19 displays the raw point clouds obtained from laser scanning and the extracted clusters of three objects used in the experiments: a hammer, a jug, and a blow torch. After the pre-processing phase the point cloud cluster of the object is triangulated to generated a triangle mesh. A two-step approach is followed for surface reconstruction. In the first step the Power Crust algorithm [84] is run. The Power Crust is based on Delaunay interpolation and generates a watertight triangle mesh. In the second step the mesh is smoothed by the Poisson algorithm [85], which is based on implicit methods. After reconstruction, when a complete 3D model of the object is available, the mesh of the object is segmented into connected parts. The approach for shape segmentation is based on the computation of the Reeb graph, which requires a watertight mesh. The Reeb graph (encoded as an undirected graph) represents the topology of a shape [86] as it tracks the connectivity of level sets of a scalar function defined on the mesh. The chosen scalar function is the integral geodesic function, which is invariant under rotations. Formally, given a surface $S$ and a real, continuous function $f : S \to R$ defined on it, the Reeb graph of $S$ with respect to the mapping function $f$ is the quotient space of $f$ in $S \times R$ by the equivalence relation $(X_1, f(X_1)) \sim (X_2, f(X_2))$ which holds if and only if $f(X_1) = f(X_2)$ and if the two points $X_1$ and $X_2$ are in the same connected component of $f^{-1}(f(X_1))$. Reeb graph segmentation is accepted as a method for semantic decomposition of objects made of multiple parts since it is appropriate for identifying object protrusions. Figure 2.19 also shows the segmented meshes of the three example objects, where the object's parts have different colors and are connected to each other.

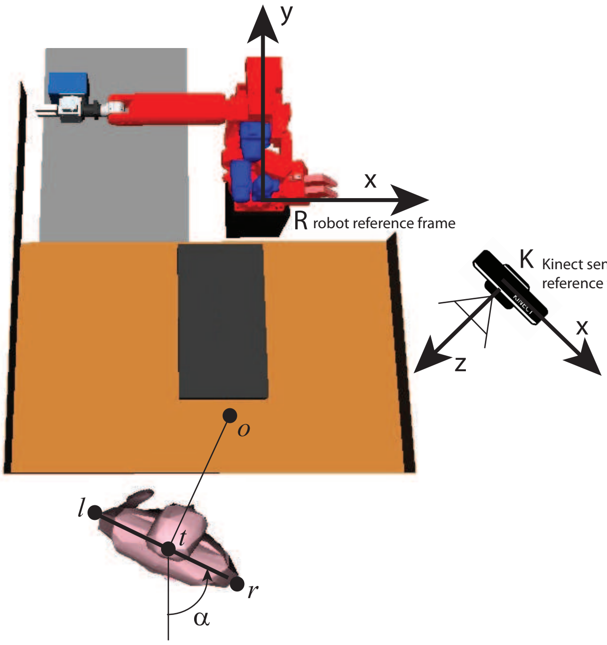


Figure 2.20: Bird's eye view of the simulation environment highlighting the reference frames an the key points.

### 2.3.3 Human detection and robot motion planning

#### Human detection

When the person approaches the robot, i.e. when he/she enters in the field of view of the fixed depth sensor (Kinect), the system performs body tracking. The system exploits the Kinect skeletal-tracking functionality (included in Microsoft Kinect SDK) which provides a 20-joint map of the human body. When the system detects that the person is standing still close to the table (a window of 4 seconds is used to check that the torso doesn't move) the position of the person is computed, as well as his/her orientation with respect to the vertical axis and the height of the body. This information is used, together with the reconstructed 3D model of the object, to build a simulation environment for robot motion planning (an example is shown in Figure 2.20 from

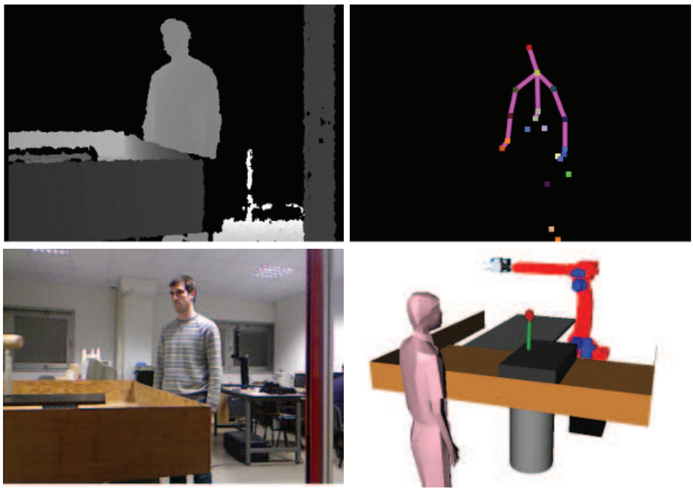


Figure 2.21: Human detection phase. Depth image acquired from the Kinect (top left), human skeleton (top right), color image from the Kinect (bottom left), and generated simulation environment for handover planning (bottom right).

a bird's eye view). Figure 2.20 also displays the main reference frames and the key points used for the handover task. A 3D human model is inserted in the simulation environment in a configuration that resembles the real pose of the person. At this point the handover task is triggered and the robot motion is planned (section 2.3.3). Figure 2.21 shows example images acquired from the Kinect sensor in the human detection phase, including the depth image, the extracted human skeleton and the generated simulation environment. To calibrate the setup, the transformation matrix ${}^{R}_{K}T$ has been determined that expresses the Kinect reference frame ($K$) with respect to the robot frame (R). The position of the human body with respect to the robot frame is computed as ${}^{R}t = {}^{R}_{K}T\,{}^{K}t$, where ${}^{K}t$ is the position of the torso with respect to the Kinect sensor. The orientation $\alpha$ of the human body about the vertical axis passing through the torso is computed from trigonometric considerations:

$$\alpha = \begin{cases} \arcsin\left(\frac{{}^{R}r_x - {}^{R}l_x}{|\overrightarrow{{}^{R}l\,{}^{R}r}|}\right) \text{ if } {}^{R}r_y \leq {}^{R}l_y, \\ \pi - \arcsin\left(\frac{{}^{R}r_x - {}^{R}l_x}{|\overrightarrow{{}^{R}l\,{}^{R}r}|}\right) \text{ if } {}^{R}r_y > {}^{R}l_y. \end{cases} \tag{2.1}$$

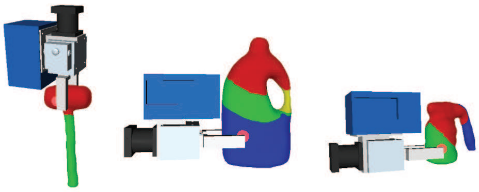


Figure 2.22: Examples of planned robot grasps.

where $({}^{R}r, {}^{R}l)$ are the 3D points located on the right and left shoulders of the human body in the robot reference frame. From ${}^{R}t$ and $\alpha$ a transformation matrix is then computed that defines the configuration of the 3D human model in the simulation environment with respect to the robot reference frame. The height of the person in the robot reference frame (along the $z$ axis) is computed as follows:

$$ {}^{R}height = \left({}^{R}_{K}T\,{}^{K}head\right)_{z} + \delta \tag{2.2}$$

where ${}^{K}head$ is the point of the skeleton located in the middle of the body's head measured in the Kinect reference frame $K$, and $\delta$ is a constant correction factor that estimates half height of a human head.

### Robot motion planning

As explained in the previous section, after the user detection phase a 3D model of the environment is generated including the reconstructed object and the estimated model of the person. The 3D environment also includes the robot model as well as static objects like the table in front of the robot. The robot to human handover task is then planned in the simulated environment. The robot planner has been developed upon the OpenRAVE engine [87]. The task consists of two steps: grasping the object with the robot hand, i.e. finding a force-closure grasp, and planning a robot trajectory so that the goal configuration of the object (when it is handed over) is comfortable for to the user. Robot grasp planning is performed offline by sampling a large number of force-closure grasps on all the segmented parts of the object but the target part that is meant to be grasped by the user. Planning robot grasps on different parts of the object than the one that is offered to the user implicitly leaves room for the user to grasp the object from the target part.

The algorithm for part-based grasp synthesis is described in detail in [81]. Object parts are automatically decomposed into convex sub-parts to refine the granularity of the model, and robot grasps are generated on each sub-part. In particular, the centroid of each sub-part and its principal axis of inertia are computed and a randomized algorithm samples grasp configurations of the robot end-effector around the principal axis of inertia of each sub-part. If the sampled grasp is force-closure it is included in a grasp set. A grasp set $\Omega_o$ of an object $o$ is a matrix data structure that has as many columns as the number of sampled force-closure grasps. Each column vector contains the 6-dof pre-grasp configuration and an offset that specifies the final distance of the palm of the end-effector from the object. Examples of planned grasps are shown in figure 2.22. When planning a collision free robot trajectory towards the goal configuration of the object all the previously sampled robot grasps are tried. If the planner finds a solution to the problem the task is then executed by the real robot as shown in sections 2.3.4 and 2.3.5. In the rest of this section, the proposed algorithm for determining a comfortable goal configuration of the grasped object is presented.

As stated in the introduction of this paragraph the proposed solution is to offer the object so that one of its parts $p$ (known a priori) is oriented towards the user. The goal configuration is expressed by a transformation matrix ${}^R_OT$ between the local reference frame of the object $O$ (centered around its centroid) and the global reference frame of the robot. The translation component of ${}^R_OT$ is defined so that the object is placed in front of the user. In particular, the centroid of the object (goal position, indicated by point ${}^Ro$ in figure 2.20) is placed at the same height of the torso and at a fixed distance from it, lying on the line $\overrightarrow{{}^Rt{}^Ro}$, which is parallel to the ground and orthogonal to line $\overrightarrow{{}^Rl{}^Rr}$ (i.e. the line passing through the shoulders). The rotation component of ${}^R_OT$ is defined so that the appropriate part $p$ of the object is oriented towards the user. Let $v_p$ be the unit vector that points towards the centroid of the part $p$ (i.e. the centroid of all the vertices of the object mesh belonging to part $p$) expressed in the local reference frame of the object and let $v_t = \frac{{}^Rt - {}^Ro}{|\overrightarrow{{}^Rt{}^Ro}|}$ be the unit vector that represents the relative displacement of the human torso with respect to the goal position ${}^Ro$. The

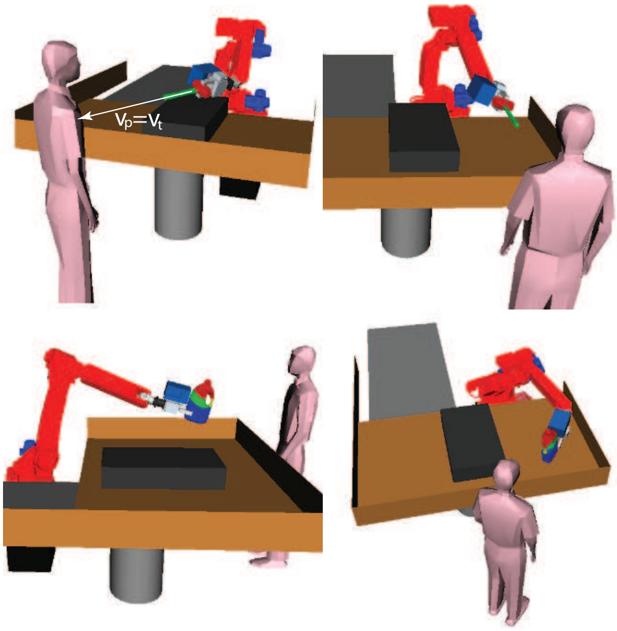


Figure 2.23: Motion planning experiments of comfortable handover tasks. Objects are placed in front of the user. The handle of the hammer and the handle of the jug are oriented towards the human torso.

rotation component is then one that rotates $v_p$ into $v_t$, which can be expressed in the axis-angle form as $(axis = v_p \times v_t, angle = \arccos(v_p \cdot v_t))$. Figure 2.23 shows examples of planned handover tasks where the robot system is programmed to deliver the hammer and the jug in a comfortable way so that the handles of these objects are the part oriented towards the user.

### 2.3.4 Grasp intention recognition

The third phase of the handover task begins when the robot completes its motion, i.e. when the grasped object reaches the planned goal configuration in front of the user. In this last phase the system is programmed to recognize the intention of the user to grasp the object being offered. When the system recognizes that the user has been

**Algorithm: Detection of human grasp**

```
Input: ${}^{K}e$: robot tool center point;
Output: returns when the user grasps the object;
grasped=0;
while grasped$\neq$ 1 do
    $C \leftarrow$ get new point cloud in sensor frame $K$;
    ${}^{K}h \leftarrow$ get human hand position;
    Extract a bounding box from $C$ centered at ${}^{K}h$;
    Extract point cloud clusters in the bounding box;
    $C^{hand} \leftarrow$ cluster closest to the center of $C$;
    if ${}^{K}e \subset C^{hand}$ then
        grasped=1;
    end if
end while
```

Figure 2.24: Algorithm for detection of human grasp.

touching the object for two seconds it opens the gripper to release it. After releasing the object the handover task is completed. Users do not need any particular instruction to take the object from the robot. In the experiments in section 2.3.5 they are just told (for safety reasons) to start moving their arm towards the object after the robot has completed its motion. Users are free to use whatever hand they are most comfortable with.

The algorithm for automatic detection of contact between the human hand and the object is illustrated in figure 2.24. The algorithm, that is iterated until contact is detected, uses range data acquired from the Kinect sensor and the tracked position of the human hand that is closer to the robot. In particular, the developed procedure combines information from both the tracked position of the human hand, that is approaching the object, and the raw depth image. Let ${}^{K}h$ be the position of the moving human hand in the Kinect reference frame that is provided by the skeletal tracker.

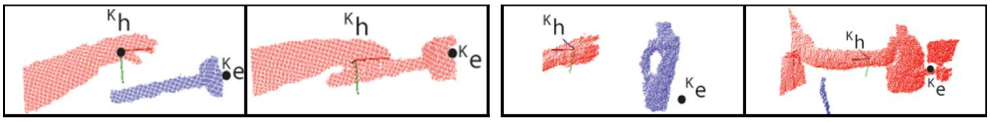


Figure 2.25: Two examples of contact detection between the human hand and the object (hammer at the left, jug at the right). Before contact the human hand and the object are detected as two separate point cloud clusters. After contact the human hand and the object belong to the same cluster.

Let also ${}^{K}e = {}^{K}_{R}T\,{}^{R}e$ be the position of the robot tool center point in the Kinect reference frame, computed through forward kinematics. The robot tool center point is located on the robot end-effector and, therefore, it is close to the object grasped by the robot. The iterative procedure starts by obtaining a new depth image expressed in the Kinect reference frame $K$ that is converted into a point cloud data structure $C$ (about 300$K$ points). Then, a bounding box is extracted from $C$ centered at ${}^{K}h$ that contains approximately 10% of the original points (line 5). The points lying in the bounding box are filtered by removing statistical outliers and clustered. Since the bounding box is centered at the human hand, the detected cluster closest to the center of the bounding box is labeled as the hand cluster (line 7). When the human hand is not in touch with the object delivered by the robot the human hand and the object are detected as two separate clusters. Conversely, when the human hand touches the object the human hand and the object are detected as a single cluster (figure 2.25). Therefore, the algorithm checks whether the robot tool center point ${}^{K}e$ lies within the human hand cluster (line 8). If the robot tool center point falls within the human hand cluster, then the system detects that the user is grasping the object and the robot releases the object. Otherwise, the robot keeps holding the object and the detection procedure is iterated.

### 2.3.5 Evaluation

A user study was conducted to test the performance of the proposed robot to human handover system. A total of 25 participants (15 males and 10 females) were recruited

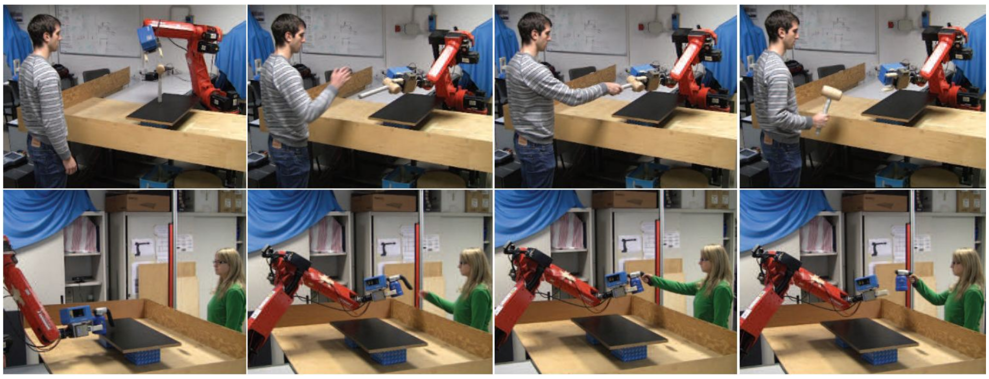


Figure 2.26: Sequences of images (left to right) of two handover experiments (hammer in the top row, blow torch in the bottom row). Objects are delivered by the robot to the user in a comfortable way by orienting the most appropriate part (handle) towards the user.

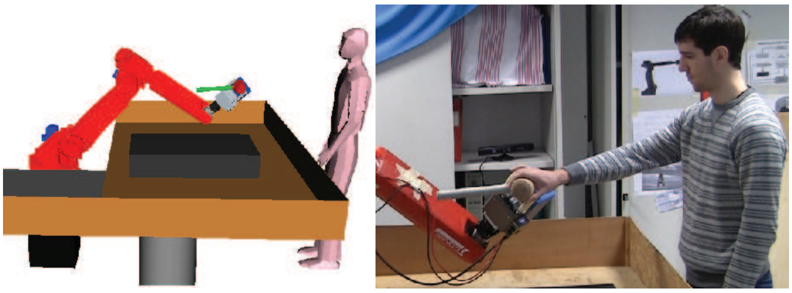


Figure 2.27: Example of a perturbed handover. The hammer is delivered to the user in an uncomfortable way.

among students and other members of the University of Parma. The age of the subjects varied between 22 and 35 years (mean age was $25 \pm 3.3$ years). About 90% of the subjects were right-handed. Each user was free to assume an arbitrary position and orientation in front of the robot. Users did not need any particular instruction to take the object from the robot. They were just told that the robot would give them an object and that they should start moving their arm to grasp the object at the end of the robot motion (for safety reasons). Each subject performed two trials of the handover

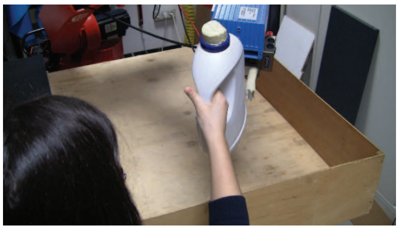


Figure 2.28: Image of a handover experiment from user's point of view. The jug is delivered to the user in a comfortable way by orienting the handle towards the human torso.

task with the same object without any practice session in order to collect an immediate and unbiased feedback. The object (hammer, jug or blow torch) was randomly chosen by the moderator. In one trial (randomly chosen) the object was delivered to the user by using the proposed approach for comfortable handover; in the other trial the same object was delivered by perturbing the goal orientation to force an uncomfortable object configuration.

Figure 2.26 displays two experiments of comfortable handover. For example, when a comfortable handover of the hammer was performed, the object was grasped by the robot from the head as shown in figure 2.26 (top row) and it was delivered so that the handle was oriented towards the user. Instead, when the perturbed handover was performed, the robot still grasped the head of the hammer but the object was delivered so that the head of the hammer was oriented towards the user as illustrated in figure 2.27. Hence, the perturbed case forced the users to grasp the object in an uncomfortable way because the head of the hammer does not afford grasping and also because users had to take the object by putting their hand close to the fingers of the robot. The most appropriate part to be oriented towards to the user is known to the system and it is specified by the moderator by using a graphical user interface. Figure 2.28 displays an image of a comfortable handover experiment from the user's point of view. After the experiments, all the participants answered a questionnaire with the purpose of assessing the level of comfort in both trials.

Table 2.3 reports the results of the questionnaire where the rating scale (Likert scale)

| **Question** | **proposed approach** | **perturbed approach** |
|---|---|---|
| Object was delivered comfortably | 4.9(±0.34) | 2.7(±0.7) |
| I felt safe | 4.7(±0.47) | 3.8(±0.75) |

Table 2.3: Questionnaire results: user average rating [1(strongly disagree)...5(strongly agree)].

ranges from 1 (strongly disagree) to 5 (strongly agree). All the users judged the proposed approach as comfortable, whereas they stated that the perturbed approach for object handover was less comfortable and not human-like. Results showed significant differences (Wilcoxon signed-rank test, $p < 0.005$). Moreover, the users felt safer when they received the object using the proposed approach with statistically significant differences (Wilcoxon signed-rank test, $p < 0.01$). It must be remarked that users were not told in advance which approach they were experimenting. Another result is that in 95% of the trials of comfortable handover users actually grasped the expected target part of the object, thus suggesting that the robot exhibited a socially aware behavior.

Table 2.4 reports the average times of the three phases of the handover task (Intel $@2.66GHz$). The first phase is very slow and, therefore, it was performed offline in absence of the user. The time required for planning 200 grasps is quite high as it includes a graphical animation. The online part of the task (starting when the user approaches the robot) required about 44 seconds for the user to receive the object with the proposed approach. The average time for the user to take the object was about $4s$ (measured from the instant when the robot stops at the goal configuration). In the perturbed approach the user spent, on average, twice the time to take the object (about $8s$). The difference suggests that in the proposed approach the user had a faster response time. The average time for taking the object has been measured by adding the time spent by the user to move his/her arm towards the object and the average time ($2s$) of the automatic grasp detection phase described in section 2.3.4. The most time

| Step | Time (s) |
|---|---|
| **First phase (offline)** | 111 |
| Scanning the environment (eye-in-hand laser) | 35 |
| 3D object modeling and segmentation | 16 |
| Robot grasp planning (200 grasps) | 60 |
| **Second phase (online)** | 40 |
| Human detection | 7 |
| Robot motion planning | 8 |
| Robot motion | 25 |
| **Third phase (online, proposed approach)** | 4 |
| **Third phase (online, perturbed approach)** | 8 |
| User hand motion towards the object (proposed) | 2 |
| User hand motion towards the object (perturbed) | 6 |
| Grasp detection phase | 2 |
| **Total (offline+online, proposed approach)** | 155 |
| **Total (offline+online, perturbed approach)** | 159 |

Table 2.4: Average times for the robot to human handover task.

consuming step in the online phase, which affected the time the user had to wait, was the motion of the robot manipulator towards the user. The robot was programmed to move slowly to exhibit a safe behavior for the user. In principle, the speed of the robot arm could be increased.

### 2.3.6 Coping with scenes with multiple objects

The proposed method for comfortable object handover has been extended to environments that include multiple objects. In such cases there are additional issues to cope with. First, the approach described in section 2.3.2 for 3D object modeling and segmentation, which requires a complete point cloud of the object, is not suitable due to inter-object occlusions. Indeed, it is difficult to obtain a complete scan of an object

**Algorithm: Object recognition**

```
Input: $C^{obj}$: object point cloud cluster; $M$: point cloud dataset;
Output: the closest prototype $M_{target}$; $H_{ICP}$: alignment transformation;
FPFH$_{C^{obj}}$ $\leftarrow$ Compute_FPFH_descriptors($C^{obj}$);
for $M_j \in M$ do
    FPFH$_{M_j}$ $\leftarrow$ Compute_FPFH_descriptors($M_j$);
    $H \leftarrow$SAC_IA($C^{obj}$,FPFH$_{C^{obj}}$,$M_j$,FPFH$_{M_j}$);
    $f = fitness(M_j, HC^{obj})$;
    if $f \leq f_{min}$ then
        $H_{IA} \leftarrow H$;
        $M_{target} \leftarrow M_j$;
        $f_{min} \leftarrow f$;
    end if
end for
$H_{ICP} \leftarrow$ ICP($H_{IA}$,$C^{obj}$,$M_{target}$);
```

Figure 2.29: Algorithm for object recognition.

when the environment contains multiple objects. To solve this problem an algorithm for object recognition from partial observations has been developed that compares the observed point clouds to a known dataset of complete point clouds. Second, a more advanced interaction technique is required to let the user select the object to be delivered by the robot. To this purpose voice recognition and text-to-speech have been used to enable natural user interaction.

### Object recognition

Object recognition is performed after extracting the point cloud clusters of the objects in the scene. The proposed algorithm is illustrated in figure 2.29. For each point cloud cluster $C^{obj}$ the algorithm finds the most similar element $M_{target}$ from a known dataset $M$ of complete point clouds. The method is based on the computation and

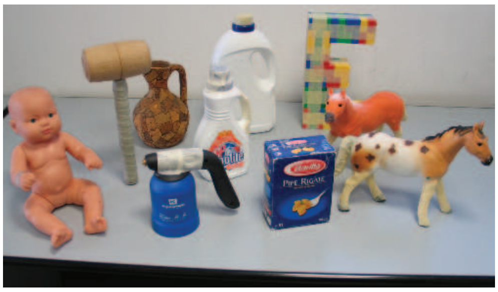


Figure 2.30: Dataset used in experiments with multiple objects.

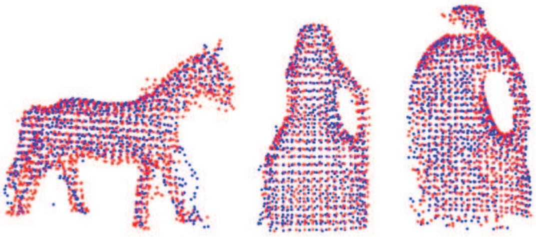


Figure 2.31: Examples of point cloud alignment (after ICP refinement) for object recognition showing the prototype point cloud in the dataset (displayed in red) and the observed point cloud cluster of the object to be recognized (displayed in blue).

the alignment of local point cloud descriptors. The dataset used in the experiments contains 10 complete point clouds that have been obtained by scanning the objects shown in figure 2.30 separately. The dataset also contains the reconstructed mesh of each object from the complete point cloud, its segmentation into parts, and the information about the most appropriate part to be oriented towards to the user. When the object is recognized the corresponding segmented model in the dataset is inserted in the simulation environment for planning the handover task.

The recognition algorithm starts by computing the Fast Point Feature Histogram ($\mathrm{FPFH}_{C^{obj}}$) of all the points of the observed point cloud $C^{obj}$ (line 1). Given a point $p$ of a point cloud and its $k$ nearest neighbors the Fast Point Feature Histogram

$FPFH(p)$ is defined as

$$FPFH(p) = SPFH(p) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot SPFH(p_k) \tag{2.3}$$

where $\omega_k$ is a weight factor, and $SPFH$ is the Simplified Point Feature Histogram [88] which is a local descriptor that encodes information about estimated surface normals of the k-neighborhood of a point. Then, for each complete point cloud $M_j$ in the dataset the algorithm computes FPFH$_{M_j}$ and it executes a sample consensus non-linear optimizer (SAC_IA, line 4) between FPFH$_{C^{obj}}$ and FPFH$_{M_j}$ on a set of correspondence triplets. The result is a transformation matrix $H$ that represents the estimated alignment between the two point clouds. The closest element $M_{target}$ in $M$ to the observed point cloud $C^{obj}$ is the one that minimizes a fitness function, that is computed as a sum of squared distances of corresponding points from $M_{target}$ to the input point cloud transformed by $H$ (lines 5-10). The transformation matrix between $C^{obj}$ and $M_{target}$ is named $H_{IA}$ (line 7). Transformation $H_{IA}$ is further refined through the iterative closest point (ICP) algorithm (line 12). Figure 2.31 shows examples of point cloud alignment after ICP refinement. After recognition, the corresponding segmented mesh in the dataset is inserted in the simulation environment for planning the handover task. The complete segmented mesh is placed in the simulation environment by applying the refined transformation matrix $H_{ICP}$. To evaluate the performance of the object recognition algorithm experiments have been conducted in different scenes by varying the number of objects and their pose (up to 5 objects per scene have been tested). A total of 100 object recognition experiments were performed with a recognition rate of 90%. The accuracy of the estimated transformation matrix has been evaluated by measuring the actual position and orientation of 20 objects. The average error in object position is $0.5 \pm 0.2cm$, while the average error in object rotation (about the vertical axis) is $1,8^\circ \pm 3^\circ$. The average errors are sufficiently small to allow robot grasping. The average time required to recognize one object (which is an additional step in the offline phase of table 2.4) is $15s$, and of course it depends on the number of elements in the dataset.

1: **phrasetype1:** <action>+"the"+<object>
2: **phrasetype2:** "it's"+...+<location>+...+<object>
3: <**action**>**:** "take"|"give me"|"get"
4: <**object**>**:** "jug"|"hammer"|"blow torch"|...
5: <**location**>**:** "front"|"behind"|"right"|"left"|...
6: <**answer**>**:** "yes"|"no"

Figure 2.32: Grammar used for speech recognition.

1: *User moves in front of the robot*
2: **robot:** "User identified, please select an object"
3: **human:** "Give me the jug"
4: **robot:** "Two jugs found, please specify which one you would like to receive"
5: **human:** "It's behind the horse"
6: **robot:** "Jug behind the horse requested, confirm?"
7: **human:** "Yes"
8: **robot:** "Request accepted"
9: *Robot gives the selected jug to the user*

Figure 2.33: Human-robot dialog: user selects the small jug located behind the horse by voice recognition (shown in figure 2.35).

### Voice interaction

Voice recognition and text-to-speech have been integrated to enable more natural user interaction allowing the user to select an object to be delivered by the robot. The voice interaction module (based on the Microsoft Kinect software development kit) is governed by a finite state machine and it was developed to recognize phrases from the grammar illustrated in figure 2.32. A first type of phrase is used for object selection (line 1). A second type of phrase is used to specify simple spatial relationships be-

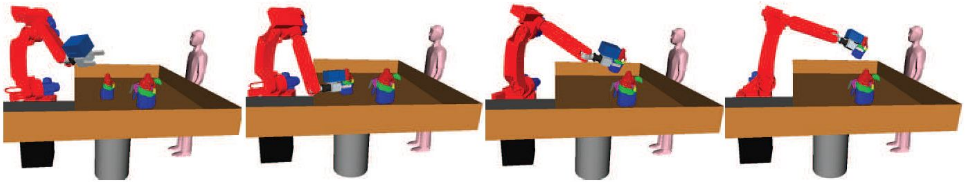


Figure 2.34: Motion planning of the handover task (with multiple objects in the scene) after selecting the small jug by vocal interaction (as shown in figure 2.33).

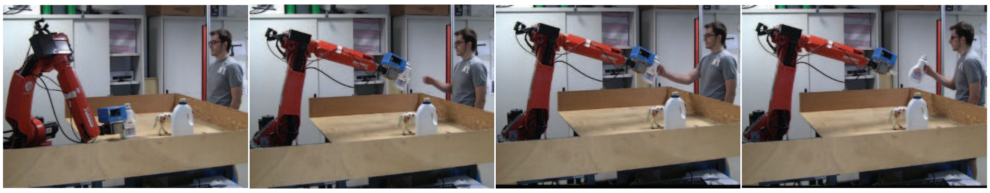


Figure 2.35: Execution of the handover task that was planned as shown in figure 2.34.

tween objects (line 2). This second type of phrase is required when the system asks the user to provide additional information to resolve ambiguous cases due to multiple objects of the same type in the scene. The Kinect speech recognition engine does not require any sort of training and it has allowance for verbal wildcards to improve flexibility.

Handover experiments performed with voice interaction required a trial period for the users to familiarize with the voice recognition interface. A new user group (with similar characteristics) was recruited with a total of 15 italian participants (9 males and 6 females). All the subjects were able to complete handover tasks (in English language) in different environments with multiple objects. The percentage of incorrect recognition of spoken words was about 10%. When the system did not recognize a word or an object not present was requested the user was invited to repeat the phrase until success.

In the rest of this section a complete handover experiment is described. Figure 2.33 reports the dialog between the user and the system. Figure 2.34 displays the motion planning phase, while figure 2.35 displays the execution of the handover experiment.

In the first phase the robot scans the environment and then object recognition is performed. The environment includes three objects: a toy horse and two jugs. When the user approaches the robot he/she is detected and the system asks which object should be delivered. The user says:“Give me the jug”, but since there are two jugs in the scene the system asks which one he wants. The user specifies that he wants the jug located behind the horse. Then the object is identified and the handover task is performed. The jug is delivered to the user so that the handle can be comfortably grasped.

## 2.4 Discussion

In this chapter two approaches for human-robot handover have been presented. The first approach, which enables human to robot handover, has been tested using the robot HERB. The second approach, which is a method for direct robot to human object handover, has been tested using a COMAU industrial manipulator. Both systems developed are *human aware* and exploit the Microsoft Kinect to perform human detection and skeletal tracking.

The former approach allows HERB to actively participate in human-robot handoffs. This approach uses human tracking and 3D point data from the Kinect sensor to determine if a handoff is desired by a human and to control the robot arm to take the object from the human. The intention of the human to perform the handover is inferred through gesture recognition. Two different control algorithms were tested: A planner and a reactive controller. While the two control methods each have their strong and weak points, I believe that the reactive controller has the best potential. The reactive behavior allows the robot to adapt and synchronize its motion to the human counterpart, which is a desirable feature in physical human-robot interaction.

The latter approach enables comfortable object delivery from a robot to a human. The novelty of the approach lies in its facilitation of user comfort when receiving the object. The proposed method achieves comfortable object handover by planning the motion of the robot so that the appropriate part of the object is oriented towards the user. In addition, with the approach proposed the robot determines when the human wants an object and which object has to be delivered through verbal communication User studies have shown that using these approaches robots and humans can effectively and intuitively work together to complete the handover without specific instruction to the human. Both systems perform object recognition and detect when the human and robot are ready to start the transfer of the object. However, since they have been developed using different robotic platforms, these functionalities have been developed in different ways.

The system developed on HERB uses range and color data provided by the Kinect to perform object recognition. Since the Microsoft Kinect has been located at 2.5 *m* from the scene, it was possible to use only global features to recognize the object. On the contrary, the system developed using the COMAU was provided with a laser scanner on the robot end effector, used to recognize the object. Since the 3D point cloud provided by the laser was pretty accurate it was possible to use more robust techniques for object recognition (such as FPFH [88] or NARF [72]) and to segment the object to make the robot aware of the object affordances. A similar approach can be useful also when receiving an object. During the user study described in section 2.2.4 it happened that the robot accidentally touched the human while grasping the object. It was not dangerous nor painful for humans, but they found it awkward. This issue could be dealt with by detecting the part of the object available for the grasp. In addition, a programming by demonstration technique [89] is useful for teaching the robot which is the most appropriate part of an object to grasp or to be offered to the user. As an alternative to the laser scanner, a second RGB-D camera on board could be used to perceive the object. In the case where two RGB-D cameras were available, the first camera would give us information about human motion in the scene and the second on board camera would give us better information around of the points of interest near the robot.

In order to detect when to transfer the object, HERB uses both force and visual sensors while the system developed on the COMAU exploits only visual information. Even if the vision-based system has shown to work properly in real cases ( all the users in the user study completed successfully the transfer of the object ), if we think about giving an object to a human we can easily identify as fundamental the tactile feedback that we feel on our fingers during the transfer of the object. If a human closes the eyes and he gives an object to another person the handoff can be still performed in a safe and reliable way. Nevertheless when it comes to robot-human handovers obtaining a tactile feedback comparable with the tactile feedback of human fingers is hard. Moreover having very accurate tactile sensors is expensive. For this reasons I believe that fusion between tactile sensors and visual sensors is a key point to build efficient and reliable handover systems. Sensor fusion is particularly important when the robot actively approaches the human hand. HERB has shown during user studies how it can approach the human hand in a “gentle” way using sensor fusion.

The availability of reliable human-tracking systems like the Kinect and smart control algorithms that use the human-tracking feedback and other sensor data can lead to rich human-robot interactions where the robot is aware of the human and they can actively collaborate together towards a final goal. In this work, the use of rich 3D RGB-D data for human-tracking and object detection, combined with incremental control methods, created handover systems that are reliable and can take some of the mental and physical burden off the human. Continued research in this area will lead to very intuitive, comfortable and efficient human-robot interaction on a level unseen before.

# Chapter 3

# Toward seamless Human-Robot Handovers

After addressing specific aspects of human-robot handovers I was interested in finding a holistic representation for this kind of interaction. A handover is a complex collaboration, where actors coordinate in time and space to transfer control of an object. This coordination comprises two processes: the physical process of moving to get close enough to transfer the object, and the cognitive process of exchanging information to guide the transfer. Despite this complexity, we humans are capable of performing handovers seamlessly in a wide variety of situations, even with strangers. This suggests at a common procedure that guides all handover interaction. This section proposes how that procedure can be encoded.

This result is achieved by studying how people hand objects to each other in order to understand their coordination process and the signals and cues that they use and observe with their partner. Based on these studies, it is proposed a coordination structure for human-robot handover, consisting of physical level and social-cognitive level coordination behaviors [90]. The structure presented describes how people establish the what, when and where of a handover: to agree that it will happen (and with what object), to establish the timing, and to decide the configuration at which the handover will occur. Then I explain how the robotic applications described in this

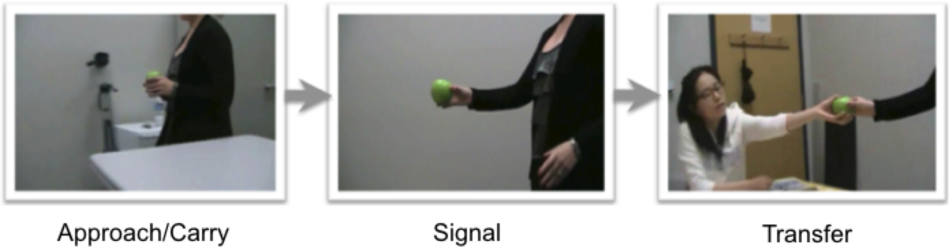


Figure 3.1: Handover activities observed in human-human handover study [91].

thesis address specific aspects of this structure and how user studies have been used to examine some specific aspects of human-robot handovers. Finally, I offer design implications for seamless human-robot handover interaction.

## 3.1 Learning handovers from humans

Handovers are complex interactions, yet humans are capable of performing handovers seamlessly and without conscious thought. This suggests people share a common procedure that guides the handover interaction.

An interesting user study has been conducted in [91]. In [91] five pairs of participants handing objects to one another in a kitchen were observed. Each person in the pair took turns being a care-giver who brought objects to a care-receiver. The care receiver was either sitting on a chair and reading magazines, or standing and packing a box on a table. For each of two scenarios, they transferred 10 objects, resulting in 40 trials for each pair. The session took about 30 minutes to complete, and was videotaped. The trials were analyzed by coding the videos based on physical activities that the subjects were engaged in: carrying, reaching out its arm to signal to indicate its readiness, and transferring. In addition it was noted who signaled its readiness to engage in handovers - giver or receiver, in order to extract coordination patterns. The three activities are shown in Figure 3.1.

**Approaching/Carrying.** When carrying objects and approaching receiver, the givers were carrying objects in distinct postures. 66% of the time, participants used both hands when carrying objects, even though the objects used in the experiment were not heavy.

**Signaling** All givers and receivers coordinated when and where the handover should happen by communicating their readiness through various communication cues:

*Giver signaling readiness.* Givers who were carrying an object with two hands, just prior to coming to a stop in front of the receiver, signaled a handover by dropping a hand and reaching out with the object. Givers using one hand, reached out with the object. Givers typically started reaching out before they came to a stop near the receiver. However, they did not perform this early reaching behavior if the receiver was not paying attention, which leads to believe that reaching out was used as a signal.

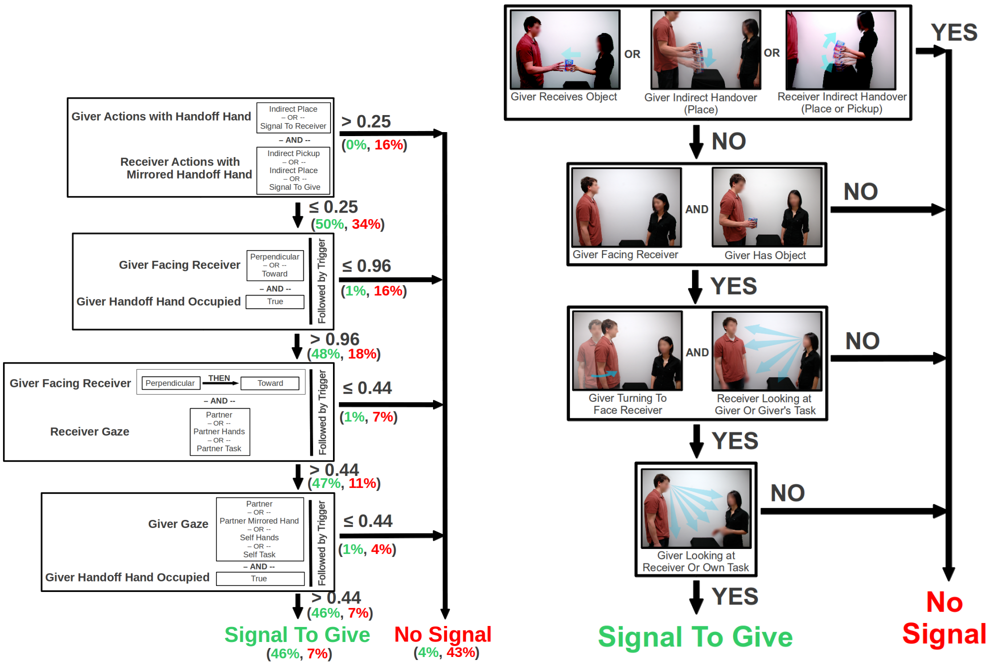
*Receiver signaling readiness.* The receiver often signaled receptivity by making a grabbing hand gesture with one arm or two. This behavior was observed in receivers significantly more often when givers were carrying a cup, pens, or a tray with a glass of water on it. These objects are more likely to be problematic if dropped (as compared with a newspaper or book, for example), so it makes sense that receivers should nonverbally reassure givers they are ready to receive the handover.

*Coordination patterns.* The most common coordination pattern (58% of trials) was givers communicating a desire to hand over an object by coming close to the receiver. The giver moved the hand holding the object toward the receiver's hand, and the receiver then would take the object. The second most common coordination pattern (34% of trials) happened when givers reached out the hand with the object at a point where the distance between the two participants was further apart than the sum of their two arm lengths. In these situations, the participants closed the gap somewhat but were further apart when the object was actually transferred. In those cases, the receiver also reached out an arm to grab the object. The giver would then move his or her hand toward the receiver's hand. Some receivers exhibited very co-operative behavior by leaning their bodies forward while reaching out their arms. The third pattern, although less common (7%), happened when the receiver waited with a grabbing hand gesture but was not looking at the giver. The givers came close to the receivers who did this and put the objects into the receiver's hands. The two less common patterns were more frequent when receivers were standing, suggesting that either the receiver's standing position and/or the busyness of the receiver (sorting items into a box) led to more signaling and intricate coordination between givers and receivers.

**Transfer.** On average, the distance between the giver and the receiver did not vary across objects. Also, all the objects were transferred at a height that was below the receivers neck, (chest level or below). A majority of the object handovers were above the waist. In 24 turns, givers turned a newspaper, book, cup, or pot so that receivers could more easily receive the object. For example, the giver would rotate the cup so that the receiver could grab the handle. This phenomenon occurred in 30% of the turns for those four objects.

The study shows three main activities that happen during human-human handovers: 1) carrying, 2) coordinating, and 3) object transfer. When givers were carrying an object, they held it with two hands, exhibiting a very distinct posture when compared to extended arms. As givers approached receivers, givers or receivers indicated whether they were ready by reaching out their arms, and their partners responded by moving their hands toward them. When givers signaled their readiness, they seemed to take into consideration the receivers attention and interruptibility (e.g., looking at givers vs. tasks at hand), and social norms (e.g., being polite by rotating a cup so that a handle faces a receiver). The results of this first study suggest that people use reaching action to communicate their intent to handover and help coordinate the timing of the handover. As soon as givers reached out their arms, receivers responded by reaching out their arms toward the object that the giver was holding. The varying coordination patterns between givers and receivers suggest that givers intentionally time when to signal rather than randomly reaching out. For example, when receivers were looking at magazines, givers did not reach out their arm until they got close to receivers. On the other hand, when receivers were looking at the givers, givers reached out their arms while still moving toward receivers.

In a second study [76], the authors analyze communication prior to givers' reaching action in order to understand how givers decide when is the right time to signal their


Giver Actions with Handoff Hand
Indirect Place – OR -- Signal To Receiver
– AND --
Receiver Actions with Mirrored Handoff Hand
Indirect Pickup – OR -- Indirect Place – OR -- Signal To Give
> 0.25 (0%, 16%)
≤ 0.25 (50%, 34%)
Giver Facing Receiver
Perpendicular – OR -- Toward
– AND --
Giver Handoff Hand Occupied
True
Followed by Trigger
≤ 0.96 (1%, 16%)
> 0.96 (48%, 18%)
Giver Facing Receiver
Perpendicular THEN Toward
– AND --
Receiver Gaze
Partner – OR -- Partner Hands – OR -- Partner Task
Followed by Trigger
≤ 0.44 (1%, 7%)
> 0.44 (47%, 11%)
Giver Gaze
Partner – OR -- Partner Mirrored Hand – OR -- Self Hands – OR -- Self Task
– AND --
Giver Handoff Hand Occupied
True
Followed by Trigger
≤ 0.44 (1%, 4%)
> 0.44 (46%, 7%)
Signal To Give (46%, 7%)
No Signal (4%, 43%)
Giver Receives Object
OR
Giver Indirect Handover (Place)
OR
Receiver Indirect Handover (Place or Pickup)
YES
NO
Giver Facing Receiver
AND
Giver Has Object
NO
YES
Giver Turning To Face Receiver
AND
Receiver Looking at Giver Or Giver's Task
NO
YES
Giver Looking at Receiver Or Own Task
NO
YES
Signal To Give
No Signal

(a) Sequence Feature Decision Tree

(b) Interpreted Decision Tree

Figure 3.2: Decision tree classifier used to predict reaching actions in human-human handovers [76].

readiness to handover. They observed 27 human pairs performing a task that required handovers. The participants were placed in a kitchen environment and tasked with putting away a bag of groceries and packing a picnic basket. Each experiment lasted an average of 250 seconds during which the participants interacted with the bag of groceries, picnic basket, kitchen cabinets, refrigerator, and each other performing an average of 9.2 handovers per experiment. The experiments were recorded using three color cameras, four depth cameras, and two microphones. From this data eye gaze, 2D position, object location, and handover actions were manually annotated at 10 Hz. A machine learning technique called feature selection was used to automatically extract sequences of events that are predictive of the physical reaching actions. Then, these *sequence features* were used in a variety of machine learning classifiers and it was found that a decision tree performs best on the evaluation data set with a classification accuracy of 89%, as shown in Figure 3.2. Finally, this decision tree was validated on a test data set where it accurately predicted 82% of the reaching actions. After interpreting the decision tree, it was determined how to predict the intent to handover. When the following four features are all true for one of the participants hands in the data set used for the study, there is a 89% probability that a handover with that hand will directly follow:

- **No previous signals:** Within the previous three seconds the hand did not receive an object and neither participant has performed indirect handovers.

- **Giver orientation and hand occupancy:** At the end of the sequence the giver must have an object in his hand and must not be facing away from his partner.

- **Giver orientation and receiver gaze:** At the end of the sequence the giver must turn to face his partner and the receiver must be looking toward him.

- **Giver gaze:** At the end of the sequence the giver is either looking at his hand or at the receiver.

The majority of misclassified examples corresponded to handovers where there was no communication of intent prior to the reaching actions. In these cases, the giver reached out to communicate the intent to handover and expected the receiver to take the object when able.

These four features can be interpreted to make the following claims. Participants do not perform both indirect and direct handovers at the same time. Joint attention, and not mutual eye gaze, is a major signal when communicating intent and coordinating reaching actions. Distance between participants is not a discriminative feature, meaning that reaching actions can be started when the participants are not near each other. These results suggest that humans often implicitly signal each other prior to reaching out, communicating their intent to handover and coordinating the start of reaching actions.

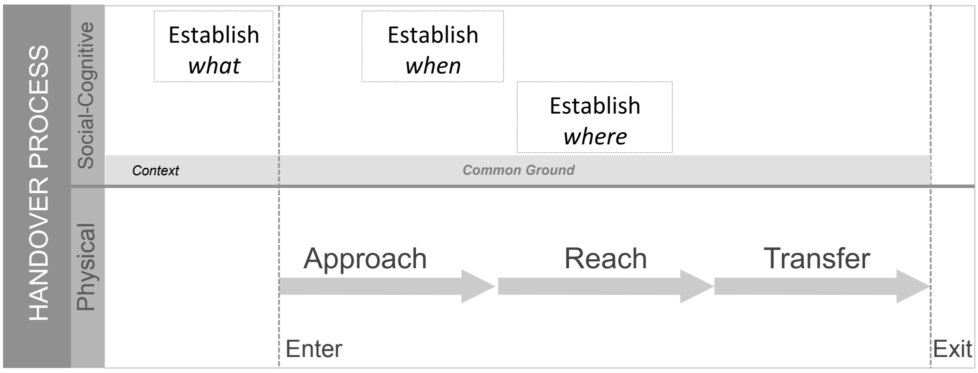


Figure 3.3: The canonical handover process (physical and social-cognitive channel) for an assistant fetching an object for a requester. The actors first agree that the handover will happen and what object will be transferred. For example, the requester could ask for the object and the assistant could verbally agree. Next, after the assistant has retrieved the object, he starts approaching the requester while carrying the object. The two actors now exchange communication cues to establish when the handover will happen. For example, as the assistant approaches, he and the requester can exchange looks, establishing that they are both ready and the handover can begin as soon as they are close enough. They start reaching at the same time, establishing where the handover will happen based on their motion and their common ground. They then transfer the object and exit the joint activity.

## 3.2 The Handover Structure

The studies on human-human handovers reported above show that people coordinate their behaviors both at physical and social/cognitive levels. Physical coordination involves actions such as approaching/carrying, reaching, and transferring of control that enable object handover. Social-cognitive coordination includes activities that establish agreement on *when/timing* and *where/location* of handovers between two people. For example, the studies show people signal their readiness to start handovers to their partner through non-verbal cues such as gaze, body orientation and starting to reach out the arm. In this section, this notion is generalized to three coordination problems in the social/cognitive level: the *what*, *when* and *where/how* of the handover.

The physical and social/cognitive level coordinations are closely intertwined. For example, an action of reaching out the arm both serves a role of moving an object closer to the receiver and communicating the intent to start the handover. For the descriptive purpose, however, I explain coordinating actions that occur at each level.

In the following section, I describe physical and social/cognitive level coordination activities involved in the handover process. According to theories of common ground and joint activity [92], context (common ground) and joint commitment are added prior to physical and social level coordination in the handover structure model below. This handover structure model is described with four exemplary situations: A *care-giver* handing over a glass of water at the patients request, a car *mechanic* reaching out while working and asking his assistant for a wrench, a *fire brigade* line in which a group of citizens is passing water buckets from a water source to a fire, and finally an employee handing out concert *fliers* on a busy university sidewalk.

The care-giver example follows a typical handover: first, *what* is established, then the care-giver approaches, the *when* is established before reaching starts, and the *where* is established during reaching – just like in Figure 3.3, which shows the preferred interaction for a typical fetching task. However, as this section will reveal, not all handovers follow this timeline, but as long as the handover structure coordinates the *what*, *when* and *where*, handovers will be seamless.

Throughout this section, I use *common ground* – the information the actors in a handover share and know they share [92] – as a foundation for the social/cognitive coordination.

### 3.2.1 Context

Context – the state of the world before entering the handover activity – is very important for handovers. Social contexts such as norms or roles [93] influence how people behave and what they expect from other people. The handover process will be different depending on what context the handover is happening in. The following examples illustrate very different contexts and their impact on the handover process.

*Examples:*

*Care-Giver:* The context contains the roles of the patient and care-giver (e.g., the care-giver is supposed to fulfill the patient's requests), and previous handover experiences (e.g., the patient has limited reaching capabilities).

*Mechanic:* The context contains the roles of the actors, as well as the fact that the mechanic is working underneath a car and cannot see the assistant.

*Fire Brigade:* The context here contains an established procedure of swinging buckets from one person to another, and the fact that the state of emergency has eradicated many of the usual social guidelines (e.g. personal space).

*Flier:* The employee has no prior relationship with the people on the sidewalk, so established social norms shape the behavior that occurs.

### 3.2.2 The Physical Channel

The physical channel is strongly tied to the social-cognitive channel: the physical channel implements what the social-cognitive channel decides: e.g. when to handover, but also what cues to use for communication.

**Carrying/Approach:** The carrying pose during approach conveys information about the object (eg. weight, fragility, importance). Depending on context/common ground, this plays a role in coordinating *when* and *where*. The social-cognitive channel might also dictate additional communication cues during approach, like gaze and body orientation.

**Reaching:** In [49] Flash at al. found that human hand trajectories often follow a minimum-jerk profile. Huber at al. [94] observed seated humans handing over objects to one another and came up with a novel trajectory generator based on a decoupled minimum-jerk profile that reproduces the reaching motions of the humans and performs similarly to the minimum-jerk profile. Reaching plays a role in coordinating both the *when* and the *where*.

**Transfer:** In the majority of everyday handovers, both actors are in direct contact with the object and the object and actors are stationary with respect to one another. In these situations the actors transfer control by the giver and receiver exchanging the object load due to external forces such as gravity and wind. After transferring the entire object load, often the receiver will retract or otherwise move the object to signal the giver that the handover is complete. Then the giver will retract his arm signaling the same, thus ending this phase and handover interaction. Chan at al. [95] found a linear relationship between grip force and load force except when either actor is supporting very little of the object load. Analysis of these grip forces suggests that the giver is responsible for the well being of the object during the transfer, while the receiver is responsible for the timing of the transfer.

### 3.2.3 The Joint Commitment on What - Agreeing to handover

Before handing over, the giver and receiver must both agree that they are willing and able to perform the handover. People come to this agreement after one actor proposes the handover and the other actor accepts it. People signal these proposals and acceptances using both actions (verbal and non-verbal) and context, and use the current common ground to decide on what is appropriate. For example, people with little common ground may need to rely on speech to propose and accept the handover, while people who handover with each other frequently have more common ground and may use more subtle and efficient signals such as gestures to propose and accept the handover. In the studies described in section 3.1, this agreement was either implicit in the task description, or the participants asked for a particular object. Once the agreement to handover is established, it enters the common ground.

*Examples:*
*Care-Giver:* The patient verbally proposes the handover and the care-giver verbally agrees.
*Mechanic:* The mechanic reaches out, as well as asks for a wrench. This, together with context (the assistant is around and his role is to fulfill the mechanic's requests).
*Fire Brigade:* The agreement is assumed based on context alone in this case: their mutual participation in the task.
*Flier:* The employee expresses his desire to interact by facing and approaching a passersby and reaching out with the flier. The joint commitment or agreement that a handover will happen, however, only takes affect when the passer-by confirms by

reaching out or by establishing mutual gaze with the employee. This is an example in which joint commitment is established very late in the handover process, after the giver has finished reaching.

### 3.2.4 Coordinating When - Signaling readiness to handover

In the user studies described in the previous section, we found that a way to establish the handover timing is to start reaching out. However, we have found that people also use communication cues before than reaching starts (e.g. gaze, body orientation) that dictate the exact moment of the reaching and establish when the handover will occur. In [96] it is shown that eye gaze can be used to infer action intention. In [97] Sebanz at al. show that joint attention helps to coordinate the initiation and progression of joint actions. Indeed, in [76], we found that readiness to handover is sometimes established by turning towards and focusing on the other actor or the item to handover. Furthermore, the study conducted in [91] indicated that when the giver is preparing for a handover, he sometimes holds the object in a carrying pose. The carrying pose conveys information about the object (eg. weight, fragility, importance). Depending on common ground, the carrying pose can immediately convey to the receiver the desire and readiness to handover. The giver may also grasp the object in a way that will facilitate the physical transfer of the object (e.g. allow the giver to present the mug's handle to the receiver), which also contributes to signaling readiness.

*Examples:*
*Care-Giver:* The care-giver focuses on the patient. When the care-giver is in view, the patient looks up to the care-giver and the glass of water. Based on their common ground, this joint attention on the handover-enabled scene signals that both actors are ready to reach, establishing the time of the handover before either party reaches.
*Mechanic:* The mechanic continuously signals his readiness by holding his hand out. However, the timing is only set when the assistant signals readiness by placing the wrench in the mechanic's hand.
*Fire Brigade:* The rhythm of the task (context) determines the timing of the handover, with no explicit actions required.

*Flier:* The timing in this example is established at the same time as the agreement to handover, once the passer-by starts reaching back.

### 3.2.5 Coordinating Where - Establishing the configuration of the handover

The handover configuration is the pose the actors have when they start transferring control of the object (e.g. arms extended and hands grasping the object). In most cases, the giver and receiver negotiate the handover configuration as they reach towards each other. During the reaching, the giver and receiver will pose their hands and their grasp on the object to communicate how they wish to transfer the object (e.g. one actor holds one end of a rod so the other actor can grasp the other end). The reaching communicates some information about the actors and the object (e.g. the object is heavy, one actor cannot reach any further, one actor is reaching more slowly than the other so the other needs to move closer). For problematic objects (e.g. glass of water), the receiver may reach out more to presumably ensure communication of readiness. In [91], we found that when a care-giver is fetching an object, the physical handover location is at torso level (between the waist and neck) and the object is presented to allow for easy grasping.

*Examples:*
*Care-Giver:* They reach toward each other based on their previous experiences with each other and meet somewhere in between. The care-giver enforces that the handover occurs with the glass upright.
*Mechanic:* The mechanic specifies how he will accept the handover with his hand pose and the assistant complies. From experience working with the mechanic, the assistant will pose the wrench in the mechanic's hand so the mechanic can immediately use the wrench.
*Fire Brigade:* The giver and receiver have established through routine an agreed upon orientation and location for the handover.
*Flier:* The passer-by reaches out to where the employee is offering the flier: the handover configuration is established by the employee.

### 3.2.6 Justification in Interaction Theory

The handover structure proposed resonates with the interaction theory in [92] by representing the handover as a sequence of phases within a joint activity (approach, reach, transfer), requiring joint commitment (the *what* - the handover agreement) and using common ground. The actors in the joint activity synchronize in the transitions between these phases, e.g. establishing the *when* nominally transitions from a preparatory phase (approach) to a reaching phase. Once committed, the failure of an actor to perform his part affects the public perception of every actor's self-worth and autonomy [92], thus creating a social consequence based on the success or failure of the handover.

## 3.3 Human-robot handovers

The knowledge of human-human handovers can be ported to robots to examine some specific aspects of human-robot handovers. In order to design a system for seamless human-robot handover all the phases of the structure presented in this chapter have to be addressed. In this section I aim to give an overview on how the phases of the handover can be addressed in Human-Robot handover. To this purpose, first I briefly discuss how the research works in the literature fit the proposed structure, then I discuss how and which phases have been addressed by the robotic applications presented in this thesis.

### 3.3.1 Human-robot handover phases in literature

I already discussed in chapter 1 the state of the art of human-robot handover. In order to give a complete overview on how the phases of the handover can be addressed in Human-Robot handover, Here I briefly discuss how the research works in the literature fit the proposed structure.

**Carrying/Approach:** In [58] Kim et al. investigated how a robot can grasp an object before handing it to a human that incorporates the object's shape, the object's function, and the safety of both the robot and human. Similarly, in [57] a planner was presented able to grasp unknown arbitrary objects for interactive manipulation tasks. In [98] Sisbot et al. developed a navigation planner that creates safe, legible, and socially acceptable paths. Takayama et al. [99] explored how personal space (proxemics) varies when approaching and being approched by a robot based on the human's experience with robots and where the robot looks during the approach. In [100] Mumm et al. studied how proxemics varies with a robot's likeability and eye gaze. Mainprice et al. [101] created a trajectory and motion planner that can vary the amount of human motion required to handover, allowing the robot to choose the best handover location based on context.

**Reaching:** In [47] Glasauer et al. investigated how a robot can convey the intent to handover and signal its readiness using human-like reaching gestures.

**Transfer:** A physical aspect of the handover is transferring control of the object. Nagata et al. [59] presented a grasping system based on force and torque feedback that senses when the humans has a stable grasp on the object, after which the robot can release the object. In [102] Sadigh et al. presented a robotic grasping controller inspired by human grasping to grasp an object with minimal normal forces while ensuring that the object does not slip.

**Common Ground:** During the handover, the actors use their common ground to decide how to communicate with each other, plan tasks, and coordinate actions. Hoffman et al. [103] created a measure of fluency for human-robot interactions and also found that anticipatory agents are more efficient than pure reactive agents. This work highlights that robots should be able to accurately predict for human actions.

**What - Joint Commitment:** Before handing over, the actors must have a *joint commitment* to handover, establishing that they are both willing and able to perform the handover. One capability that facilitates entering a joint commitment and maintaining the commitment is the recognition of engagement. Rich et al. [104] observed engagement in human interactions, used these observations to derive four types of events that contribute to the perceived engagement, and created a computational pipeline that robots can use to detect these events and determine human engagement.

**When:** The handover process requires the actors to coordinate *when* the handover will occur. From the study in [76], we know that eye gaze is very important when signaling when to handover in human-human handovers. Mutlu et al.[105] examined the effectiveness of gaze cues performed by robots that are designed with abstracted human-like features. In [106] Cakmak et al. confirm that arm extension is an important signal to communicate readiness to handover. They also suggest that having a distinct carrying posture prior to extending the robot arm, is critical for people to understand when they can grab an object from the robot. Grip positions may matter less in terms of signaling its readiness to hand over an item, though it may make it more convenient for people take an object out of robots' hands.

**Where:** The handover process requires the actors to coordinate *where* the handover will occur. In [60] Edsinger et al. found that during a handover humans will pose an object in the robot's stationary hand regardless of the robot's hand pose, demonstrating that humans adapt to the robot's hand pose. Pandey et al. [61] investigated how a robot can predict where the human will handover and then proactively move to this location. Sisbot et al. [63] developed a manipulation planning framework that chooses handover locations based the human's safety, accessibility, field of view, posture, and preferences. In [11] Cakmak et al. suggest that robot should choose the best handover configuration taking into consideration the human's preferences, i.e. choose the best object location, object pose, and arm configuration.

### 3.3.2 Human-robot handover phases addressed in this thesis

The two systems described in the previous chapter implement various aspects of human-robot handovers and the user studies conducted evaluate their performance. These aspects are all parts of the physical process of handing over, specifically how to present and negotiate the physical handover, but they also are used to coordinate when and where the handover should occur.

The system described in section 2.2 was used to explore how to perceive human readiness and how to negotiate when and where to hand over. This issue has been already addressed in [61] in cases where the robot suggests the hand over location. In my study, I focused on cases where the human suggests the hand over location while the robot complies, such as in the mechanic example where the robot is the assistant. The system developed is able to infer the human *readiness* by detecting reaching gestures of the human, to negotiate the *where* by tracking the position of the object that the human is holding, and to convey the robot readiness to start the transfer by reaching out and softly touching the object. These skills have been achieved using an accurate perception system and a reactive controller to move the robot arm which, during a user study, has proved to be suitable for this kind of interaction. Since the robot actively negotiates where to handover up to the contact with the object, it needs to behave in a way which is accepted by the human and that makes the human feel comfortable and safe. In the user study described in section 2.2 I found out that in order to obtain such a suitable behavior the following four factors are crucial during the interaction when negotiating the *where* and *when* of the handover:

**Forcefulness:** The robot must be able to apply the proper force to the object in the final part of the reaching and during the transfer of the object.

**Aggressiveness:** The robot must tune its velocity. It is important that the robot does not move too fast when close to the human. Its velocity should be directly proportional to the distance from the object.

**Predictability:** Predictability makes humans more comfortable around the robot because they can plan into the future and know that it won't do anything strange.

**Timing:** Human-human handover are fast. Robots should be as fast as humans to obtain a seamless interaction.

The system described in section 2.3 was exploited to study *where* the robot should hand over an object when delivering it to a human, how it should present the object and how the robot can detect the human *readiness* to start the transfer of the object. In addition, the system has been used to study how the *joint commitment* can be obtained (through verbal interaction) in cases where the robot does not have enough information to infer a handover based on human cues and the context does not contain information about which object has to be delivered. User studies showed that in order to obtain a seamless interaction the robot should include the following considerations in its behavior:

- the robot should perceive the human and his/her position and configuration in order to propose the object in front of him when it is possible.

- the robot should take into account the object grasp affordances and deliver the object so that its most appropriate part (e.g. a handle) is oriented towards the user. The user study also shows that the average time for the user to take the object is lower when this behavior is adopted. This suggests that the proposed approach makes the transfer easier for the user and may help to convey the robot readiness (even if less than the robot reaching gesture).
- In oder to detect the readiness of the user to start the transfer the robot can use its vision system (to detect the grasp of the person). An effective and reliable system can be obtained using the vision system together with tactile sensors.
- Explicit verbal communication can be used to agree to hand-over. When the robot does not have enough information to infer a handover based on human cues, the robot should be able to understand simple verbal requests from the human. Very simple grammars can enable intuitive and effective interactions. A simple but effective grammar has been presented in section 2.3.6.

## 3.4 Discussion

Human-human handovers suggests that humans handing off items to humans coordinate their handover process using context and cues that are physically and verbally communicated, and that robots can adopt some of these signals to coordinate handover activities with people. Throughout my studies, I found that people could easily understand human-like cues when performed by a robot, and that they preferred these cues over machine-like ones.

I offer the following design recommendations for seamless human robot handovers. First, HRI designers should rely on human-like gestures and cues for seamless handovers. Second, HRI researchers should model social norms that are well-codified and heavily relied on in certain social settings. Third, HRI designers should implement robots with capabilities to detect how people want to establish the what, when, and where of handovers. It is also important for robots to respond using human-like gestures and signals (so that people know that robots are responding to their signals).

Two exemplary scenarios for seamless human-robot handovers are presented below:
**When the robot is a giver:**

- The robot receives a request from a person about what to retrieve
- The robot carries the object in a distinctive carrying posture while approaching the person
- When the robot is getting near the person, the robot observes her eye gaze (i.e., whether she is looking at the robot) and interruptibility (i.e., not holding objects)
- Upon finding a good moment to interrupt, the robot reaches out the object toward the torso of the person
- If the person reaches out its arm toward the robot before the robot reaches out, the robot reaches out in response to hand the object out to the person
- If the robot cannot find a good moment to interrupt while traveling, it stands next to the person and reaches out its arm toward the person's hand

**When the robot is a receiver:**

- The robot receives a person's request to give an object to the robot
- The robot approaches the person with arms close to its body (in order to communicate that the robot is not ready to receive)
- When the robot is getting near people, the robot observes the person to see whether she reaches out her arm with an object
- Once the person starts to reach out her arm, the robot responds by reaching out its arm toward the object
- If the person does not reach out her arm, the robot reaches its arm and opens its hand to signal its readiness to receives an object

Like any study, the research presented in this dissertation has many limitations. Human-robot handovers were observed and evaluated through laboratory experiments. Furthermore, human-robot behaviors were implemented to test specific aspects of the handover structure. Future work will implement and evaluate the entire seamless human-robot handover structure in more general settings. In addition, applying the proposed framework and guidelines to handovers will uncover further design and research questions that are of interest. For example, in human-robot handovers, how can seamlessness be maintained when robots have a primitive arm, are not anthropomorphic, or have very simple sensing and actuation capabilities? An interesting question is how robots should behave, based on their limited knowledge of the social and cognitive context of a situation. Also, how can robots learn social and cognitive context?

This work suggests new opportunities for research in human-robot handovers, and for exploring the role of social information, cognitive information, and context, in improving interactions between people and the robots that work closely with them.

# Chapter 4

# Conclusions

Until a few years ago robots were confined to factories and other industrial settings. However, robots are currently in the progress of moving out from the factories and into our homes and offices, for example, to run errands for us or otherwise assist us. In this context Human-Robot Interaction (HRI) is particularly relevant. During my research activities I focused on physical HRI. In particular I addressed the issue of the hand-over of an object between a human and a robot. This highly collaborative task is very common in everyday life, and robots that will operate in domestic environment will inevitably have to deal with it.

The hand-over task has been vastly investigated in literature. Traditionally, robot put most of the effort during the interaction on the human partner. This happened because until very recently it was hard for robots to have a strong perception of the human, and therefore they could not reliably plan actions in the proximity of a human being. In order to accomplish the hand-over, robots showed their intention through a reaching gesture and let the human take the object from their hands or position the object into their hands. Recently, the fast advancement in computer vision and robotic technology has enabled the development of perception systems that are able to accurately perceive humans. With the availability of these new systems it is possible to break the human-centric interaction, that depends mostly on human effort, and allow the robot to take initiative by acting in the proximity of the human body, thus decreasing the cognitive and physical load of interaction on human side.

In this thesis I have presented two human-aware robotic systems that enable a robot to perform hand-over tasks. The first system provides the robot with the capability to receive an object from a human. The system is able to detect the intention of the human to hand the object over and to start a reaching behavior in the robot that takes the object from the human hand. The intent to hand over the object is detected performing gesture recognition. The robot behaves actively until the end of the interaction reaching out up to the contact with the object. This is achieved using a reactive controller and a perception system that tracks the position of the object that the human is holding. This behavior is suitable in situation where the human is busy and can not focus on the hand-over, hence he needs help from the robot to accomplish the task. In a user study the system has proven to work properly completing most of the handovers attempted. Furthermore, observing the interaction and analyzing a survey questionnaire filled out by subjects, I found that four features of the robot behavior are critical during the interaction: forcefulness, aggressiveness, predictability, and timing. By tuning these features the robot can perform a behavior which is accepted by humans and perceived as safe and comfortable. These properties, which have been explained in section 2.2, are intuitively critical for many tasks that require physical interaction. For example a care-giver robot that has to lift a patient in order to put him on a wheelchair should properly tune its force when it touches the human body (forcefulness), should not approach the patient too fast (aggressiveness), should not perform strange trajectories (predictability) and should take a reasonable time to complete the task (timing).

The second system described in this thesis enables a robot to deliver objects to humans in a way which is comfortable for humans. Comfort is taken into account by delivering the object so that the most suitable of its constituent parts (e.g. a handle) is oriented towards the user. This idea is supported by the observation that most ordinary objects (such as tools) have grasp affordances, i.e. preferential and comfortable ways of grasping an object to achieve a particular function. A user study has confirmed that users feel that the robot delivers the objects in a comfortable way that complies with their expectations. In addition, using this approach the reaction time of the person that receives the object is shorter and the handover is completed faster. In order to determine when the robot has to deliver the object and which object has to be delivered the robot is provided with voice recognition and text-to-speech systems. These systems enable natural and effective interaction.

Although these robotic systems have achieved good results during user studies, they have some limitations. First, they have been tested through laboratory studies. Second, they address specific aspects of human-robot handovers. The first system focuses on the detection of the human intention to hand over an object and the negotiation of the hand over location, but it does not address how the robot should approach the human when they are apart before the interaction or how the robot can perform a stable grasp during the transfer of the object. The second system focuses on how, when and where the object should be offered to the human and how the robot can detect the human readiness to start the transfer. However, it does not address which arm trajectory the robot should perform.

In section 3.2 I proposed a holistic representation of handovers. The handover structure proposed describes handovers at physical and social/cognitive levels. Physical coordination involves actions such as approaching/carrying, reaching, and transferring of control that enable object handover. Social-cognitive coordination includes activities that establish agreement on when/timing and where/location of handovers between two persons. Effective and seamless human-robot handovers can be obtained by addressing all these aspects. Future work will implement and evaluate the entire seamless human-robot handover structure. Robots are close to effectively and seamlessly participate in human-robot handovers. Continued research in human-robot interaction and physical human-robot interaction will lead to personal robots that are able to effectively interact with people in a way accepted by humans.

# Appendix A

# Tools for Gesture Recognition

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the goal of conveying information, either in place of speech or together and in parallel with spoken words. They are an interesting subspace of the possible human motions. Generally, there are many-to-one mappings from concepts to gestures and vice versa. Gestures vary among individuals, and even for the same individual among different situations. Moreover, gestures are often language and culture specific. For these reasons, gestures can be *ambiguous* and *incompletely specified*. Nevertheless, in several cases a gesture can be successfully used to extract useful information or to realize the intention of the subject who is performing it.
Typically, the meaning of a gesture can be dependent on several factors:

- spatial information: where it occurs;
- path information: the path it takes;
- symbolic information: the sign it makes;
- context information: the context in which it occurs.

Gestures can be static (the user assumes and maintains a certain pose or configuration) or dynamic (with prestroke, stroke, and poststroke phases). Moreover, some gestures have both static and dynamic elements, as in sign languages. Gestures can be classified in the following types:

- *hand and arm gestures*: recognition of hand poses and hand trajectories, often used for sign languages and entertainment applications;

- *head and face gestures*: some examples are direction of eye gaze, raising the eyebrows, opening the mouth to speak, winking, looks of surprise, happiness, disgust, fear, anger, sadness, contempt, etc.; this information is used to detect emotions or perform gaze-following.

- *body gestures*: involvement of full body motion, as in: tracking movements of two people interacting outdoor, analyzing movements of a dancer or recognizing human gaits for medical rehabilitation and athletic training. This information is often used in medical, human-robot-interaction or entertainment applications.

In this chapter, the main sensors and the main techniques used to perform gesture recognition are reported.

## A.1 Devices and sensors for gesture recognition

In order to detect gestures the human body position, configuration (angles and rotations), and movement (velocities) need to be sensed. A large set of devices ranging from instrumented data gloves and body suits to RGB and depth cameras are used for gesture recognition. Each sensing technology varies along several dimensions, including accuracy, resolution, latency, range of motion, user comfort and cost. During my Phd I conducted research activities to design and develop gesture recognition systems suitable for interaction with 3D environments. I developed and tested prototypes based on infrared and RGB cameras [107], touchsreens [108] and accelerometers [109].

This paragraph reports an overview of the main sensors used to perform gesture recognition.

(a) Apple iPad.

(b) Microsoft Surface.

Figure A.1: Commercial devices that make use of touchscreen interfaces. Figure (a) shows the iPad, a successful tablet from Apple. Figure (b) shows Microsoft Surface, a touchscreen table.

### A.1.1 Touchscreens and Multi-Touch

Touchscreens are devices that enable the user to interact with a computer simply by touching the screen. Touchscreens are able to detect a single contact point and track the movement of this point on the screen to recognize gestures, whereas multi-touch devices can sense several touches. Multi-touch capable displays are one of the central emerging technologies in Human Computer Interfaces and many commercial applications like the Apple iPad or the Microsoft Surface (Fig. A.1) already take advantage from the benefit of this interaction technique.

When a generic user touches a multi-touch device, the contact generates an event. Several touches generate a continuous stream of events. The amount of concurrent events in these interfaces is limited only by the data type holding the number of fingers inputs. Since they are not the kind of "on/off" inputs we are used to in traditional interfaces, there are needs for new ways to interpret and analyze the inputs type and the gestures they make out. Different approaches for multi-touch sensing have been published. The Diamondtouch [110] uses capacitive sensing. In contrast to most optical systems it allows to distinguish different users. However, contact with a special mat is required. Other approaches are based on optical touch sensing. These systems can be used with rear-projected images and allow shadow free interaction. The reactable [111], for example, illuminates the display with infrared light from be-

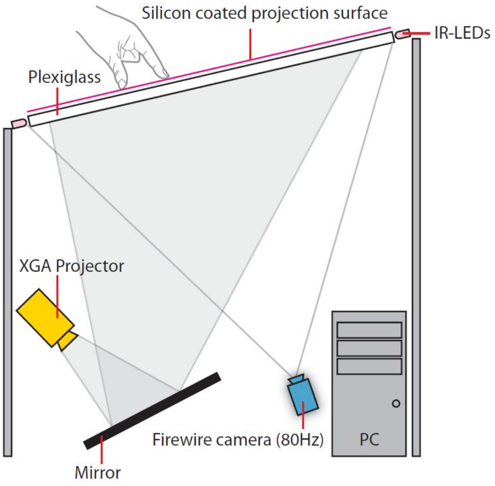


Figure A.2: The Hardware Architecture of *The DabR*

low the surface. An advantage besides the high spatial resolution obtained by using a camera is that also objects can be used for tangible interaction. But problems with detected objects slightly above the surface can lead to confusion during interaction. The multi-touch system “Surface” from Microsoft uses a similar technique to sense fingers and objects. A different promising technique for optical multi-touch sensing was presented by Han [112]. A plexiglas surface is illuminated from the sides with infrared light. Touching the surface leads to frustrated total internal reflection (FTIR): the infrared light escapes and gets captured by a infrared sensitive camera below the display. Hence, only clear contacts and nothing above the screen is detected. In [113] Edelmann at al. use a similar architecture (Fig A.2 ) to sense touches on a screen in order to detect easy and intuitive multi-touch gestures to control a virtual 3D camera.

### A.1.2 Accelerometers

Typically accelerometer sensors measure the acceleration of an object along 3 axes. Accelerometers have been extensively used in the last years for gesture recognition. With the rapid development of the MEMS (Micro Electrical Mechanical System)

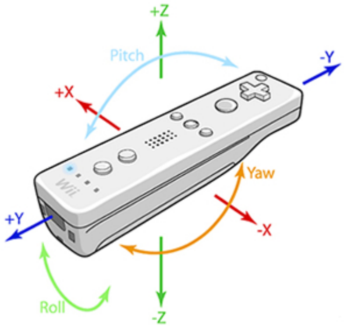


Figure A.3: Wiimote - An ADXL330 accelerometer provides acceleration values along 3 axes.

technology, people can wear/carry one or more accelerometer-equipped devices in daily life such as the Apple iPhone and the Nintendo Wiimote. These wireless-enabled mobile/wearable devices provide new possibilities for interacting with a wide range of applications, such as home appliances, mixed reality, etc. Since this sensor provides information about the motion, it is suitable to detect dynamic gestures.
To recognize a gesture from the captured data, researchers have applied diverse machine learning and pattern recognition techniques. The main algorithms in the literature are Dynamic Time Warping (DTW)-based approaches [114, 115, 116] and Hidden Markov Model (HMM)-based approaches [117, 118, 119, 120, 121]. Both sets of algorithms process the acceleration data in the time domain.
The Nintendo Wiimote (Fig A.3), one of the most successful controllers of the last decade, uses an ADXL330 accelerometer to perform gesture recognition. The Wiimote is a controller developed by Nintendo for the company's home video game console, Wii. In [117] a Wii controller is used to recognize 3D gestures.

### A.1.3 Cameras

Images are usually meaningful but it's hard to extract information from them. Camera-based systems have to cope with several problems such as the occlusion of parts of the user's body, changes in shape and size of the gesture-generating object (that varies between individuals), other moving objects in the background, light variations, and noise. Vision-based techniques can also vary among themselves in: 1) the number of cameras used; 2) their speed and latency; 3) the structure of environment (restrictions such as lighting or speed of movement); 4) any user requirements (whether the user must wear anything special); 5) the low-level features used (edges, regions, silhouettes, moments, histograms); 6) whether 2-D or 3-D representation is used; and 7) whether time is represented.

Vision based gesture recognition is a multi-steps process that consists of:

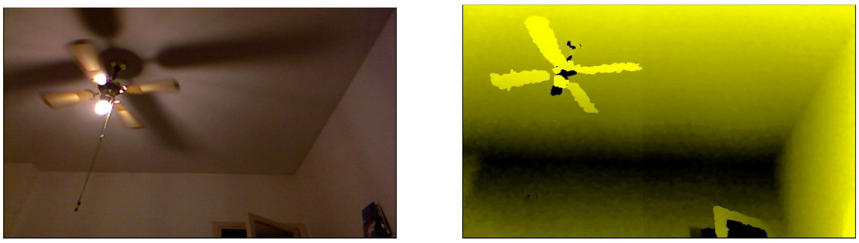
- $Pre-processing$: Usually, before an image can be exploited for gesture recognition, it is necessary to apply some filters or other transformations. This step adjusts the image so that it satisfies certain assumptions implied by the methods that are being used in the following steps.

- $Segmentation$: This phase consists of extracting from an image, points or regions which are relevant for further processing.

- $Feature\ extraction$: In this step, the goal is to extract features like contours, fingertips, etc. These features are then used as characteristic patterns for the recognition engine.

- $Model/Classifier$: This is the place of the recognition procedure. Often modeling tools like Hidden Markov Models are used for gesture training and recognition. Finally, a classification algorithm is used.

The great advantage of vision based gesture recognition is the freedom that is given to the person whose gestures are recognized. Cameras have been used extensively for gesture recognition. In [122] Wang at al. present a system that performs real time hand tracking, using a camera and a color glove. The system is able to reconstruct the pose of the hand from a single image of the hand wearing a multi-colored glove. In [123] Salti at al. present an approach for 3D arm pose estimation from a monocular video. Their proposal has been designed to provide real-time and realistic reconstruction of the user motion, as required by advanced Human Computer Interaction (HCI) applications. Both a 2D arm tracking and a 3D arm pose estimation



(a) RGB Image.

(b) Depth Image.

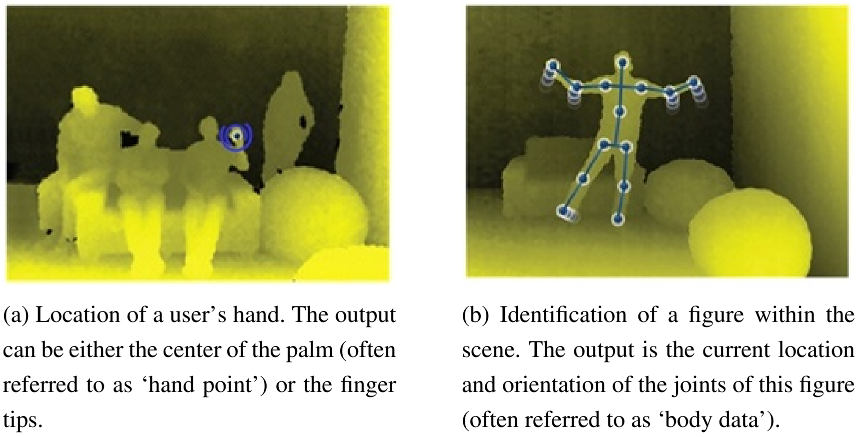
Figure A.4: RGB-D camera images.

algorithm are introduced and discussed.

Recently the availability of affordable RGB-Depth cameras, with real-time synchronized color and dense depth, has dramatically improved and fundamentally changed system capabilities to perceive and interact with people and environments. The last years have seen notable improvements in the capabilities of RGB-Depth cameras (RGB-D), which provide per pixel RGB and Depth information. This sensors enable a low-cost reconstruction of 3D images and image processing at high frame rate. The most famous RGB-D camera is the Microsoft Kinect. This camera provides synchronized color and depth images with a resolution of 640x480 pixels (Fig A.4). Using this data reliable gesture recognition systems can be developed. An example is given by the OpenNI framework [124] that provides tools for reliable gesture recognition and people detection using an RGB-D camera (Fig A.5).

### A.1.4 Data Gloves: Tracking sensors and flex sensors

Tracking sensors can reliably track the movements of a user enabling gesture recognition. These sensors provide information about the position and the orientation of an object with respect to a fixed frame of reference. These sensors can be used to build instrumented data gloves. Often a motion tracker, such as a magnetic tracking device or inertial tracking device, is attached to capture the global position/rotation data of the glove. These movements are then interpreted by the software that accompanies the glove, so each movement can mean any number of things. In addition, modern



(a) Location of a user's hand. The output can be either the center of the palm (often referred to as 'hand point') or the finger tips.

(b) Identification of a figure within the scene. The output is the current location and orientation of the joints of this figure (often referred to as 'body data').

Figure A.5: Human body tracking with RGB-D Camera and OpenNI Framework.

data gloves use flex sensors to detect how the fingers are bent. An example is the CyberGlove (Fig A.6). There are different versions of the CyberGlove which can have up to 22 flex sensors and a motion tracker in the data glove wristband. Flex sensors are thin, flexible, and sewn into the lightweight elastic glove fabric. As each strip is bent, its electrical resistance changes; this datum is used to calculate the angle of that joint. In literature there are many works on dataglove-based gesture recognition. In [125] Saggio at al. use The HiTEg Glove V4, developed by the Health Involved Technical Engineering Group (HiTEg) at the University of Rome "Tor Vergata" [126], to perform the classification of 20 different gestures, evaluating three different methodologies: Support Vector Machines, Mahalanobis and Euclidean based classifiers. In [127] Lu at al. describe the design and evaluation of a new calibration protocol for motion-capture gloves, which is designed to make the process more efficient and to be accessible for participants who are deaf and use American Sign Language (ASL).

## A.2 Techniques for gesture recognition

Gesture recognition is an ideal example of multidisciplinary research area. There are different tools for gesture recognition, based on approaches ranging from statistical

Figure A.6: CyberGlove motion capture data glove.

modeling, computer vision and pattern recognition, connectionist systems, etc.
Most of the problems have been addressed based on statistical modeling, such as Principal Component Analysis (PCA), Hidden Markov Models (HMM) [128, 129, 130], Kalman filtering [131], more advanced particle filtering [132, 133] and condensation algorithms [134, 135, 136]. Finite State Machines (FSM) have been effectively employed in modeling human gestures in several works [137, 138, 139, 140]. Connectionist approaches [141], involving multilayer perceptrons (MLP) [29, 142], timedelay neural networks (TDNN) [143], and radial basis function networks (RBFN) [29, 144], have been utilized in gesture recognition as well. While static gesture (pose) recognition can be often accomplished by template matching, standard pattern recognition, and neural networks, the dynamic gesture recognition problem involves the use of techniques such as time-compressing templates, dynamic time warping, HMMs, and TDNN. This paragraph describes some of these popular approaches used in gesture recognition. For a more comprehensive review refer to [145].

### A.2.1 Hidden Markov Model

HMMs are usually used to analyze or to predict time series. A Hidden Markov Model [128, 129] (Fig. A.7) is a Markov chain with a finite number of states which are not directly observable. A Markov chain is a Bayes Network represented by a sequence of states that evolve over time, and each state depends only on the previous state in the network. This specific kind of “memorylessness” is called the Markov property.

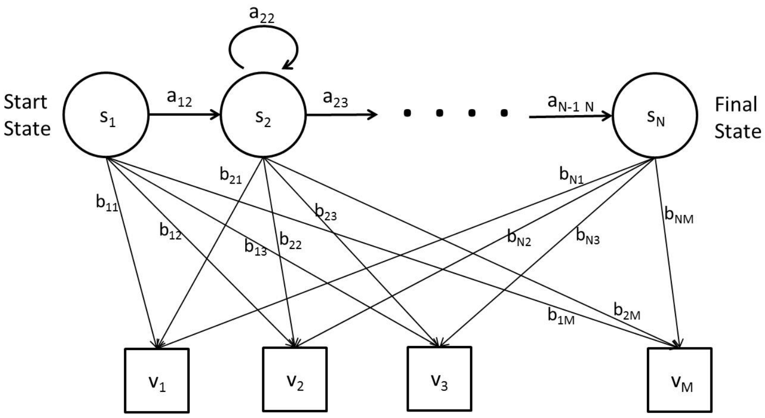


Figure A.7: A discrete HMM network.

This is a useful assumption to make, when considering the positions and orientation of the hands of a gesturer through time. What distinguishes a HMM from a Markov model is the fact that each state also emits a measurement, so rather than being able to observe the state itself, what you get to see are measurements (observations). Each state is characterized by two sets of probabilities: a transition probability, and either a discrete output probability distribution or a continuous output probability density function. The output probability distribution and the output probability function, given the state, define the condition probability of emitting each output symbol from a finite alphabet or a continuous random vector. In the case of gesture recognition discrete HMMs are employed.

A HMM is expressed as $\lambda$=$(A,B,\Pi)$ and is defined as follows:

- a set of $N$ states $\{s_1,...,s_N\}$;
- a set of $M$ discrete observation symbols $\{v_1,...,v_M\}$;
- a set of observation strings $O = \{O_1,...,O_T\}$, where $t = 1,...T$ and $O_t \in \{v_1,...,v_M\}$ ;
- a state transition matrix $A = \{a_{ij}\}$, where $a_{ij}$ is the transition probability from

state $s_i$ at time $t$, to state $s_j$ at time $t + 1$

$$A = \{a_{ij}\} = Prob(s_j \ at \ t+1|s_i \ at \ t), \qquad for \ 1 \leq i, j \leq N$$

- an observation symbol probability matrix $B = \{b_{jk}\}$, where $b_{jk}$ is the probability of generating symbol $v_k$ from state $s_j$

$$B = \{b_{jk}\} = Prob(v_k|s_j), \qquad for \ 1 \leq j \leq N \ and \ 1 \leq k \leq M$$

- an initial probability distribution for the states

$$\Pi = \{\pi_j\}, j = 1, 2, ..., N \ where \ \pi_j = Prob(s_j \ at \ t = 1)$$

For a discrete HMM, $a_{ij}$ and $b_{jk}$ have the following properties:

$$a_{ij} \geq 0, \ b_{jk} \geq 0, \quad \forall i, j, k,$$

$$\sum_j a_{ij} = 1 \quad \forall i,$$

$$\sum_k b_{jk} = 1 \quad \forall j.$$

The key idea of HMM-based gesture recognition is to use multi-dimensional HMM representing the defined gestures. The parameters of the model are determined by the training data. The trained models represent the most likely human performance and are used to evaluate new incoming gestures. The HMM-based gesture recognition approach can be described as follows:

1. *Describe each gesture in terms of a HMM* - A multi-dimensional HMM is employed to model each gesture. As shown in Fig A.8, the global structure of the HMM is constructed by parallel connections of each HMM $(\lambda_1, \lambda_2, ..., \lambda_N)$, whereby insertion (or deletion) of a new (or existing) HMM is easily accomplished. Here $\lambda$ corresponds to a constructed HMM model for each gesture, where N is the total number of gestures being recognized. Note that only the structures of A and B are determined in this step and the values of elements in A and B will be estimated in the training process

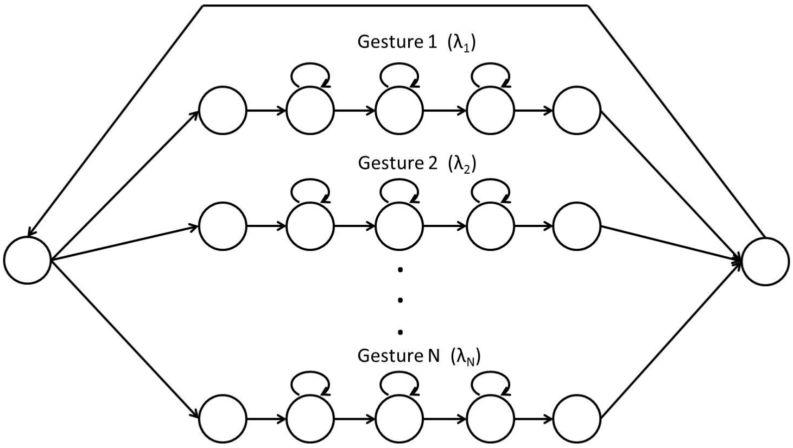


Figure A.8: N parallel HMMs connected together to recognize N gestures.

2. *Collect training data* - In the HMM-based approach, gestures are specified through the training data. It is essential that the training data are represented in a concise and invariant form. Raw input data are preprocessed before they are used to train the HMMs.

3. *Train the HMMs through training data* - Training is one of the most important procedures in a HMM-based approach. The model parameters are adjusted in such a way that they can maximize the likelihood $P(O|\lambda)$ for the given training data. No analytic solution to the problem has been found so far. However, the Baum-Welch algorithm [146] can be used to iteratively reestimate model parameters to achieve the local maximum.

4. *Evaluate gestures with the trained model* - The trained model can be used to classify the incoming gestures. The Forward-Backward algorithm or the Viterbi algorithm [128, 147] can be used to classify isolated gestures. The Viterbi algorithm can also be used to decode continuous gestures.

The goal in a recognition process is to retrieve the input gestures which are represented by a sequence O. The process is to find the HMM with the highest probability given a sequence, i.e.,

$$g = \arg\max_{all\ \lambda} P(\lambda|O)$$

The Forward-Backward algorithm is able to evaluate the probability of the observation sequence generated by a HMM, i.e., $P(O|\lambda)$. However, the problem of recognition is to compute $P(\lambda|O)$. From the Bayes formula, a posteriori probability given the sequence can be written as

$$P(\lambda|O) = \frac{P(O|\lambda)P(\lambda)}{P(O)}$$

Because P(O) is a constant for a given input, only $P(O|\lambda)P(\lambda)$ needs to be computed. $P(\lambda)$ is the probability of the gesture being used, i.e., it characterizes the likely sequence of gestures in certain rules. For example, if the gestures are used for sign language, $P(\lambda)$ can then be determined through a language model. In the simplest case, all the gestures are equally likely to be used, then only the term $P(O|\lambda)$ is a variable.

### A.2.2 Finite State Machine

A finite state machine (FSM) or finite state automaton, is a mathematical model of computation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. The machine is in only one state at a time. The state it is in at any given time is called the current state. When a triggering event or condition occurs, the machine can change from one state to another. This is called a transition. Formally, a finite state machine is defined as a 5-tuple $(S, I, f, S_0, F)$ where,

- $S$ is a finite set of states.
- $I$ is a finite set of symbols or the alphabet.
- $f : S \times I \rightarrow S$ is the transition function.
- $S_0$ is an element of $S$ called the start state, and

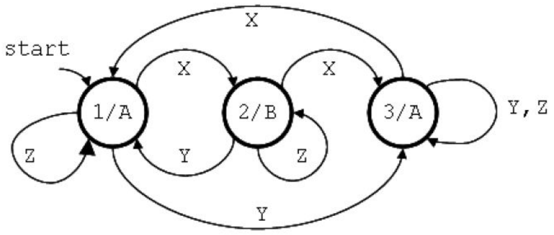


Figure A.9: An Example Moore Machine.

- $F$ is a subset of $S$ called the set of accept states.

The Moore and Mealy machines are extensions of the FSM that add an output alphabet and a function to generate the output.
A Moore machine is a 6-tuple $(S, I, O, f, S_0, g)$ where,

- $S$ is a finite set of states.

- $I$ is a finite set of symbols called the input alphabet.

- $O$ is a finite set of symbols called the output alphabet.

- $f : S \times I \rightarrow S$ is the transition function.

- $S_0$ is an element of $S$ called the start state, and

- $g : S \rightarrow O$ is the output function mapping the current state to the output.

In the Moore machine, the set of accept states, F, has been replaced with an output function giving the symbols to be generated. In a Moore machine, an output symbol is generated each time a state is entered. This output symbol does not depend on how the state was entered, which means that the output is strictly a function of the state being entered and not of the input symbol being read. Figure A.9 shows a diagram of a simple Moore machine. Each state label includes the state number followed by the output symbol to be generated when this state is entered. Each of the six elements of this example Moore machine are given below,

- $S = 1,2,3$.
- $I = i$, which can have values X,Y,Z.
- $O = o$, which can have values A,B.
- $$\begin{aligned} f = (&[1,X] \rightarrow 2, [1,Y] \rightarrow 3, [1,Z] \rightarrow 1,\\ &[2,X] \rightarrow 3, [2,Y] \rightarrow 1, [2,Z] \rightarrow 2,\\ &[3,X] \rightarrow 1, [3,Y] \rightarrow 3, [3,Z] \rightarrow 3). \end{aligned}$$
- $S_0 = 1$
- $g = (1 \rightarrow \text{A}, 2 \rightarrow \text{B}, 3 \rightarrow \text{A})$.

Sometimes it would be useful to generate a different output depending upon the input symbol being read and the state from which the transition is occurring. The Mealy machine offers this capability. A Mealy machine is a 6-tuple $(S, I, O, f, S_0, h)$ where,

- $S$ is a finite set of states.
- $I$ is a finite set of symbols called the input alphabet.
- $O$ is a finite set of symbols called the output alphabet.
- $f : S \times I \rightarrow S$ is the transition function.
- $S_0$ is an element of $S$ called the start state, and
- $h : S \times I \rightarrow O$ is the output function mapping the current transition to the output.

The Mealy machine is the same as the Moore machine except that the output function $g$ has been replaced with the output function $h$, which maps the Cartesian product of the set of states $S$ and the set of input symbols $I$ to the set of output symbols $O$. This means that the symbol being output depends on the transition rather than the state being entered. Figure A.10 shows a Mealy machine designed to produce the same output as the Moore machine shown in Figure A.9.

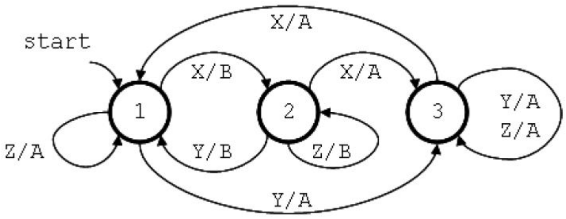


Figure A.10: An Example Mealy Machine.

In the FSM approach, a gesture can be modeled as an ordered sequence of states in a spatio-temporal configuration space [137, 138, 139, 140]. The number of states in the FSM may vary between applications. Generally a gesture is represented by a prototype trajectory defined as a set of points (e.g., sampled positions of the head, hand, and eyes) or as a set of motion properties (speed, direction, etc.).

The training of the model is done off-line, using many possible examples of each gesture as training data, and the parameters (criteria or characteristics) of each state in the FSM are derived. The recognition of gestures can be performed online using the trained FSM. When input data (feature vectors such as trajectories) are supplied to the gesture recognizer, the latter decides whether to stay at the current state of the FSM or jump to the next state based on the parameters of the input data. If it reaches a final state, we say that a gesture has been recognized.

The state-based representation can be extended to accommodate multiple models for the representation of different gestures, or even different phases of the same gesture. Membership in a state is determined by how well the state models can represent the current observation. If more than one model (gesture recognizer) reach their final states at the same time, we can apply a winning criteria to choose the most probable gesture.

### A.2.3 Dynamic Time Warping

Dynamic Time Warping (DTW) is one of the most popular algorithms used for gesture recognition [116, 148, 149, 150]. DTW is an algorithm for measuring similarity

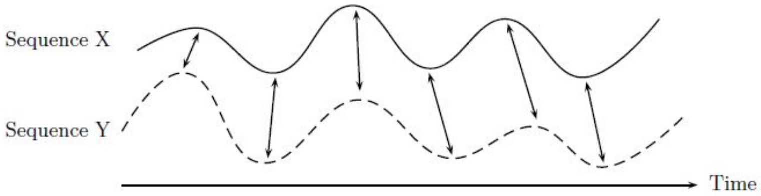


Figure A.11: Time alignment of two time-dependent sequences. Aligned points are indicated by the arrows.

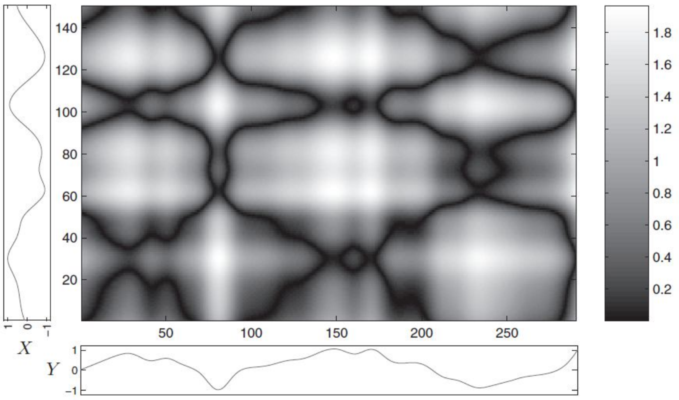


Figure A.12: Cost matrix of the two real-valued sequences X (vertical axis) and Y (horizontal axis) using the Manhattan distance (absolute value of the difference) as local cost measure c. Regions of low cost are indicated by dark colors and regions of high cost are indicated by light colors.

between two sequences which may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another he or she were walking more quickly, or even if there were accelerations and decelerations during the course of one observation.

The goal of the DTW algorithm is to match a given sequence of sensor values to a stored prototype. The stored prototype is collected earlier during a special training procedure. Training sequences are combined into prototype sequences for each type of gesture. During the recognition phase sensor readings from the prototype and input sequence are then compared using a distance function. To account for amplitude differences, the sequence matching algorithm tries matching several versions of the prototype with differently scaled amplitudes. The sequence is classified as the gesture with the highest score.

The objective of DTW is to compare two (time-dependent) sequences $X := (x_1, x_2, ..., x_N)$ of length $N \in \mathbb{N}$ and $Y := (y_1, y_2, ..., y_M)$ of length $M \in \mathbb{N}$. These sequences may be discrete signals (time-series) or, more generally, feature sequences sampled at equidistant points in time. In the following, we fix a $feature\ space$ denoted by $\mathscr{F}$. Then $x_n, y_m \in \mathscr{F}$ for $n \in [1 : N]$ and $m \in [1 : M]$. To compare two different features $x, y \in \mathscr{F}$ a *local cost measure* (sometimes also referred to as $local\ distance\ measure$) is needed. The local cost measure is defined as a function:

$$c : \mathscr{F} \times \mathscr{F} \to \mathbb{R} \geq 0$$

Typically, $c(x,y)$ is small (low cost) if $x$ and $y$ are similar to each other, otherwise $c(x,y)$ is large (high cost). Evaluating the local cost measure for each pair of elements of the sequences $X$ and $Y$, it is possible to obtain the cost matrix $C \in \mathbb{R}^{N \times M}$ defined by $C(n,m) := c(x_n, y_m)$ (see Fig. A.12).

Then the goal is to find an alignment between X and Y having minimal overall cost. Intuitively, such an optimal alignment runs along a "valley" of low cost within the cost matrix C (Fig. 4.4) called *warping path*. An $(N,M)$-warping path (or simply referred to as warping path if $N$ and $M$ are clear from the context) is a sequence $p = (p_1, ..., p_L)$ with $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ for $l \in [1 : L]$ satisfying the following three conditions:

- $Boundary\ condition : p_1 = (1,1)\ and\ p_L = (N,M).$
- $Monotonicity\ condition : n_1 \leq n_2 \leq ... \leq n_L\ and\ m_1 \leq m_2 \leq ... \leq m_L.$
- $Step\ size\ condition$: $p_{l+1} - p_l \in \{(1,0),(0,1),(1,1)\}\ for\ l \in [1 : L-1]$.

Note that the step size condition implies the monotonicity condition, which nevertheless has been quoted explicitly for the sake of clarity. An $(N,M)$-warping path

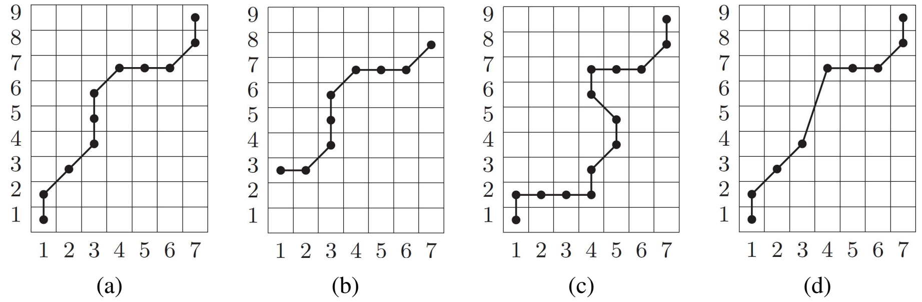
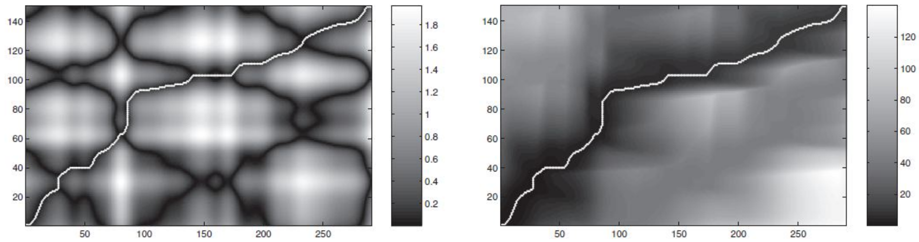


Figure A.13: Illustration of paths of index pairs for some sequence $X$ of length $N = 9$ and some sequence $Y$ of length $M = 7$. (a) Admissible warping path satisfying the three conditions (b) Boundary condition is violated. (c) Monotonicity condition is violated. (d) Step size condition is violated

$p = (p_1, ..., p_L)$ defines an alignment between two sequences $X = (x_1, x_2, ..., x_N)$ and $Y = (y_1, y_2, ..., y_M)$ by assigning the element $x_{n_l}$ of $X$ to the element $y_{m_l}$ of Y . The boundary condition enforces that the first elements of $X$ and $Y$ as well as the last elements of $X$ and $Y$ are aligned to each other. In other words, the alignment refers to the entire sequences $X$ and $Y$ . The monotonicity condition reflects the requirement of faithful timing: if an element in X precedes a second one this should also hold for the corresponding elements in Y , and vice versa. Finally, the step size condition expresses a kind of continuity condition: no element in $X$ and $Y$ can be omitted and there are no replications in the alignment (in the sense that all index pairs contained in a warping path $p$ are pairwise distinct). Fig A.13 illustrates the three conditions. The total cost $c_p(X,Y)$ of a warping path $p$ between $X$ and $Y$ with respect to the local cost measure c is defined as

$$c_p(X,Y) := \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}).$$

Furthermore, an *optimal warping path* between $X$ and $Y$ is a warping path $p^*$ having minimal total cost among all possible warping paths. The DTW *distance* $DTW(X,Y)$



(a) (b)

Figure A.14: (a) Cost matrix C as in Fig. 4.2 and (b) accumulated cost matrix D with optimal warping path $p^*$ (white line)

between $X$ and $Y$ is then defined as the total cost of $p^*$:

$$\begin{aligned} DTW(X,Y) &:= c_{p^*}(X,Y) \\ &= \min\{c_p(X,Y) \mid p\ is\ an\ (N,M)-warping\ path\} \end{aligned} \tag{A.1}$$

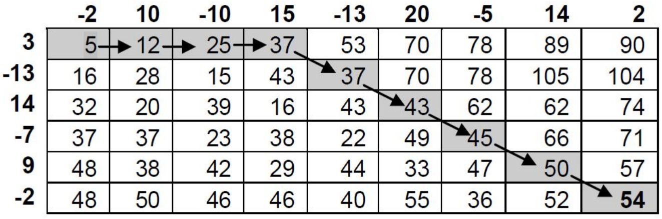
A way to determine an optimal path $p^*$, could be to test every possible warping path between $X$ and $Y$ . Such a procedure, however, would lead to a computational complexity that is exponential in the lengths N and M. Typically, *dynamic programming* algorithms are employed to reduce the computation time. Following is described a common algorithm used to figure out an optimal warping path.

Given two feature sequences $X$ and $Y$, it is possible to define the prefix sequences $X(1:n) := (x_1,...,x_n)$ for $n \in [1:N]$ and $Y(1:m) := (y_1,...,y_m)$ for $m \in [1:M]$ and set

$$D(n,m) := DTW(X(1:n),Y(1,m)).$$

The values $D(n,m)$ define an $N \times M$ matrix $D$, which is also referred to as the *accumulated cost matrix* . Obviously, D(N,M) = DTW(X, Y ). The first step of the algorithm is to compute the matrix $D$. It can be proved that the matrix D can be computed efficiently satisfying the following identities:

- $D(n,1) = \sum_{k=1}^{n} c(x_k,y_1)\ for\ n \in [1:N]$
- $D(1,m) = \sum_{k=1}^{m} c(x_1,y_k)\ for\ m \in [1:M]$



|  | -2 | 10 | -10 | 15 | -13 | 20 | -5 | 14 | 2 |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 5 | 12 | 25 | 37 | 53 | 70 | 78 | 89 | 90 |
| -13 | 16 | 28 | 15 | 43 | 37 | 70 | 78 | 105 | 104 |
| 14 | 32 | 20 | 39 | 16 | 43 | 43 | 62 | 62 | 74 |
| -7 | 37 | 37 | 23 | 38 | 22 | 49 | 45 | 66 | 71 |
| 9 | 48 | 38 | 42 | 29 | 44 | 33 | 47 | 50 | 57 |
| -2 | 48 | 50 | 46 | 46 | 40 | 55 | 36 | 52 | **54** |

Figure A.15: An example of accumulated cost matrix and optimal warping path (grey cells).

- $D(n,m) = \min\{D(n-1,m-1), D(n-1,m), D(n,m-1)\} + c(x_n, y_m)$ $for\ 1 < n \leq N\ and\ 1 < m \leq M$

In particular, $DTW(X,Y) = D(N,M)$ can be computed with $O(NM)$ operations. The matrix $D$ can be computed recursively. The initialization can be simplified by extending the matrix $D$ with an additional row and column and formally setting $D(n,0) := \infty$ for $n \in [1:N]$, $D(0,m) := \infty$ for $m \in [1:M]$, and $D(0,0) := 0$.
Once the entire $(N \times M)$-matrix D is available the optimal warping path $p^*$ is computed in reverse order of the indices starting with $p_L = (N,M)$. Suppose $p_l = (n,m)$ has been computed. In case $(n,m) = (1,1)$ the algorithm has reached the end, otherwise

$$p_{l-1} = \begin{cases} (1, m-1) & ifn = 1 \\ (n-1, 1) & ifm = 1 \\ \arg\min\{D(n-1,m-1), D(n-1,m), D(n,m-1)\} & \text{otherwise} \end{cases}$$

Figure A.14 shows the optimal warping path $p^*$ (white line) for the sequences of Fig. A.12. Note that $p^*$ covers only cells of $C$ that exhibit low costs. The resulting accumulated cost matrix $D$ is shown in Fig. A.14b. In figure A.15 a numeric example of accumulated cost matrix and optimal warping path for two short sequences is reported. For a more detailed description of DTWs see [151, 152, 153, 149].

# Bibliography

[1] S. Srinivasa, D. Berenson, M. Cakmak, A. Collet Romea, M. Dogar, A. Dragan, R. A. Knepper, T. D. Niemueller, K. Strabala, J. M. Vandeweghe, and J. Ziegler. Herb 2.0: Lessons learned from developing a mobile manipulator for the home. *Proceedings of the IEEE*, 100(8):1–19, July 2012.

[2] S. Srinivasa, D. Ferguson, J. M. Vandeweghe, R. Diankov, D. Berenson, C. Helfrich, and H. Strasdat. The robotic busboy: Steps towards developing a mobile robotic home assistant. In *International Conference on Intelligent Autonomous Systems*, July 2008.

[3] M. Dogar and S. Srinivasa. A framework for push-grasping in clutter. In Nick Roy Hugh Durrant-Whyte and Pieter Abbeel, editors, *Robotics: Science and Systems VII*. MIT Press, July 2011.

[4] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann. Integrated grasp and motion planning. In *International Conference on Robotics and Automation*, May 2010.

[5] K. Dautenhahn, C. L. Nehaniv, M. L. Walters, B. Robins, H. Kose-Bagci, N. A. Mirza, and M. Blow. Kaspar-a minimally expressive humanoid robot for human-robot interaction research. *Applied Bionics and Biomechanics*, 6:369–397, 2009.

[6] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao. Perception, planning, and execution for mobile manipulation in unstructured environments. *IEEE*

*Robotics and Automation Magazine, Special Issue on Mobile Manipulation*, 2012.

[7] S. Coradeschi, H. Ishiguro, M. Asada, S. C. Shapiro, M. Thielscher, C. Breazeal, M. J. Mataric, and H. Ishida. Human-inspired robots. *IEEE Intelligent Systems*, 21(4):74–85, July 2006.

[8] T. Mukai, S. Hirano, H. Nakashima, Y. Kato, Y. Sakaida, S. Guo, and S. Hosoe. Development of a nursing-care assistant robot riba that can lift a human in its arms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5996–6001. IEEE, 2010.

[9] T. W. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots: Concepts, design, and applications. Technical Report CMU-RI-TR-02-29, Robotics Institute, Pittsburgh, PA, December 2002.

[10] H. A. Yanco. Classifying human-robot interaction: An updated taxonomy. In *IEEE Transactions on Systems, Man, and Cybernetics*, pages 2841–2846, 2004.

[11] M. Cakmak, S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In IEEE, editor, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2011.

[12] T. Mukai, S. Hirano, M. Yoshida, H. Nakashima, S. Guo, and Y. Hayakawa. Whole-body contact manipulation using tactile information for the nursing-care assistant robot riba. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2445–2451. IEEE, 2011.

[13] C. Breazeal. Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, 59(1-2):119–155.

[14] T. L. Chen, C. King, A. L. Thomaz, and C. C. Kemp. Touched by a robot: an investigation of subjective responses to robot-initiated touch. In Aude Billard,

Peter H. Kahn Jr., Julie A. Adams, and J. Gregory Trafton, editors, *Proceedings of the 6th international conference on Human-robot interaction*, pages 457–464. ACM, 2011.

[15] A. Bicchi, M. Bavaro, G. Boccadamo, D. De Carli, R. Filippini, G. Grioli, M. Piccigallo, A. Rosi, R. Schiavi, S. Sen, and G. Tonietti. Physical human-robot interaction: Dependability, safety, and performance. In *10th IEEE International Workshop on Advanced Motion Control.*, pages 9 –14, March 2008.

[16] D. Kulic and E. A. Croft. Real-time safety for human-robot interaction. *Robotics and Autonomous Systems*, 54(1):1 – 12, 2006.

[17] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, September 2008.

[18] S. Sanan, M. H. Ornstein, and C. G. Atkeson. Physical human interaction for an inflatable manipulator. In *Engineering in Medicine and Biology Society, Annual International Conference of the IEEE*, pages 7401 –7404, sept 2011.

[19] J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C.W. Nielsen. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(4):438 – 449, July 2005.

[20] T. B. Sheridan. *Humans and Automation: System Design and Research Issues*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[21] M. R. Endsley, B. Bolte, and D. G. Jones. *Designing for Situation Awareness: An Approach to User-Centered Design*. Taylor and Francis, 2003.

[22] G. Klein, P. J. Feltovich, J. M. Bradshaw, and D. D. Woods. *Organizational Simulation*, chapter Common Ground and Coordination in Joint Activity, pages 139–184. John Wiley & Sons, Inc., 2005.

[23] M. Baker, B. Keyes, and H. A. Yanco. Improved interfaces for human-robot interaction in urban search and rescue. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, pages 2960–2965, 2004.

[24] M. W. Kadous, R. Sheh, and C. Sammut. Effective user interface design for rescue robotics. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, HRI '06, pages 250–257, New York, NY, USA, 2006. ACM.

[25] C. W. Nielsen and M. A. Goodrich. Comparing the usefulness of video and map information in navigation tasks. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, HRI '06, pages 95–101, New York, NY, USA, 2006. ACM.

[26] S. Waldherr, S. Thrun, and R. Romero. A gesture-based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.

[27] H. D. Yang, A.Y. Park, and S. W. Lee. Gesture spotting and recognition for human-robot interaction. *IEEE Transactions on Robotics*, 23(2):256–270, April 2007.

[28] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 708–713, August 2005.

[29] M. Sigalas, H. Baltzakis, and P. Trahanias. Gesture recognition based on arm tracking for human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5424 –5429, oct. 2010.

[30] M. Salem, K. Rohlfing, S. Kopp, and F. Joublin. A friendly gesture: Investigating the effect of multimodal robot behavior in human-robot interaction. In *20th IEEE International Symposium on Robot and Human Interactive Communication*, pages 247–252, August 2011.

[31] J. H. Hong, Y. S. Song, and S. B. Cho. A hierarchical bayesian network for mixed-initiative human-robot interaction. In *IEEE International Conference on Robotics and Automation*, pages 3808–3813, April 2005.

[32] D. Perzanowski, A. C. Schultz, W. Adams, and E. Marsh. Goal tracking in a natural language interface: towards achieving adjustable autonomy. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 208–213, 1999.

[33] B. Bramas, Y. M. Kim, and D. S. Kwon. Design of a sound system to increase emotional expression impact in human-robot interaction. In *International Conference on Control, Automation and Systems.*, pages 2732–2737, October 2008.

[34] R. J. Stone. Haptic feedback: A brief history from telepresence to virtual reality. In *Haptic Human-Computer Interaction*, pages 1–16, 2000.

[35] K. Lay, E. Prassler, R. Dillmann, G. Grunwald, M. Hägele, G. Lawitzky, A. Stopp, and W. Von Seelen. Morpha: Communication and interaction with intelligent, anthropomorphic robot assistants. In *International Status Conference Lead Projects Human-Computer Interaction*.

[36] T. Shibata, T. Tashima, and K. Tanie. Emergence of emotional behavior through physical interaction between human and robot. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 2868–2873, 1999.

[37] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka. Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *IEEE Transactions on Robotics*, 28:899–910, August 2012.

[38] D. Perzanowski, A. C. Schultz, W. Adams, E. Marsh, and M. Bugajska. Building a multimodal human-robot interface. *Intelligent Systems, IEEE*, 16(1):16–21, 2001.

[39] R. Stiefelhagen, H. K. Ekenel, C. Fugen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, and A. Waibel. Enabling multimodal human-robot interaction for the karlsruhe humanoid robot. *IEEE Transactions on Robotics*, 23(5):840–851, October 2007.

[40] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, 3rd edition, September 1999.

[41] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann. Using gesture and speech control for commanding a robot assistant. In *11th IEEE International Workshop on Robot and Human Interactive Communication.*, pages 454–459, 2002.

[42] T. Salter, K. Dautenhahn, and R. Boekhorst. Learning about natural human-robot interaction styles. *Robotics and Autonomous Systems*, 54(2):127–134, February 2006.

[43] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2422–2427, 2004.

[44] M. Cakmak and A. L. Thomaz. Designing robot learners that ask good questions. In *Human Robot Interaction (HRI)*, March 2012.

[45] M. Lawitzky, A. Mortl, and S. Hirche. Load sharing in human-robot cooperative manipulation. In *International Symposium on Robot and Human Interactive Communication*, September 2010.

[46] S. Kajikawa and E. Ishikawa. Trajectory planning for hand-over between human and robot. In *9th IEEE International Workshop on Robot and Human Interactive Communication.*, pages 281–287. IEEE, 2000.

[47] S. Glasauer, M. Huber, P. Basili, A. Knoll, and T. Brandt. Interacting in time and space: Investigating human-human and human-robot joint action. In *19th*

*IEEE International Symposium on Robot and Human Interactive Communication*, pages 252–257. IEEE, September 2010.

[48] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Prentice Hall, 3rd edition, 2005.

[49] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985.

[50] P. Basili, M. Huber, T. Brandt, S. Hirche, and S. Glasauer. Investigating Human-Human approach and Hand-Over. In Helge Ritter, Gerhard Sagerer, RÃijdiger Dillmann, and Martin Buss, editors, *Human Centered Robot Systems*, volume 6, pages 151–160. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[51] S. Shibata, K. Tanaka, and A. Shimizu. Experimental analysis of handing over. In *4th IEEE International Workshop on Robot and Human Communication.*, 1995.

[52] Kheng Lee Koay, E. A. Sisbot, D. S. Syrdal, M. L. Walters, K. Dautenhahn, and R. Alami. Exploratory study of a robot approaching a person in the context of handing over an object. AAAI Press, 2007.

[53] M. Jindai, S. Shibata, T. Yamamoto, and A. Shimizu. A study of Robot-Human system with consideration of individual preferences. *JSME International Journal. Series C Mechanical Systems, Machine Elements and Manufacturing*, 46(3):1075–1083, 2003.

[54] A. Agah and K. Tanie. Human interaction with a service robot: mobile-manipulator handing over an object to a human. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 575–580. IEEE, April 1997.

[55] J. Mainprice, E. A. Sisbot, T. Simeon, and R. Alami. Planning Safe and Legible Hand-over Motions for Human-Robot Interaction. In *IARP Workshop on Technical Challenges for Dependable Robots in Human Environments*, 2010.

[56] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortes, R. Alami, and T. Simeon. Planning human-aware motions using a sampling-based costmap planner. In *IEEE International Conference on Robotics and Automation*, pages 5012–5017, May 2011.

[57] E. Lopez-damian, D. Sidobre, S. De La Tour, and R. Alami. Grasp planning for interactive object manipulation. In *Proceedings of the International Symposium on Robotics and Automation.*

[58] J. Kim, J. Park, Y. Hwang, and M. Lee. Advanced grasp planning for handover operation between human and robot: Three handover methods in esteem etiquettes using dual arms and hands of Home-Service robot. In *2nd International Conference on Autonomous Robots and Agents*, pages 34–39, 2004.

[59] K. Nagata, Y. Oosaki, M. Kakikura, and H. Tsukune. Delivery by hand between human and robot based on fingertip force-torque information. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 750 –757 vol.2, October 1998.

[60] A. Edsinger and C. C. Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *The 16th IEEE International Symposium on Robot and Human interactive Communication.*, 2007.

[61] A. K. Pandey, M. Ali, M. Warnier, and R. Alami. Towards multi-state visuo-spatial reasoning based proactive human-robot interaction. In *International Conference on Advanced Robotics*, pages 143 –149, June 2011.

[62] Hyeg Joo Choi and Leonard S. M. Scaling affordances for human reach actions. *Human Movement Science*, 23(6):785–806, 2004.

[63] E. A. Sisbot and R. Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(99):1045–1057, 2012.

[64] E. Guizzo and E. Ackerman. The rise of the robot worker. *Spectrum, IEEE*, 49(10):34 –41, October 2012.

[65] V. Micelli, K. W. Strabala, and S. Srinivasa. Perception and control challenges for effective human-robot handoffs. In *Robotics: Science and Systems. Workshop on RGB-D Cameras*, 2011.

[66] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet Romea, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. Vandeweghe. HERB: a home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, January 2010.

[67] A. Collet Romea, M. Martinez, and S. S. Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *International Journal of Robotics Research*, 2011.

[68] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *IEEE International Conference on Robotics and Automation. Workshop on Open Source Software*, 2009.

[69] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[70] O. Chapelle, P. Haffner, and V.N. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5), 1999.

[71] M. J. Swain and D. H. Ballard. Indexing via color histograms. In *Third International Conference on Computer Vision.*, 1995.

[72] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard. NARF: 3D range image features for object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics.*, Taipei, Taiwan, 2010.

[73] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, May 2010.

[74] D. Berenson, S. Srinivasa, and J. Kuffner. Task Space Regions: A framework for pose-constrained manipulation planning. *International Journal of Robotics Research*, 2011.

[75] O. Khatib. The operational space formulation in the analysis, design, and control of manipulators. In *Proceedings of the International Symposium on Robotics Research*, December 1986.

[76] K. Strabala, Min Kyung Lee, A. Dragan, J. Forlizzi, and S. Srinivasa. Learning the communication of intent prior to physical collaboration. In *Robot and Human Interactive Communication*, 2012.

[77] R. Bischoff and V. Graefe. Design principles for dependable robotics assistants. *International Journal of Humanoid Robotics*, 1(1), 2005.

[78] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, October 2007.

[79] E. A. Sisbot, L. F. Marin, and R. Alami. Spatial reasoning for human robot interaction. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems*, pages 2281–2287, November 2007.

[80] J. Aleotti, V. Micelli, and S. Caselli. Comfortable Robot to Human Object Hand-Over. In *Robot and Human Interactive Communication*, September 2012.

[81] J. Aleotti and S. Caselli. A 3D Shape Segmentation Approach for Robot Grasping by Parts. *Robotics and Autonomous Systems*, 60(3):358–366, 2012.

[82] J. Aleotti, D. Lodi Rizzini, and S. Caselli. Object Categorization and Grasping by Parts from Range Scan Data. In *IEEE International Conference on Robotics and Automation*, St. Paul, USA, 2012.

[83] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (pcl). In *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.

[84] N. Amenta, S. Choi, and R. K. Kolluri. The Power Crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266, 2001.

[85] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson Surface Reconstruction. In *Eurographics Symposium on Geometry Processing*, 2006.

[86] S. Berretti, A. Del Bimbo, and P. Pala. 3D Mesh decomposition using Reeb graphs. *Image and Vision Computing*, 27(10):1540–1554, 2009.

[87] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner. Grasp planning in complex scenes. In *7th IEEE-RAS International Conference on Humanoid Robots*, pages 42–48, Nov. 2007.

[88] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3217, May 2009.

[89] J. Aleotti and S. Caselli. Part-based robot grasp planning from human demonstration. In *IEEE International Conference on Robotics and Automation*, pages 4554–4560, Shanghai, China, 2011.

[90] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S.S. Srinivasa, M. Cakmak, and V. Micelli. Toward seamless human-robot handovers. *In Journal of Human-Robot Interaction, Special Issue: HRI System Studies*, 2012.

[91] Min Kyung Lee, J. Forlizzi, S. Kiesler, M. Cakmak, and S. Srinivasa. Predictability or adaptivity?: Designing robot handoffs modeled from trained dogs and people. In *Human-Robot Interaction*, 2011.

[92] H. H. Clark. *Using Language*. Cambridge University Press, 1996.

[93] Erving Goffman. *The presentation of self in everyday life*. Doubleday, Garden City, N.Y., 1959.

[94] M. Huber, H. Radrich, C. Wendt, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. Evaluation of a novel biologically inspired trajectory generator in human-robot interaction. In *Robot and Human Interactive Communication*, pages 639–644. IEEE, October 2009.

[95] W. P. Chan and C. A. C. Parker. Grip forces and load forces in handovers: implications for designing human-robot handover controllers. *International Conference on Human-Robot Interaction*, pages 9–16, 2012.

[96] U. Castiello. Understanding other people's actions: Intention and attention. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):416–430, 2003.

[97] N. Sebanz, H. Bekkering, and G. Knoblich. Joint action: bodies and minds moving together. *Trends in Cognitive Sciences*, 10(2):70–76, February 2006.

[98] E. A. Sisbot, R. Alami, T. Simeon, K. Dautenhahn, M. Walters, and S. Woods. Navigation in the presence of humans. In *5th IEEE-RAS International Conference on Humanoid Robots*, pages 181–188. IEEE, 2005.

[99] L. Takayama and C. Pantofaru. Influences on proxemic behaviors in human-robot interaction. *Intelligent Robots and Systems*, pages 5495–5502, 2009.

[100] J. Mumm and B. Mutlu. Human-robot proxemics: physical and psychological distancing in human-robot interaction. In *Human-robot Interaction*, pages 331–338, 2011.

[101] J. Mainprice, M. Gharbi, T. Simeon, and R. Alami. Sharing effort in planning human-robot handover tasks. In *Robot and Human Interactive Communication*, 2012.

[102] M. J. Sadigh and H. Ahmadi. Safe grasping with multi-link fingers based on force sensing. In *IEEE International Conference on Robotics and Biomimetics*, pages 1796–1802. IEEE, December 2009.

[103] G. Hoffman and C. Breazeal. Cost-based anticipatory action selection for human-robot fluency. *IEEE Transactions on Robotics*, 23(5):952–961, October 2007.

[104] C. Rich, B. Ponsler, A. Holroyd, and C. L. Sidner. Recognizing engagement in human-robot interaction. In *Human-Robot Interaction*, pages 375–382, March 2010.

[105] B. Mutlu, F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita. Nonverbal leakage in robots: communication of intentions through seemingly unintentional behavior. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, HRI '09, pages 69–76. ACM, 2009.

[106] M. Cakmak, S. Srinivasa, Min Kyung Lee, S. Kiesler, and J. Forlizzi. Using spatial and temporal contrast for fluent robot-human hand-overs. In *Human-Robot Interaction*, 2011.

[107] V. Micelli, F. Rosiello, J. Aleotti, and S. Caselli. Design of gestural interfaces for simulated smart environments. In *IntTech Workshop, 3rd European Conference on Ambient Intelligence*, Salzburg, Austria, 2009.

[108] V. Micelli, A. Arca, J. Aleotti, J. A. Diaz-Nicolas, S. Caselli, and M. T. Arredondo. Evaluation of hand interfaces for intelligent virtual environments. In *4th Symp. on Ubiquitous Computing and Ambient Intelligence (UCAmI2010)*, Valencia, Spain, September 2010.

[109] V. Micelli, A. Avanzi, J. Aleotti, and S. Caselli. Evaluation of a wiimote-based user interface for immersive environments. In *First International Joint Conference on Ambient Intelligence (AmI10). Workshop on Interaction Techniques in Real and Simulated Assistive Smart Environments*, Malaga, Spain, 2010.

[110] P. Dietz and D. Leigh. Diamondtouch: a multi-user touch technology. In *ACM User interface software and technology (UIST2001)*, 2001.

[111] M. Kaltenbrunner, S. Jorda, G. Geiger, and M. Alonso. The reactable: A collaborative musical instrument. In *Workshop on Enabling Technologies*, 2006.

[112] J.Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *User interface software and technology*, 2005.

[113] J. Edelmann, A. Schilling, and S. Fleck. The DabR - a multitouch system for intuitive 3D scene navigation. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 1–4, may 2009.

[114] G. Niezen and G. P. Hancke. Gesture recognition as ubiquitous input for mobile phones. In *International Workshop on Devices that Alter Perception, conjunction with Ubicomp*, 2008.

[115] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. In *Proceedings of IEEE PerCom*, 2009.

[116] D. H. Wilson and A. Wilson. Gesture recognition using the XWand. *Technical Report CMURI-TR-04-57, CMU Robotics Institute*, 2004.

[117] T. Schlömer, B. Poppinga, N. Henze, and S. Boll. Gesture recognition with a Wii controller. In *International Conference on Tangible and Embedded Interaction*, pages 11–14, Bonn, Germany, February 2008.

[118] J. Mäntyjärvi, J. Kela, P. Korpipää, and S. Kallio. Enabling fast and effortless customization in accelerometer based gesture interaction. In *Proceedings of*

*the 3rd International Conference on Mobile and Ubiquitous Multimedia*, pages 25–31. ACM Press, New York, October 27-29 2004.

[119] V. M. Mäntyla. Discrete hidden markov models with application to isolated user-dependent hand gesture recognition. *VTT publications*, 2001.

[120] F. G. Hofmann, P. Heyer, and G. Hommel. Velocity profile based recognition of dynamic gestures with discrete hidden markov models. In *Lecture Notes in Computer Science*, volume 1371, pages 81–95. Springer, Heidelberg, 1998.

[121] J. Kela, P. Korpipaa, J. Mantyjarvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca. Accelerometer-based gesture control for a design environment. In *Personal Ubiquitous Computing 10*, pages 285–299, 2006.

[122] R. Y. Wang and J. Popovic. Real time hand-tracking with a color glove. In *The 36th International Conference and Exhibition on Computer Graphics and Interactive Techniques*, July 2009.

[123] S. Salti, O. Schreer, and L. Di Stefano. Real time 3D arm pose estimation from monocular video for enhanced hci. In *Proceeding of the 1st ACM workshop on Vision networks for behavior analysis*, October 2008.

[124] OpenNI organization. *OpenNI User Guide*, November 2010. Available from: `http://www.openni.org/documentation`.

[125] G. Saggio, P. Cavallo, A. Fabrizio, and S. O. Ibe. Gesture recognition through HITEG data glove to provide a new way of communication. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, ISABEL '11, New York, NY, USA, 2011. ACM.

[126] G. Costantini, G. Saggio, and M. Todisco. A glove based adaptive sensor interface for live musical performances. In *1st International Conference on Sensor Device Technologies and Applications*, Venice, Italy, July 26-28 2010.

[127] P. Lu and M. Huenerfauth. Accessible motion-capture glove calibration protocol for recording sign language data from deaf subjects. In *Eleventh International Conference on Computers and Accessibility*, Pittsburgh, PA, USA, October 26-28 2009.

[128] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[129] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 379–385, Champaign, IL, June 1992.

[130] F. Samaria and S. Young. HMM-based architecture for face identification. *Image and Vision Computing*, 12(8):537 – 543, 1994.

[131] G. Welch and G. Bishop. An introduction to the kalman filter. *Design*, 7(1):1–16, 2001.

[132] S. Arulampalam, S.l. Maskel, S. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, February 2002.

[133] C. Kwok, D. Fox, and M. Meila. Real-time particle filters. *Proceedings of the IEEE*, 92(3):469 – 484, March 2004.

[134] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of the 4th European Conference on Computer Vision - Volume I*, ECCV '96, pages 343–356, London, UK, 1996. Springer-Verlag.

[135] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[136] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag, 2001.

[137] J. Davis and M. Shah. Visual gesture recognition. *IEEE Proceedings - Vision, Image and Signal Processing*, 141(2):101–106, April 1994.

[138] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1325 –1337, December 1997.

[139] M. Yeasin and S. Chaudhuri. Visual understanding of dynamic hand gestures. *Pattern Recognition*, 33(11):1805–1817, 2000.

[140] P. Hong, M. Turk, and T.S. Huang. Gesture modeling and recognition using finite state machines. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition.*, pages 410–415, March 2000.

[141] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[142] Chenglong Yu, Xuan Wang, Hejiao Huang, Jianping Shen, and Kun Wu. Vision-based hand gesture recognition using combinational features. In *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing.*, pages 543 –546, October 2010.

[143] P. Modler and T. Myatt. Recognition of separate hand gestures by time-delay neural networks based on multi-state spectral image patterns from cyclic hand movements. In *IEEE International Conference on Systems, Man and Cybernetics.*, pages 1539 –1544, October 2008.

[144] D. K. Ghosh and S. Ari. A static hand gesture recognition algorithm using k-mean based radial basis function neural network. In *8th International Conference on Information, Communications and Signal Processing*, pages 1 –5, December 2011.

[145] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews.*, 37(3):311 –324, May 2007.

[146] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[147] Jr. Forney, G.D. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[148] F. Bettens and T. Todoroff. Real-time DTW-based gesture recognition external object for Max/MSP and Puredata. In *Proceedings of the 6th Sound and Music Computing Conference*, July 2009.

[149] G. A. ten Holt, M. J. T. Reinders, and Hendriks E. A. Multi-dimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, June 13-15 2007.

[150] T. Darrell and A. Pentland. Space-time gestures. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, pages 335–340, June 1993.

[151] M. Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[152] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, October 2007.

[153] E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *First SIAM International Conference on Data Mining*, 2001.