Technische Universität München
Lehrstuhl für Echtzeitsysteme und Robotik

Context-aware human-robot collaboration as a basis for future cognitive factories

Claus Lenz

Vollständiger Abdruck der von der Fakultät der Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. Daniel Cremers
Prüfer der Dissertation:	1. Univ.-Prof. Dr. Alois Knoll
	2. Prof. Dr. Tomohiro Shibata (Nara Institute of Science and Technology)

Die Dissertation wurde am 24. Mai 2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 30. September 2011 angenommen.

Zusammenfassung

Die verarbeitende Industrie stößt aufgrund der stetig ansteigenden Produktvarianten bei stark verkürzten Produktzyklen derzeit mit konventionellen Optimierungsstrategien an ihre Grenzen. Die zukünftige Wettbewerbsfähigkeit der Hersteller hängt von deren Fähigkeit ab, sich flexibel und zeitnah an die Gegebenheiten sich schnell verändernder Märkte anzupassen. Die Zusammenarbeit von Mensch und Roboter wird hierbei als eine vielversprechende Möglichkeit angesehen, konventionelle Strategien zu ergänzen. Wenn Mensch und Roboter als Team zusammenarbeiten, ergänzen sich die Stärke, Ausdauer und Effizienz des Roboters und die Fingerfertigkeit und kognitive Fähigkeit des Menschen zu einem flexiblen und leistungsstarken Gesamtsystem.

Diese Arbeit beschäftigt sich mit der Identifizierung benötigter Funktionalitäten und Mechanismen, um die Zusammenarbeit von Mensch und Roboter effizient und für den Menschen natürlich zu gestalten. Um dem Menschen einen leistungsfähigen Kooperationspartner zur Verfügung zu stellen, orientiert sich die Arbeit an den Ergebnissen psychologischer Forschung. Diese legen den Schluss nahe, dass die Fähigkeit, Aktionen des Kooperationspartners vorherzusagen und eigene Aktionen diesen Vorhersagen anzupassen, der Schlüssel zu einer erfolgreichen Zusammenarbeit ist. Dies setzt voraus, dass der Roboter in der Lage ist, seine dynamische Umwelt wahrzunehmen und den damit einhergehenden Kontext zusammenzutragen und zu repräsentieren. Desweiteren müssen die Aktionen des Menschen erkannt werden, um dadurch in geeigneter Weise dynamische und adaptive Aktionen des Roboters auszuführen. Diese Fähigkeiten wurden in einem verteilten Software Framework unter Verwendung von modell-basiertem Tracking und aufgaben-basierter Robotersteuerung umgesetzt.

Auf der Roboterplattform *JAHIR*, die im Laufe der Arbeit in Zusammenarbeit mit Projektpartnern entwickelt und aufgebaut wurde, wurden Kollaborationsexperimente zwischen Mensch und Roboter durchgeführt. Die Ergebnisse untermauern, dass die Umsetzung der genannten Mechanismen zu einer Verbesserung der Zusammenarbeit in Bezug auf Effizienz, Adaptivität und Flexibilität des Systems beiträgt.

Abstract

Today, manufacturing industry is faced with increasing product variants while product cycles decrease. Since conventional optimization strategies are saturating, future competitiveness of manufacturers will be highly related to their ability and flexibility to adapt to these essential market requirements. The collaboration between human and robot has been announced as a promising approach to solve these challenges, because it teams the strength and the efficiency of robots with the high degree of dexterity and the cognitive capabilities of humans into a flexible overall system.

This thesis identifies required functionality and mechanisms for both efficient and natural human-robot collaboration. To give the human an appropriate and efficient collaboration partner, the work of this thesis bases on recent psychological research about cognitive processes of joint-action among humans. These psychological studies interpret, that the success of efficient collaboration depends on the team-partners' abilities to predict actions and to affect own actions using the prediction. This implies, that the robotic system is capable of perceiving, gathering and representing contextual information such as the environment, recognizing current actions of the human, and by this producing context-aware dynamic and adaptive actions. These demanded capabilities are realized in a distributed software framework using model-based visual tracking and task-based robot control.

Human robot collaboration experiments were performed using the integrated demonstration platform *JAHIR*, that was co-developed along with this work. The results validate the benefits regarding efficiency, adaptability, and flexibility.

Acknowledgements

I want to thank Prof. Dr.-Ing. habil. Alois Knoll for giving me the opportunity to be part of his group, to work on this thesis topic, and for supporting discussions.

My sincere thanks go to Prof. Dr. Tomohiro Shibata for his helpful comments and ideas during the writing of the thesis.

Additionally, I want to thank Prof. Dr. Gordon Cheng for interesting discussions and his wise remarks.

Special thanks go to my colleagues Thorsten Röder[1], Manuel Giuliani[2], Markus Huber[3], Markus Rickert[4], Giorgio Panin and the OpenTL team[5], all colleagues at the Robotics and Embedded Systems Lab, and to all my students who contributed to this work.

Dr. Gerhard Schrott and the Secretoids Monika Knürr, Amy Bücherl, and Gisela Hibsch gain a special thank for their support in all affairs.

I want to acknowledge my project colleagues within CoTeSys[6]—especially Jürgen Blume, Alexander Bannat, Tobias Rehrl, Prof. Dr.-Ing. Frank Wallhoff (MMK), Wolfgang Rösel (*iwb*), and Christoph Mayer (I9).

This work was supported by the DFG cluster of excellence *CoTeSys* within the projects *JAHIR (Joint Action for Humans and Industrial Robots)*, *BAJA (Basic Aspects of Joint Action)*, and *ITrackU (Image-based Tracking and Understanding)*.

[1] for fruitful discussions, good collaboration, and proof reading

[2] who shared the office with me at all locations

[3] for the prosperous collaboration

[4] for his advises regarding robotics, help with efficient C++ programming, and his *RL*

[5] for interesting discussions and the joint development of OpenTL

[6] `www.cotesys.org`

Contents

Chapter 1

Introduction

Contents

As industrial robots lack a lot of capabilities such as high flexibility or adaptability towards the human, robotic systems need to be advanced in that direction to create new flexible ways in the production. Additionally, it is important to keep and/or integrate the human in the loop *to also profit from the human's capabilities. For future production processes it is of high importance to enable a real "human–(industrial) robot collaboration" to open new ways of (industrial) processes in* factories of the future *(Section 1.1). After stating the main aims, an overview of the contributions that come with this thesis (Section 1.2) and the structure is presented (Section 1.3).*

1.1 Motivation

Robots have proven their success since their introduction in the industry in the late 60ties. Mainly because robots are reconfigurable and therefore applicable for a variety of tasks. At the end of 2008, more than one million robots have been in operation world-wide [1] with most of the robots following static action programs to handle repetitive production tasks and working isolated or surrounded by fences.

Since manufacturing industry is faced with increasing product variants while production cycles decrease, conventional strategies to optimize production steps are saturating. Future competitiveness of manufacturers will be highly related to their ability and flexibility to adapt to these essential market requirements [2]. This flexibility is hardly reachable with fully automated production processes. Especially, if small lot sizes of units or prototypes with a high variety and high task complexity are needed, current automation strategies are not cost efficient [3].

Although robots are flexible and multi-purpose machines, this flexibility heavily reduces due to their static task programming. If the task, the product or the environment changes, their movements often need to be re-programmed from scratch. The costs of task-adequate robots and the effort to set-up, program, and integrate them into existing production lines amortize only with a large number of manufactured products, because the costs of the integration of a robot are approximately ten times the price of the robot itself [4].

The collaboration between human and robot has been announced as a promising approach to solve these challenges, because it teams the strength and the efficiency of robots with the high degree of dexterity and the cognitive capabilities of humans into a flexible overall system. As consequence of current flexible automation techniques including *flexible manufacturing systems (FMS)* and *reconfigurable manufacturing systems (RMS)* [5], a recent trend in robotics focuses on new generations of robots with the capability to directly assist humans. This bridges the gap between fully automated systems and fully manual workstations [6]. Highly related is that a significant amount of research has been done in the area of physical human-robot interaction (pHRI) [7, 8] that can also be applied to collaborative tasks between human and robot. Additionally, the introduction of cognitive capabilities [9] for assistive robotic systems, new robot control schemes, and advances in artificial perception, to name only a few, are enabling factors to bring human and robot further together in a shared workspace. Especially, technical systems need to improve their performance with respect to unforeseen events, flexibility in their use, and field of application through cognitive capabilities [2]. This creates more flexible and (partly) autonomous machines that are able to directly cooperate and support the human co-worker [10, 11]. Further, keeping the human *in the loop* of production processes for highly flexible assembly advances the skills of the overall system due to cognitive and senso-motoric advantages of the human. Hence, the human is part of production processes when he is needed and can concentrate on other tasks to improve the overall system performance [12].

The support of humans by robotic systems can then lead on the one hand to more ergonomic work places and on the other hand to more time-efficient production processes. Additionally, the amount of fixed production costs in relation to variable costs can be reduced [13]. The advantage of the potentials for humans and robots to work together as a team is only in early stages and needs a safe, robust and efficient realization [14]. Once this is reached, the subsequent flexibility and adaptability of human and robot collaborating as a team allows production scenarios in permanently changing environments as well as the manufacturing of highly customized products in *factories of the future*.

1.2 Thesis aims and contributions

1.2.1 Aims

The use of industrial robots is usually characterized by a strict workspace separation in space or time to guarantee the safety of the human. But the combination of human flexibility and machine efficiency offers several advantages including ergonomic and efficiency improvements. As many algorithmic and technical challenges are still unsolved, the usage of collaborative systems is not standard in the industry, but approaching fast. Up to now, systems that combine humans with robots are referred to as *hybrid assembly* systems. These systems divide the task in simple tasks suited for the robot as well as complex and changing tasks for the human [13].

This thesis aims to advance the collaboration of humans and robots in production environments for assembly tasks by stating and implementing required factors. Considering that collaboration on a shared assembly task consists of a sequence of coordinated actions in space and time, the success depends on the collaboration partners' abilities to adapt to and to predict actions of the partner [15, 16]. These abilities are achieved by the human by combining of several mechanisms [16] including *joint attention* to steer the concentration and to share representations about events and objects, *task sharing* to be able to predict the next steps based on the expected behavior of the partner before an action can be observed, *action observation* to predict the next goal based on the current behavior of the partner, and *action coordination* to adjust own actions in space and time according to the behavior of the partner.

The overall aim of this thesis is to implement and transfer the described mechanisms to a robotic system to form a basis of covered functionality to enable manlike collaboration with a human. In this thesis, it is assumed that the assembly plan is known to both human and robot a-priori and that the attention of the human lays on the joint assembly task. Under these assumptions, the mechanisms *task sharing* and *joint attention* are disregarded. Hence, this thesis targets especially on the mechanisms *action observation* and *action coordination* as presented in Chapters 3 and 4.

1.2.2 Contributions

To realize the mechanisms *action observation* and *action coordination*, the robotic system needs to be endowed with capabilities to perceive the environment, to gather and represent contextual information, to recognize human actions, and to produce appropriate dynamic and adaptive actions using the gathered context. Therefore, this thesis contributes in particular to these aspects:

Perceiving the environment: *OpenTL*[1] was co-developed and used as software backend to perform the computer vision and tracking tasks. *OpenTL* is a general purpose tracking library with a lot of well-known and new computer vision and tracking algorithms including many inter-exchangeable Bayesian filters (see Section 3.2). In this way, the software modules and algorithms created here can be adapted and parameterized to fulfill or to be integrated in other algorithms or modules. Further, perceiving the environment also inherently depends on the integration of multiple sensor information. Therefore, sensors including depth cameras, color cameras, the Microsoft Kinect, and infrared cameras (see Section 5.1.2 for details) are used to gain data and corresponding processing modules perceive the surrounding, update the geometric representation (see Section 5.2), and deliver information about e.g. the hand positions (see Section 3.3).

Representing contextual information: Since contextual information mainly includes in this work geometric surrounding and information such as the position of the hands, a dynamic way to manage this information is needed. The resulting geometric representation is based on the *Robotics Library*[2] developed at the Robotics and Embedded Systems Lab with the extension, that every sensing module can add, update, and remove geometrical shapes via a standardized communication channel (see Section 5.2). An example that visualizes the geometric representation of an overall *cognitive factory* scenario is depicted in Figure 1.1 with the human worker being tracked by the Microsoft Kinect sensor.

[1] http://www.opentl.org

[2] http://roblib.sourceforge.net

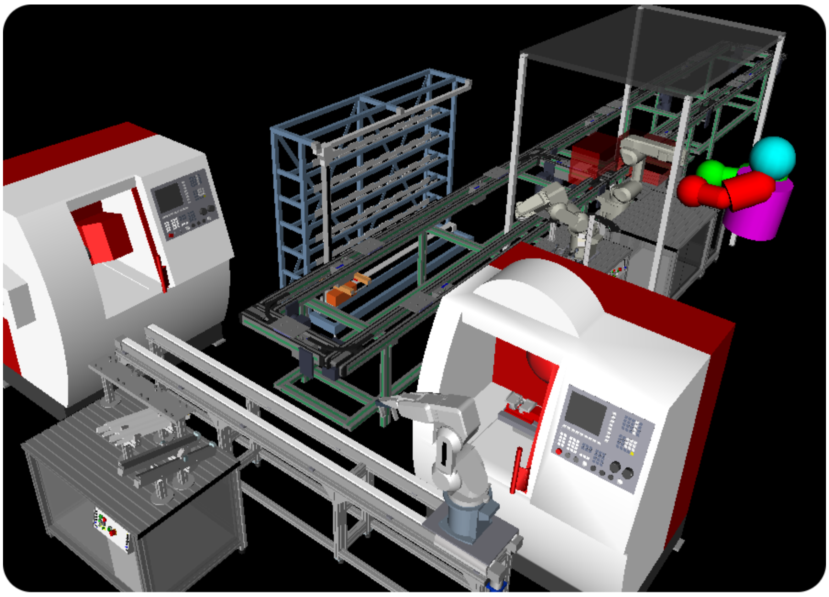


Figure 1.1: Geometric representation - The whole scenario can be represented geometrically including the pose of human workers gained from diverse sensors

Recognizing human actions: The observation of human actions by an assistive robotic system can, for example, be used in an assembly task to act pro-actively with the preparation of future steps based on the current action of the human. Therefore, an assembly experiment with humans that were not influenced by a robot or any other technical device was used as a basis to train models that can recognize the current action of the human (see Section 3.1). The experiments showed, that with the right abstraction level regarding sensor information, the actions can be recognized by means of hand velocity, acceleration, and jerk. These enables the transfer of the recognition models to the human robot collaboration scenario *JAHIR* . Due to the gained flexibility, the models performed well, although the sensors and the arrangement of the set-up changed (see Section 5.3).

Producing appropriate dynamic actions using available information: In this thesis, the concept of task-based hierarchical control has been accomplished with position controlled, closed architecture industrial robots. With the seamless integration of the geometric context representation, a collision avoidance module for the robot was created, that preserves movements of the robot that are not in conflict with a given avoidance strategy (see Sections 4.1 and 5.4.1). Further, it has been experimentally validated in the *JAHIR* scenario, that the choice of the motion velocity profile influences the unconscious adaption of the human. In a hand over experiment, humans showed varying reaction times depending on the velocity profile, although they could not see a difference in the motion as questioned after the experiment (see Sections 4.2 and 5.4.2). Additionally, it has been verified with experiments using the *JAHIR* set-up, that the right timing of actions in e.g. hand-over tasks enables a seamless interaction without waiting times for human *and* robot (see Section 4.3) with an improved overall task efficiency.

Demonstration platform: In order to show the mechanisms and the inter-working of required functionality, the demonstration platform *JAHIR* [17] was created together with project partners from the electrical and mechanical engineering department[1]. This includes—beside the used hardware (see Section 5.1)—the creation of several processing modules and a generic software architecture to inter-connect these single processing modules among multiple computers (see Section 5.2).

1.3 Structural design of the thesis

Based on the aims of the thesis to present, implement, and proof required basic mechanisms and concepts to enable an advanced collaboration between human and (industrial) robot (see Section 1.2), the thesis is organized as follows:

Chapter 2 gives an overview and shows the large desire to make progress in the area of human-(industrial) robot collaboration by reviewing related systems and projects in this sector (Section 2.1). As background information important factors of human-industrial robot collaboration are presented. This includes a summary of international industrial norms and strategies to enhance the safety for the human (Section 2.2). Important functional aspects based on psychological findings are given (Section 2.3), that could improve the collaboration between human and robot. Further, the demand to bring all these points together in a generic software framework under certain design principles is presented (Section 2.4).

Chapter 3 sets the basis of transferring the psychological mechanism *action observation* of joint action to a robotic system. This mechanism enables an assistive robotic system to predict the next goal based on the current behavior of the opponent to assist with pro-active behavior. Hence, the system needs to recognize the

[1] namely: Wolfgang Rösel, Jürgen Blume, Alexander Bannat, Frank Wallhoff

current action of the human using captured information form sensory data. An exemplary assembly situation was designed and afterwards executed by multiple subjects (Section 3.1). The recorded data was used to train models that analyze the workflow of the assembly. The experiments revealed that it is sufficient to use only information derived from the hand position. Therefore, a hand tracking method was developed (Section 3.3) using model-based visual tracking (see Section 3.2).

Chapter 4 sets the basis in order to transfer the mechanism *action coordination* as fundamental principle adjusting own actions in space and time according to the behavior of the collaboration partner or the perceived context. The coordination of robotic movement based on a hierarchy of atomic tasks including the geometric awareness of the robot, that integrates static and dynamic geometric representations of the surrounding, is presented (Section 4.1). The coordination of actions also incorporates many conscious and unconscious aspects, which need to be considered. Therefore, Section 4.2 presents a handing over experiment that shows that the reaction times of humans can unconsciously be influenced in a positive manner by choosing appropriate robotic motion profiles. Parameters that have to be negotiated by the subjects during a hand over also include the right timing of actions. Hence, Section 4.3 shows how the robot can use the observations of human actions to efficiently coordinate actions in time to increase the collaboration fluency.

Chapter 5 presents the integrated demonstration platform *JAHIR* that evolved throughout this thesis. This includes the hardware set-up (Section 5.1) and the software architecture behind it (Section 5.2) to glue the developed software modules together. Both form a powerful demonstration platform that was also used beyond the scope of this work for experiments and demonstrations. The effects of transferring the psychological motivated mechanisms action observation and action coordination to the *JAHIR* platform are presented and evaluated (Sections 5.3, 5.4.2 and 5.4.3) along with sample applications of the task-based controller (Section 5.4.1).

Chapter 6 concludes the thesis. A review of the presented work and the corresponding contributions is given. Additionally, possible improvements and a glance towards future work are presented.

Chapter 2

Background

Contents

This chapter gives an overview and shows the large desire to make progress in the area of human-(industrial) robot collaboration by reviewing related systems and projects in this sector (Section 2.1). As background information important factors of human-industrial robot collaboration are presented. This includes a summary of international industrial norms and strategies to enhance the safety for the human (Section 2.2). Important functional aspects based on psychological findings are given (Section 2.3), that could improve the collaboration between human and robot. Further, the demand to bring all these points together in a generic software framework under certain design principles is presented (Section 2.4).

2.1 State of the art and related work

2.1.1 Collaborative (industrial) robotic systems

Robotic systems that assist human workers in production processes as well as in production environments are an active research field with a variety of applications. The following overview gives an impression about systems that have been introduced in the past to tackle human-robot collaboration in the production.

A robotic system consisting of multiple impedance-controlled robots is introduced by Kosuge in [18]. With this system, a joint object manipulation of human(s) and (multiple) robot(s) in a dynamic way is possible. The control scheme assumed that the interaction takes always place through an object. A force sensor attached to the wrist of each robot measured the external forces on the robots through the object. The human commanded the movement of the robots by applying forces to the object. This approach was later extended to the mobile robotic assistant *MR Helper* as presented in [19].

Khatib presented in [20] several strategies to support workers in physical tasks for compliant motion and cooperative manipulation. In addition to the controlling of multiple arms corresponding to the applied forces, multiple holonomic mobile platforms were coordinated to have a fully flexible mobile assistant.

For situations that involve large interaction forces as it is for example present in the automobile production, *Cobots* have been introduced by Colgate in [21]. These specialized mechanical devices provide guidance to human operator's motion. The *cobots* act passively with virtual fixtures and virtual walls to support and guide the human collaborator without the intention to act autonomously.

PowerMate—introduced by Schraft in [12]—is another example of a system designed to give the human worker a robotic assistant for handling and assembly tasks. The system follows current safety norms and works with normal velocity, if no human is present. In presence of a human, the velocity is limited and with the confirmation of the human, the robot can be guided to place heavy good using force/torque sensing. With this mode, it is possible to pull the robot on its gripper to a desired position.

A rather application oriented approach to ease and speed up the programming of industrial robots is presented by Pires in [22]. The approach is object-oriented and based on a client-server architecture. It is claimed that the underlying concept is general enough to be applied to organize and program overall flexible manufacturing cells.

Flexible, adaptive, and *cognitive* robots are especially needed, if small lot sizes of units or prototypes with a high variety and high task complexity are required. That means, that future industrial robotic assistants should be flexible and safe on the one side and *clever helpers* in manufacturing environments on the other side. Hägele describes this in [14] as the evolution*from robots to robot assistants*. To show the concept presented in [14], the mobile robot assistant *rob@work* has been developed by Helms and Hägele [23, 24] as direct interacting and flexible device for assistance. The collection of functionality includes automatic path planning, obstacle avoidance, precise positioning, and several safety concepts of the robotic arm motions. Further, the ease of use to instruct tasks was demonstrated.

A cooperative assembly cell system using a four-axis scara robot was presented by Thiemermann in [25, 26]. The system enhanced standard tasks of the assembly robot with new functionality to help the human worker with the work-piece positioning or with other parts and tools. A camera-based system is used to adapt the working velocity of the robot according to the distance between human and robot.

A mobile assistive robot for flexible and interactive manufacturing is presented by Stopp in [27]. Due to safety aspects, the human instructor teaches interactively the robot an order-picking from outside its workspace using a laser pointer and a handheld computer. This has been extended to a safety concept using dynamic sensor-based surveillance of the robot workspace and multiple safety regions with (possibly) different safety levels [28].

Iossifidis presents in [29] the stationary 7 dof robotic assistant *CoRA*, that is able to cooperatively solve an assembly task using diverse inter-connected components including speech, object, and gesture recognition. Additionally, the robot has been prepared with an artificial skin to allow a touching and positioning of the robot by the human.

Gecks presents in [30] *SIMERO*, a stationary industrial robot system. Several stationary cameras that detect obstacles to dynamically adapt motions accordingly supervise the workspace of the robot. This approach has later been enhanced by the use of multiple depth cameras [31] to perform a three-dimensional collision avoidance for unknown objects. One master computer evaluates the synchronously acquired data of slave computers and employs a geometrical model to revise the data points and to adjust the robot velocity according to distance of human and robot. In this way collisions with obstacles or the human can be damped or even prevented.

A pro-active collaboration between human and robot based on the recognition of the intentions of the human is described by Schrempf in [32]. As the recognition of intentions is quite uncertain, the robot resolves this uncertainty by pro-active execution to minimize the overall costs. For the system, Dynamic Bayesian Networks (DBNs) were used.

The approach by Rickert presented in [33] describes a scenario in which a human and the *JAST* robot—a robot with two arms in a human-like arrangement—build together a wooden model of an aircraft using a distributed architecture divided into the high-level components *input, interpretation, representation, reasoning,* and *output* with several functional modules.

Another example is the DLR lightweight arm presented in [34, 35] that was transferred to KUKA[1] and is now commercially available. This lightweight robot is especially designed for interaction with unknown environments and with humans [36]. The integrated compliance, virtual fixtures, high interpolation rate, and many more make the robot very promising to work alongside with humans. Haddadin uses this arm to build a sensor-based robotic co-worker for a safe and close cooperation and presents strategies for safe interaction with the human [37].

Chuang presents in [10] a study on human robot collaboration design for robot assisted cellular manufacturing and identifies that collaboration planning, collaboration safety, mental workload management, and a good man-machine interface as four main concepts that need to be met to enable human robot collaboration. Further the authors present in [38] experiment evaluations of different supportive information formats and show, that it is important to display information near relevant object and in the visual attention field of the human. Additionally, the combination of information sources such as text and images is important.

2.1.2 Related large scale research projects

Since the development potential and the demand of the industry is high, research concentrates with large-scale projects on that topic with special interest. The following (incomplete) list gives an overview of currently running or recently finished projects:

Morpha: The project *Morpha*[2] (1999 to 2002) worked towards enabling a robot assistant to cooperate with and assist the human user in a variety of tasks using intuitive and natural ways to communicate.

SMERobot: The project *SMERobot*[3] (2005 to 2009) was funded within the 6th Framework Program of the European Union with the scope to create a new family of robots suitable for small and medium sized manufacturing enterprises.

[1] `http://www.kuka.com`

[2] `http://www.morpha.de`

[3] `http://www.smerobot.org`

PHRIENDS: The project *PHRIENDS – Physical Human-Robot Interaction: DepENDability and Safety*[1] (2006 to 2009) worked towards a new physically interactive and safe generation of robots and robotic components.

rosetta: *rosetta – RObot control for Skilled ExecuTion of Tasks in natural interaction with humans; based on Autonomy, cumulative knowledge and learning*[2] (2007 to 2013 funded by European Union under the FP7) investigates human-centric technology for industrial robots to cooperate naturally with human workers.

Custom Packer: The EU funded project *Custom Packer – Highly Customizable and Flexible Packaging Station for mid- to upper sized Electronic Consumer Goods using Industrial Robots*[3] (2010 to 2013) tries to develop a scalable and flexible packaging assistant to support human workers in packaging mid to upper sized and mostly heavy goods.

JAST: The EU FP6 project *JAST – Joint-Action Science and Technologie*[4] (2004 to 2009) directed to develop autonomous systems that communicate and act jointly with a human on mutual tasks in dynamic unstructured environments.

CoTeSys: The *Cognitive Factory* with a production line for individualized manufacturing including direct human robot collaboration and joint action is one of the central demonstration scenarios within the Cluster of Excellence *Cognition for Technical Systems—CoTeSys*[5] (2006 to 2012).

But not only research tries to bring together human and robot in the same workspace, also industry sees the potential and the great benefit for human robot teaming in industrial production [39]. With new developments of robotic manufacturer in the direction of *safe robot*, it seems to be clear, that human operators and robots will soon be working together in one workspace without the need for the robotic system to suspending its work when humans come too close to the robot [40].

[1] http://www.phriends.eu

[2] http://www.fp7rosetta.org/

[3] http://www.custompacker.eu

[4] http://www6.in.tum.de/Main/ResearchJast

[5] http://www.cotesys.org

2.2 Safety issues in human–industrial robot collaboration

Industrial robots are possibly big, heavy, fast, and powerful. Hence, they can generate high forces and can heavily injure a human if they come into contact. But as robotic applications aiming towards a direct physical human-robot interaction and a close human-robot collaboration emerge more and more in multiple fields including industrial production and automation, the safety for the human and the dependability of collaborative robotic systems become crucial questions [41]. Sharing the same workspace might lead to situations where human and robot collide with each other. This can be very harmful to the human, especially, when standard industrial robots are used [42, 43].

An objective classification of danger/safety is needed to indicate and identify the risk of the collaboration for the human. To find such categories, possible dangerous situations, impacts, and injuries have been evaluated in the literature using different safety measure schemes including the *Wayne State University Tolerance Curve (WSTC)*, *Manipulator Safety Index (MSI)*, or the *Head Injury Criteria (HIC)* to name only a few [44, 45, 46, 47, 48]. In addition to find the right measures, the risk analysis has to include aspects regarding the placement and surrounding of the robot, the mounting (e.g. is the work performed below a robot), the robot type (e.g. force, speed, energy), the end effector demands (e.g. sharp edges), process dependent hazards (e.g. temperature), personal safety devices (e.g. protective clothing), and the construction and installation of control elements (e.g. reachability, ergonomic aspects) [49].

To ensure the safety of the human, industrial norms dealing with the usage of industrial robots along with humans are highly constraining. In Section 2.2.1, a short overview about current industrial norms is given. The introduction of such norms shows, that the industry has also identified a huge potential for the collaboration between human and robot and tries to find rules, how this can be applied in the future.

2.2.1 Norms and industrial standards for human robot collaboration

In order to prevent dangerous situations, impacts, and the corresponding physical damages of humans, international safety standards and norms regulate the use of physically cooperating robots. The ISO committee TC184/SC2[1] is concerned with the devel-

[1] http://www.iso.org/iso/iso_technical_committee.html?commid=54138

opment and revision of such standards. Existing ISO robot standards (e.g. the *ISO 10218-1* [50]) have been developed with a limiting focus on industrial use, because robots have formerly only been considered as valuable tools for manufacturing in industrial environments [51]. Therefore, these norms are mostly only applicable to static industrial tasks such as lifting heavy parts, machining various metal and non-metal components, and joining large panels. Such applications demand the use of robots with large and powerful machines, which result in highly hazardous collaboration partners. Such risks for the human are avoided by separating the workspace of robots with real or virtual cages or in time. With respect to human-robot collaboration, *ISO 10218-1* [50] covers the topics:

Stopping functions: (5.10.2) specifies that the robot has to perform protective or emergency stops when humans are in the robot's workspace.

Guiding the robot by hand: (5.10.3) allows the guidance of a robot with equipment closely mounted to the end effector including an emergency and a confirmation button with a speed limitation of $0.25\,\mathrm{m\,s^{-1}}$ max.

Speed and position control: (5.10.4) fixes the maximum allowable speeds of robot arms and end effectors when humans are in the robot's workspace to $0.25\,\mathrm{m\,s^{-1}}$ max.

Power and force control: (5.10.5) limits the maximum allowable power and forces applied by robot arms and end effectors to $150\,\mathrm{N}$ when humans are in the robot's workspace.

While such arrangements keep the human at distance, new emerging applications need the human as collaboration partner close to the robot. Therefore, the international standards are advanced [52] and extended to handle safety requirements for robots that allow autonomous work or collaboration with humans. *ISO 10218-2* [49] currently covers the following issues from which at least one needs to be fulfilled:

Design of collaborative operation workspaces: (5.11.3) sets the requirements for the layout design of the workspace around the robot, including safeguarded spaces (where humans are separated from the robot and protected by safeguards) and collaborative spaces where humans are not separated from the robot and hence the robot shall apply the control limits mentioned above. The switching between autonomous and collaboration mode needs to be performed in a way, that no person is endangered (5.11.4).

Collaborative operation modes: (5.11.5) specifies the operating modes that must be designed into the robot's control function when collaborating with a human in the collaborative workspace. Possible modes are: the robot is stopping, if a human is in the collaboration space; the robot is guided by a human; the robot acts autonomously with no human in danger; the robot moves with limited speed $0.25\,\mathrm{m\,s^{-1}}$ if a human is in the collaboration space; the robot acts autonomously and keeps a security distance to humans in the collaboration space and adjusts the velocity accordingly to prevent possibly collisions.

2.2.2 Safety strategies

There are several ways to deal with the safety problem in human robot collaboration. The regulation of industrial norms tries to increase safety by separating the workspace of humans and robots in space or time or by limiting the robot velocities to minimize possible impacts. This is needed, because industrial robots do not comply in case of a collision due to their stiffness. [53] divides safety related approaches into active and passive safety. Passive safety—realized as warning lights, signs, boundary chains, painted boundaries, fences, barriers or robot cages—is static and simple to design; therefore it is very reliable but also very limited. Active safety devices include laser curtains, pressure mats, infrared barriers and capacitive devices to sense and react to changes in the cell's environment. Certified sensor systems give only sparse data and can hardly cope with complex dynamic scenarios. This may allow the human to be the avoiding or careful part, but do not make the robot safe. The differentiation here will be done by physical and/or logical means to improve the collaboration safety with the introduction of physical safe structures, algorithms, sensors, or control strategies.

Physical safety

[54] claims that making a stiff and possibly heavy robot to behave gently and safely seems to be an unrealistic task and presents several principles of compliant actuators. Compliant actuator design is also seen as key to increase safety and flexibility in human robot collaboration [55]. [56] presents with the DM2 arm an actuator principle with a drastically reduced effective impedance that enables essential characteristics for intrinsic safety. The biologically inspired, lightweight and elastic robot arm *BioRob* [57] is able to estimate the weight of payload, to detect collision and touch in a similar way as the human arm. Further, a direct teach-in or a hand guided robot movement becomes possible.

To reduce the impact, the masses of the robot can also be reduced. A first lightweight arm for service applications was proposed in [58] with the whole-arm manipulator. Another example is the DLR lightweight arm [34, 35] that is used among many other things for *Justin*, a lightweight robot especially designed for interaction with unknown environments and with humans [36].

Such approaches mainly have the disadvantage, that the there is a trade-off between safe (hardware) design and absolute accuracy of the robot, which makes it often uninteresting for industrial use. Additionally, the development and industrial investments in regular industrial robots would be to some extend lost if they had to replace all robots for human-robot interaction (HRI). Further, intrinsically compliant robots are hardly available outside of scientific labs.

Logical safety

Logical safety can introduce ways to use also for example standard robots along with humans. The safety is here ensured using algorithms, external sensors, and corresponding software modules. A discussion about safety issues in human robot interaction along with an overview of currently safety devices can be found in [59]. Which and how sensors are used in the industry is given in [53]. As an advanced sensor system, that is eminently applicable in safe human-robot interaction, proximity skins should be named here. An early example is given in [60].

More on the software side [61, 62] presents a very promising integrated human-robot interaction strategy that ensures the safety of the human by a coordinated suite of safety components. The components anticipate and respond to varying time horizons for potential hazards and varying expected levels of interaction. The three main components are:

1. a *safe path planning* that includes a measure of danger based on the robots inertia and the distance between human and robot to find safe trajectories;

2. a *safe controller*, that evaluates the safety of the planned trajectories at each control step and deviates from the planned trajectory in case of danger using the robot configuration based on a danger index;

3. *human monitoring* based on visual to estimate the focus-of-attention of the human and physiological data to measure stress-levels.

The co-worker scenario would greatly benefit from the use of both physical and logical strategies since a compliant robot as with its use the problem of physically safe collaboration may be reduced. In this work the focus does not lay on the use or the development of this kind of robots, but in the creation of an overall framework to enable such a collaboration. This includes more generic components that give information about the current state of the human or his position. With a robot, that is inherently safe, such sensor-based components can then be used to support the system to solve the overall goal re-using e.g. some logical components.

2.3 Psychological aspects

To successfully integrate collaborative assembly systems such as *JAHIR* in today's processes, high demands regarding safety, efficiency, ergonomics, flexibility, programmability, and adaptability need to be met. As humans being experts in physical interaction and collaboration, the investigation of high-level joint-action strategies between humans is important to find strategies for robotic systems [63]. Further, the research on especially humanoid robotics can on the reverse side also contribute to understand how humans behave [64]. Psychological research on cognitive processes of joint-action among humans [16] lists *task sharing, joint attention, action observation* and *action coordination* as important mechanisms that influence the efficiency:

- *joint attention* to steer ones concentration and to share representations about events and objects
- *task sharing* to be able to predict the next steps based on the expected behavior of the opponent before an action can be observed
- *action observation* to predict the next goal based on the current behavior of the opponent
- and *action coordination* to adjust own actions in space and time to the behavior of the opponent

That means, that in collaborating human-human teams, an efficient coordination requires participants that plan and execute their actions in relation to what they expect from the opponents based on observations [15]. Action of the team members are observed and *evaluated* to coordinate own actions appropriately in space and time. Hence, humans negotiate unconsciously various parameters to optimize the co-operation during the collaboration [65]. During repetitions of the same action, the coordination becomes smoother and more accurate and leads to a maximum in comfort and efficiency.

Additionally, enabling factors are the abilities of the system to act autonomously in the environment according to sensor and context information. The communication and the explanations from the assistive system should also follow psychological aspects to decrease the distraction and the cognitive load of the human [66, 67].

Further, technical systems should make use of information provided by multiple sensors, (multiple) actuators, that are embedded in and aware of the *real world* to perceive, reason, learn and plan in a cognitive way. Along with reflectiveness about their own capabilities and limitations, cognitive technical systems know *what* they are doing, *how* things can be done, and *about* the human collaboration partner. This leverages higher flexibility, adaptivity, interaction and collaboration capabilities of the systems [68].

Another important issue constitutes, that collaboration of partners can be defined as being based on achieving a common goal together with commitments of every participating partner. This differs significantly from short-term interaction, where partners have no shared (long-term) goal [69]. Although, it is hard to distinguish these two terms in many cases at first glance, the difference becomes clear if errors occur: partners that act jointly and collaborative with the same global goal in mind can support each other and assist, because both know *what* needs to be done [70].

Therefore, to manage a predefined goal—e.g. the joint assembly of a product—the robot-system needs to know about the task in the production process as well as the human worker (*task sharing*). If the representation of the subtasks is as generic as possible, the role allocation can dynamically change even during the execution. A common representation of human and robot capabilities is an important issue in order to assign tasks according to specific skills [71]. With the knowledge about a shared plan, the system is able to predict possible next action steps and prepare these steps pro-actively.

For cooperation between robot and human, it is important that both partners coincide on the same objects and topics and create a perceptual common ground with a shared representation [16]. Considering *joint attention*, the system needs to have the ability to control the focus-of-attention of the human to direct to relevant objects or has to be able to estimate the human's focus-of-attention directly. This can also include pointing gestures as integrated in [33] or the head orientation of the human worker.

The transfer of these basic mechanisms of joint-action to a robotic assistive system can improve the collaboration of human and robot. This also includes the unconscious adaption of own actions using anticipatory knowledge about the actions of the team member (see Sections 4.2 and 4.3) and the recognition of what the team partner is currently doing (see Section 3.1). The benefit of transferring anticipatory action to a human-robot context is also shown in [72], where a significant improvement of task efficiency compared to reactive behavior was possible. The effects of transferring the deeper investigated aspects *action observation* and *coordination* to the robotic system *JAHIR* are evaluated in Sections 5.3 and 5.4.

2.4 Software design aspects

2.4.1 Design principles

The previous sections presented important factors and aspects that need to be considered for and that influence the design of flexible, dynamic, and adaptive collaborative system. The safety issues and the theoretical cognitive psychological aspects need to go hand in hand. In addition with the demand to have a directed behavior of the robot to reach a goal, robots need to connect perception, action, inference, and management components. Hence, relevant principles for the software design are presented to cover these demands.

- *Concurrent modular processing*,
- *structured management of knowledge*,
- and *dynamic contextual processing*

have been found to be important aspects to design software for robotic systems [73]. To especially allow safer and more efficient collaboration between human and robot considering the psychological and safety aspects, these principles have been supplemented in [74] with guidelines regarding

- *robustness*,
- *fast reaction time*,
- and *context awareness*.

All of these design principles are important for robotics in general, but especially crucial for collaborative robotic systems. For a detailed discussion how these principles are realized in the *JAST* system please refer to [74]. Deeper insights on the realization in the *JAHIR* set-up are given in Section 5.2.

Concurrent modular processing

Robot architectures incorporate many single processing modules that run in parallel and are often distributed among multiple computers. In this way, complicated tasks can be solved by combining higher-level information as a result of solving sub-tasks by specialized parts of the system, which can be executed in parallel. To give an example, the vision system may track a human with a specialized tracking module while the actuator control makes sure that the robot does not collide with the human based on the position data coming from the tracking results.

This design principle also implies that the system can consist of several hierarchical or parallel specialized sub-architectures to structure complex processing. Using the same interfaces or representation of data for similar information, modules can be renewed or extended easily and flexibly due to the modular architecture design. This includes also the intrinsic possibility to distribute modules over several computers employing modern middlewares (see Section 2.4.2) to cope with high computational needs of the modules.

Structured management of knowledge and dynamic contextual processing

Multiple processing modules produce data that might be needed by other modules to perform their processing. The flow of information between the single parts of the system and the form of representation of information needs to be organized, steered, and controlled. Information inside the architecture is defined by sub-architecture ontologies and general ontologies. The term *ontology* is considered as a general expression for any kind of representation format [73]. This means, each sub-component of the system has its own representation for knowledge in the most appropriate way to perform and solve given tasks. A perception module might represent known objects in terms of features such as key-points or histograms along with information about the shape. Other modules are possibly only interested in the positional information and the shape of the object. Therefore, externally the perception module might communicate only positional and body structure data to other sub-architectures.

Although the *internal* module representation is specialized, the *inter-module* communication needs to follow standardized interfaces. Hence, functional blocks share the same communication channel to distribute their knowledge system-wide to interested modules. This way, a seamless integration of new input and processing modules becomes possible without the need to adapt existing modules. In order to realize a goal-oriented behavior, certain control mechanisms need to steer the information flow of the concurrently working system components. [73] proposes that sub-components have to announce their intent to process information while a controlling system component grants a permit if they are allowed to do that or not.

Robustness

Robustness is associated with *robust behavior* of the system at all times. That means a robot must be aware of possible unclear, ambiguous, or unexpected situations, and react in a reasonable way. In the context of robot controlling (see Section 4.1), robust robot behavior includes

- that the trajectories are computed on-line to be able to react quickly on new perceived events,
- that the complex behavior of the robot is decomposed into atomic tasks that can be arbitrarily hierarchically arranged and that can suspend the motion in the case of errors or critical situations,
- and that motions are suspended before actual collisions can occur—e.g. by employing an appropriately chosen internal world representation.

This can also be supplemented with a coordinated suite of safety components as proposed in [61, 62].

Since there are a lot of errors potential sources in such a complex robotic system, the system must be able to compensate for erroneous or missing input. When errors occur, the system has to be able to identify these errors and to develop strategies to solve them. As presented in [75], a robot that shows signs of incompetence, looses significantly the human's trust in the system which leads to a decrease in the efficiency of the overall collaboration performance.

Robustness also includes the use of natural cues as robust input modalities. Natural does not always mean, that control via speech is the best choice. Natural means that the human should be allowed to use the communication strategies that are known and trained for a specific case. Alternative modalities include drawing a sketch and using simple gestures if they fit [66]. This way, the input to the system becomes more natural and more robust.

Although it seems, that robustness is mostly a matter for input processing modules that have to interpret the robot's environment in a stable way, the robot architecture itself can also contribute to increase the overall robustness of the system. For this, the architecture has to guarantee the transmission of information in the data flow as well as to provide mechanisms that help sub-components of the architecture to recover in case of breakdowns [74].

Fast reaction time

Fast reaction time of a robotic system relates in this work to the time needed to react in different situations [74]. This also includes the fast reaction on input signal as e.g. speech commands or changes in the environment. Consequently, interaction capabilities of a system are highly related with the ability react and to transmit information. The interaction will suffer, if this transmission is designed inappropriately or disturbed by delays, jitter, noise, and so on [76]. This also includes non-verbal aspects, which trigger unconscious adaption as shown in Section 4.2, where the choice of the motion profile of the robot has influence on the reaction time of the human. That means, that also the way robots move communicates important information to the user and can be used—if the correct information is transmitted—to increase the collaboration efficiency.

For a goal-oriented task, timing is a crucial point that can confuse the interaction [76]. Section 4.3 shows the positive effects if the timing is adapted towards the human. With an evaluation of timing the fluency and therefore the performance of the collaboration between human and robot can be improved.

Robots have to process and react fast to input from their environment. This has mainly two reasons: on the one hand, humans will only accept robots if they are reacting fast to what they say or do. Most confusion regarding the interaction between human and robot arises when the robot is needs too much time to react to the human's utterances or does not react at all. On the other hand, the robot needs to quickly detect dangerous situations for the human. Hence, the robot architecture is important for fast reaction times in two ways:

1. Without a suitable infrastructure even fast system sub-components cannot contribute to a fast reaction due to e.g. latency in the communication. The architecture has to take care that information between components flows fast and reliably.

2. The architecture has to provide dedicated fast processing channels for security-related parts of the system. For example, the architecture needs a specially designed connection for robot control. Over this connection the robot can be stopped at all times. This can be done for example from outside the system by pressing an emergency button or by dedicated input processing modules. For instance a specialized module that measures the loudness of the human utterances could stop the robot in case the loudness reaches a certain threshold, which would indicate an emergency situation.

Context awareness

In order to increase the safety for the human, the robot has to react to unknown situations fast and also with a reaction that takes into account the partially available context of this situation as fast as possible. This means that the robot has to *understand* the current situation rather than to just react on signal input. Also, only with a reasonable fast reaction time of the robot the human can judge the robot's actions, which also inherently increases the safety for the human.

Therefore, robotic software architectures need to be designed with these principles in mind and to combine reaction-based methods (for *short-term goals*) with high-level methods for reasoning (to reach *long-term goals*). Reaction-based or low-level methods can be seen as part of the embodiment movement as described in [77], whereas high-level methods are part of more traditional artificial intelligence (AI). This mixture of new and old approaches is also in agreement with [78], that argues that an architecture for a truly cognitive technical system has to combine methods from embodiment and traditional AI, which also shows similarities to how human perceive and reason about their environment [74].

When robots process input of sensor modules, there will be inevitably situations in which the input is ambiguous or where not enough input information is available to compute a complete hypothesis of which action to execute next. In these situations, robots have to make use of context information. Therefore, robots need suitable representation formats for the knowledge about the world and defined interfaces to the software modules that provide the information about the surrounding.

Context awareness has to be a built-in feature of a robot architecture in order for the robot to interact with its environment in a reasonable and safe way. Therefore, context awareness should be included in two ways:

1. The architecture provides the infrastructure for the input modules so that they can publish their recognition results system-wide—and thus in a sense *generate* the context for the robot.

2. Since the safety of the human co-worker of the robot is also part of the context, in which the robot is working in, the architecture should have built-in mechanisms to increase the safety for the human.

2.4.2 Robotic middlewares

Since the application field of robots evolve towards the usage in the *real world* with many uncertainties, dynamics, and unstructured environments, computational needs, inter-module, inter-computer, and even inter-programming language and inter-operating system become relevant constraints that need to be fulfilled to solve the high task complexity and therefore the computational demands. This is also followed by the need to integrate a high number of hard- and software modules that form the complex robotic system.

Therefore, high granularity of modules that communicate with each other via the same interfaces and data structures enable fast implementation of new applications, improve flexibility, maintainability, and exchange of modules [79]. In the area of robotics, many approaches exist to create this kind of *glue* between modules and hardware components with so called robotic middlewares. One of the major problems of most of the currently available robotic middlewares is the not adequately solved issue about security mechanism within the communication [80].

Many robotic middlewares were created in the past. Well-known middlewares are among others [80, 79, 81]:

- Orca[1] is an open-source framework for developing component-based robotic systems based on *ICE* the *Internet Communications Engine*[2] as middleware backbone. The component act stand-alone and communicate via defined interfaces.
- Player[3] provides a distributed access to a variety of robots and sensor hardware. The information sharing takes place via client-server connections.
- OpenRTM[4] is a robotic technology middleware developed and distributed by Japan's National Institute of Advanced Industrial Science and Technology to build robots and their functional parts in a modular structure.
- YARP[5] is middleware that clearly decouples modules and devices in a distributed manner.
- At the moment, the recently introduced *Robot Operating System* (ROS)[6] by Wil-

[1] http://orca-robotics.sourceforge.net/

[2] http://www.zeroc.com

[3] http://playerstage.sourceforge.net/

[4] http://www.openrtm.org/

[5] http://eris.liralab.it/yarp/

[6] http://www.ros.org

low Garage[1] is very popular in the robotic community with an increasing number of modules available. As ROS provides a software management and design tool chain, the integration of new and/or own modules eases up.

[1] `http://www.willowgarage.com`

Chapter 3

Action observation

Contents

This chapter sets the basis of transferring the psychological mechanism action observation *of joint action to a robotic system. This mechanism enables an assistive robotic system to predict the next goal based on the current behavior of the opponent to assist with pro-active behavior. Hence, the system needs to recognize the current action of the human using captured information form sensory data. An exemplary assembly situation was designed and afterwards executed by multiple subjects (Section 3.1). The recorded data was used to train models that analyze the workflow of the assembly.*

The experiments revealed that it is sufficient to use only information derived from the hand position. Therefore, a hand tracking method was developed (Section 3.3) using model-based visual tracking (see Section 3.2).

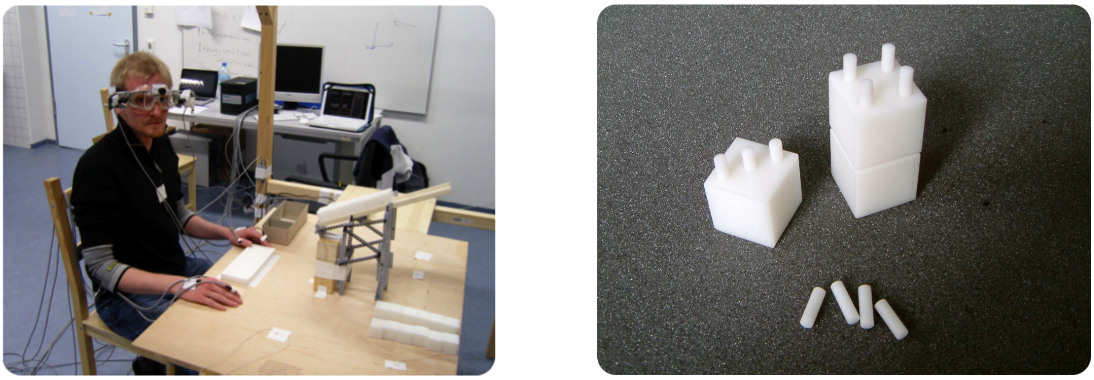
3.1 Workflow analysis of assembly tasks

Dynamic workflows with pro-active support of the robotic assistive system can only be achieved, if the system is able to *recognize* the actions of the human counterpart. In an assembly task, the sequence of actions—i.e. the assembly workflow—underlies certain constraints. Although, there are multiple ways to assemble a specific product, the number of ways is limited and the goal is unique. Therefore, if an assistive system knows, in which state of assembly the human is acting, future steps can be pro-actively prepared in advance.

In the domain of modeling and monitoring actions in surgeries, workflow analysis has been successfully employed [82, 83, 84]. Context-aware operating rooms are able to assist the surgeon with context-sensitive user interfaces [83]. In the area of robotics, the work of [85], for example, focuses on learning by demonstration and to replay demonstrated actions. Since, related work proofs that Hidden Markov Models (HMMs) can be successfully employed, the method to analyze the collaborative assembly tasks between human and robot also bases upon HMMs.

3.1.1 Experimental design

In an experiment originally designed to investigate the timing of actions [87, 86], 25 subjects were instructed to assemble a tower by combining six cubes with several bolts. Each cube features one to five holes on two opposing sides as shown in Fig. 3.1(b). With the number of bolts needed to stack two cubes, the complexity of the assembly step varies. As depicted in Fig. 3.1(a), subjects were sitting on a desk and had to build towers upon a board. In total, every subject mounted 6 towers in a row. A box containing the bolts was positioned to the left. Cubes were available to the human from a slide placed in a way that the foremost cube laid at roughly the handover-position of an imaginary cooperation partner [87]. The sequence of the cubes on the slide with respect to the number of holes was varied among the persons. The number of holes of two subsequent cubes always matched each other. Furthermore, the board in front of the subjects initially contained the correct number of bolts for the first cube. That way,



(a)

(b)

Figure 3.1: Baja experimental set-up - Subjects assemble a tower by combining six cubes (b) provided by a cube vendor in front of them with several bolts taken from a box on their left (a). During the experiment the position and orientation of thumbs, forefingers, back of both hands, head, torso, and gaze was recorded [86]

the assembly task incorporated the taking of six cubes and the connecting with bolts for five times.

The movements of the subjects were recorded by a Polhemus Liberty tracking device[1] which measures the position and orientation of eight sensors at a sample rate of 240 Hz. The sensors were attached to the thumbs, the forefingers, the back of both hands, the head, and the torso. Moreover, the intersection of the person's gaze with the table was recorded with the eye-tracking device "EyeSeeCam" [88].

The input data was smoothed and the numeric approximations of velocity, acceleration, and jerk were computed. Since the usage of pure positional data did not satisfy the demand to generalize the model for the task, because small changes in the set-up let pure positional trained models easily fail, the positional data is only used to activate different table zones. The table zones were defined around the cube slide, around the box of bolts and around the assembly place. With the application on the *JAHIR* set-up in mind this enables high flexibility regarding the arrangement of the setting while being able to use the same trained models. Additionally, flexibility is gained in the sensory input. With an adaption of the update rates and measuring units, the same models can be used to analyze the same task in slightly changed set-ups.

[1] `http://www.polhemus.com/?page=Motion_Liberty`

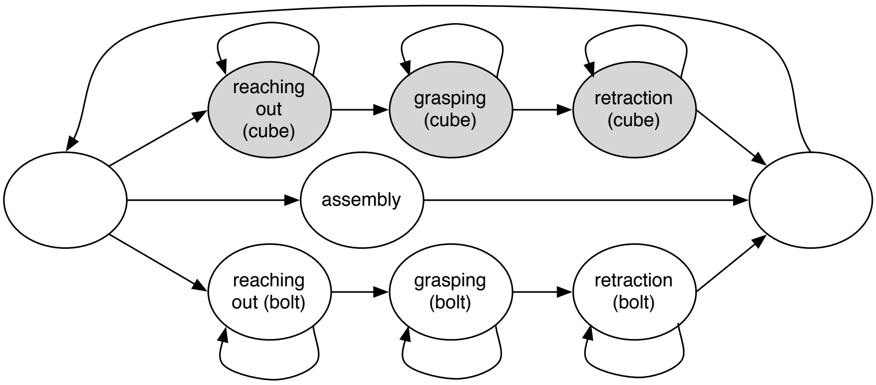


Figure 3.2: Composite HMM - Individual continuous HMMs are trained and connected in a composite HMM for each hand. Grey action HMMs are only available for the right hand model [90]

3.1.2 Structure of the workflow models

The tower assembly task of the experiment can be divided to seven different actions that need to be recognized by the system:

1. reaching out for a cube,
2. grasping a cube,
3. retraction of the cube for the assembly,
4. reaching out for a bolt,
5. grasping a bolt,
6. retraction of the bolt,
7. and performing the assembly itself

For each of the seven actions a left-to-right continuous HMM using the Baum-Welch algorithm [89] has been trained. Experimental results revealed that six states with skipping transitions per model and two normal distributions for each state are appropriate for all individual action HMMs. The skipping transitions allow that unnecessary states fall out during the training phase. Hence, the number of states for each action ranges between three and six states.

These action HMMs are then connected by means of a grammar and form the workflow analysis composite HMM as depicted in Figure 3.2. Since the left and the right hand act mostly in parallel and lead to a large number of movement combinations, a composite HMM per hand was used instead of a single composite HMM covering both hands. The grammar allows taking cubes for the right hand, taking bolts and assembling for both hands to follow in an arbitrary manner. It restricts the three minor movements of taking a cube to succeed in the correct order. The same holds with respect to taking a bolt.

3.1.3 Applying the workflow models

The experiment has been accomplished with 22 subjects using a 11-fold cross-validation. That means, the HMMs were trained on 20 persons and tested on the remaining two ones. Three out of the total number of 25 subjects performed the assembly task incorrectly and were excluded from the dataset. Although each person assembled six towers, a closer look into the data sets revealed that some subjects dropped bolts near the cube slide, wrongly decided to take a cube, or stopped the execution of movements in the middle. These wrong trials were also excluded as training sequence to have only correct motions for training the single action HMMs. But at least three towers were always assembled per subject. The testing was done in a single run on the data set.

Different permutations of the available sensor data were individually investigated in nine experimental data sets as shown in Table 5.1. The data of the torso sensor was never taken, because the subjects hardly moved due to the stationary sitting position in front of the table. In the experiments the accuracy of the workflow recognition was evaluated. The accuracy was defined as the percentage of correct recognized action labels. Further, if the recognized label is in the same compound action (see Section 3.1.2), it is also judged as being correct.

The usage of all available input data (table zones, gaze data and velocity, acceleration and jerk of head, thumbs, forefingers and back of hands) results in a 70 dimensional feature vector. With all information available, an average accuracy of $(95.67 \pm 5.07)\,\%$ was achieved for the right hand and $(87.68 \pm 5.32)\,\%$ for the left hand. Although the table zones seem to describe the task workflow sufficiently, the recognition results using only the table zones are very low with $(56.27 \pm 12.64)\,\%$ for the left and $(74.18 \pm 14.23)\,\%$ for the right hand. By reducing the data in the feature vector to the table zones and the velocity, acceleration and jerk of the back of both hands, the average accuracy scores high with $(95.11 \pm 5.20)\,\%$ for the right hand and $(83.48 \pm 7.39)\,\%$ for the left hand. Compared to the results with full dimensionality, these values are not remarkably lower. Comparable results can also be reached by using all data except the table zones ($(87.66 \pm 4.67)\,\%$ for the left hand and $(90.61 \pm 3.46)\,\%$

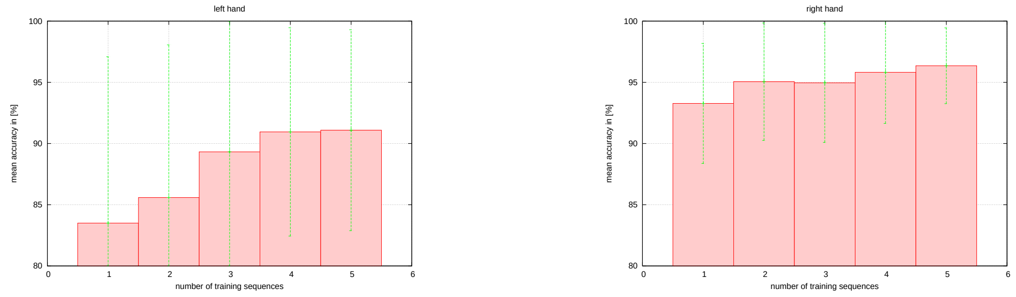
Table 3.1: Recognition results for different data sets - The table shows the accuracy of the workflow recognition for the left and the right hand along with the used data sources and the corresponding dimensionality of the feature vector. The standard deviation results from the 11-fold cross-validation [90]

data set	**dimensions**	**accuracy (%)**	
		left hand	**right hand**
all data	70	87.68 ± 5.32	95.67 ± 5.07
gaze+hands+fingers+head	65	87.66 ± 4.67	90.61 ± 3.46
table zones	5	56.27 ± 12.64	74.18 ± 14.23
table zones+gaze	7	56.15 ± 11.12	78.26 ± 13.48
table zones+head	14	46.36 ± 10.78	75.17 ± 19.60
table zones+gaze+head	16	52.50 ± 9.02	75.56 ± 11.03
table zones+hands	23	83.48 ± 7.39	95.11 ± 5.20
table zones+fingers	41	85.38 ± 6.47	95.86 ± 4.79
table zones+hands+fingers	59	87.08 ± 5.18	95.94 ± 5.22

for the right hand). This indicates, that the different data sources incorporate redundant information.

Further, the results show that it is sufficient to focus on the hands to be able to reconstruct the actions of the human. Although robustness might get lost due to the lack of redundant information, a more important aspect is gained: flexibility. The task is abstracted from high input dimensionality and the experimental installation. Additionally, this finding enables the transfer of the trained models to the *JAHIR* set up as shown in Section 5.3. All tested data set permutations can be found in Table 3.1 along with the recognition accuracy for both hands.

Additionally, an experiment was performed to show the adaption capabilities of the model. 12 subjects with complete data sets—i.e. six towers were assembled—were chosen for the experiment using the table zones plus hand data as input. Each subject was trained independently using one to five complete tower buildings and tested on the remaining five to one sequences. Figure 3.3 shows the resulting accuracy and corresponding variance for the experiments. The 12 single recognition results show in average that the accuracy increases with the number of training sequences while the variance in the resulting accuracy reduces. The results show, that the system is able



(a) (b)

Figure 3.3: Adaption of HMM - This diagram shows the mean accuracy for 12 subjects (red) and the corresponding standard deviation (green bars). Each subject was trained independently using one to five complete tower buildings and tested on the remaining five to one sequences

to estimate the workflow for the applied assembly task with 83.5 % for the left and 93.28 % for the right hand after the first run and reaches an average accuracy of 91.1 % for the left and 96.35 % for the right hand after 5 assembly sequences. Hence, with a well-defined structure of possible basic actions of the human, the system is able to recognize the succession of actions after only few training sequences.

3.2 Model-based visual tracking

To observe actions of humans, the robotic system needs have certain perception capabilities at best without augmenting the human. An efficient way to do that is to use models that describe and encapsulate a-priori knowledge that is "useful" to solve the perception task. The results of the workflow analysis experiments in Section 3.1.3 show, that it is sufficient to focus on information derived from the hand position for the presented assembly task. Hence, a model-based visual tracking approach was chosen to robustly estimate the hand position.

Model-based visual tracking deals with the problem of localizing one or more objects in a sequence of images employing computer vision techniques along with a-priori knowledge about the object and knowledge from past estimations. Models are used to form this a-priori knowledge and describe the *acquisition* of the visual data (Section 3.2.1), the *degrees of freedom* (Section 3.2.2) and the expected *motion* of the

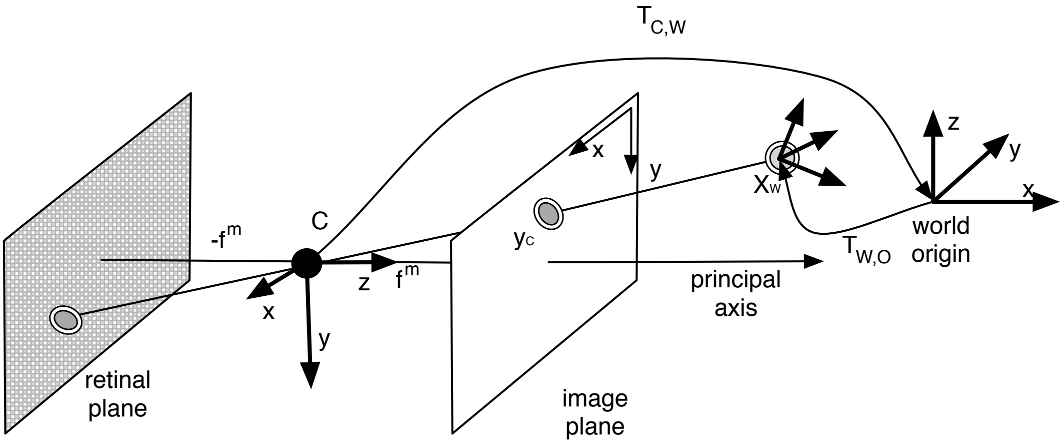


Figure 3.4: The pinhole camera model - The scheme depicts the projection of a point X_W described in the world coordinate frame onto the image plane of a pinhole camera (y_C)

target object (Section 3.2.3). Further, prior information may consist of shape, appearance, deformation parameters, kinematic structure, as well as any useful information about sensors and context that may be specified in advance or refined during the task itself. The sequential estimation is then done by applying Bayesian tracking schemes (Section 3.2.4).

The background knowledge of model-based visual tracking presented in this Section follows the book [91] about this topic and the OpenTL[1] software framework, which was co-developed and supported by the work performed throughout this thesis. OpenTL is a hierarchical, object-oriented general-purpose software library written in C++ for visual tracking tasks. With this generic software framework as backbone, the approaches are applicable to many other scenarios by e.g. re-parameterizing or the exchange of a specific model (e.g. the object to be tracked).

3.2.1 From world to camera space: the projection model

Visual tracking is mostly done using cameras as input sensors. Cameras map the three-dimensional world to two-dimensional views through their optics on a chip area. To track three-dimensional objects or objects that move in the three-dimensional space, the mapping between world and camera needs to be applied to know how these objects look in specific camera views.

[1] http://www.opentl.org

To describe the mapping of points (X_W) given in a three-dimensional metric world space to the two-dimensional pixel based screen space of a given camera (y_C) [92] as depicted in Figure 3.4, we need the acquisition model of the camera (*intrinsic parameters*) and the translation $t_{C,W}$ and orientation $R_{C,W}$ of the camera to the world origin (*extrinsic parameters*) denoted by

$$T_{C,W} = \begin{bmatrix} R_{C,W} & t_{C,W} \\ 0 & 1 \end{bmatrix}, \tag{3.1}$$

resulting in a 4 × 4 transformation matrix.

If cameras have negligible lens distortions or undistorted images, the projection into screen coordinates can be described by the *intrinsic* projection matrix of the camera

$$K = \begin{bmatrix} f_x & \sigma & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{3.2}$$

where f_x and f_y are the focal lengths in x- and y-direction. The metric focal length f^m given in meter is normalized with the size of the pixels (p_x^m and p_y^m) also given in meter to define the unitless focal length in horizontal and vertical direction

$$f_x = \frac{f^m}{p_x^m} \tag{3.3}$$

$$f_y = \frac{f^m}{p_y^m}. \tag{3.4}$$

c_x and c_y are the pixel location of the principle point of the camera, which is the intersection of the optical axis with the camera plane (*retinal plane*). The principle point can be expressed in pixels by normalizing the actual chip size (c_x^m and c_y^m) with the size of the pixels:

$$P_p = (c_x, c_y) = \left(\frac{c_x^m}{p_x^m}, \frac{c_y^m}{p_y^m} \right). \tag{3.5}$$

The screw factor σ is related to the axis displacement α of the single pixels:

$$\sigma = tan(\alpha) \cdot f_y. \tag{3.6}$$

If the vertical and horizontal axis of the pixels are perpendicular, the screw factor $\sigma = 0$ corresponding to the *pinhole camera model.* This model is often appropriate enough and therefore used in later described applications to describe the intrinsic projection of cameras.

Now, that the intrinsic projection characteristics of the camera is described, the projection of point X_W described in the world coordinate frame to the point y_C in camera space is given by

$$
\begin{aligned}
y_C^h &= K \cdot T_{C,W} \cdot X_W^h && (3.7)\\
y_C^h &= P_{C,W} \cdot X_W^h && (3.8)
\end{aligned}
$$

with $P_{C,W}$ being the cumulative projection that includes the extrinsic and intrinsic mapping. Please note, that the two points y_C^h and X_W^h are given in homogeneous coordinates, where a dimension is added to allow a direct multiplication with the projection matrix. To transform the homogeneous coordinates back to non-homogeneous ones, the non-linear operation

$$
\begin{aligned}
\pi^2(x,y,h) &= \begin{bmatrix} \frac{x}{h} & \frac{y}{h} \end{bmatrix}^T && (3.9)\\
\pi^3(x,y,z,h) &= \begin{bmatrix} \frac{x}{h} & \frac{y}{h} & \frac{z}{h} \end{bmatrix}^T && (3.10)
\end{aligned}
$$

is applied for two-dimensional and three-dimensional points respectively:

$$
\begin{aligned}
y &= \pi^2(y^h) && (3.11)\\
X &= \pi^3(X^h). && (3.12)
\end{aligned}
$$

In model-based visual tracking, complex three-dimensional models are often used to describe the target object as for example an airplane in [93]. These models consist of multiple points described in a local coordinate system of an object (X_O). Hence, to project these points, also the local transformation to the object coordinate system needs to be considered:

$$
\begin{aligned}
y_C^h &= K \cdot T_{C,W} \cdot T_{W,O}(p_O) \cdot X_O^h && (3.13)\\
y_C^h &= P_{C,O}(p_O) \cdot X_O^h. && (3.14)
\end{aligned}
$$

As described in the next section, the transformation of an object in the world can efficiently be expressed by means of pose parameters p_O .

3.2.2 Modeling degrees of freedom of objects using pose parameters

Pose parameters p are a vectorial representation of the degrees of freedom of an object. These parameters control a homogeneous transformation matrix $T(p)$, that may be a sub-group of the general linear group $GL(n)$ of invertible $(n \times n)$ matrices, closed under matrix multiplication [91]:

$$T(p) = \begin{bmatrix} A(p) & t(p) \\ v(p)^T & 1 \end{bmatrix} \tag{3.15}$$

with $v(b)$ being a three-dimensional vector, describing the homographic mapping[1], $t = [t_x, t_y, t_z]^T$ being the translations along the coordinate axis, and $A(p)$ being a generic 3×3-matrix representation for linear transformation:

$$A(p) = R(p) \cdot R_s(p)^{-1} \cdot S(p) \cdot R_s(p). \tag{3.16}$$

$S(p) = diag\,(s_x(p), s_y(p), s_z(p))$ defines the scaling transformation, $R_s(p)$ defines the rotation into the coordinate frame where the scaling shall take place, and $R(p)$ describes the rotation ($R(p)^T R(p) = I$).

Opposite to a full matrix representation, pose parameters model only the actual degrees of freedom. As an example, consider the Euclidean group $SE(3)$. The motion of a rigid body is representable in this case with the homogeneous transform given by

$$T(p) = \begin{bmatrix} R(p) & t(p) \\ 0 & 1 \end{bmatrix}. \tag{3.17}$$

Since this specification offers 3 rotational and 3 translatory degrees of freedom, the resulting pose parameters are:

$$p = (\alpha, \beta, \gamma, x, y, z). \tag{3.18}$$

If the motion of the object is restricted to have only translations, then the pose reduces to

$$p = (x, y, z), \tag{3.19}$$

corresponding to the transformational representation

$$T(p) = \begin{bmatrix} I & t(p) \\ 0 & 1 \end{bmatrix}. \tag{3.20}$$

The optimization and/or estimation of only *active* parameters of the transformation matrix results in more efficient computation and the ability to describe easily the dynamical updates in a compositional way.

[1] for non-homographic transformations $v(p) = [0, 0, 0]^T$

Since rigid rotations using Euler angles $R(\alpha,\beta,\gamma)$ have multiple values for p that lead to the same rotation, a local parameterization solves the problem to find the corresponding transformation matrix

$$T_t = T_{t-1} \cdot \delta T(\delta p_t). \tag{3.21}$$

This *compositional* update[1] [94] is singularity-free around $\delta T(\delta p = 0) = I$ due to its incremental characteristic. In this way, the tangent space can be expressed in a Lie algebra [95]. The Euclidian group $SE(3)$ (see (3.17)) leads to the parameterization

$$M(\delta p) = \sum_{d=1}^{6} G_d \delta p_d \tag{3.22}$$

where G_d are (4×4) Lie generators for the active translational and rotational motion parameters in local coordinates.

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad G_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad G_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Any matrix in $SE(3)$ can be obtained through the exponential mapping

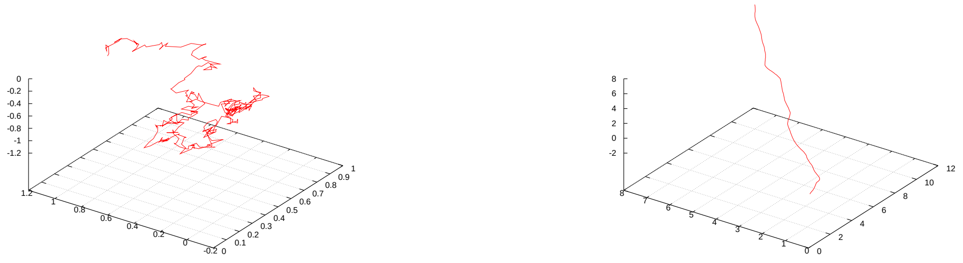
$$\delta T = \exp(M(\delta p)) \tag{3.23}$$

which is singularity-free around $\delta p = 0$ [91].

3.2.3 Modeling object dynamics by auto-regressive processes

If an object should be followed in a sequence of images, useful information is the knowledge about the dynamical behavior of the target object. The better this behavior can be defined in advance, the more accurate is the generation of possible locations of the object for future measurements and the more misleading measurements can be excluded from evaluation.

[1] in comparison to an additive update $p_t = p_{t-1} + \delta p_t$ performed in absolute parameter space



(a)

(b)

Figure 3.5: Different motion models - The diagrams show the three-dimensional motion trajectory of an object that moves with (a) Brownian motion, and (b) with a constant velocity white noise acceleration model (CWNA)

Since the precise object dynamic is often unknown, it can be approximated by *auto-regressive* (AR) processes ($p_k = F^1 p_{k-1} + F^2 p_{k-2} + \ldots + W^0 w_k$) [91]. The past is propagated using time dependent matrices for the transition of the past (A_t), time dependent noise gain (B_t), and the unit covariance w_t

$$s_t = A_t s_{t-1} + B_t w_t \tag{3.24}$$

with $s_t = [p_t, \dot{p}_t, \ldots]$, in terms of the pose parameters p_t and the corresponding time derivatives. The process noise of the motion Q is given by

$$Q = B_t W_t B_t^T \tag{3.25}$$

where W_t is the noise covariance matrix corresponding to w_t.

A first-order example of such a process where the state only incorporates the pose ($s_t = [p_t]$) is the *Brownian* motion. Figure 3.5(a) depicts the generated brownian trajectory of an object with three degrees of freedom (see (3.19)). The transition matrix and the noise gain matrix for this motion is given by

$$\begin{aligned} F &= [I] && (3.26) \\ A_t &= [I] && (3.27) \\ B^0 &= [I] && (3.28) \\ B_t &= \left[W^0 \tau\right], && (3.29) \end{aligned}$$

with $\tau = t - (t-1)$ being the time lag between the current state and the past state estimation. This leads to the state prediction

$$s_t = [p_t] = p_{t-1} + w_t. \tag{3.30}$$

A second-order example is given with a constant velocity motion model with white noise acceleration (CWNA). This model uses the first time derivative $s_t = [p_t, \dot{p}_t]$. The process matrices for this kind of motion that is depicted in Figure 3.5(b) are given by

$$\begin{aligned} F^0 &= [2.0I] && (3.31) \\ F^1 &= [-1.0I] && (3.32) \\ B^0 &= [I] && (3.33) \end{aligned}$$

and result in the time dependent matrices

$$\begin{aligned} A_t &= \begin{bmatrix} I & I\tau \\ 0 & I \end{bmatrix} && (3.34) \\ B_t &= \begin{bmatrix} 0.5 \cdot I\tau^2 \\ I\tau \end{bmatrix}. && (3.35) \end{aligned}$$

This leads to the current state prediction

$$s_t = \begin{bmatrix} p_t \\ \dot{p}_t \end{bmatrix} = \begin{bmatrix} p_{t-1} + \dot{p}_{t-1}\tau + \frac{1}{2} I w_t \tau^2 \\ \dot{p}_{t-1} + I w_t \tau \end{bmatrix}. \tag{3.36}$$

Further, with the representation of motions in this way, motion models can also be learned from ground-truth sequences [96].

Since the singularity free compositional update should be used for the state dynamics (see (3.21)), the dynamical model needs to be formulated in terms of δp and its temporal derivatives [97]. Therefore, the state is expressed in an incremental manner $\delta s = (\delta p, \delta \dot{p}, \delta \ddot{p}, \ldots)$:

$$\delta s_t = f(T_{t-1}, \delta s_{t-1}) + B_t w_t \tag{3.37}$$

where the previous incremental state δs_{t-1}, is referred to T_{t-1}, while the noise w_t is additive in δs_t. Although incremental parameters refer to local coordinates, the same dynamical properties of the original model are kept [91]. The AR model formulates in terms of incremental states in the following way

$$\delta s_t = A_t \delta s_{t-1} + B_t w_t \tag{3.38}$$

with a compositional update after each state update on the transformation matrix

$$\begin{aligned} T_t &= T_{t-1}\delta T\left(\delta p_t\right) && (3.39) \\ \delta p_t &= 0. && (3.40) \end{aligned}$$

3.2.4 Sequential estimation with Bayesian tracking schemes

The main advance of tracking systems compared to frame-by-frame detection methods is the integration of past information with current knowledge to perform the state estimation. That means, that the dynamical model with corresponding state hypotheses gets combined with the current measurement z_t. z_t can be instantiated as

- raw sensory data such as an image (*pixel-level measurement*),
- extracted features associated with the target object (*feature-level measurement*),
- or a direct estimation of the object's state (*object-level measurement*).

All past measurements Z^{t-1} are also incorporated in two probabilistic filter stages [98, 99]

1. *Prediction* (Kolmogorov-Chapman equation):

$$P\left(s_t|Z^{t-1}\right) = \int_{s_{t-1}} P\left(s_t|s_{t-1}\right) P\left(s_{t-1}|Z^{t-1}\right) \tag{3.41}$$

2. *Correction* (Bayes' rule):

$$P\left(s_t|Z^t\right) = kP\left(z_t|s_t\right) P\left(s_t|Z^{t-1}\right). \tag{3.42}$$

In the following, concepts and application conditions of several tracking filter following this scheme are described. A general review of Bayesian filters can be found for example in [100] or the OpenTL book [91]. For more detailed descriptions please refer to these references.

Kalman filters

If the object dynamics and the measurement model are linear and Gaussian processes with zero-mean, white Gaussian noise variables

$$s_t = A_t s_{t-1} + B_t w_t \tag{3.43}$$

$$z_t = H_t s_t + C_t v_t, \tag{3.44}$$

the Kalman Filter [101, 91] is the optimal solution and instantiates the prediction/correction scheme as follows

1. *Prediction*:

$$s_t^- = A_t s_{t-1} \tag{3.45}$$

$$\Sigma_t^- = A_t \Sigma_{t-1} A_t^T + Q_t \tag{3.46}$$

2. *Correction*:

$$s_t = s_t^- + K_t \left(z_t - H_t s_t^- \right) \tag{3.47}$$

$$\Sigma_t = \left(I - K_t H_t \right) \Sigma_t^- \tag{3.48}$$

where the *Kalman gain* matrix is given by $K_t = \Sigma_t^- H_t^T \left(H_t \Sigma_t^- H_t^T + R_t \right)^{-1}$, Σ^- and Σ are the prior and posterior covariance matrix, and Q_t and R_t are the overall noise covariance matrices for the motion and the measurement process.

Since the object dynamics and the measurement model are possibly non-linear, but with additive Gaussian noises

$$s_t = f_t \left(s_{t-1} \right) + w_t \tag{3.49}$$

$$z_t = h_t \left(s_t \right) + v_t, \tag{3.50}$$

the assumptions of the standard Kalman filter are no longer valid. An approximated sub-optimal solution can be found, if the dynamic and the measurement process functions f, h can be linearized around the current mean state. Using the corresponding Jacobian matrices F_t and H_t for the dynamic and measurement process, the resulting prediction-correction scheme forms the *Extended Kalman Filter* (EKF) [102, 91]

1. *Prediction*:

$$s_t^- = f_t \left(s_{t-1} \right) \tag{3.51}$$

$$\Sigma_t^- = F_t \Sigma_{t-1} F_t^T + Q_t \tag{3.52}$$

2. *Correction*:

$$s_t = s_t^- + K_t \left(z_t - h_t \left(s_t\right)\right) \tag{3.53}$$

$$\Sigma_t = \left(I - K_t H_t\right) \Sigma_t^- . \tag{3.54}$$

Another solution that can handle non-linear state estimations is the *Unscented Kalman Filter* (UKF) [103, 104, 91]. The UKF statistically approximates the state distribution by weighted state hypothesis around the principal axes of the covariance matrix using an *unscented transform* with so called *sigma points*. In the non-weighted case, *sigma points* can be expressed by

$$s^i = \breve{s} + \left(\sqrt{n\Sigma^s}\right)_i \tag{3.55}$$

$$s^{i+n} = \breve{s} - \left(\sqrt{n\Sigma^s}\right)_i \tag{3.56}$$

$$\pi_m^i, \pi_c^i = 1/2n;\ i = 1, \ldots, 2n \tag{3.57}$$

leading to

$$\breve{s} = \frac{1}{2n} \sum_{i=1}^{2n} s^i \tag{3.58}$$

$$\Sigma^s = \frac{1}{2n} \sum_{i=1}^{2n} \left(s^i - \breve{s}\right) \left(s^i - \breve{s}\right)^T \tag{3.59}$$

and

$$\breve{z} = \sum_{i=0}^{2n} \pi_m^i z^i \tag{3.60}$$

$$\Sigma^z = \sum_{i=0}^{2n} \pi_c^i \left(z^i - \breve{z}\right) \left(z^i - \breve{z}\right)^T$$

for the mean and covariance of the estimated state and corresponding measurement [91]. The derivative for the measurement process needs not to be computed. This makes this kind of filter more robust to noise. The *unscented transform* formulation leads to a linear increase of the number of needed hypotheses $(2n+1)$ with increasing state dimension.

Unfortunately, the standard formulation of the Kalman filters uses a gain to update the posterior distribution. This includes the need to invert the innovation covariance matrix, that is possibly very high dimensional for a large set of features in the measurement process as present in the pixel-level case. This makes them better applicable for a low dimensional measurement space [105, 106, 91]. For higher dimensions, a dual formulation of *information filters*

$$\begin{aligned} Y_t &= (\Sigma_t)^{-1} && (3.61)\\ y_t &= Y_t s_t && (3.62) \end{aligned}$$

avoids the matrix inversion in the update step. An inversion needs to be only computed in the prediction step where the dimension—e.g. the degrees of freedom—are small and with constant size. Additionally, the dual formulation allows with an additive update a sequential data fusion [107, 108] and distributed multi-sensor schemes. The dual formulation can be used with the EKF leading to the *Extended Information Filter* (EIF) [109, 110, 91]

1. *Prediction*:

$$\begin{aligned} s_t^- &= f_t\left(s_{t-1}\right) && (3.63)\\ Y_t^- &= \left(F_t Y_{t-1}^{-1} F_t^T + Q_t\right)^{-1} && (3.64)\\ y_t^- &= Y_t^- s^- t && (3.65) \end{aligned}$$

2. *Correction*:

$$\begin{aligned} Y_t &= Y_t^- + H_t^T R_t^{-1} H_t && (3.66)\\ y_t &= y_t^- + H_t^T R_t^{-1}\left(z_t - h\left(s_t^-\right) + H_t s_t^-\right) && (3.67)\\ s_t &= Y_t^{-1} y_t. && (3.68) \end{aligned}$$

The dual formulation can also be applied to the UKF to instantiate the *Unscented Information Filter* (UIF) [111, 91] leading to the prediction/correction scheme

1. *Prediction*:

$$\begin{aligned} s_t^{i,-} &= f_t\left(s_{t-1}^i\right);\ i = 1,...,2n && (3.69)\\ \breve{s}_t^- &= \sum_i \pi_m^i s_t^{i,-} && (3.70)\\ \Sigma_t^{s,-} &= \sum_i \pi_c^i \left(s_t^{i,-} - \breve{s}_t^-\right)\left(s_t^{i,-} - \breve{s}_t^-\right)^T + Q_t && (3.71)\\ Y_t^- &= \left(\Sigma_t^{s,-}\right)^{-1} && (3.72)\\ \breve{y}_t^- &= Y_t^- - \breve{s}_t^- && (3.73) \end{aligned}$$

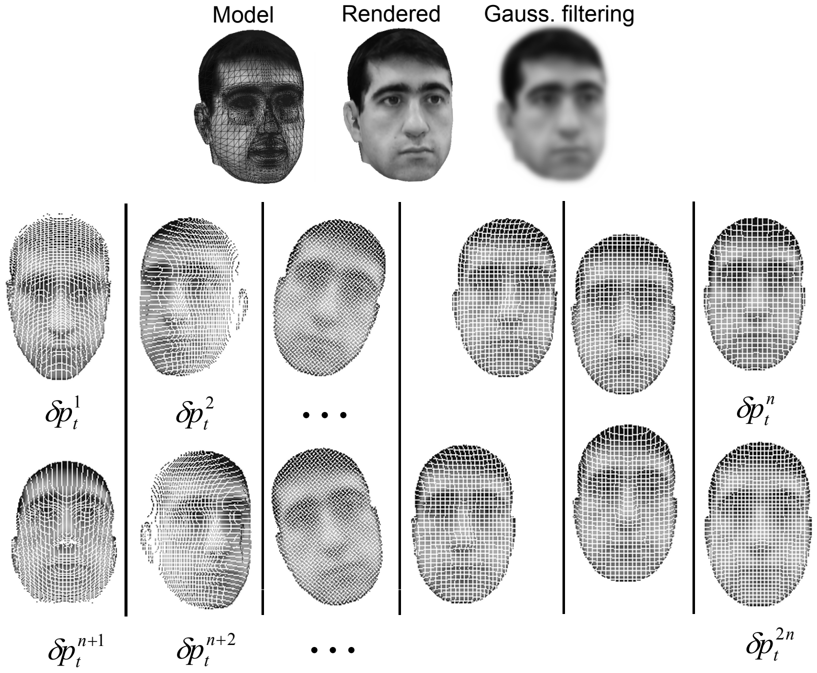


Figure 3.6: Lagrangian approach to template tracking - a sparse set of points, sampled on a rendered image and back-projected in 3D at the average state, and possibly at different resolutions, are re-projected at different hypotheses (sigma points) [112]

2. *Correction*:

$$
\begin{aligned}
e_t^i &= z_t - h\left(s_t^{i,-}\right);\ i = 0, ..., 2n && (3.74)\\
\breve{e}_t &= \sum_i \pi_m^i e_t^i && (3.75)\\
\Sigma_t^{sz} &= \sum_i \pi_c^i \left(s_t^{i,-} - \breve{s}_t^-\right)\left(\breve{e}_t - e_t^i\right)^T && (3.76)\\
\Sigma_t^z &= \sum_i \pi_c^i \left(e_t^i - \breve{e}_t\right)\left(e_t^i - \breve{e}_t\right)^T + R_t && (3.77)\\
\mathcal{H}_t^T &\equiv Y_t^- \Sigma_t^{sz} && (3.78)\\
\mathcal{R}_t &\equiv \Sigma_t^z - \mathcal{H}_t \Sigma_t^{s,-} \mathcal{H}_t^T && (3.79)\\
Y_t &= Y_t^- + \mathcal{H}_t^T \mathcal{R}_t^{-1} \mathcal{H}_t && (3.80)\\
\breve{y}_t &= \breve{y}_t^- + \mathcal{H}_t^T \mathcal{R}_t^{-1}\left[\breve{e}_t + \mathcal{H}_t \breve{s}_t^-\right] && (3.81)
\end{aligned}
$$

A face-tracking example shortly presents and validates three presented Bayesian filter instantiations. The face-tracking example was chosen due to the motivation given

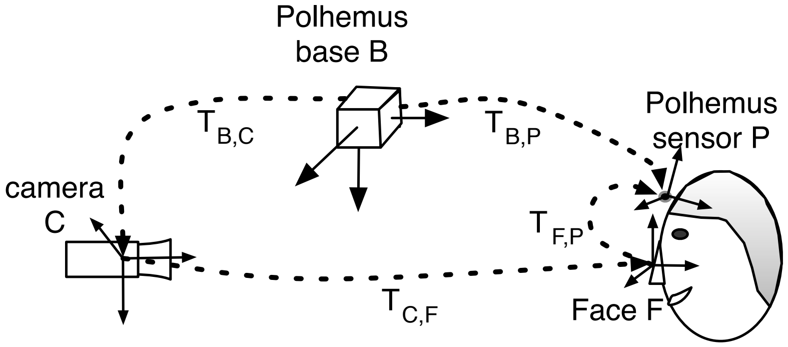


Figure 3.7: Face-tracking set-up - Synchronized ground-truth data were recorded from a magnetic sensor (Polhemus) [112]

in Section 2.3 about the psychological aspect *joint attention*. *Joint attention* of an assistive system and the human cooperation partner is important for an efficient joint-action [16] to share representations about events and objects, to steer the attention, and to estimate the focus-of-attention. Behavioral studies show that the head orientation is directly connected to the related focus-of-attention [113].

The three different Bayesian tracking approaches were tested and evaluated on real video sequences. As basis, texture templates as shown in Figure 3.6 are used along with

1. a least-square optimization of feature residuals in state-space integrated in a standard Kalman filter,

2. a feature-level Extended Information Filter using Lie algebras,

3. and the feature-level and incremental state based Unscented Information Filter

to track the face. A more detailed description with more experiments can be found in [112].

One of the crucial issues in order to evaluate and benchmark the tracking results is the generation of ground truth data. Therefore, a set-up was built up as depicted in Figure 3.7, using a commercial magnetic field tracker as ground-truth estimator (Polhemus Liberty[1]). A 6-dof sensor was taped to the subjects' forehead. With an update rate of $240Hz$, the system delivers position and orientation of the sensor, with respect to the base, that is denoted by $T_{B,P}$.

[1] `http://www.polhemus.com/?page=Motion_Liberty`

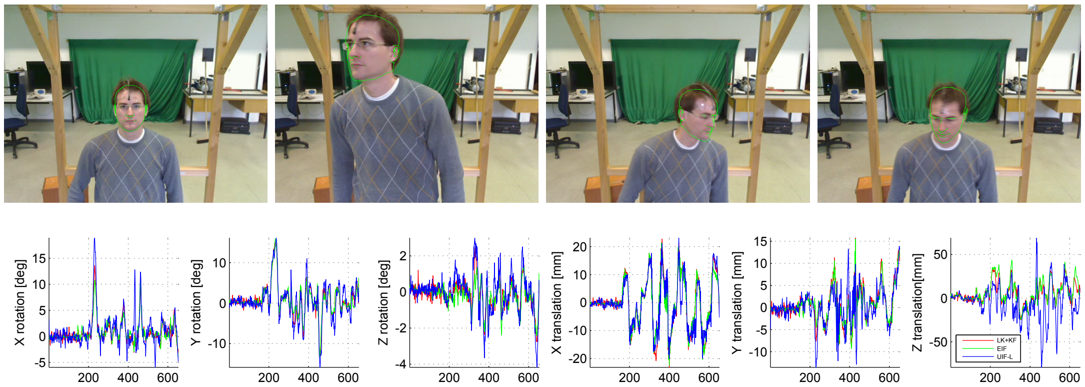


Figure 3.8: Face-tracking in 3D with CWNA motion model and two-resolution - Position errors (absolute parameters) are given by a comparison of estimated poses with ground-truth data [112]

Simultaneously, a calibrated USB camera recorded video data at 30 fps that are subsequently synchronized to the magnetic tracker data. Moreover, the position and orientation of the Polhemus base was calibrated with respect to the camera, $T_{C,B}$ by acquiring a small set of images, where the position of the Polhemus sensor was annotated by hand, while its 3D position with respect to the base was given by the last column of the $T_{B,P}$ matrix. By assuming a perfect estimation of $T_{B,P}$ and noisy image measurements, a standard pose estimation procedure can be applied [114] to compute $T_{C,B}$.

The next step in order to obtain the face-to-camera ground-truth was to compute the face-to-sensor transformation, $T_{F,P}$. This has been be done by manually aligning the first, frontal pose of the subject in camera space, $T_{C,F}$; that is also the pose at which the texture is acquired, and therefore corresponds to a perfect matching, with zero image residuals. Together with the knowledge about the sensor pose at this frame, the wanted transformation can be computed by

$$T_{F,P} = T_{C,F}^{-1} \cdot T_{C,B} \cdot T_{B,P}. \tag{3.82}$$

More accurate calibration procedures could also be applied to this problem, such as the method presented [115] for computing $T_{F,P}$, followed by averages on the Euclidean space [116] to compute $T_{C,B}$. However, this requires an accurate labeling of multiple 3D camera poses, which is difficult for a monocular system especially in the depth direction.

Table 3.2: Tracking results for face-tracking - Position RMS errors for the face-tracking sequence shown in Figure 3.8) [112]

	LK+KF	EIF	UIF-L
X rotation	2.46470	2.19878	3.41815
Y rotation	3.89136	3.65396	4.56358
Z rotation	0.75107	0.70378	1.07299
X translation	9.05872	8.31842	8.46676
Y translation	4.01319	3.64263	4.03052
Z translation	12.90897	14.93438	19.54911

Afterwards, each ground-truth transform k can be computed by

$$T_{C,F}^{true} = T_{C,B} \cdot T_{B,P}(k) \cdot T_{F,P}^{-1}. \tag{3.83}$$

Figure 3.8 shows the resulting position errors for the three filter used to track face on three sequences along with exemplary screenshots from the EIF filter. For these sequences, a CWNA model as described in Section 3.2.3 has been used in order to cope with variable motion of the head. Moreover, the sample set of visible control points has been re-collected at each frame, due to the fact that the head is a highly non-planar model. Table 3.2 resumes the average position errors. All filters have performed well on the tested sequences with robust results regarding fast motion and blurred images, as well as partial occlusions.

Monte Carlo filters

For non-linear, non-Gaussian processes and multi-modal distributions including pixel-level measurements, where Jacobian matrices are not available or too costly to compute, Monte Carlo or particle filters [117, 118] can be applied, where the state statistics are represented by a set of N weighted particles

$$P\left(s_t|Z^t\right) = \{s_t^n, \pi_t^n\}\,;\; n = 1\ldots N \tag{3.84}$$

where $\sum_n \pi_t^n = 1$. One well-known specification is the Sampling-Importance-Resampling (SIR) scheme [119]:

1. Sample from previous posterior: $s_t^n \sim P\left(s_t | s_{t-1}^n\right)$. This means, that the dynamic model as described in Section 3.2.3 is applied to all N state hypotheses to generate the prior.

2. Weight the particle according to the likelihood of the measurement $\pi_t^n \propto P\left(z_t | s_t^n\right)$ and normalize the weights, so that $\sum_n \pi_t^n = 1$.

3. Re-sample the particles $s_t^{n'} \leftarrow s_t^n$ with n' randomly selected according to $\{\pi_t^n\}$ and reset the weights to $\pi_t^n = 1/N$.

To track multiple objects at the same time with only one filter the SIR scheme can be extended to particles with multiple object hypothesis (MOSIR) [120]. Therefore, one particle contains a set of $i = 1 \ldots I$ targets and forms a complete hypothesis scene $\mathbf{s}_t$.

$$\begin{aligned} P\left(\mathbf{s}_t | Z^t\right) &= \left\{\{s_{t,i}^n\}_{i=1}^I, \pi_t^n\right\}_{n=1}^N && (3.85) \\ &= \{\mathbf{s}_t^n, \pi_t^n\}_{n=1}^N . && (3.86) \end{aligned}$$

The weight π_n of each particle $n = 1 \ldots N$ is computed by comparing its hypotheses scenes $\mathbf{s}_t^n = \{s_{t,i}^n\}_{i=1}^I$ with the current measurement z_t of the scene. Here, target interactions can be modeled with an appropriate likelihood model.

Unfortunately, the computational costs increase exponentially if many objects and many object dimensions are involved. Therefore, a particle filter approach based on the Monte Carlo Markov Chain (MCMC) principle can be used to efficiently handle multiple targets, target interactions, and uncertain data association for non-Gaussian processes [121]. At each frame iteration the MCMC particle filter generates a new particle set forming a Markov chain for estimating the posterior state. For generating the chain from an initial seed particle state the Metropolis-Hastings rejection-sampling algorithm can be used.

The MCMC approach uses only equally weighted particles during the sampling step to approximate the posterior distribution. In order to approximate the posterior distribution from a more stationary distribution, the MCMC filter uses additional burn in iterations (burn in samples) that are sampled in each iteration but they are not involved for estimating the posterior distribution. Since the prior distribution models all targets simultaneously, the computational costs are dependent on the number of particles and the number of concurrent targets tracked [122].

The MCMC was used for example used in [123] to track multiple persons on an experimental area of approximately 100 square meters arranged and furnished with a kitchen and a living room [124] with 40 GigE cameras. Given this large amount of cameras, distribution of the computational processing among multiple computers is required, which is addressed using 14 disk-less client processing nodes operating up to three cameras each. Therefore, a management of target detection, target tracking and target transfer between processing nodes was developed accordingly.

3.3 Hand tracking using pixel-level likelihoods

The communication among humans incorporates many non-verbal aspects such as the usage of dynamic and static hand gestures [125, 126, 127, 128, 129, 130]. Thus, the determination of the hand position(s) and further the continuous estimation of the hand motion is a very crucial step. It is essential not only in the gesture recognition domain, but also to coordinate appropriate actions for the robot (see Chapter 4) and to recognize the current actions of the human as presented in Section 3.1. Therefore, this section presents two approaches to determine the hand position of a human (3.3.1 and 3.3.2).

In principle, the tracking of hands can be divided in approaches that work in the 2D image space or in the 3D space. [131] gives a good overview of approaches for both "worlds" and names contours [132, 119, 133], silhouettes [134, 135], fingertips [136], colors [137, 138, 139], or a combination of multiple cues [140, 141, 142] as the main features used for 2D hand tracking. The tracking of hands in 3 dimensional space has the advantage to directly provide position information in global coordinates. Complex approaches even estimate the articulation of the fingers (e.g. [105]). Such approaches lead to high computational complexity and are not applicable to be used on-line today.

3.3.1 GPU supported particle filtering

Many examples of particle filters for single and multiple object tracking are well-known in the literature. Most of these systems use *feature-level* likelihoods, where model features including color statistics [143] or contour points [119] are selected and matched with the current image under a given state hypothesis, applying diverse likelihood models. In particular, these filters achieve different degrees of precision and robustness, as well as tracking speed, according to the specificity of the feature being

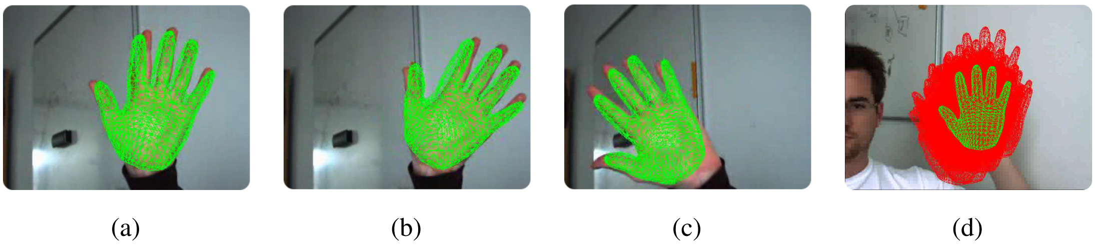


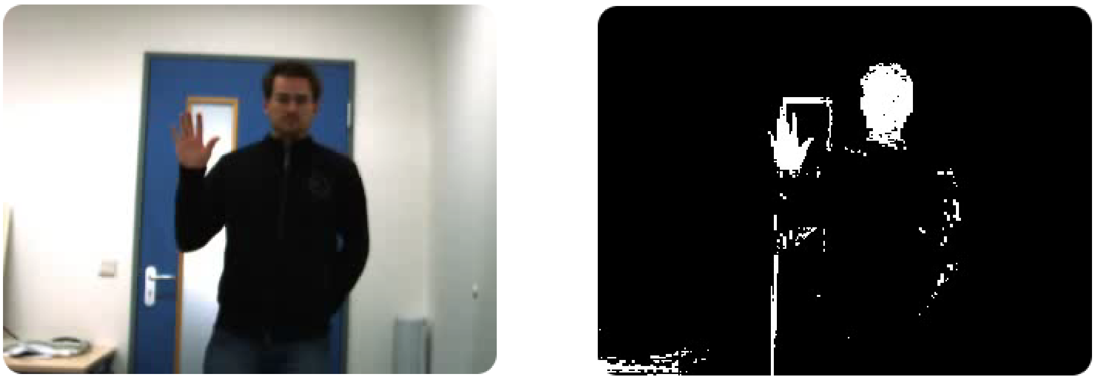
Figure 3.9: Hand tracking using a 3D mesh model - The screenshots from a video sequence show the tracking results using a particle filter framework. By applying a pixel-level likelihood the rotations of the hand can also be extracted (a-c). The estimation of the hand is approximated by a weighted average of the distributed particle set (d) [135]

measured. For real-time applications, simple statistics including color histograms are usually preferred [144, 145, 143], and provide good results for 2D problems where a precise shape localization is not required: usually a rough estimation of translation and scale parameters is obtained by using rectangular or elliptical models.

[135] presents a generic particle filter approach to do the tracking directly on pixel-level supported by the graphics hardware with generic object shapes including the possibility to estimate planar rotations. The basis of the particle filters used is given in Section 3.2.4. This approach is well suited to be used for hand tracking tasks with both an elliptic shape approximation or complex mesh models of the hand.

As the evaluation on pixel-level involves computing and matching the filled *model silhouette* for each pose hypothesis, it can be extremely time-consuming for generic object shapes if performed on the CPU. The time consuming computational steps are shifted to the graphics hardware, where the execution time is much faster due to independent parallel rendering pipelines. A well-known bottleneck of current GPUs is given by the back-transfer of the data on the main bus. Hence, the pixel-level likelihood are also computed “on-board” and only the matching results are transferred back to the CPU memory, in order to update the particle weights.

Without loss of generality, a skin-color segmentation in the HS color space is considered as input source to compute the pixel-level likelihood, as skin-color is a robust and distinct feature to determine hand and face regions of humans. Gaussian Mixture Models (GMMs) have been widely used to do foreground segmentation [146, 147, 148, 149, 150], because of their efficient training and evaluation procedures. The separation of pixel luminance from the pure color channels (hue-saturation) pro-



(a) (b)

Figure 3.10: Skin-color segmentation - An input video frame (a) and the corresponding segmented image using a GMM (b).

vides more robustness against illumination changes. Fig. 3.10 gives an example of an input video frame and the corresponding possible skin-color regions.

A GMM is composed of K Gaussian probability density functions (pdfs), described by the following equation:

$$p(\mathbf{c}_j|C_{skin}) = \sum_{k=1}^{K} w_k p_k(\mathbf{c}_j|C_{skin}) \tag{3.87}$$

where p_k is the k^{th} mixture component, with weights w_k normalized so that $\sum_{k=1}^{K} w_k = 1$, and each component p_k is described by a bi-variate Gaussian

$$p_k(\mathbf{c}_j|C_{skin}) = \frac{1}{2\pi\sqrt{|\Sigma_k|}} e^{-\frac{1}{2}(\mathbf{c}_j-\mu_k)^T \Sigma_k^{-1}(\mathbf{c}_j-\mu_k)} \tag{3.88}$$

with $\mathbf{c}_j$ the 2-dimensional (H,S) color of screen pixel j. Mean and covariance matrix (μ_k, Σ_k) for each component, as well as the mixture weights w_k, are learned from a given training set, via the Expectation-Maximization algorithm [151]. The number of components necessary for skin detection has been widely discussed in the literature, and ranges according to [152] from $K = 2$ [149] to $K = 16$ [153]. The GMM model used here was built from a training data set of labeled skin-pixels, and consists of $K = 2$ mixture components following [149].

To classify a color pixel j, its GMM likelihood can e.g. compared to a suitable

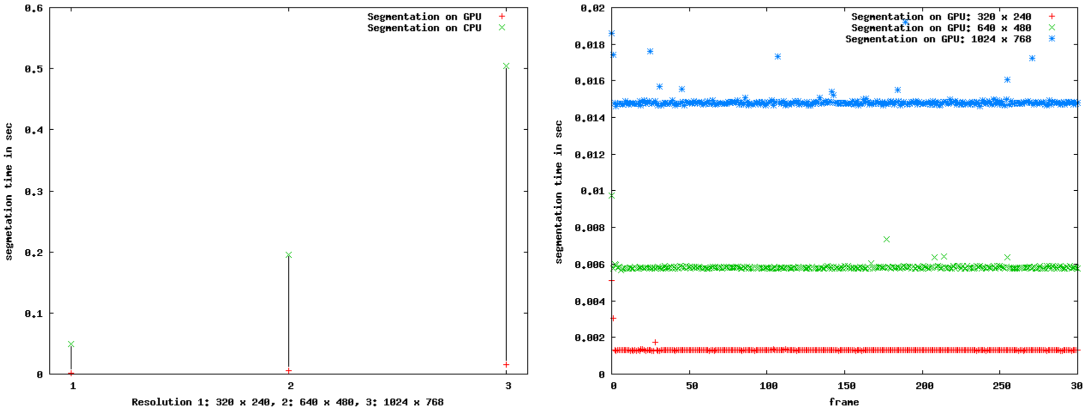


Figure 3.11: Comparison of segmentation times - The GPU outperforms the segmentation speed of the CPU dramatically (left) with constant segmentation times (right).

value p_{min}

$$z(j) = \begin{cases} 1 & \text{if } p(\mathbf{c}_j|C_{skin}) > p_{min} \\ 0 & \text{if } p(\mathbf{c}_j|C_{skin}) \leq p_{min} \end{cases} \tag{3.89}$$

which results in a binary image (Figure 3.10), which constitutes our pixel-level measurement z_t for tracking.

Although this segmentation step is performed only once per frame, it results in pixel-wise expensive computations that leaves less time for the subsequent tracking steps. As modern graphics card are becoming very popular today, because of their computational power, the low costs, and the emerging of high-level languages such as CUDA [154], Cg [155], or the OpenGl Shader Language [156] that allow a general purpose use of the graphics hardware, these operations were implemented on the graphics processing unit (GPU), using at the same time the power of the rendering engine and the parallel pixel-pipelines. [157] presents a good survey of the possibilities and limitations of the GPU. To make use of the possibilities, both computations (3.87),(3.89) have been implemented on the GPU (NVidia GeForce 8800) by using the OpenGL shader language [156]. The speed improvements gained by the usage of the GPU are dramatic and increase with the image size as shown in Fig. 3.11.

In the update step of the Bayesian tracking scheme (see Section 3.2.4), the weight π_n of each particle $n = 1 \dots N$ is computed by comparing its state hypothesis s_t^n with the current segmentation image z_t. The projected filled object silhouette h_t^n at each pose hypothesis has to be computed (Fig. 3.12 (middle)) providing a binary map which

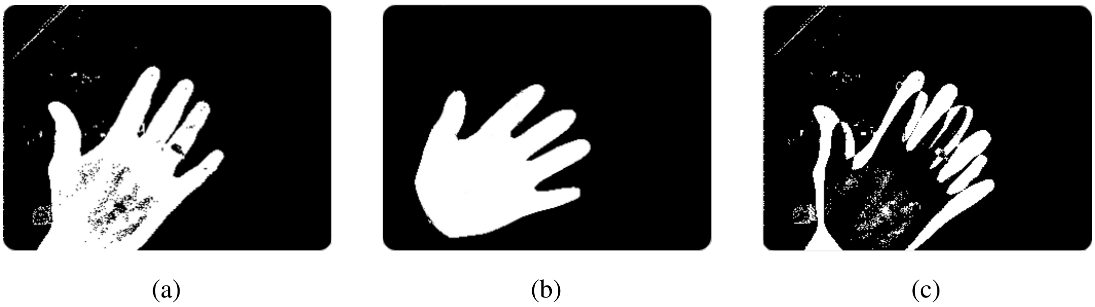


Figure 3.12: Residual computation - The segmented binary image (a) and the hypothesis silhouette image (b) are compared with each other (c). The resulting white pixels are related to the error of the hypothesis [135]

represents the expected measurement for an ideal, noise-free segmentation under the given pose hypothesis s_t^n.

Afterwards, the residual with the current measurement z_t is computed by the cost function

$$e_t^n = \sum_y \left[h_t^n(y) - z_t(y)\right]^2 \tag{3.90}$$

that is equivalent to a pixel-wise XOR (see Fig. 3.12 (right)) followed by a sum of the non-zero pixels for the binary images h_t^n, z_t.

For this step, also the GPU was used, because the computation of h^n can be very expensive if performed on the CPU where no pixel parallelization can be exploited while comparing it with z_t.

The residual value (3.90) is normalized to the range $[0,1]$ by dividing it by the number of pixels, and the likelihood is evaluated with the likelihood model

$$\pi_n = P(z_t|s_t^n) = \exp(-\frac{e_t^n}{2r^2}) \tag{3.91}$$

where the measurement variance r, providing the new particle weights π_n, afterwards normalized so that $\sum_n \pi_n = 1$. Deterministic re-sampling of the particle set [119] is applied after each update, in order to keep a well-distributed particle set. Figure 3.9(d) shows the weighted average of the particle set

$$\hat{s}_t = \sum_{n=1}^{N} \pi_h s_t^n \tag{3.92}$$

that approximates the hand position along with the distributed particle set. The result of the hand tracking employing a 3D mesh model is depicted in Fig. 3.9(a) - 3.9(c). The

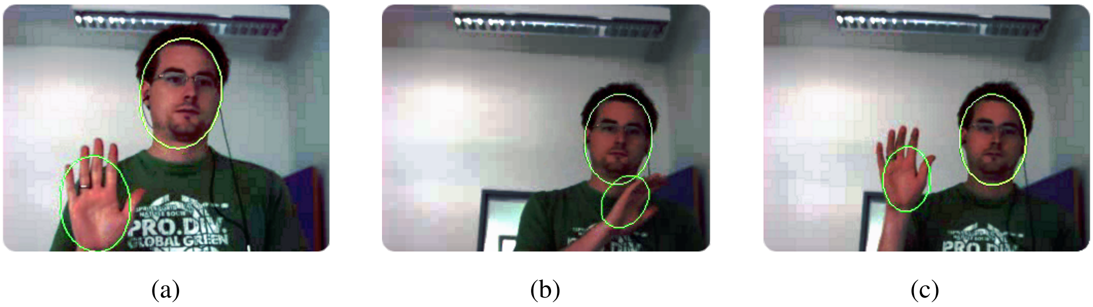

(a) (b) (c)

Figure 3.13: Multi-target tracking - The simultaneous tracking of multiple skin-colored targets with an elliptic shape that approximates both hand and head [120]

related publication on the GPU-accelerated particle filter work was also recognized by multiple authors that further improved and extended this approach [158, 159].

Nevertheless, if multiple skin-colored objects have to be tracked simultaneously such as two hands or hands and head as depicted in Fig. 3.13, the same approach can be used by applying the multiple object particle filter scheme (MOSIR) described in Section 3.2.4. The computation of the residual and therefore the weighting of the particles follows Eq. (3.90) with the difference, that multiple object hypotheses are projected. The tracking of multiple targets increases the dimensionality of the overall system, which leads to the need of more particles to be used to approximate this increase in dimensionality.

In scenarios where human and robot share the same workspace the robot can occlude the hand(s) in a single camera view. Additionally, in interaction scenarios, the hand positions need to be often evaluated in the 3 dimensional world space. Both aspects can be solved by the usage of multiple cameras and/or exponentially more particles to approximate the increasing dimensions. Unfortunately, the generic particle filter approach does not scale properly with multiple cameras and objects as the dimensionality increases and each hypothesis scene needs to be compared to multiple camera views.

Since the results in Section 3.1 show, that the workflow for the tower assembly task can be analyzed using the three-dimensional position and derived information, the basic principles of the presented GPU particle filter approach are used while optimizing the estimation and tracking process by the usage of a three-dimensional grid-based approach as presented in Section 3.3.2.

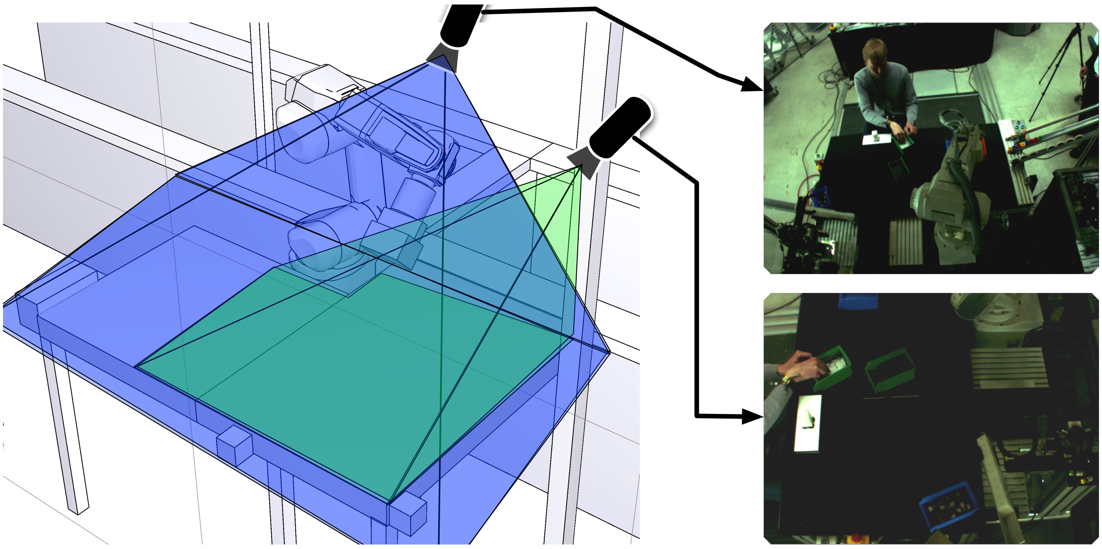


Figure 3.14: Occupancy grid tracking set up - Two cameras were mounted on the sensor scaffold to have multiple views of the human hands. The right images show example views of the two cameras [90]

3.3.2 Three-dimensional occupancy grid tracking

Occupancy grids are a well-known technique used in mobile robotics to solve path planning and localization problems [160, 161, 162, 163, 164]. Recently, the application of such grid maps was found to be well suitable to be employed in tracking tasks. [165] uses discretized areas on the ground plane and fits GMMs to estimate the likelihood of persons standing on a specific location. The motion of humans is modeled by a Kalman filter. A combination of a probabilistic occupancy maps with models of color and motions is presented in [166]. To follow and distinguish multiple persons in the synchronized camera streams, the Viterbi algorithm and a greedy approach is used. [167] uses hierarchical likelihood grids based on intensity edges followed by a global nearest neighbor data association approach to perform the tracking of multiple persons in a multiple camera set-up. All these approaches also use generative and generic models to compare "ideal" measurements with the real—and probably noisy—sensory data on discretized locations on one layer.

The discretization of the problem space allows a pre-computation of expected measurements for all possible (discrete) locations. This reduces the computational complexity during run-time and makes this approach linear scalable to multiple cameras.

With the extension of the occupancy grid to three dimensions [90], a reliable, fast, and robust hand tracking in world coordinates becomes possible. For the following evaluation, two intrinsically and extrinsically calibrated cameras mounted on the *JAHIR* set-up were used. One camera is facing the human from the front and the other is facing towards the workspace from the side as depicted in Fig. 3.14.

The volume in which the hands of the human are likely positioned and are to be tracked can be defined. This volume is set in the world coordinate frame that is located at the left corner of the desk. The volume of interest starts at $x = 0.3\,\mathrm{m}$, $y = -0.1\,\mathrm{m}$, $z = 0.0\,\mathrm{m}$ and has the width $w = 1.1\,\mathrm{m}$, the depth $d = 0.5\,\mathrm{m}$, and the height $h = 0.3\,\mathrm{m}$. This results with a discretization step of 0.05m in 1694 locations (22 in x; 11 in y; 7 in z direction) as shown in Fig. 3.15(a).

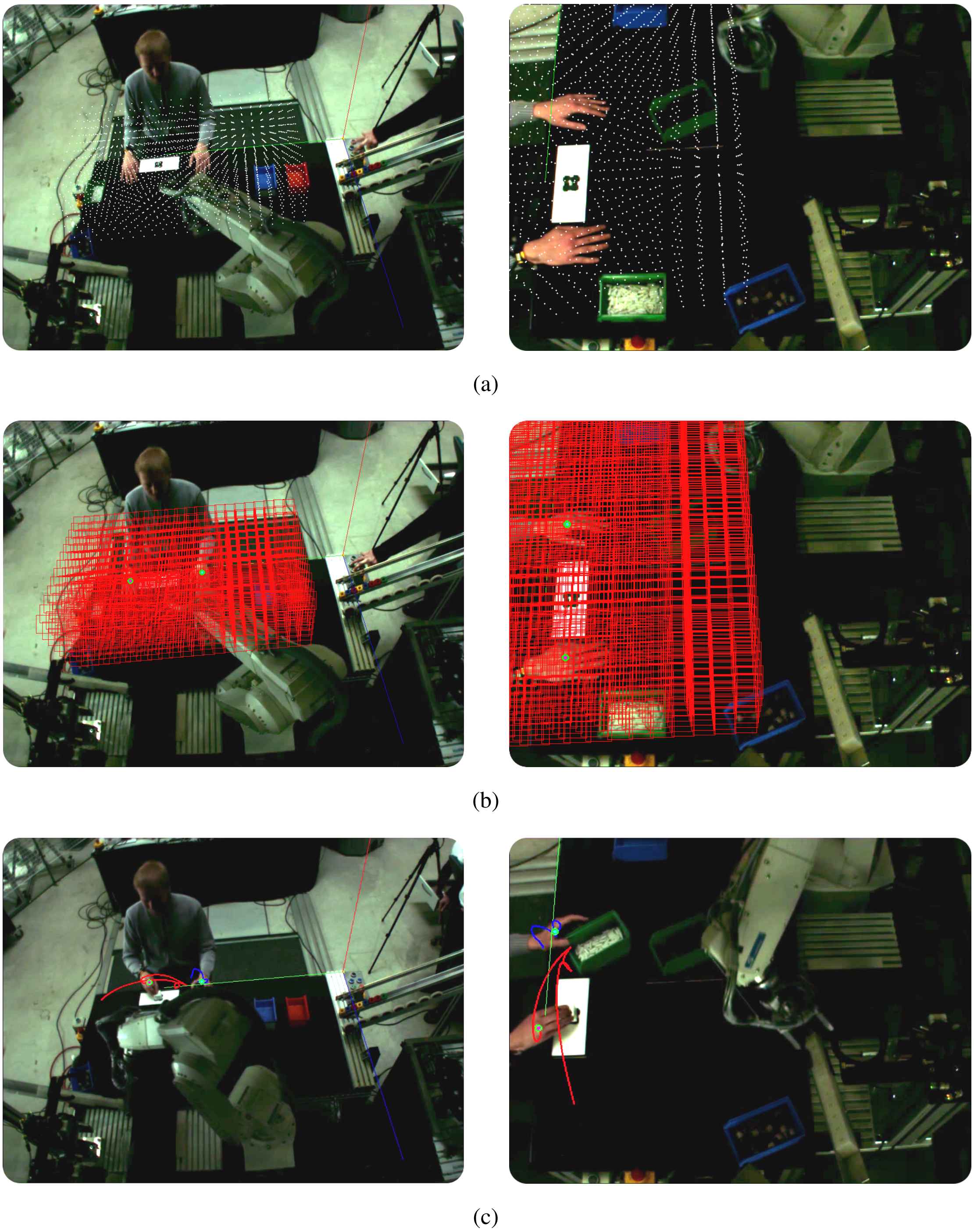
A cube with the length of 0.05 m approximates the hand of a human. That is roughly the dimension of the palm. This model is projected using the intrinsic and extrinsic camera parameters as described in Section 3.2.1 to all 1694 locations in each camera resulting in screen rectangles. If a projected model is not visible in one camera, the corresponding screen rectangle is marked as invisible and will not be evaluated in this camera. Partly visible screen rectangles are truncated to fit the camera screen. The screen rectangles can be interpreted as expected measurement of a hand being at a specific location. The projection to the two camera views used here is depicted in Fig. 3.15(b). The chosen camera arrangement offers also the advantage, that the views are approximately aligned with two world axes, which leads to axis aligned screen rectangles. All of these steps are computed off-line.

During the on-line tracking, every incoming image is first transformed into a scale space and then segmented by skin-color (see e.g. Fig. 3.10). Every screen rectangle $S^{1\ldots C}_{1\ldots R}$ is tested on the binary image z^c of camera c and the likelihood of a hand being positioned in the rectangle r in camera view c is evaluated by

$$P(S^c_r|z^c) = \frac{F(S^c_r, z^c)}{A(S^c_r)} \tag{3.93}$$

where $F(S^c_r, z^c)$ estimates the number of white pixels in the screen rectangle with an integral image approach [168] and $A(S^c_r)$ is the area covered by the screen rectangle. The overall likelihood for a hand in a specific rectangle is then given by

$$P(S_r|z) = \prod_{c=1}^{C} P(S^c_r|z^c). \tag{3.94}$$



(a)

(b)

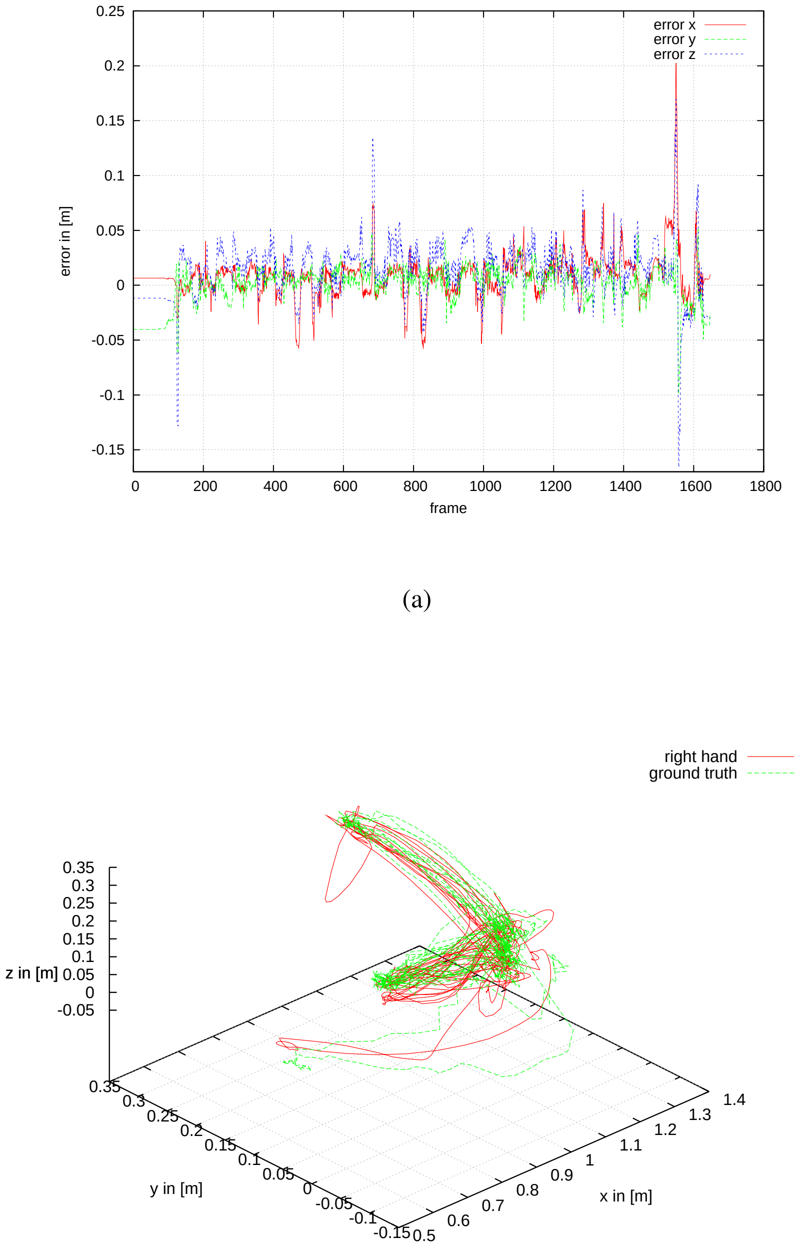
(c)

Figure 3.15: Overview occupancy grid tracker - (a): The discretized volume of interest starting results with a discretization step of 0.05m in 1694 locations (22 in *x*; 11 in *y*; 7 in *z* direction); (b): The approximated hand shape is projected to the 1694 locations and forms the expected appearances; The screenshots in (c) show the projection of parts of the hand trajectory to the 2 camera views [90]

Given this three-dimensional likelihood distribution, all rectangle candidates that are above a chosen global prior value are used to compute the weighted average. This average is used to divide the data set to left hand and right hand candidates. This assumption is valid because it is assumed to have exact two hands in the volume of interest. For the hand candidates again a weighted average is applied leading to the three-dimensional position of both hands.

These positions are then used as input values for two standard Kalman filters as described in Section 3.2.4. Since all operations are performed directly in three-dimensional space (*object level*), it complies with the linearity and Gaussian requirements of a Kalman filter [101]. The motion of the hands is modeled by a constant velocity motion model with white noise acceleration (CWNA) (see Section 3.2.3).

The resulting position errors of the tracking are depicted in Fig. 3.16 and show that the presented tracking approach delivers a good estimation of the hand position compared to ground truth data. To gain the ground truth, the hand positions were labeled in every frame of every camera and then the 3D position was reconstructed using the *Direct Linear Transformation* (DLT) algorithm [92]. The standard deviation of the position error is given with 0.0207 m in x, 0.0179 m in y, and 0.026 m in z direction which is a really good result considering the discretization of the space to base points with a distance of 0.05 m. The errors at the end of the sequence result, because the subject is leaving the volume of interest and then the view of one camera, which makes the estimation in three-dimensions hard. This can be e.g. solved, if hand targets are added and removed automatically by the tracker. The approach works in real-time with over 20 fps on a standard machine.



(a)

(b)

Figure 3.16: Hand tracking results - The graph (a) show the absolute position error for x, y, and z-direction for the tracking of the right hand compared to ground truth data. Graph (b) shows the three-dimensional trajectory for the right hand (red) along with the ground truth trajectory (green) [90]

Chapter 4

Action coordination

Contents

This chapter sets the basis in order to transfer the mechanism action coordination *as fundamental principle adjusting own actions in space and time according to the behavior of the collaboration partner or the perceived context. The coordination of robotic movement based on a hierarchy of atomic tasks including the geometric awareness of the robot, that integrates static and dynamic geometric representations of the surrounding, is presented (Section 4.1). The coordination of actions also incorporates many conscious and unconscious aspects that need to be considered. Therefore, Section 4.2 presents a handing over experiment that shows that the reaction times of humans can unconsciously be influenced in a positive manner by choosing appropriate robotic motion profiles. Parameters that have to be negotiated by the subjects during a hand over also include the right timing of actions. Hence, Section 4.3 shows how* *the robot can use the observations of human actions to efficiently coordinate actions in time to increase the collaboration fluency.*

4.1 Task-based robot controller using orthogonal projection

For a dynamic collaboration, the actions that should be accomplished by the robot need to be intuitively specifiable. According to [169], actions consist of several atomic tasks that are arranged in a task-oriented way. Hence, the stacking of hierarchical tasks together forms the hierarchical control structure for the robot [120].

Tasks can be seen as basic modules that are made to solve a specific problem including moving to a certain position and *emit* control signals like velocity commands for the robot. Since tasks can competitively interfere with each other and lead to uncontrollable or unwanted behavior of the robot, nullspaces and constraint least-square optimization are employed to project task velocities in safe (orthogonal) subspaces of higher priority tasks. Based on the syntax of [170], an action A can be formulated as a compound of tasks T_k with a projection rule $\lhd_k$ that ensures the behavior of the higher priority task:

$$A = \langle T \rangle_0^n = T_n \lhd_n T_{n-1} \lhd_{n-1} \ldots \lhd_1 T_0. \tag{4.1}$$

As most industrial robots—including the one used in the *JAHIR* set up—are only controllable on the position- or velocity-level, the task descriptions are expressed in terms of joint velocities ($\underline{\dot{q}}$):

$$\underline{\dot{q}}_{\text{Action}} = \underline{\dot{q}}_{T_n} \lhd_n \underline{\dot{q}}_{T_{n-1}} \lhd_{n-1} \ldots \lhd_1 \underline{\dot{q}}_{T_0}. \tag{4.2}$$

According to the defined control structure, mainly three points need to be considered for each task:

- How can the velocity to solve the single task be computed?
- What kind of (static or dynamically changing) constraints should limit lower priority task velocities?
- How can the lower priority task velocity be safely projected into a subspace of the current task velocity respecting the constraints?

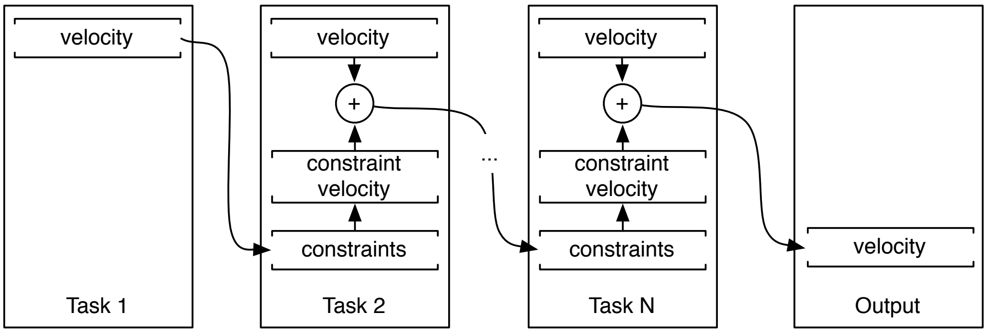


Figure 4.1: Actions are defined through task compositions - The associated constraints are respected through projections into corresponding subspaces, with execution priorities ranging from lowest (left) to highest (right)

For the latter *nullspaces* with orthogonal projectors are used as described e.g. in [169, 171] in the posture and the operational position task and a constraint least-square optimization for the collision avoidance task as e.g. done for accelerations in [172, 173]. With these projectors tasks get decoupled from each other (see Figure 4.1) and only the single goals and constraints need to be defined, while the rest is internally computed. Application examples of the task-based controller applied in the *JAHIR* set-up are given in Section 5.4.1.

4.1.1 Joint position task

The goal of the joint position task is to drive the robot to a certain joint configuration $\underline{q}_{goal}$. It can also be used to give the robot a specific posture. The velocity for this task can be calculated, for example, by

$$\underline{\dot{q}}_{po} = C\left(\frac{\underline{q}_{goal} - \underline{q}(t)}{\Delta t}\right) \tag{4.3}$$

with

$$C(\underline{\dot{q}}) = s(\underline{q}, \underline{\dot{q}}, \underline{\ddot{q}}) \cdot \underline{\dot{q}}. \tag{4.4}$$

being a function, that limits the joint velocities depending on the current state of the robot and corresponding limits (such as joint limits, velocity limits, and acceleration limits) *without* changing the trajectory of the motion.

To constrain certain degrees of freedom in joint space, that cannot be influenced by lower priority tasks, a $n \times n$ matrix S_{po} is used to select them, with n being the number of joints. This means the guaranteed velocity can be expressed as

$$\underline{\dot{q}}^*_{po} = S \cdot \underline{\dot{q}}_{po}. \tag{4.5}$$

Using the projector N_{Po} of the selected constrained joints, the projection of an input velocity $\underline{\dot{q}}_{in}$ is done according to:

$$\underline{\dot{q}}^* = \underline{\dot{q}}^*_{po} + N_{po} \cdot \underline{\dot{q}}_{in} \tag{4.6}$$

with

$$N_{po} = I - S_{po}. \tag{4.7}$$

4.1.2 Cartesian position task

The Cartesian position task drives the tool center point of the robot to a defined goal position and/or orientation $\underline{x}_{goal}$ in Cartesian coordinates. The Cartesian velocity can for example be computed according to:

$$\underline{\dot{x}} = \frac{\underline{x}_{goal} - \underline{x}(t)}{\Delta t}. \tag{4.8}$$

After the velocity in Cartesian space is calculated, constraints can be set using a diagonal selection 6 × 6 matrix S_{Op} to select the degrees of freedom (in Cartesian space) that should not be influenced by lower priority tasks. Transformed into limited joint velocities using a singularity robust pseudo-inverse $J_e^\dagger$ of the Jacobian J_e of the end-effector, resulting in

$$\underline{\dot{q}}^*_{op} = C(J_e^\dagger \cdot S_{op} \cdot \underline{\dot{x}}_{op}). \tag{4.9}$$

Using the nullspace N_{op} of the selected constraints, the orthogonal projection of an input velocity $\underline{\dot{q}}_{in}$ leads to:

$$\underline{\dot{q}}^* = \underline{\dot{q}}^*_{op} + N_{op} \cdot \underline{\dot{q}}_{in} \tag{4.10}$$

with

$$N_{op} = I - J_e^\dagger S_{op} J_e. \tag{4.11}$$

4.1.3 Collision avoidance task

Collisions need to be avoided for static (i.e., the workbench) and dynamic environment (i.e., the human and moving obstacles). In this task, the avoidance is done in a reactive

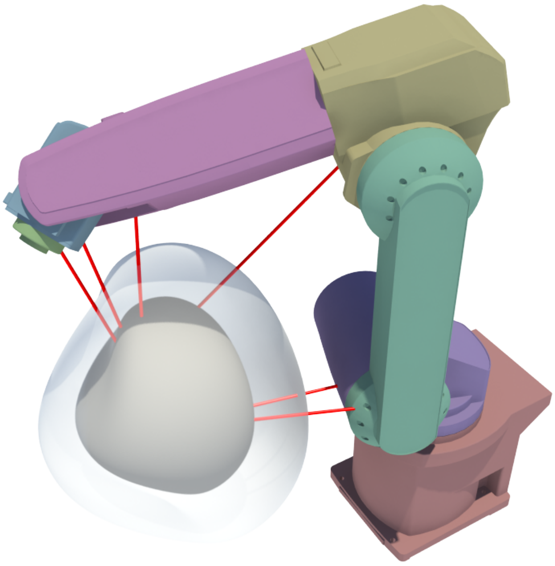


Figure 4.2: Computing distances between obstacles and robot - The red lines illustrate the minimum distances of an obstacle to the body parts of the robot in a given joint configuration. If a distance is below a chosen threshold (transparent bubble), the distance is used to compute virtual forces on the robot using potential fields [174]

way with dynamically updated collision scenes that can be interfaced with a variety of sensors as presented in Section 5.2.2. The main challenge that arises here, is that the planned motion and the avoidance motion must be handled in a way where they do not interfere with each other. Therefore, the potential field methodology to repel the robot from the obstacle is combined with a constraint least-square optimization that restricts the motion of the robot to safe orthogonal subspaces of the collision avoidance motion.

Virtual forces

To compute the velocity that repels the robot from surrounding obstacles, the minimum distances of all objects in the environment model (including self-collision) to all body parts of the robot need to be measured. Figure 4.2 depicts the body parts of the used robot in different colors along with an example of the minimum distances d_i (red lines) from an obstacle to a given joint configuration.

Opposite to simplified and only approximated models of manipulators (e.g., used in the skeleton algorithm presented in [175]), the distances of arbitrary shapes to a convex version of the real CAD-model of the robot can be measured in order to reach a high precision of virtual forces. With an efficient implementation of the GJK algorithm [176], these distances can be computed faster than the update rate of the robot controller. After calculating the minimum distance vectors $v_{x,i}$ in Cartesian space (i.e. the direction of the applied virtual force), the corresponding velocities in joint space are computed to find the overall motion of the robot that avoids the collision. This is done according to

$$\underline{\dot{q}} = \sum_{i}^{I} \underline{\dot{q}}_i = \sum_{i}^{I} J_{P_r(i)}^T \cdot F_{rep,i}(q) \cdot v_{x,i}(q), \tag{4.12}$$

with I being the number of bodies of the robot, the current joint configuration of the robot q, the Jacobian of the minimum distance point on the robot $J_{P_r(i)}$ and $F_{rep,i}$ being the virtual force on the robot body according to the repelling potential function $U_{rep,i}$ [177] :

$$U_{rep,i}(q) = \begin{cases} \frac{1}{2}\eta_i \left(\frac{1}{d_i(q)} - \frac{1}{Q^*}\right)^2 & \text{if } d_i(q) \leq Q^* \\ 0 & \text{if } d_i(q) > Q^* \end{cases} \tag{4.13}$$

$$F_{rep,i}(q) = \begin{cases} \eta_i \left(\frac{1}{d_i(q)} - \frac{1}{Q^*}\right) \frac{1}{d_i(q)^2} \nabla d_i(q) & \text{if } d_i(q) \leq Q^* \\ 0 & \text{if } d_i(q) > Q^* \end{cases} \tag{4.14}$$

with Q^* being the distance at which the potential field function is applied (see transparent bubble in Figure 4.2).

Constraint least-square minimization

To ensure that the lower priority task is projected in an orthogonal subspace of the collision avoidance task, the mathematical framework of quadratic programming [178, 172, 173] is used to minimize the quadratic error between optimal velocity of the lower priority task subject and the constraints of the higher priority task. The low-priority task execution ($\underline{\dot{q}}_{in}$) is optimized regarding *must have* constraints of the higher priority task. The projection is described according to:

$$\min_{\underline{\dot{q}}_{in}} ||J_e \cdot \underline{\dot{q}}_{in} - \underline{\dot{x}}_t||^2, \tag{4.15}$$

where $\underline{\dot{x}}_t$ is the ideal linear and angular velocity to solve the lower priority task subject to the linear constraints of the form

$$C^T \underline{\dot{q}}_t \geq 0 \tag{4.16}$$

with the constraint matrix

$$\underline{C}^T = \begin{pmatrix} \underline{\dot{q}}_1^T \\ \vdots \\ \underline{\dot{q}}_I^T \end{pmatrix} \tag{4.17}$$

and $\underline{\dot{q}}_i$ calculated according to (4.12). This means only those velocities are valid, that are orthogonal to the direction of the collision avoidance velocities or point in a direction that leaves the defined safety region. The output of the minimization process is then equal to the constrained joint velocity $\underline{\dot{q}}^*$.

4.2 Unconscious adaption of action parameters by the human

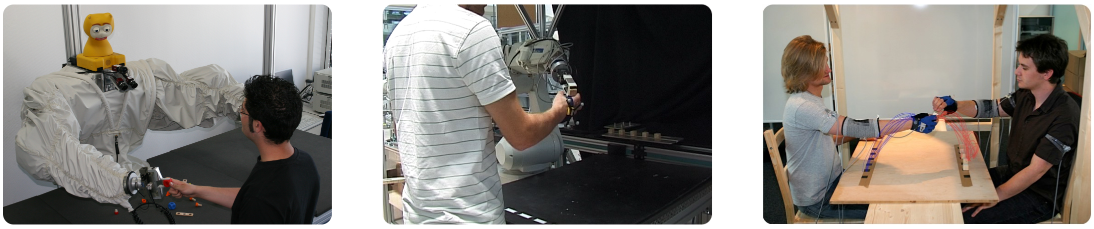
Since the handing over of objects needed for assembly steps is a central action in the collaboration between human and robot, investigations on how to improve the collaboration by means of robotic motions has been accomplished. To examine effects regarding the adaptation of human reaction, a repetitive hand-over task was designed in [179]:

Subjects were instructed to take six cubes from their opponent and to put each cube on marked positions in front of them as depicted in Figure 4.3(c). The motion of the subjects was recorded and subsequently analyzed to determine the time needed to perform the task (*task effectiveness*) during the repetitions. The hand-over action was especially analyzed regarding the reaction time of the subjects.

The experiment was conducted and evaluated in:

- a *human-humanoid* set-up, where the *JAST* platform [33] was used as depicted in Figure 4.3(a) [179],
- the *human-industrial robot JAHIR* set-up as depicted in Figure 4.3(b) [180].
- and a *human-human* set-up as depicted in Figure 4.3(c) [179, 180],

In the robotic set-ups, a force/torque sensor mounted on the tool center point of the robot was used to determine whether the human has grasped the cube. After the hand-over, the robot moved to a resting position and waited between zero and four seconds before starting the next hand-over, so that the subject was not able to adapt to



(a) (b) (c)

Figure 4.3: Hand-over experiments - A repetitive handing over task between human and robot was designed and evaluated on the human-humanoid set-up *JAST* [33] (a) and on the *JAHIR* set-up [180] (b) and compared to human-human experiments [179] (c)

a periodical behavior. In contrast to the *human-human* experiment presented in [179], the giving subject was triggered by a headphone to start the handing over, leading to shorter reaction times in average than in the previous experiment. With this change in the experiment, a direct comparison of the reaction times becomes possible, because the robot was triggered with the same time values. The hand-over position of the robot stayed fixed throughout the experimental runs in a position related to the hand-over position measured in the *human-human* experiment.

In both robotic set-ups, two different methods were used to generate the robotic motions: a trapezoidal velocity profile in joint coordinates and a minimum jerk velocity profile in Cartesian coordinates. In the trapezoidal profile, trajectories are calculated with a constant acceleration $\underline{\ddot{q}}_a$ and deceleration $\underline{\ddot{q}}_d$ phase of the joints in a given acceleration and deceleration time (t_a, t_d):

$$\underline{\dot{q}}(t) = \begin{cases} \underline{\ddot{q}}_a t + \underline{\dot{q}}_0, & 0 \leq t < t_a \\ \underline{\ddot{q}}_a t_a + \underline{\dot{q}}_0, & t_a \leq t < t_d \\ \underline{\ddot{q}}_a t_a + \underline{\ddot{q}}_d (t - t_d) + \underline{\dot{q}}_0, & t_d \leq t < t_e \end{cases} \tag{4.18}$$

Opposite to the interpolation in joint space, the minimum jerk velocity profile interpolates in the Cartesian space with straight lines in the motion of the robot's tool center point. The minimization of corresponding the objective function

$$c(\underline{x}) = \frac{1}{2} \int_0^{t_e} \left| \frac{d^3 \underline{x}}{dt^3} \right|^2 dt \tag{4.19}$$

leads to a fifth-order polynomial. With given initial and end position, velocity and acceleration for the trajectory, the polynomial coefficients can be specified. The deriva-

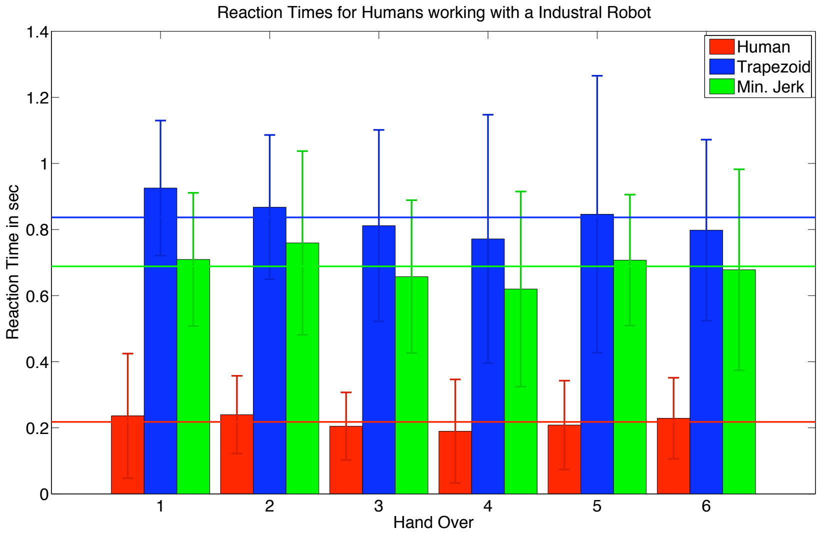


Figure 4.4: Reaction time for all six trials - The diagram shows the reaction time for the human-human handover (red), for the *JAHIR* robot using the trapezoid profile (blue), and for the *JAHIR* robot using the minimum jerk profile. Error bars indicate standard deviation, the straight lines indicates the mean over all trials [180]

tion of this equation results in the velocity profile [180],

$$\dot{\underline{x}}(t) = (\underline{x}_0 - \underline{x}_e)\left(60\frac{t^3}{t_e^4} - 30\frac{t^4}{t_e^5} - 30\frac{t^2}{t_e^3}\right) \tag{4.20}$$

with $\underline{x}_0$ and $\underline{x}_e$ being the initial and end position of the tool center point of the robot and t_e being the desired time to reach the end position.

By using the different velocity profiles, the influence of motion trajectories on the effectiveness of the task performance was determined. The experiments pointed out, that the minimum jerk profile [181] leads to shorter reaction times (0.69 s vs. 0.86 s). Hence, the task effectiveness can be positively influenced by how the robot actually moves. Current developments in generating human-like movements [182, 183] might even improve this effect. Table 4.1 lists the evaluated times needed for the specific parts of the hand-over task and the overall time needed for a single hand-over. As expected, the reaction times are minimal for human-human hand-overs with 0.22 s and the overall time for the hand-overs is one second faster compared to the *JAHIR* set up. A detailed view on the reaction times for the single cube hand-overs is given in Figure 4.4.

Table 4.1: Average duration of the reaction times of the human during a handover for the minimum jerk and the trapezoid velocity profile in seconds [180]

set-up	velocity profile	reaction time
human–human	—	(0.22 ± 0.02) s
human–humanoid	trapezoidal	(0.50 ± 0.06) s
	minimum jerk	(0.39 ± 0.04) s
human–*JAHIR*	trapezoid	(0.86 ± 0.03) s
	minimum jerk	(0.69 ± 0.03) s

Additionally, the experiments turned out significant differences in the reaction times between the two robotic systems: the reaction times (0.50 s for trapezoid and 0.39 s for the minimum jerk profile) were faster in the humanoid set-up than in the industrial *JAHIR* set-up (0.86 s/0.69 s). An explanation for this effective is that humans are able to predict the motions of other humans and the attributed goals by covertly simulating observed behavior using their own mind in a simulation mode in order to understand what intentions are coupled to the observed actions [184]. Therefore, a human like arrangement (i.e. the *JAST* platform) showed better performance, because the subjects could unknowingly predict the motion of the robot better during the procedure [179, 180]. Further evaluations of the experiments related to the *JAHIR* set-up are given in Section 5.4.2.

4.3 Active adaption of action parameters by the robotic system

Section 3.1 has focused on the observation and the subsequent recognition of the human's actions. Additionally, actions of the robotic system can on the reverse side be efficiently coordinated based on the observation of the human's actions, if a robotic system is able to adapt its own parameters to the human counterpart.

Therefore, [186] employed a bank of Kalman Filters to determine the complexity of an assembly step and the related time needed to perform the step. With an estimation of

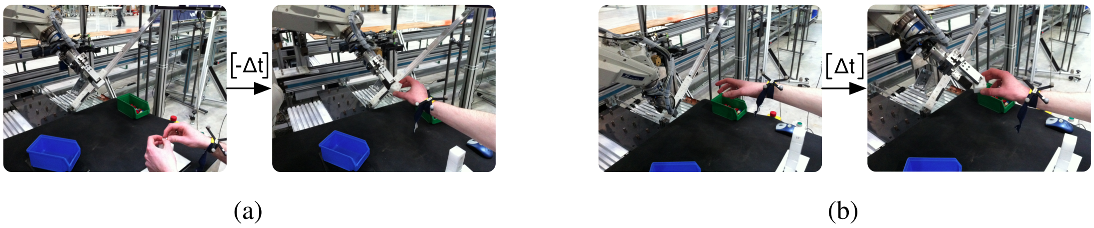


Figure 4.5: Waiting times - Either the robot needs to wait in the hand-over position if the predicted time was too short (a) or the human needs to wait for the object—i.e., triggers the robot to perform the hand-over—if the predicted time was too long (b). This error estimation is one input value for the prediction module [185]

the time needed for an assembly step, the robot can coordinate its actions to reduce or even avoid waiting times for the human in hand-overs. This enables a fluent collaboration and—as shown in this section—the possibility to generate dynamic workflows for the robot, because the robot can use the time prediction to perform preliminary tasks. The timing of actions has been further investigated in previous work [86] by evaluating the assembly task experiment, that was also used and described in Section 3.1. For the sake of completeness, the experiment is shortly repeated here: subjects were instructed to build towers using cubes that differ in the number of bolts needed to combine them (Figure 3.1(b)). The cubes were delivered by a slide and are available for the subjects at any time.

The analysis of data gained in the assembly experiment leads to the assumption, that the right time to hand-over a cube, is the point in time when the subjects reached out for the omnipresent cubes. Based on this assumption and the data gained in the assembly experiment, in [86] a method was developed to predict the assembly durations using a probabilistic Bayesian framework. The predictor was realized as a bank of Kalman filters with continuously updating parameters that describe the workers assembly behavior using an underlying relationship between duration and complexity of the assembly step. Additionally, the complexity of known component types can be estimated using the inverse relationship of such a duration-prediction framework. A recursive interlacing allows the framework to be applied to any kind of assembly task, without exact a-priori knowledge of the component's complexities. These findings are then transferred and applied to the *JAHIR* set up as presented in Section 5.4.3.

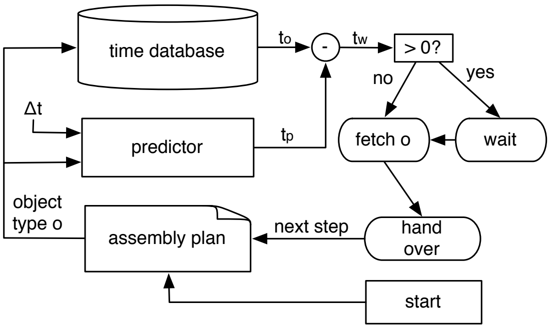


Figure 4.6: Schematic overview for single task - The timing error Δt and the needed object type o (from an assembly plan) are the input values to the *predictor*, resulting in a time estimation to the next hand-over t_p. This is compared with the time t_o needed to fetch and hand-over object o, resulting in the expected waiting time for the robot t_w

4.3.1 Reducing the waiting times of the human

The predicted duration of an assembly step can be used to perform the hand-overs from robot to human just-in-time. A schematic cycle of the collaboration using predicted hand-over times is depicted in Figure 4.6. The time error Δt of the previous hand-over and the object type are the input values to the *predictor*, resulting in a time estimation t_p for the next hand-over.

A *time database* delivers the time needed to perform a hand-over of a specific object t_o (*object time*), including the time required to move the robot to the storage position of the object, the grasping, and then the motion to the hand-over position. The relationship between objects and their time consumption t_o can be previously estimated or roughly approximated and updated during run-time. These two times are compared in $(t_p - t_o) > 0$, resulting in the decision whether to fetch the object if the predicted time is smaller than the needed *object time*, or if the robot needs to wait to be able to perform the hand-over *just-in-time*. A successful hand-over initiates a step forward in the assembly plan and the cycle proceeds. The actual hand-over time point is measured by a force-torque sensor attached to the gripper and defines the timing error Δt. As depicted in Figure 4.5, either the robot needs to wait in the hand-over position or the human needs to wait—i.e. triggers the robot to perform the hand-over. The triggering of the robot and the estimation of the waiting time can be measured using the estimated hand position. This information is communicated via the corresponding communication channel (see Section 5.2).

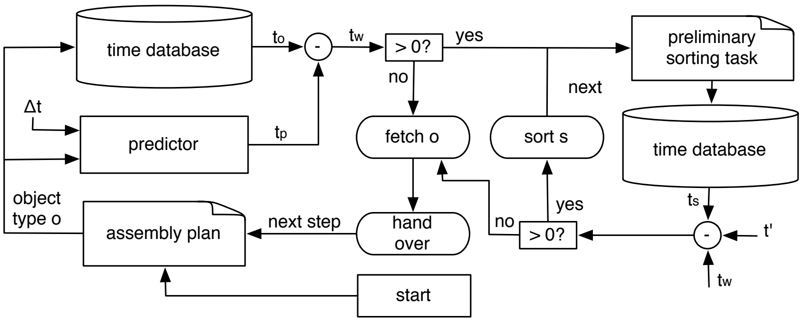


Figure 4.7: Schematic overview for multiple tasks - A preliminary task fills the waiting times of the robot—if $t_w > 0$ is true—to use this time efficiently. The steps of this task are transformed to their time consumption t_s and evaluated if a step fits in the waiting time of the robot, including the time that has passed since the start of the current task t'. If enough waiting time is available, the secondary cycle can be executed multiple times

4.3.2 Using the waiting times of the robot for preliminary tasks

The waiting time $t_w = (t_p - t_o)$ of the robot, if $t_w > 0$, can also be interpreted as a downtime of the robot. If the performance of the overall system should be increased, this downtime of the robot needs to be decreased. With the resulting estimation of the waiting time or free time of the robot, the robot can be used to perform preliminary tasks during this time, including the presorting of parts that are needed in the future.

The preliminary task can be performed, whenever the assembly duration prediction calculates a time interval, which is big enough to perform the task. It has to be mentioned, that the effective time for planning preliminary tasks is the predicted duration minus the time the robot needs to do the assistance. To give an example, the time needed by the robot to do the assistance is the time it takes the robot to move to the next component, grasp it, and present it at the handing-over position. The schematic cycle of Figure 4.6 can be extended by substituting the waiting block with the preliminary task performance (Figure 4.7). If $t_w > 0$ is true, the cycle for the preliminary task can be initiated. The steps of this task are transformed to their time consumption t_s and evaluated if one step fits into the waiting time of the robot. If enough time is available, the preliminary task can be executed multiple times. Therefore, the elapsed time t' from starting one action up to now in included in the comparison. Hence, if $(t_w - t' - t_s) > 0$, the robot can execute step s of the preliminary task; if not, the fetching of object o and therefore the hand-over is initiated.

Chapter 5

Bringing things together – the *JAHIR* platform

Contents

This chapter presents the integrated demonstration platform JAHIR *that evolved throughout this thesis. This includes the hardware set-up (Section 5.1) and the software architecture behind (Section 5.2) to glue the developed software modules together. Both form a powerful demonstration platform that was also used beyond the scope of this*

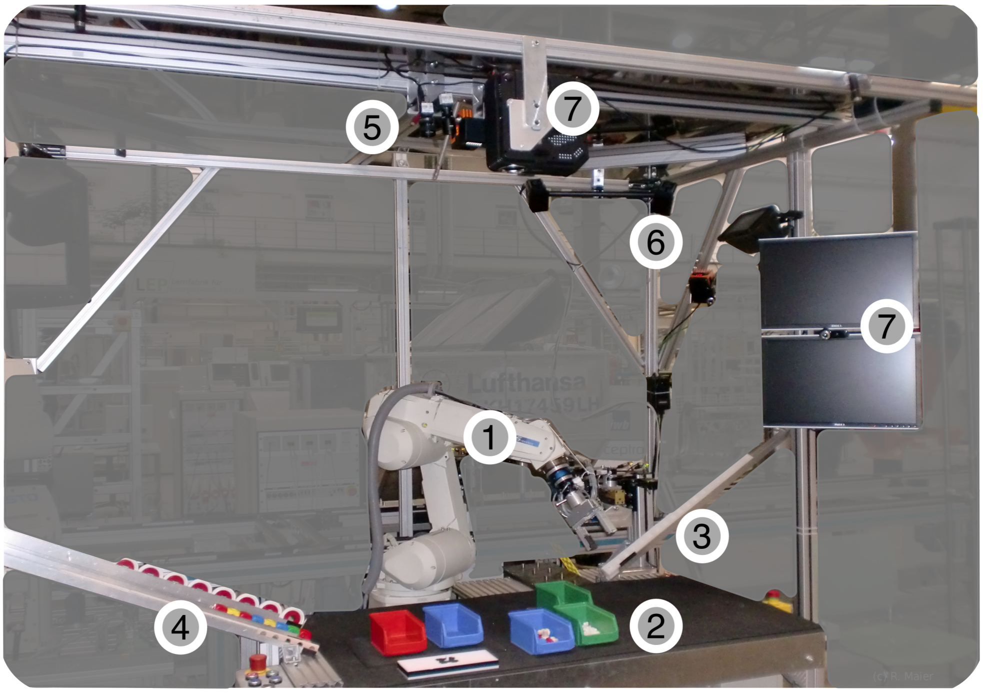


Figure 5.1: Hardware platform *JAHIR* - ① industrial robot, ② shared workbench, ③ slide for tower parts, ④ slides for car parts, ⑤ CCD and depth camera devices, ⑥ depth camera and *ARTrack* system to estimate e.g. the human worker's hand, ⑦ output devices (2 monitors, 1 projector) to present the human information about the next steps and the internal representation of the robot

work for experiments and demonstrations. The effects of transferring of the psychological motivated mechanisms action observation and action coordination to the JAHIR *platform are presented and evaluated (Sections 5.3, 5.4.2 and 5.4.3) along with sample applications of the task-based controller (Section 5.4.1).*

5.1 Hardware design

The demonstration platform *JAHIR* —Joint-Action for Humans and Industrial Robots— has been created and embedded in the *Cognitive Factory* scenario of CoTeSys[1] in close cooperation with project partners from the *Institute for Machine Tools and Industrial*

[1] `http://www.cotesys.org`

Management–iwb[1] and the *Institute for Human-Machine Communication–MMK*[2]. The *Cognitive Factory* is one of the central demonstration scenarios of CoTeSys and focuses on advancing today's production processes with cognitive capabilities. To cover all aspects of production, a fully automated production, a manual working desk, and the *JAHIR* set-up as central in-between station, that combines both *worlds* have been created.

The *JAHIR* set-up was publicly presented at several events including the Automatica 2008[3] and the Schunk Expert Days 2009[4].

5.1.1 The robotic platform

The *JAHIR* system has been designed as a generic robotic system to analyze and show a variety of concepts regarding collaboration aspects between human and robot. As depicted in Figure 5.1, a standard position controlled industrial robot (Mitsubishi RV-6SL) is placed on a working table. The robot has six degrees of freedom and has a maximum payload of six kilograms. Since the robot has a manipulation sphere of 0.902m, the robot can reach almost the whole desk, which is 1.4×0.7m. The tool center point of the robot is extended with a force-torque-sensor from *JR3*[5] and a tool-change unit from *Schunk*[6]. Two tool change stations with three tool ports each are placed in the workspace. With this, the robot is able to change, store and use different kinds of end effectors according to the task. Currently the following tools are available to be changed autonomously by the robot:

- to imitate gluing operations with the robot, a pen can be mounted as tool
- to perform a close-up inspection, a gripper holds a camera
- to jointly screw, an electrical drill gripper can be attached to the robot. The drilling speed is controlled by the applied force

[1] `http://www.iwb.tum.de`

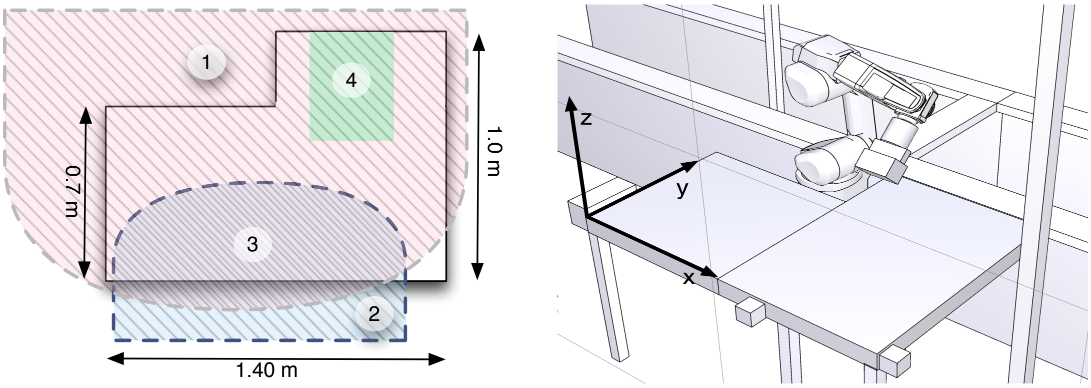
[2] `http://www.mmk.ei.tum.de`

[3] `http://www.cotesys.de/news/articles/cotesys-at-automatica-2008.html`

[4] `http://www.cotesys.de/news/articles/schunk-expert-days.html`

[5] `http://www.jr3.com/`

[6] `http://www.schunk.de`



(a) (b)

Figure 5.2: Different workspaces of human and robot - left: ① workspace of robot; ② workspace of human; ③ shared workspace; ④ storage for robot; right: The world coordinate frame is on the left corner of the shared working desk

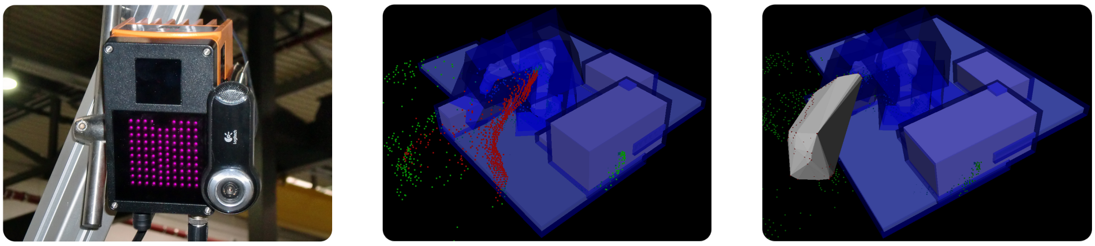
- with a pallet gripper, the robot can fetch parts and tools from the assembly line in the back of the set-up
- with a big parallel gripper, the robot can grasp boxes on the working table

Depending on the requested robot operation, a gripper management module initiates the changing towards the appropriate gripper autonomously.

Human and robot can jointly use a workbench and partly share the same workspace. In this way, both human and robot have areas where they can work for their own and on the other side where both partners can work together. Figure 5.2(a) depicts an exemplary split-up with ① being the workspace of robot; ② being the workspace of human; ③ being the intersected shared workspace; ④ being storage for robot that is not reachable for the human. Hence, human and robot are brought closely together for a diversity of collaborative assembly applications.

5.1.2 Input and output devices

To perceive the environment, several types of sensors were evaluated and integrated. This includes CCD cameras, depth cameras, Microsoft Kinect cameras, and an infrared based marker-tracking system. Further, microphones capture utterances of the human to allow a natural way of interaction with the system. Beside the robotic arm,



(a) (b) (c)

Figure 5.3: Building obstacles from depth data - Three-dimensional data points delivered here by a extrinsically calibrated PMD camera (a) are tested against the static and dynamic environment to include unknown obstacles into the environment model. In this example, a worker is grasping inside the shared workspace (b). The green points represent either points that come from already known objects or are irrelevant, because they are outside the working range of the robot. The red points are unknown obstacles combined to a new obstacle (c).

two monitors, a speech synthesis unit, and a projection unit are mounted on a scaffolding around the shared workspace (see Figure 5.1) to inform the human worker. As it has been described and explained throughout the thesis, techniques are employed to observe and understand relevant actions in the surrounding of the robotic system including tracking parts of the human (see Section 3.3). This information is then combined with a task-based control of the industrial robot (see Section 4.1) to produce dynamic motions respecting e.g. geometric constraints and to be used to adaptively coordinate the actions performed by the robot (see Sections 4.2 and 4.3).

CCD cameras

Several color CCD cameras are mounted on different locations around the set-up and are used for a variety of tasks. This includes the position estimation of colored boxes on the table as explained in [187] or the triggering of so-called *soft buttons* as described in [188]. Further the tracking of the worker's hands are performed using multiple camera views as described in Section 3.3.2.

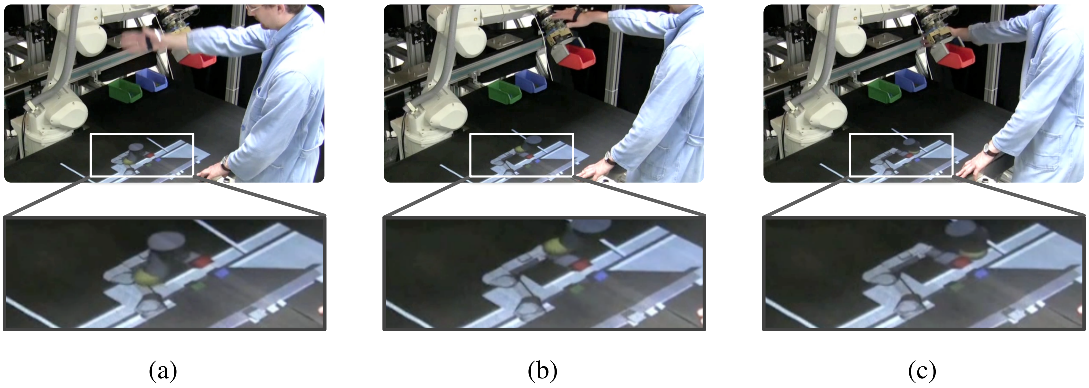


Figure 5.4: Arm tracking with the ARTrack - The human arm was tracked with the ARTrack marker-based tracking device using markers for hand, lower, and upper arm. The results were taken by a specific module that interfaced the three-dimensional scene representation.

PMD cameras

This kind of camera is bases on the time-of-flight principle [189, 190]. It is employing a solid state imaging unit comparable to a CMOS device and a modulated light illumination rather than a single laser beam as laser scanners use. Each pixel measures the round-trip-time of the modulated light and computes the three-dimensional data point. While the resolution is limited and different colored areas (black/white) lead to an increase in the measurement noise, the PMD technology seems to be the most appropriate technology to survey shared workspaces [191]. Therefore, an industrial PMD camera[1] is integrated in our scenario.

The very limited resolution of the used PMD camera (64 x 50 pixel) obviates a direct extrinsic calibration that is needed to associate the 3D data points with the static and dynamic environment model. Therefore, a webcam was attached to the PMD camera to perform the extrinsic calibration (Figure 5.3(a) left). The transformation between PMD camera and webcam can be better estimated, because the distance is very small and the calibration pattern can be seen in both cameras in an appropriate size. An additional depth calibration step can be performed following [192].

As shown in Figure 5.3(b), the three-dimensional points from the depth camera are tested against the environment model using efficient bounding box tests to classify

[1] ifm efector pmd O3D200 (`http://www.ifm.com/ifmde/web/dsfs!O3D200.html`)

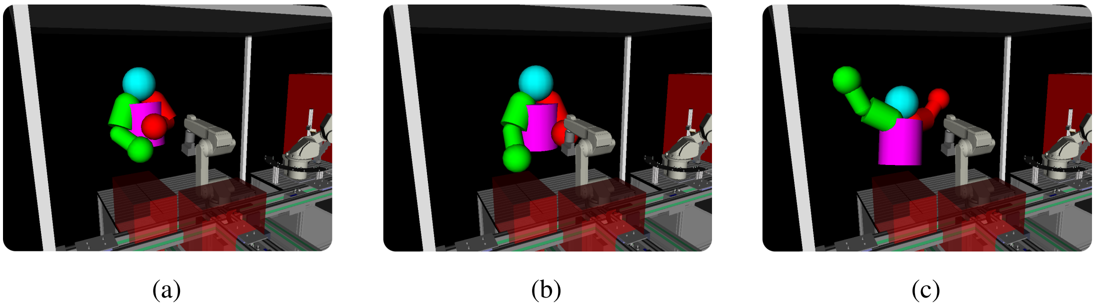


Figure 5.5: Skeleton Tracking with Microsoft's Kinect - The human worker is tracked using Microsoft Kinect and the OpenNI[1] framework. The results are taken and update the corresponding objects in the scene representation.

whether the measurement point originates from an known or irrelevant object (green points) or from an unknown (red). The unknown points form new obstacles using the *quickhull* algorithm [193].

ARTracking device

A high-end commercial marker-based tracking system from *ARTrack*[2] can be used as reliable data source with a high update rate (60Hz) for experiments. The passive markers can be attached to the human to enable e.g. the tracking of his hands or the body center. An example, where the markers were integrated in a smock to estimate the position of the full arm is depicted in Figure 5.4. As can be seen in the figure, the arm is tracked and the tracking module updates the geometric representation of the robot, that is projected on the table.

Microsoft Kinect

Recently, Microsoft released the Kinect camera originally designed for their Xbox. The Kinect camera has become a popular sensor in the robotics community, because it incorporates a high-resolution depth camera and a color CCD camera. Additionally, with the software from the OpenNI[2] a full body skeleton tracking for multiple persons is performed directly by the camera. For the *JAHIR* set-up, one Kinect was added to

[2] http://www.ar-tracking.de/
[2] http://www.openni.org

estimate the pose of the human worker. An exemplary result is given in Figure 5.5, where the upper body is tracked and corresponding body part approximations (cylinders, spheres) are added to the scene representation and updated. In combination with the task-based controller (see Section 4.1), collisions can be avoided with all estimated body parts. A demonstration video can be accessed online[1].

Output devices

The *JAHIR* system can communicate visually and auditory with the human. Two monitors mounted on the height of the human head, give the system the ability to display for example important information or assembly instructions. Further, a projector facing down from the sensor scaffolding can project directly onto the workbench and can be used for the same informative purpose. Additionally, virtual buttons can be adaptively arranged and displayed directly on the workbench. A button manager uses the information of the hand tracking modules to estimate if a button is triggered and broadcasts this information.

5.2 Software architecture

5.2.1 Applied middleware

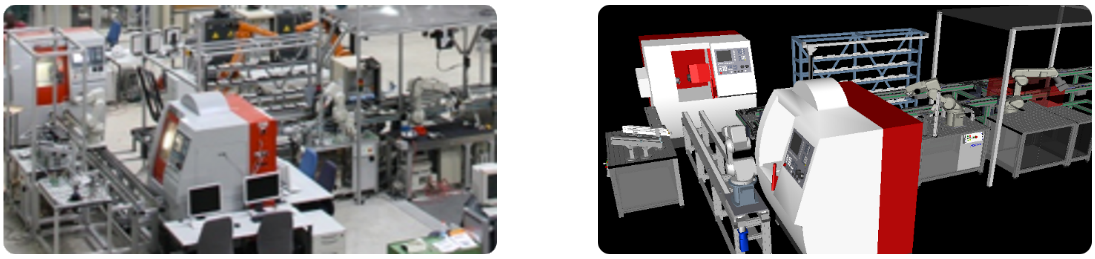
The software architecture of *JAHIR* is oriented on the design principles presented in Section 2.4. This includes among the other principles, multiple processing modules that are inter-connected via dedicated communication channels are used. The integration of the different software components and processing modules was done using not a specialized robotic middleware as the ones presented in Section 2.4.2, but the generic middleware *Internet Communications Engine (ICE)*[2] that allows a seamless distribution of modules among multiple computers.

Although, the number of already available modules makes other middlewares such as ROS[3] very interesting, the *Internet Communications Engine–ICE* was directly used as middleware, because it supports almost all operating systems and most object oriented programming languages. This is not supported by other middlewares including

[1] `http://www.youtube.com/watch?v=MsNbEBSC4UU`

[2] `http://www.zeroc.com`

[3] `http://www.ros.org`



(a) (b)

Figure 5.6: Reality and Simulation - Impression of the overall *Cognitive Factory* scenario in reality (left) and represented in the three-dimensional simulation on the right

ROS up to now. Additionally, the fundamental basic of the software architecture that is used in the *JAHIR* set-up was started with the *JAST* system [33]. Therefore, this work based on the *JAST* architecture and refined and improved it accordingly. But, since the middleware is “just” the glue between the modules, the ideas and concepts presented here can also be mapped to other middlewares.

5.2.2 Modules and inter-module connections

As mentioned in Section 2.4, the information flow needs to be steered and controlled in the system. This idea was not directly applied here, because processing modules broadcast their results system-wide via communication channels. Modules that are interested in specific information *listen* to the needed information. In this way, we get implicit connections between modules without the need of a central instance to explicitly control the information flow. This so-called *publish/subscribe* principle is used to communicate results and information to an unknown number of interested modules. Since this is a one-way communication pattern, the broadcasting module does not come to know if information is actually used. To encapsulate different kinds of information, the meta-channels *scene, input*, and *output* are established.

Scene modification

In collaborative tasks, robots are acting in highly dynamic environments with objects that appear and disappear or humans that move inside the robots workspace. Especially the area of the workspace that can be accessed by both, the human and the robot (see

Figure 5.2), needs to be reliably perceived by multiple sensor devices. But besides the actual capturing of this information (see Section 5.1 and Chapter 3), the robot needs to *know* the static and dynamic surrounding. An appropriate representation for this information is to directly *transfer* the three-dimensional position and physical dimensions of objects. This representation can then be used to avoid collisions with both static and dynamic objects including the human and other moving obstacles.

The used scene representation is based on the scenegraph of the *Robotics Library (RL)*[1] [194]. The scenegraph is built up from the nodes *Model, Body* and *Shape*. The unique base node *scene* can have multiple *models*. Models join multiple *bodies* together and bodies encapsulate a compound of possibly multiple *shapes*. Shapes can be instantiated as one of several basic blocks including boxes, spheres, and cylinders or be advanced like polygons or convex hulls. The implementation in *RL* abstracts several possible backends while using the same file format (*VRML*) and and opens therefore the possibility to use the same models for visualization (using *OpenInventor*), distance computation (using *Solid*) as needed in the robot controller, or for physic simulation

[1] available under `http://roblib.sourceforge.net`

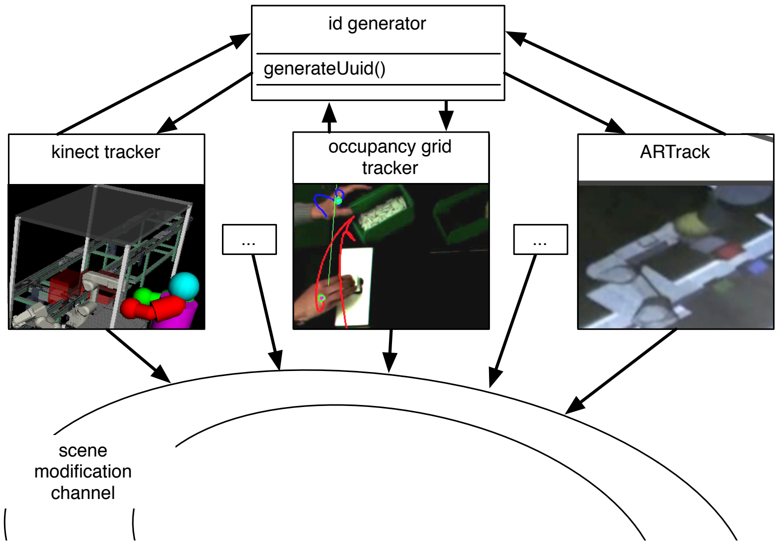


Figure 5.7: Publishers of the scene modification channel - Every sensor module that gathers information about the geometric environment (e.g. the ARTracker or the Kinect skeleton tracker) can add, update, and remove objects. The objects are added with a unique id that is generated by a centralized instance: the *id generator*

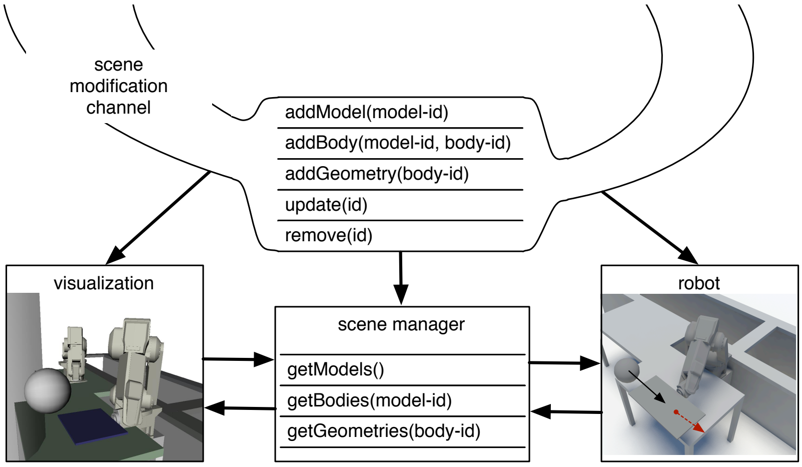


Figure 5.8: Subscribers of the scene modification channel - Modules that are interested in changes of the geometric environment (*scene*) subscribe to the *scene modification channel*. The *scene manager* contains the current context and can provide this information to modules that been e.g. crashed to provide the missing context to their representation

(using *Bullet* or *ODE*).

In the *JAHIR* set up with its dynamic environment characteristic, the scene representation needs to be influenceable at all times to add, update, or delete objects. Therefore, the scene can be interfaced from all modules that provide relevant information about the surrounding. Each module can modify its objects through a *scene modification channel* as shown in Figure 5.7. New objects are added to the scene with generated ids and are therefore uniquely identified within the scene. The update on objects takes place on the basis of these unique ids. Since the used publish/subscribe methodology is event-driven, modules that have stopped due to unforeseen events might loose information about the current context. Therefore, a *scene manager* was integrated to preserve the current context. During the start up of a module, it asks the scene manager for all known objects and adds them to the own representation. This is related to the design principle *robustness* as presented in Section 2.4.

The robot has an optimized geometric representation of the scene that allows the computation of distances between the parts of the robot and the objects in the surrounding as shown in Figure 4.2 to compute virtual forces that repel the robot from obstacles as described in Section 4.1. The robot is connected to the *scene modification channel* and *listens* therefore to all modifications on objects. An example is given in Figure 5.4,

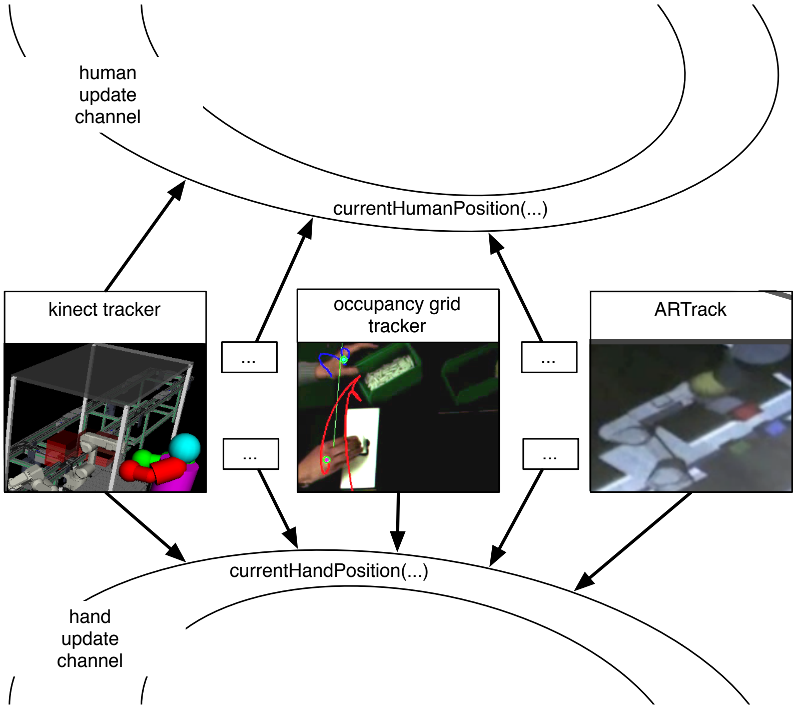


Figure 5.9: The human and hand update channels - Every sensor module that estimates the position of the human body or the human hands, communicates these results over the *human update* and the *hand update* channel. Modules that deliver such information include the Kinect skeleton tracker, the ARTracker, and the occupancy grid tracker. Further, safety mats or a laser range finder under the table can also deliver estimation about the human's position

where the arm of the human worker is perceived employing a marker-based approach. The module that handles the estimation of the marker positions adds cylinders for upper and lower arm and a sphere for the hand to the scene. Further, the processed arm estimation is directly projected onto the tabletop of the workbench.

Input

The *input*-channels incorporate all communication channels that deliver input information to the system. This includes for example the position of the human and his hands. Due to the chosen architecture, different modules that do hand tracking are exchangeable as long as they communicate their results over the same communication channel as depicted in Figure 5.9. The information that is communicated includes the current

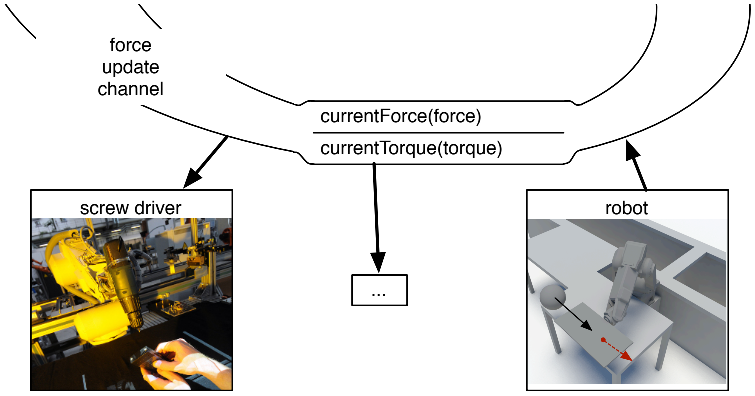


Figure 5.10: **The force update channel** - The force/torque sensor mounted on the hand of the robot broadcasts current measurements for force and torque via the *force update channel*. The screwdriver module listens for example to it and controls the turning speed according to the current force

position and orientation of the hands, the distinction between left and right hand, and a confidence value to enable the possibility to fuse the data of multiple modules that offer information about the same hand. The confidence value is given by the modules itself and can be computed e.g. from the covariance of the Bayesian filter or the distribution of weights in case of a particle filter. Information about the sensor module — e.g. the sensor id or the friendly name of the module— that delivers the information can be added. The same holds for the broadcasting of the position and orientation of the human body center.

Further, the force/torque sensor that is attached to the robot delivers the current forces and torques to interested modules as shown in Figure 5.10. When the robot has the screwdriver mounted on its hand and is in the drilling position, the screwdriver module takes the current forces to adaptively control the turning speed and direction according to the applied force.

The information about the current robot configuration is for multiple modules of interest. Beside the illustration purpose of the visualization module, the configuration is used to filter data points in the surveillance module. This information can also be used along with the estimations about human body position to compute the current distance between human and robot. The estimated distance can then be used to adaptively control the robot's velocity according to e.g. the current norms presented in Section 2.2.1.

Output

Opposite to the event-driven publish/subscribe principle that is used in the other communication channels to broadcast e.g. sensory data, the connection to the robot is client/server based. With this connection, the calling module can get return values to know, if errors occurred in the accomplishment of the task. Further, a direct connection can be established to stop the motion of the robot at all times.

The robot offers several basic functionality through its interface. This includes abilities to pick up an object at given position on the table, or the hand over of an object that is in the hand of the robot. For a hand over, the robot moves to a hand over position and waits for the human to grasp the object. If the force/torque sensor measures the grasping, the robot opens the gripper and moves away. By concatenating multiple basic operations, a complete building plan can be teached-in. This can also e.g. be done using a speech recognition module as shown in [187].

Additionally, the output channel includes the triggering of so called *soft buttons* as firstly presented in [188] and improved and integrated in this work. A soft button server is connected to the projection unit. External components can add and remove wanted buttons on the projection area. Using the information about the current hand positions, the buttons can be triggered. Further, the triggered buttons are communicated via an own channel to initiate connected behaviors.

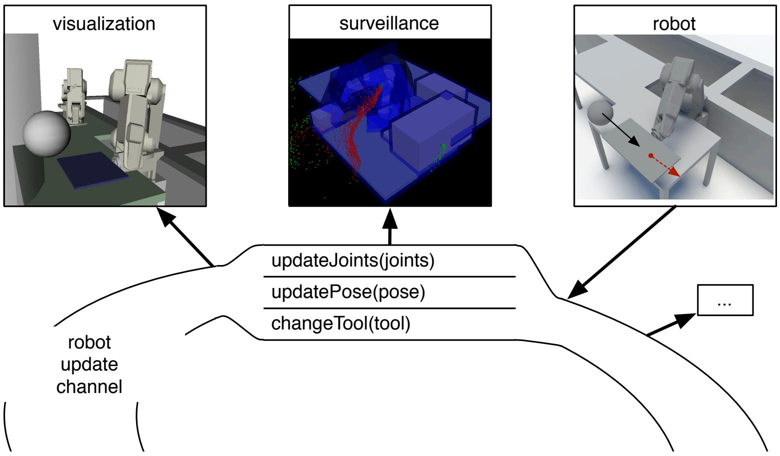


Figure 5.11: The robot update channel - The robot controller broadcasts the current joint positions via the *robot update channel*. Modules that listen to this information are e.g. the *visualization* that shows the current situation, or the surveillance module

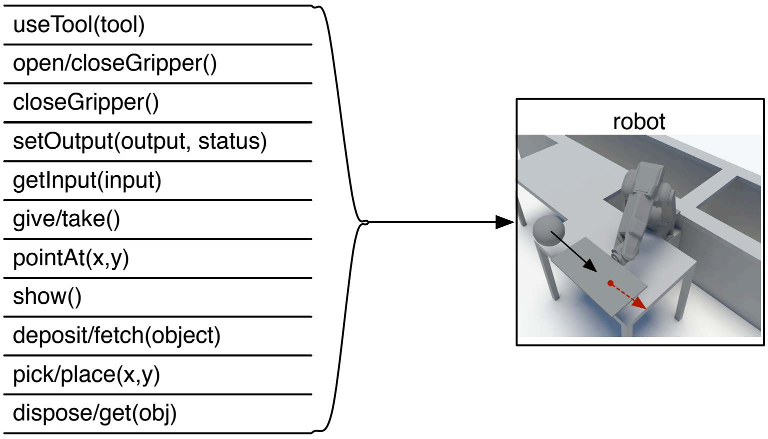


Figure 5.12: Connection to the robot - The connection to the robot is based on a client/server pattern. Modules can call basic skills of the robot via the presented interfaces

First order logic system control

In *JAHIR* a first order logic system[1] as knowledge-based event controller based on facts and rules. Facts represent all kinds of information. This includes knowledge about the environment (e.g. the position of objects) and about skills of the connected modules. E.g. the robot registers its basic actions including *move to position, open gripper*, and its higher-level actions including the picking up of an object from a given position on the table. Hence, the system controller can access these abilities without any knowledge about the real hardware. Further, facts include events that can be triggered by external signals. All kinds of alterations in the working memory of the rule engine including updating, deleting, and creating facts or triggering events, initiate the check of associated rules.

The available abilities of the system can be combined and build a complex workflow description. A worker can teach in such descriptions via speech using a speech recognition program with a grammar that is automatically adapted to the abilities of the system. This plan can then be executed respecting environmental information and events. To get an impression videos about the teach in step[2] and the execution step[3]

[1] The Java based rule engine *Jess* (`http://www.jessrules.com/`) was used here.

[2] `http://www.youtube.com/watch?v=6Jnuqxa6PGc`

[3] `http://www.youtube.com/watch?v=hN92vOxzKu4`

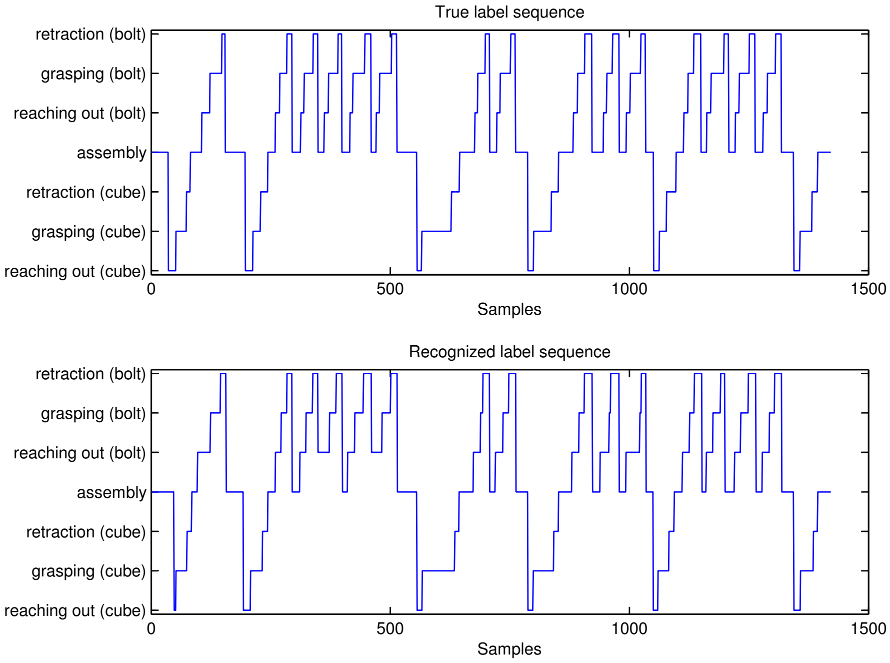


Figure 5.13: Recognized Workflow - Ground truth label sequence (upper diagram) and recognized one (lower diagram) of the right hand using the camera tracking dataset [90]

are available online. For a detailed description on the knowledge-based system components please refer to [187].

5.3 Transferring action observation

The observation of human actions by an assistive robotic system enables pro-active behavior based on the current action of the human. The realization of this concept advances the collaboration between human and robot in the context of production processes by being able to prepare future steps, to analyze the action sequence if errors occurred, or to warn the user if a step was possibly not performed [195].

In an industrial setting such as the *JAHIR* scenario, it is desirable to extract the information needed to analyze and recognize the workflow by means of non-invasive methods. Opposite to [195], where body worn accelerometers and microphones were used as input to classify actions and to estimate the progress of an assembly task, no attached sensors or artificial markers are used to analyze the workflow of the human

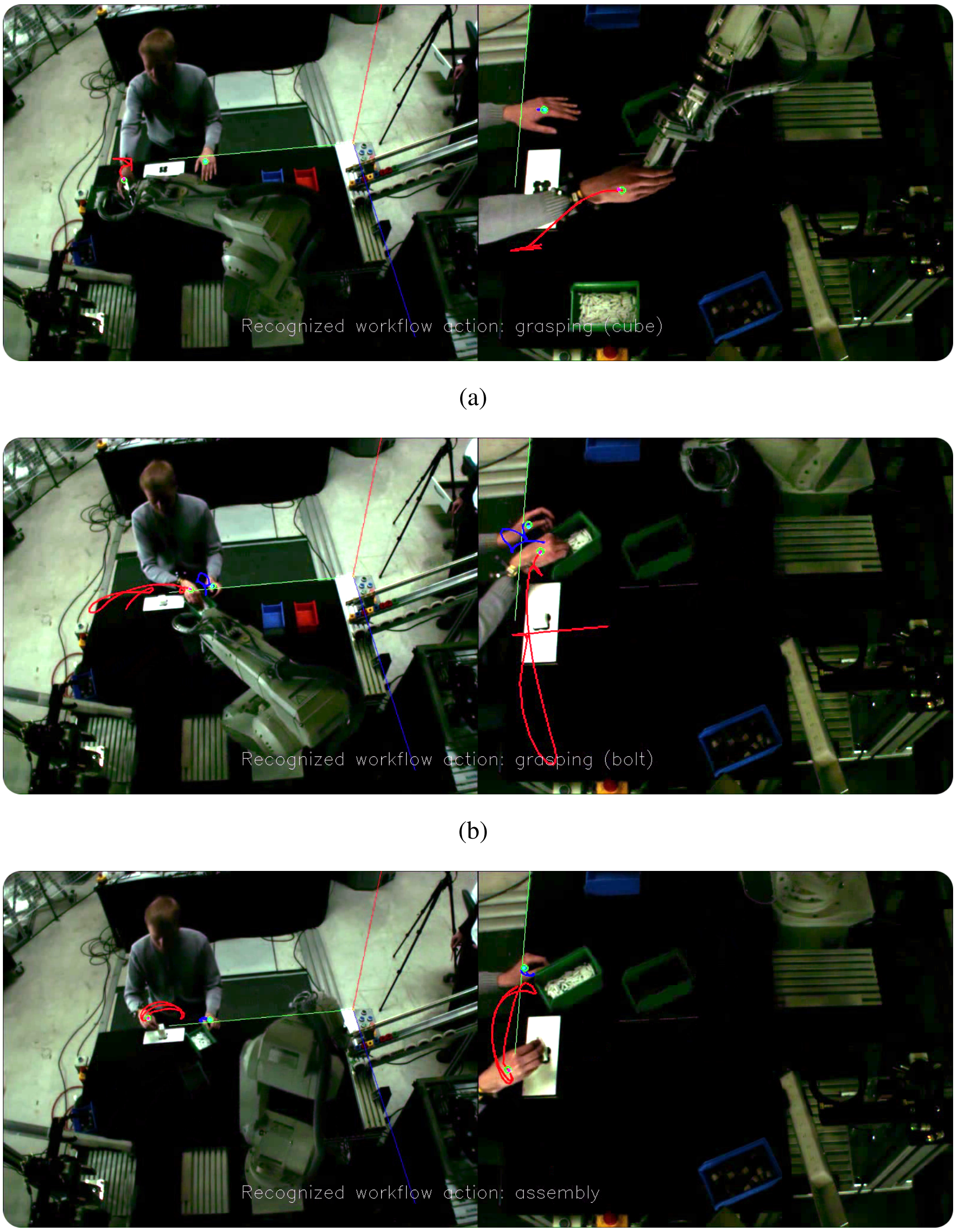
Table 5.1: Workflow recognition results for the *JAHIR* set-up - The accuracy is the percentage of labels, which correspond to the true ones. Correspondence also includes being in the same general movement. The results of the recognition rates for the right hand show that a transfer of the trained models to another set-up is possible [90]

set-up	data set	accuracy (%)
BAJA	all data	95.67 ± 5.07
BAJA	hands	95.11 ± 5.20
JAHIR	camera tracking data	92.26

here. As described in Section 3.1.3 the workflow of the tower building task can be sufficiently recognized by employing only the three-dimensional velocity, acceleration, and jerk of the hands and the activation level of table zones. Having the three-dimensional position of both hands, all necessary data for the feature vector can be approximately derived knowing the update rate: velocity, acceleration and jerk. The location of the active table zones needs to be adjusted to the geometry of the recorded workspace. Thereupon, the activation can be determined from the position of the hands. An alternative to find the activation level of a table zone would be for example to use extra cameras or sensors that recognize hand in the area of the associated camera. Due to its accurate and fast results, the three-dimensional occupancy grid approach presented in Section 3.3.2 is used to estimate the position of the hands. Additionally, every sensor that provides information about the current hand position could be used for an online version of the analysis module, if it is connected to the hand update channel.

To analyze the workflow of the collaborative assembly task between human and the *JAHIR* robot, the experiment presented in Section 3.1 with subjects that were not influenced by a robot or any other technical device were used as basis. Since the workflow analysis is now used in the *JAHIR* environment, HMMs were trained on all 22 persons of the basic experiment 3.1.1 and tested on camera tracking data. This corresponds to using a pre-trained model for the real setting.

An accuracy of 92.26 % was reached for the right hand. Compared to the previous results presented in Table 3.1), the estimation for the right hand is just marginally lower. In fact, a direct comparison between the true and the recognized label sequence shows that all grasps of cubes and bolts are correctly identified. The left hand was


Recognized workflow action: grasping (cube)
Recognized workflow action: grasping (bolt)
Recognized workflow action: assembly

(a)

(b)

(c)

Figure 5.14: Recognized workflow actions - The figures show snapshots of the workflow action recognition. The human is grasping the cube in (a), a bolt in (b), and puts the bolts in the tower cubes (assembly) in (c)

not analyzed, because in the recorded test sequence, the subject was only using his right hand. As shown in Figure 5.13, only the boundaries of the movements were not exactly recognized. Figure 5.14 illustrates the recognition results. The experiments show that the approach of combining three-dimensional occupancy grid hand tracking with pre-learned HMMs worked well in the experiments. The results also show that the recognition of assembly actions can be abstracted from the sensory input data and the set-up. This allows a transfer of pre-learned models to be applied on real environments [90].

5.4 Transferring action coordination

5.4.1 Applications of the task-based robot controller

As stated in Section 4.1 with the presented task-based hierarchical control structure of the industrial robot, a magnitude of tasks that can be used in production scenarios can be intuitively solved. In this Section, two sample applications are presented using the *JAHIR* platform.

Mobile Storage Box

In manual production, the efficiency of the current production step depends highly on the availability of parts. If different parts needed for a certain step are always within reach, the human can take them efficiently. On the other side, parts that are pre-assembled and not needed at the moment, need to be placed somewhere where they can be accessed easily when required. In the first application scenario, we use the industrial robot as a mobile storage box. Parts can be placed in the box and the robot needs to guarantee they are not falling off. To be always within reach, the box follows the human hand, but avoids collision with it and the surrounding environment. Regarding these requirements, the controllers are arranged to compose action A_1 as follows:

$$A_1 = T_{\text{orientation}} \lhd T_{\text{avoidance}} \lhd T_{\text{position}} \lhd T_{\text{posture}}. \qquad (5.1)$$

$T_{\text{orientation}}$ is the task with the highest priority, which takes care of keeping the box always in a horizontal orientation. The operational position controller is used here

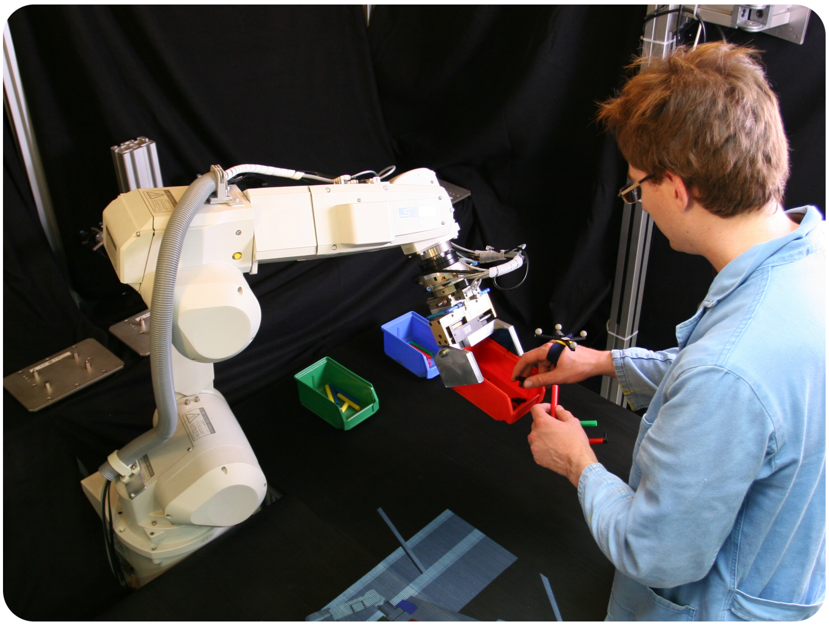


Figure 5.15: Mobile storage box (real) - The robot has a storage box as tool which follows the human hand in safe distance and avoids collision with the hand and the surrounding, so that the human can pick up parts or place assembled products in there [174]

with the selection matrix

$$S_{\text{orientation}} = \operatorname{diag}(0,0,0,1,1,1) \tag{5.2}$$

to fix the orientation of the box. Task $T_{\text{avoidance}}$ avoids collisions with the surrounding environment and the human hand. The position task T_{position} follows the human hand through updates of the hand tracking system to the goal position x_{goal} that should be 0.1 m below and in front of the hand. To keep the position fixed, the selection matrix

$$S_{\text{position}} = \operatorname{diag}(1,1,1,0,0,0) \tag{5.3}$$

is used. In the posture task, we defined that the robot should have an upright joint configuration. Because the posture task has the lowest priority, we can include all joints in the velocity calculation. The result of this experiment is depicted in Figure 5.15. The robot is carrying a red box with parts in its gripper, so that the human can grasp out of the box. To be able to grasp something, the motion of the robot is stopped through the avoidance task that measures the distances. This is done in the controller with a defined distance, to stop the current motion, if the distance of robot and obstacle (i.e. hand) is touching or is below.

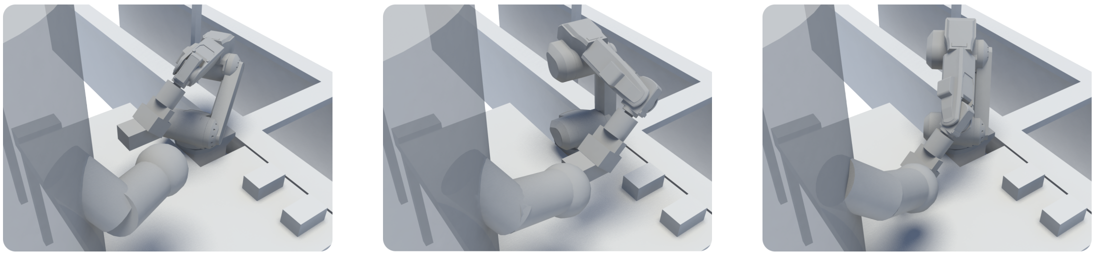


Figure 5.16: Mobile storage box (internal representation) - (a), (b) and (c) show the three-dimensional internal representation of the surrounding and the robot behavior at some instances in time [174]

As the human starts to move his hand from left to right in the workspace, the box starts to follow the hand. On the way back, the robot avoids the collision with the hand (Figure 5.16(a) and Figure 5.16(b)) and converges in Figure 5.16(c) to the resting position of the hand 0.1 m below and in front the hand.

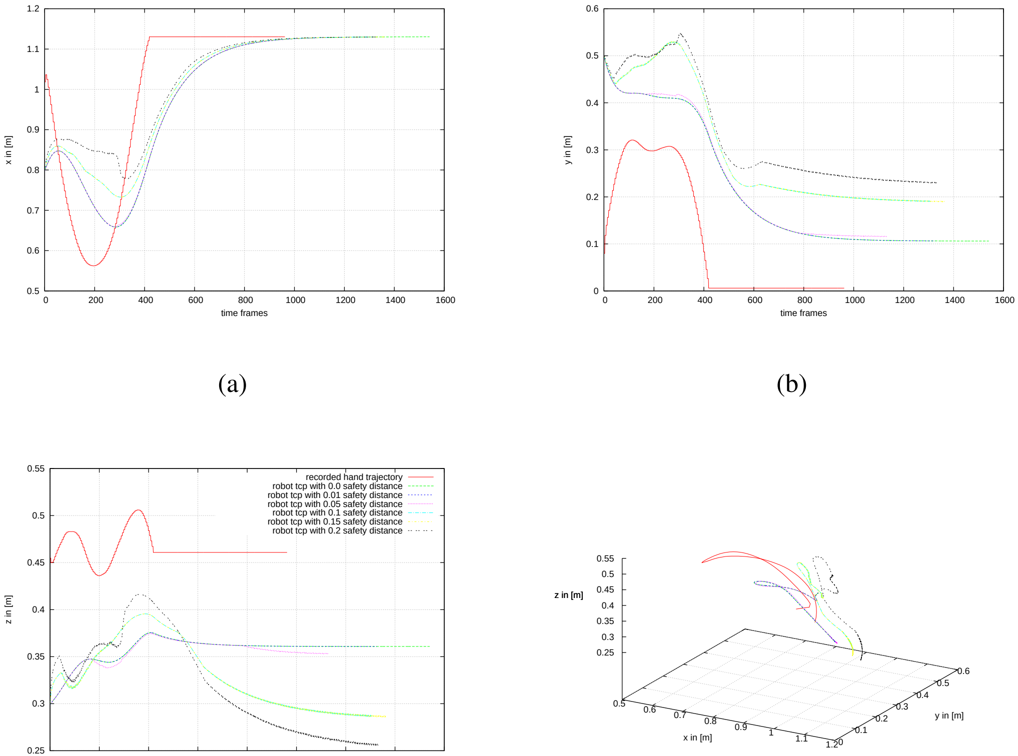
The human hand is added to the geometric representation as sphere with a diameter of 0.1 m. The distance computation between the hand of the human and the robot starts at the surface of the sphere. Additionally, the choice of a safety distance influences the respond of the potential field. As shown in Figure 5.17, where a recorded hand trajectory was used to enable a comparison, the robot follows the hand and avoids collision with it with different strength depending on the safety distance. It can also be seen that even with no given safety distance, the robot converges in its position 0.1 m below and in front of the hand due to diameter of the hand. Hence, the safety of the human is secured in two ways:

1. The size of the shape is bigger than the real hand and if collision shapes touch, the velocity of the robot is suspended until the collision is cleared.

2. The area in which the virtual forces are applied to the robot can be adapted and even updated during run-time.

Direct Line-of-Sight

In the second demonstration scenario, a top-mounted camera directed towards the working table is used to recognize, inspect and track objects lying on the table as de-



(a)

(b)

(c)

(d)

Figure 5.17: Influence of safety distance on the robot's trajectory - The diagrams show the influence of the safety distance on the robot's trajectory in x, y, and z direction. The hand trajectory was recorded and tested with the mobile storage box task. The safety distance in which the potential field generates virtual forces was varied. The diagrams show that the robot follows the trajectory of the hand with some latency, avoids collisions with the hand, and converges at different positions below and in front of the hand depending on the chosen safety distance. The hand is approximated with a sphere with a diameter of 0.1 m. This distance is always secured, because, if the robot would touch the sphere, the motion would be suspended

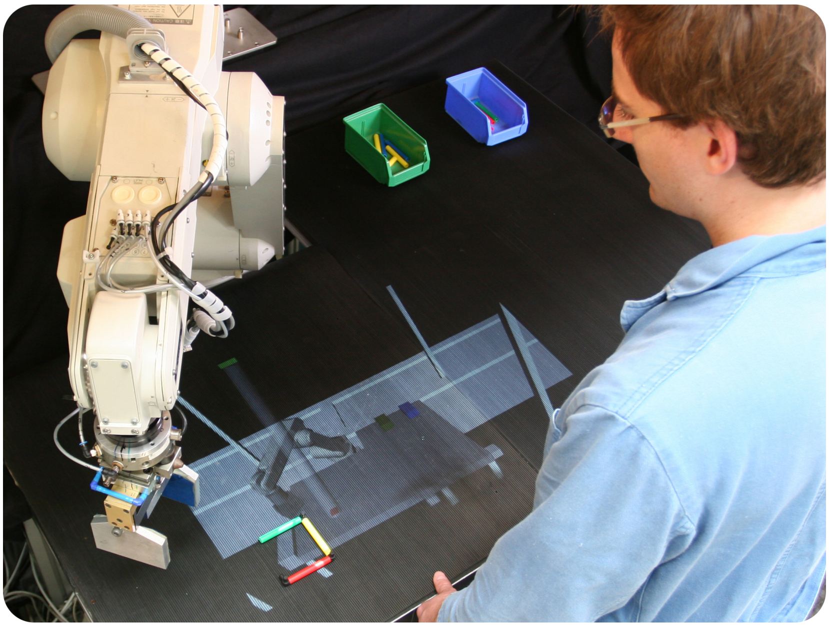


Figure 5.18: Direct line-of-sight (real) - The robot picks up an object at a position and needs to place it on another specified position. A top-down camera is inspecting another object on the table and needs to have always direct line-of-sight for this task. Therefore, the robot needs to find a way around the line from camera to object, respecting also the human working in the same workspace [174]

picted in Figure 5.18. To recognize objects reliably or to inspect objects according to defects, the camera needs direct line-of-sight for a certain amount of time. The robot should not be stopped, because it can fulfill other tasks in the meantime—including picking up an object at a position and placing it somewhere else or handing over tools needed for the next assembly steps to the human.

What needs to be considered here is the issue that the collision avoidance with the environment has to have a higher priority than to avoid the crossing of the line-of-sight of the camera with the robot. Therefore, two collision avoidance controllers with different collision scenes need to be specified.

If the action of the robot is again described according to the controller scheme, the action A_2 can be defined as:

$$A_2 = T_{\text{environment}} \lhd T_{\text{orient.}} \lhd T_{\text{line-of-sight}} \lhd T_{\text{position}} \lhd T_{\text{posture}}. \tag{5.4}$$

$T_{\text{environment}}$ is the collision avoidance controller without the line-of-sight as depicted in Figure 5.19(c).

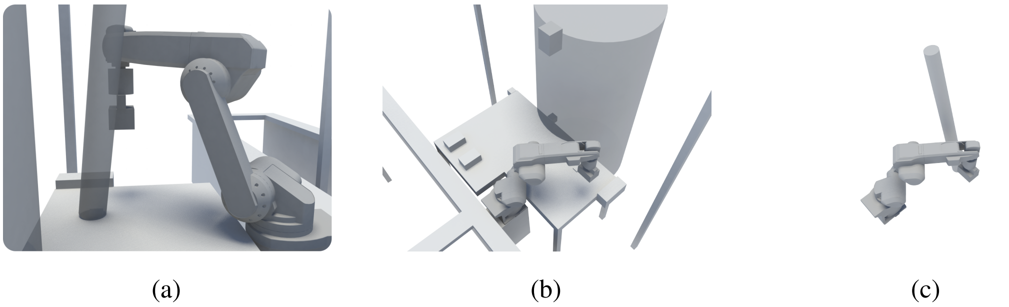


Figure 5.19: Direct line-of-sight (internal representation) - (a) shows the overall three-dimensional internal representation. To avoid collisions with the environment (b) and the line-of-sight of the camera (c) different three-dimensional scenes are used in two instances of the collision avoidance controller [174]

$T_{\text{orientation}}$ is the controller taking care of the orientation of the gripper with the same selection matrix as in the previous experiment. In the collision controller $T_{\text{line of sight}}$, a cylinder approximates the line-of-sight from the camera to the object on the table as depicted in Figure 5.19(c). The task T_{position} drives the robot to the goal position. The posture task T_{posture} is the same as in the previous experiment.

The images in Figure 5.18 and 5.19 show the behavior of the robot moving from position $(0.1\,\text{m}, 0.2\,\text{m}, 0.3\,\text{m})$ to the goal position $(0.8\,\text{m}, 0.3\,\text{m}, 0.1\,\text{m})$. As depicted in Figure 5.19(b) and 5.19(c), the potential field generated by different collision scenes repels the robot.

5.4.2 Subjective feeling of safety

The hand over experiments presented in Section 4.2 were carried out on a human-human, a human-humanoid, and the *JAHIR* set-up to measure, evaluate and compared timing characteristics such as the reaction times of the human using a trapezoidal and a minimum jerk velocity profile. It was shown that the motion trajectories have an influence on the effectiveness of the task performance with a minimum jerk profile in Cartesian space leading to shorter reaction times (0.86 s) than a trapezoidal velocity profile in joint space (0.69 s) to shortly repeat the results presented in Section 4.2.

In the experiments, the gripper moved at very high velocities with up to $1.74\,\text{m}\,\text{s}^{-1}$ for a trapezoid velocity profile and $1.67\,\text{m}\,\text{s}^{-1}$ for a minimum jerk profile. These velocities were used to reproduce the same duration of the movements as it was determined

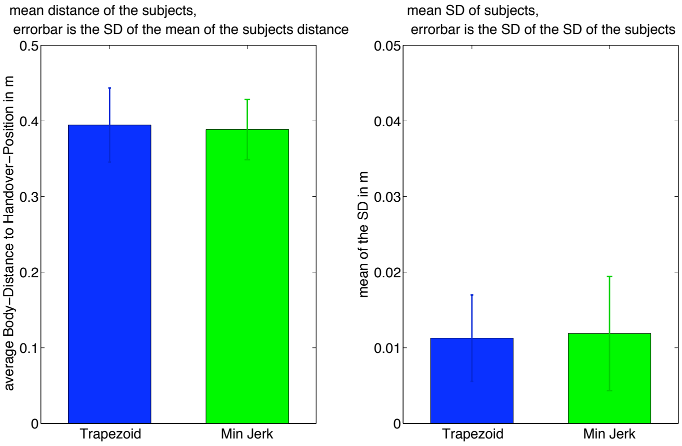


Figure 5.20: Body position of the subjects - The body position in the direction towards the robot in relation to the handing over position is shown on the left. The subjects show only insignificant little body-movements during the experiment towards the robot (right) [180]

for human movements. Observations e.g. in [196] show that the peak velocity of humans arm movements increases as an function of the movement distance to keep the duration of the movement constant around 1.2 s.

The speed limitations of the robot to $0.25\,\mathrm{m\,s^{-1}}$ as defined in current industrial norms (see Section 2.2.1) limit the impact in case of a collision with the human. But if the efficiency is considered, the speed limitation retards an efficient collaboration between human and robot especially in the case of hand-overs. Even the high motion velocities used in the experiments to mimic the time of needed for a human to perform the hand-over could not reach the same task performance as in the compared human-human experiment. This can be partly explained with the high technical disadvantages like the usage of a parallel gripper of the robotic systems compared to humans.

During the experiment on the *JAHIR* robot the middle of the subject's chest was measured to estimate the most comfortable relative position of the subject and the fixed hand-over position. The estimated mean body position of all subjects reveals only insignificant small body-movements during the experiment towards the robot (0.01 m for both velocity profiles as shown in Figure 5.20. This indicates that the hand-over position that was transferred from the human-human experiments is also comfortable and valid in the human-robot collaboration scenario.

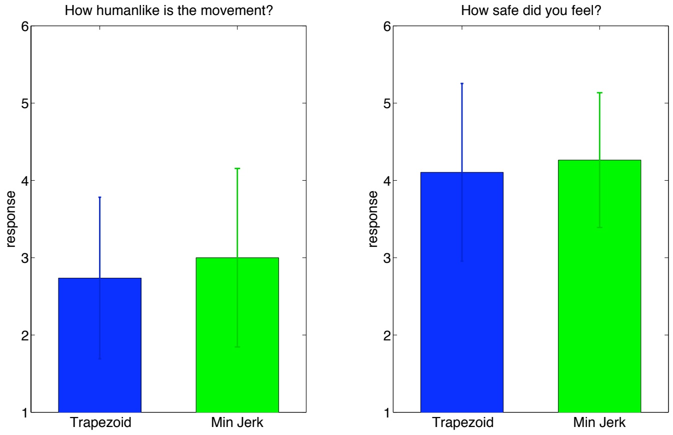


Figure 5.21: Interview of the subjects after the hand over experiments - The subjects had to answer (from 1 to 5) how human-like they thought the robot movement was and how safe they felt during the experiment [180]

The subjects kept the same distance to the table during the experiment for both tested profiles of the robot. The mean distance to the handing over point was 0.39 m for both profiles. The standard deviation of the mean body-position for the subjects hereby is 0.05 m for the trapezoid profile, respectively 0.04 m for the minimum jerk profile (see left of Figure 5.20). The results reveal that there was no discomfort even for the first movement of the robot towards the subjects. Hence, it leads to the interpretation that as soon as the hand-over position is in a region of comfort, humans do not need to further optimize their body position. Further, the results show that the subjects were not surprised during the first hand-overs despite the high absolute velocities (max. $1.74\,\mathrm{m\,s^{-1}}$ for the trapezoid profile, max. $1.67\,\mathrm{m\,s^{-1}}$ for the minimum jerk profile) of the robot gripper moving directly towards the subject. If the robot movement has created any discomfort or fear to the subjects, a significant adjustment or change in the body position would be expected.

In addition to this, the subjects were interrogated on how they interpret different velocity profiles in *JAHIR* set-up regarding *human-like movements* of the robot and the *subjective safety*. The evaluation of the answers in the industrial setting show that there are neither preferences in terms of how human-like the robot movements were nor in a subjective feeling of safety (see Figure 5.21. Despite of the high maximum velocities of the robot system the questionnaire indicated a relatively high feeling of subjective

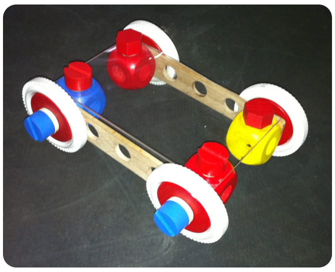


Figure 5.22: Secondary assembly task - A Baufix toy has to be assembled as secondary, more complex object

safety in both profiles (averaged 4.1 scores out of 5 for the trapezoid profile, 4.3 scored out of 5 for the minimum jerk profile).

The post-experiment questionnaire filled out by subjects in the human-humanoid experiment [179] on the *JAST* system also revealed that no difference was recognized by the subjects between the two profiles in terms of a human-like motion. But, in this case, the subjective safety was significantly higher for the minimum jerk profile (Wilcoxon matched pairs test, $p = 0.013$). The different configuration types of the human-like *JAST* and the industrial-like *JAHIR* set-up might lead to the discriminative results of the experiment. In the human-like arrangement of the robot as present in the *JAST* set-up, the motor resonance of the human might be better activated [184].

An explanation for the overall high subjective feeling of safety could be that the subjects were instructed about the task. In this way, the subjects were prepared and could previously adapt based their expectation and were therefore not surprised at all as also shown in the invariance of the body center position. Further that means that if a robotic system can communicate its next steps and move in a predictable way, the safety can be increased, because the robotic motions gets integrated in the prediction and expectation of the human.

5.4.3 Applying predictive timing to *JAHIR*

In [86], the assembly duration prediction has been developed based on the assembly experiment performed by humans without assistance as described in Section 3.1.1. To examine the benefits of transferring timing prediction and corresponding time management to the *JAHIR* set-up, the same tower building assembly task was repeated with the *JAHIR* robot as assistant. In order to evaluate and compare the task efficiency [197]

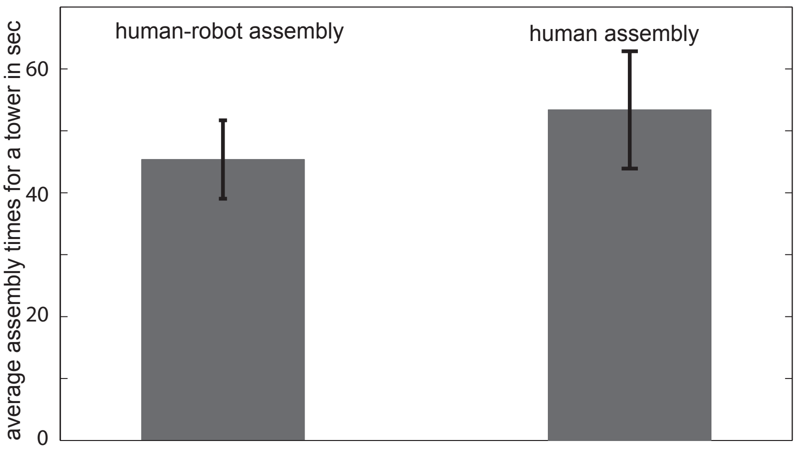


Figure 5.23: Performance evaluation - Duration of the average tower assembly time, with robot assistance (left) and without robot assistance (right). The results show that the efficiency for the tower building task could be increased compared to the case without assistance, but with omnipresent components [185]

of the assistive system with the "human" system, the *time-to-completion* of four towers was used in an experiment with 37 subjects (15 female, 22 male). A video of the tower building task can be accessed online[1].

As shown in Figure 5.23, it took the subjects on average (45.37 ± 6.33) s in order to assemble a tower with robotic assistance. The average assembly time measured with the robot assistance is significantly shorter $(t - test; p < 0.001)$ than the average assembly time (53.39 ± 9.47) s in the experiment presented in [86], where a human had to perform the task alone[2]. This is astonishing, because in the robot-human experiment, additional waiting times for both robot and human—due to small errors in the predictions of the assembly durations—were present as opposed to omnipresent parts in the "human" experiment.

The result indicates that the working speed of the human does not only depend on factors like stress, skill level, and fatigue, but also differs if the task is performed alone or with assistance. Similar to these, an increase of movement speed is also reported in [198] if a pick and place task is performed with a partner.

Further, the adaptability of the predictor was tested. To do so, the first tower was

[1] `http://www.youtube.com/watch?v=J3u-v39vBbA`

[2] `http://www.youtube.com/watch?v=tfW4L7Idpqk`

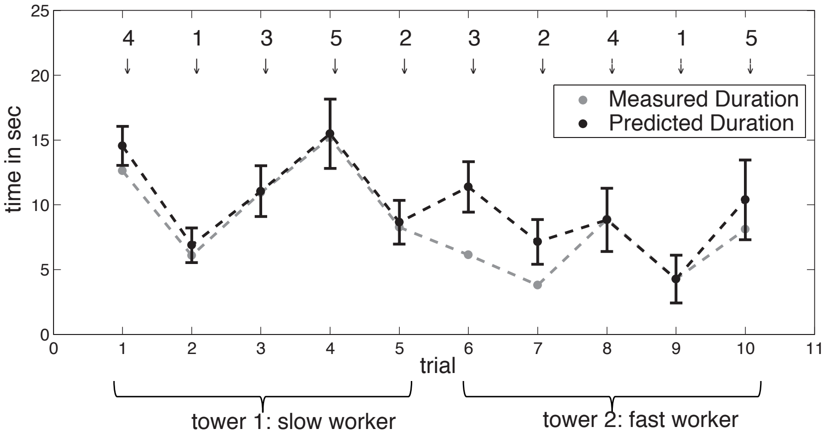
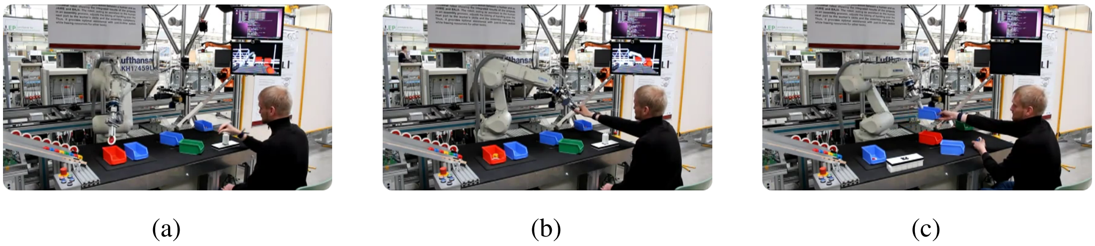


Figure 5.24: Adaptive time prediction - Empirical (black) and predicted duration (gray) over all working steps. The first tower was built very slowly using only one hand while the second tower was built as fast as possible using both hand

built by a subject using only one hand, mimicking a very slow worker. Since the initial parameters of the predictor model a slow performance of the worker, the assembly durations can be precisely predicted at the beginning. The second tower was built using both hands and as fast as possible. The sudden change of the worker's behavior challenges the adaption of the assembly duration predictor. The predictor assumes a slow performance at the beginning of the second tower assembly. This results in an error Δt for the first two cubes of the changed performance. After two imprecise assistive actions, the algorithm predicts the timings correctly again for the fast worker. Figure 5.24 depicts the predicted durations (black) for the slow and fast tower building task along with the measured durations (gray) over the trials. The error bars indicate the standard deviation of the predicted duration. The numbers above the durations indicate the number of bolts the subject had to use in the assembly steps.

In an additional, more realistic experiment, a second product assembly task needs to be assembled: a Baufix toy car as depicted in Figure 5.22. The complexity of the assembly steps of the second task consists of multiple sub-steps and cannot be intuitively specified as for the tower building task. However the generic concept of the predictor allows to continuously update the complexities, even for uncertain assembly steps or a set of assembly-steps. The overall scenario, including both tasks and the time management, is depicted in Figure 5.25. A video of this application scenario is



(a) (b) (c)

Figure 5.25: Application scenario - While the human is performing his assembly task, the robot starts the presorting of objects for future assembly steps (a). If the next cube is needed, the robot reaches the hand-over point just-in-time (b). For the second assembly task, the robot hands over the boxes with presorted parts and can again prepare future steps during the complex assembly steps

also available online[1].

In the second task, the assembly of a Baufix car, the robot hands over boxes with presorted parts needed for the current assembly step. The complexities of the presorted components are far more uncertain than in the previous tower building scenario. Although, the predictor is able to update imprecise initial complexity parameters. The method can therefore be applied to a broader range of applications, as e.g. for the assembly of a Baufix car. For this task, the experiments were repeated with a human acting as a slow worker for the first car. For the second car the human suddenly changes his behavior and to assembles the car as fast as possible. The same ability of the robot to adapt to changes in the human worker's behavior can be observed. During both tasks, the robot uses the predicted hand-over times in order to estimate the time available for presorting the parts into boxes for the Baufix car assembly, according to the description in Section 4.3.2.

The results of the experiments show that working with a robot assistant partner increases the overall task performance. This indicates that the robot partner is not only perceived as artificial assistant, but also as competitive partner. Finishing a task faster than someone expects it might be motivation for the subjects. However, it seems unlikely to find such a motivational effect in long term interactions with an assistive robot system. Further, the experiment showed that the method to predict the assembly duration could be implemented in a human-robot assembly scenario. Furthermore, the efficiency for the tower building task was better than in the case without assistance, but

[1] `http://www.youtube.com/watch?v=Lf2n6HKrNNU`

with omnipresent components. Additionally, the duration prediction method enables the robot system to perform preliminary tasks while it is not needed for assistance. Based on the predictions, a sense-full time management can be developed, thus further increasing the efficiency.

Chapter 6

Conclusion

To conclude the thesis, a review of the presented work and the corresponding contributions are given. Additionally, possible improvements and a glance towards future work are presented.

6.1 Summary

The collaboration between human and robot constitutes a promising approach to supplement current automation and optimization strategies in industrial production. The strength and the efficiency of robots together with the high degree of dexterity and the cognitive capabilities of humans complement one another and can lead to a high-producing team. To investigate the potentials, the demonstration platform *JAHIR* was created together with project partners from the electrical and mechanical engineering department along with this thesis. It is embedded into an overall cognitive factory scenario equipped with one fully automated and one fully manual assembly station for investigating further optimization strategies that exploit cognitive capabilities.

Starting from the assumption that collaboration in a shared assembly task constitutes in the ideal case a well-coordinated sequence of actions in space and time, the thesis follows previous psychological findings, that describe how humans perform joint actions. It was observed, that the success of human-human collaboration depends on certain abilities including the ability to prediction and the ability to affect own actions based on these predictions. Several mechanisms are responsible for these abilities. To realize these mechanisms, the robotic system requires several capabilities including perceiving, gathering and representing contextual information such as the environment, recognizing current actions of the human, and from this deducing context-aware dynamic and adaptive actions. These demanded capabilities are conceptually realized in a distributed, modular software framework.

The work of this thesis co-developed and contributed to the general purpose tracking library *OpenTL*[1] which includes well-known as well as novel computer vision algorithms and many Bayesian filter concepts. In particular, real-time approaches for a robust tracking of hands have been developed to be applied in this work for action observation. Further, multiple sensors have been integrated and used to perceive the geometric context.

With the realization of proposed architectural and software design principles, a modular and generic framework based on a publish/subscribe principle was created. The flow of information for inter-module communication is encapsulated into communication channels such as e.g. the scene modification or the hand update channel. This allows a seamless integration of additional processing and sensor models including the newly released Microsoft Kinect with an integrated full body skeleton tracking into the overall system.

Based on reference experiments with humans, models were trained to recognize the current assembly action of the human. It was experimentally evaluated, that with a small decrease in the robustness, it is sufficient to focus on the hand position and especially on derived data such as the velocity, acceleration, and jerk. With this level of abstraction from the task itself, from sensory input, and the set-up configuration, it was possible, to transfer the pre-learned models to the *JAHIR* set-up using only a three-dimensional occupancy grid tracking approach as exchanged input source.

Further, the concept of a task-based hierarchical robot control based on joint velocities was accomplished with a closed architecture industrial robot. With the seamless integration of the geometric context representation, a collision avoidance module for the robot was created, that preserves movements of the robot that are not in conflict with the avoidance strategy. The avoidance strategy is based on the information of multiple sensors and the computation of virtual forces based on the minimal distance of objects to each part of the robot. With this flexible controller, it is possible to easily define actions of the robot based on the priority of the task execution.

Previous findings that state that the choice of the motion velocity profile influences

[1] `http://www.opentl.org`

the unconscious adaption of the human motion, have been experimentally validated in the *JAHIR* scenario. The reaction times of humans to start a hand over motion were lower with a minimum jerk velocity profile compared to a trapezoidal velocity profile. Further, preceding evaluations regarding the right timing of actions in e.g. hand-over tasks were also verified by experiments on the *JAHIR* set-up. In this way, a seamless collaboration without waiting times for human *and* robot has been achieved. It has been shown that with the correctly timed robot assistance an assembly task can be more efficiently solved compared to if humans assemble on their own.

6.2 Future work

This work has conceptually exhibited how a subset of psychologically known mechanisms in human-human collaboration for assistive robotic systems can be successfully adopted. Although in this work the topic “human-robot collaboration” was not covered in an all-encompassing way, the results should seriously be considered for future collaborative systems in order to create a more natural collaboration with robots. Additional psychological mechanisms should be considered, realized and evaluated in the future, so that robotic systems are recognized and considered as adequate human collaboration partners. This should include the dynamic allocation of tasks based on the specific skills of human and robot.

Since in this work the context information was mainly restricted to geometric information of the surrounding and e.g. the hand positions, future work should take advantage of additional verbal and non-verbal aspects to gather a more detailed and encompassing context. On the output side, more capabilities to communicate with the human should be added. Current context-based and -suitable output modalities should be used adaptively—e.g. if suddenly noise occurs, the speech output could switch to a text-based notification. With a robust and fast six-dimensional face-tracking module, for example, it would be possible to directly estimate the focus-of-attention of the human. This information could then be used to adaptively show important information according to the attention of the human or to measure the human’s awareness regarding the robot. The exploitation of more and faster sensors along with advanced data fusion and machine learning approaches would also enlarge the system’s capabilities. This could also include an expansive recognition of complex actions of the human. Additionally, this work is based on the assumption, that the assembly plan is known a-priori. A probabilistic learning of plans and new action primitives of the robot would enrich the flexibility of the system even further.

Since the safety for the human has been conceptually considered to be an integrated part of the presented hierarchical robot controller, a focus of future work should be to find ways that guarantee the safety for the human. Beside the alleviation of for example the teach-in of new actions, the usage of compliant robots would greatly increase the safety for the human. Further, a promising technology to robustly survey the shared workspace is given with high-resolution depth sensing cameras. With the use of multiple of these cameras, all obstacles for the collision avoidance strategy of the robot controller could be estimated.

Selected Publications

Several parts and aspects presented were published alongside the work on the thesis. The following itemization lists a selected subset of these publications ordered by the publication date:

(1) M. Huber, **C. Lenz**, C. Wendt, B. Färber, A. Knoll, and S. Glasauer. Efficient Robot-supported Assembly using a Predictive Concept of Assistance (submitted). In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, May. 2012.

(2) G. Panin, **C. Lenz**, and A. Knoll. Real-time template tracking with Bayesian information filters on Lie algebras (submitted). *Computer Vision and Image Understanding*, 2012.

(3) **C. Lenz**, A. Sotzek, T. Röder, H. Radrich, M. Huber, S. Glasauer, and A. Knoll. Human workflow analysis using 3D occupancy grid hand tracking in a human-robot collaboration scenario. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3375–3380, San Francisco, CA, USA, Sept. 2011

(4) **C. Lenz**, T. Röder, M. Eggers, S. Amin, T. Kisler, B. Radig, G. Panin, and A. Knoll. A distributed many-camera system for multi-person tracking. In *Proceedings of the 1st International Joint Conference on Ambient Intelligence*, Malaga, Spain, Nov. 2010.

(5) M. Giuliani, **C. Lenz**, T. Müller, M. Rickert, and A. Knoll. Design principles for safety in human-robot interaction. *International Journal of Social Robotics*,

2(3):253–274, Sept. 2010.

(6) C. Staub, **C. Lenz**, G. Panin, A. Knoll, and R. Bauernschmitt. Contour-based surgical instrument tracking supported by kinematic prediction. In *Proceedings of the IEEE/RAS International Conference on Biomedical Robotics and Biomechatronics*, Tokyo, Japan, Sept. 2010.

(7) F. Wallhoff, J. Blume, A. Bannat, W. Rösel, **C. Lenz**, and A. Knoll. A skill-based approach towards hybrid assembly. *Advanced Engineering Informatics*, 24(3):329 – 339, Aug. 2010. The Cognitive Factory.

(8) A. Bannat, T. Bautze, M. Beetz, J. Blume, K. Diepold, C. Ertelt, F. Geiger, T. Gmeiner, T. Gyger, A. Knoll, C. Lau, **C. Lenz**, M. Ostgathe, G. Reinhart, W. Rösel, T. Rühr, A. Schuboe, K. Shea, I. S. genannt Wersborg, S. Stork, W. Tekouo, F. Wallhoff, M. Wiesbeck, and M. F. Zäh. Artificial cognition in production systems. *IEEE Transactions on Automation Science and Engineering*, PP(99):1–27, July 2010.

(9) M. Wojtczyk, G. Panin, T. Röder, **C. Lenz**, S. Nair, R. Heidemann, C. Goudar, and A. Knoll. Teaching and implementing autonomous robotic lab walkthroughs in a biotech laboratory through model-based visual tracking. In *IS&T / SPIE Electronic Imaging, Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques: Autonomous Robotic Systems and Applications*, San Jose, CA, USA, Jan. 2010.

(10) **C. Lenz**, M. Rickert, G. Panin, and A. Knoll. Constraint task-based control in industrial settings. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3058–3063, St. Louis, MO, USA, Oct. 2009.

(11) **C. Lenz**, G. Panin, T. Röder, M. Wojtczyk, and A. Knoll. Hardware-assisted multiple object tracking for human-robot-interaction. In F. Michaud, M. Scheutz, P. Hinds, and B. Scassellati, editors, *HRI '09: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 283–284, La Jolla, CA, USA, Mar. 2009. ACM.

(12) M. Wojtczyk, G. Panin, **C. Lenz**, T. Röder, S. Nair, E. Roth, R. Heidemann, K. Joeris, C. Zhang, M. Burnett, T. Monica, and A. Knoll. A vision based human

robot interface for robotic walkthroughs in a biotech laboratory. In F. Michaud, M. Scheutz, P. Hinds, and B. Scassellati, editors, *HRI '09: Proceedings of the 4th ACM/IEEE international conference on Human Robot Interaction*, pages 309–310, La Jolla, CA, USA, Mar. 2009. ACM.

(13) **C. Lenz**, G. Panin, and A. Knoll. A GPU-accelerated particle filter with pixel-level likelihood. In *International Workshop on Vision, Modeling and Visualization (VMV)*, Konstanz, Germany, Oct. 2008.

(14) M. Huber, **C. Lenz**, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. Human preferences in industrial human-robot interactions. In *Proceedings of the International Workshop on Cognition for Technical Systems*, Munich, Germany, Oct. 2008.

(15) G. Panin, E. Roth, T. Röder, S. Nair, **C. Lenz**, M. Wojtczyk, T. Friedlhuber, and A. Knoll. ITrackU: An integrated framework for image-based tracking and understanding. In *Proceedings of the International Workshop on Cognition for Technical Systems*, Munich, Germany, Oct. 2008.

(16) S. Nair, G. Panin, M. Wojtczyk, **C. Lenz**, T. Friedelhuber, and A. Knoll. A multi-camera person tracking system for robotic applications in virtual reality tv studio. In *Proceedings of the 17th IEEE/RSJ International Conference on Intelligent Robots and Systems 2008*. IEEE, Sept. 2008.

(17) **C. Lenz**, S. Nair, M. Rickert, A. Knoll, W. Rösel, J. Gast, and F. Wallhoff. Joint-action for humans and industrial robots for assembly tasks. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, pages 130–135, Munich, Germany, Aug. 2008.

(18) T. Müller, **C. Lenz**, S. Barner, and A. Knoll. Accelerating integral histograms using an adaptive approach. In *Proceedings of the 3rd International Conference on Image and Signal Processing*, Lecture Notes in Computer Science (LNCS), pages 209–217, Cherbourg-Octeville, France, July 2008. Springer.

(19) G. Panin, **C. Lenz**, S. Nair, E. Roth, M. Wojtczyk, T. Friedlhuber, and A. Knoll. A unifying software architecture for model-based visual tracking. In *IS&T/SPIE 20th Annual Symposium of Electronic Imaging*, San Jose, CA, Jan. 2008.

List of Figures

List of Tables

References

[1] **World Robotics, Executive Summary of 1. Industrial Robots and 2. Service Robots**. online, 2009. 2

[2] A. BANNAT, T. BAUTZE, M. BEETZ, J. BLUME, K. DIEPOLD, C. ERTELT, F. GEIGER, T. GMEINER, T. GYGER, A. KNOLL, C. LAU, C. LENZ, M. OSTGATHE, G. REINHART, W. RÖSEL, T. RÜHR, A. SCHUBOE, K. SHEA, I. S. GENANNT WERSBORG, S. STORK, W. TEKOUO, F. WALLHOFF, M. WIESBECK, AND M. F. ZÄH. **Artificial Cognition in Production Systems**. *IEEE Transactions on Automation Science and Engineering*, **PP**(99):1–27, July 2010. 2

[3] M. ZÄH, M. BEETZ, K. SHEA, G. REINHART, K. BENDER, C. LAU, M. OSTGATHE, W. VOGL, M. WIESBECK, M. ENGELHARD, C. ERTELT, T. RÜHR, AND M. FRIEDRICH. **Changeable and Reconfigurable Manufacturing Systems**, chapter The Cognitive Factory, pages 355–371. Springer, 2009. 2

[4] S. BARTSCHER. **Mensch-Roboter-Kooperation in der Montage**. In H. HOFFMANN, G. REINHART, AND M. F. ZÄH, editors, *Münchner Kolloquium–Fachformum Automation und Montagetechnik*, pages 33–40, 2010. 2

[5] Y. KOREN, U. HEISEL, F. JOVANE, T. MORIWAKI, G. PRITSCHOW, G. ULSOY, AND H. V. BRUSSEL. **Reconfigurable Manufacturing Systems**. *CIRP Annals - Manufacturing Technology*, **48**(2):527 – 540, 1999. 2

[6] J. KRÜGER, T. LIEN, AND A. VERL. **Cooperation of Human and Machines in Assembly Lines**. *CIRP Annals - Manufacturing Technology*, **59**, 2009. 2

[7] R. ALAMI, A. ALBU-SCHAEFFER, A. BICCHI, R. BISCHOFF, R. CHATILA, A. D. LUCA, A. D. SANTIS, G. GIRALT, J. GUIOCHET, G. HIRZINGER, F. INGRAND, V. LIPPIELLO, R. MATTONE, D. POWELL, S. SEN, B. SICILIANO, G. TONIETTI, AND L. VILLANI. **Safe and Dependable Physical Human-Robot Interaction in Anthropic Domains: State of the Art and Challenges**. In *Proceedings IROS'06 Workshop on pHRI - Physical Human-Robot Interaction in Anthropic Domains*. IEEE, 2006. 2

[8] V. Duchaine and C. Gosselin. **Safe, stable and intuitive control for physical human-robot interaction**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3383–3388, Kobe, Japan, May 2009. 2

[9] E. Hauck, A. Gramatke, and K. Henning. **A Software Architecture for Cognitive Technical Systems Suitable for an Assembly Task in a Production Environment**. In *Automation Control - Theory and Practice*, 2009. 2

[10] J. T. C. Tan, F. Duan, Y. Zhang, K. W. abd Ryu Kato, and T. Arai. **Human-Robot Collaboration in Cellular Manufacturing: Design and Development**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 29–34, St. Louis, USA, oct 2009. 2, 12

[11] T. Ogure, Y. Nakabo, S. Jeong, and Y. Yamada. **Risk Management Simulator for Low-powered Human-collaborative Industrial Robots**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 49–54, St. Louis, USA, oct 2009. 2

[12] R. D. Schraft, C. Meyer, C. Parlitz, and E. Helms. **PowerMate–A Safe and Intuitive Robot Assistant for Handling and Assembly Tasks**. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4074–4079, 2005. 2, 10

[13] J. Krüger, R. Bernhardt, D. Surdilovic, and G. Spur. **Intelligent Assist Systems for Flexible Assembly**. *CIRP Annals - Manufacturing Technology*, **55**(1):29 – 32, 2006. 3

[14] M. Hägele, J. Neugebauer, and R. Schraft. **From robots to robot assistants**. In *Proc. of the 32nd ISR (International Symposium on Robotics)*, pages 19–21, 2001. 3, 11

[15] G. Knoblich and J. S. Jordan. **Action coordination in groups and individuals: learning anticipatory control.** *J Exp Psychol Learn Mem Cogn*, **29**(5):1006–1016, September 2003. 3, 18

[16] N. Sebanz, H. Bekkering, and G. Knoblich. **Joint action: bodies and minds moving together**. *Trends in Cognitive Sciences*, **10**(2):70–76, February 2006. 3, 18, 19, 46

[17] C. Lenz, S. Nair, A. Knoll, W. Rösel, J. Gast, F. Wallhoff, and M. Rickert. **Joint-Action for Humans and Industrial Robots for Assembly Tasks**. In *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication*, pages 130–135. IEEE, August 2008. 6

[18] K. Kosuge, H. Yoshida, D. Taguchi, T. Fukuda, K. Hariki, K. Kanitani, and M. Sakai. **Robot-human collaboration for new robotic applications**. In *20th International Conference on Industrial Electronics, Control and Instrumentation*, **2**, 1994. 10

[19] K. Kosuge, M. Sato, and N. Kazamura. **Mobile robot helper**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, **1**, pages 583–588, 2000. 10

[20] O. Khatib. **Mobile manipulation: The robotic assistant**. *Robotics and Autonomous Systems*, **26**(2-3):175–183, 1999. Field and Service Robotics. 10

[21] E. Colgate, W. Wannasuphoprasit, and M. A. Peshkin. **Cobots: Robots for collaboration with human operators**. In *Proceedings of the International Mechanical Engineering Conference And Exhibition*, **58**, pages 433–439, 1996. 10

[22] J. Pires and J. Sá da Costa. **Object-oriented and distributed approach for programming robotic manufacturing cells**. *Robotics and Computer Integrated Manufacturing*, **16**(1):29–42, 2000. 10

[23] E. Helms, R. D. Schraft, and M. Hägele. **rob@work: Robot assistant in industrial environments**. In *Proceedings of the 11th IEEE International Workshop on Robot and Human interactive Communication*, pages 399–404, Berlin, Germany, 2002. 11

[24] M. Hägele, W. Schaaf, and E. Helms. **Robot Assistants at Manual Workplaces: Effective Co-operation, and Safety Aspects**. In *In Proceedings of the 33rd International Symposium on Robotics*, Stockholm, Schweden, oct 2002. 11

[25] S. Thiemermann and O. Schulz. **team@work Mensch-Roboter-Kooperation in der Montage**. *Automatisierungstechnische Praxis atp: Praxis der Mess-, Steuerungs-, Regelungs-, und Informationstechnik*, **45**(11):31–35, 2003. 11

[26] S. Thiemermann. **Direct man-robot-cooperation in assembly of small volume products with a SCARA-robot**. PhD thesis, Fraunhofer IPA, 2005. 11

[27] A. Stopp, T. Baldauf, S. Horstmann, and S. Kristensen. **Dynamic work space surveillance for mobile robot assistants**. In *Proceedings of the 12th IEEE International Workshop on Robot and Human interactive Communication (ROMAN)*, pages 25–30, Millbrae, California, USA, October 2003. 11

[28] A. Stopp, T. Baldauf, S. Hantsche, S. Horstmann, S. Kristensen, F. Lohnert, C. Priem, and B. Rüscher. **The manufacturing assistant: Safe, interactive teaching of operation sequences**. In *Proceedings of the 11th IEEE International Workshop on Robot and Human interactive Communication (ROMAN)*, Berlin, Germany, 2002. 11

[29] I. Iossifidis, C. Bruckhoff, C. Theis, C. Grote, C. Faubel, and G. Schoner. **CORA: An anthropomorphic robot assistant for human environment**. In *Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication*, pages 392–398, 2002. 11

[30] T. Gecks and D. Henrich. **SIMERO: Camera Supervised Workspace for Service Robots**. In *2nd Workshop on Advances in Service Robotics (ASER)*, Feldafing, Germany, may 2004. 11

[31] M. Fischer and D. Henrich. **3D Collision Detection for Industrial Robots and Unknown Obstacles Using Multiple Depth Images**. In T. Kröger and F. M. Wahl, editors, *Advances in Robotics Research*, pages 111–122. Springer Berlin Heidelberg, 2009. 11

[32] O. Schrempf, U. Hanebeck, A. Schmid, and H. Wörn. **A novel approach to proactive human-robot cooperation**. In *IEEE International Workshop on Robot and Human Interactive Communication, 2005. ROMAN 2005.*, pages 555–560, 13-15 Aug. 2005. 11

[33] M. Rickert, M. E. Foster, M. Giuliani, T. By, G. Panin, and A. Knoll. **Integrating Language, Vision and Action for Human Robot Dialog Systems**. In C. Stephanidis, editor, *Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction, HCI International, Part II*, **4555** of *Lecture Notes in Computer Science*, pages 987–995, Beijing, July 2007. Springer. 12, 19, 67, 68, 83

[34] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer. **On a new generation of torque controlled light-weight robots**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, **4**, pages 3356–3363, 2001. 12, 17

[35] R. Burger, S. Haddadin, G. Plank, S. Parusel, and G. Hirzinger. **The driver concept for the DLR lightweight robot III**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5453–5459, October 2010. 12, 17

[36] C. Ott, O. Eiberger, W. Friedl, B. Bäuml, U. Hillenbrand, C. Borst, A. Albu-Schäffer, B. Brunner, H. Hirschmüller, S. Kielhöfer, R. Konietschke, M. Suppa, T. Wimböck, F. Zacharias, and G. Hirzinger. **A humanoid two-arm system for dexterous manipulation**. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, pages 276–283. IEEE, 2006. 12, 17

[37] S. Haddadin, M. Suppa, S. Fuchs, T. Bodenmüller, A. Albu-Schäffer, and G. Hirzinger. **Towards the Robotic Co-Worker**. In C. Pradalier, R. Siegwart, and G. Hirzinger, editors, *Robotics Research*, **70** of *Springer Tracts in Advanced Robotics*, pages 261–282. Springer Berlin / Heidelberg, 2011. 12

[38] J. T. C. Tan, Y. Zhang, F. Duan, K. Watanabe, R. Kato, and T. Arai. **Human factors studies in information support development for human-robot collaborative cellular manufacturing system**. In *The 18th IEEE International Symposium on Robot and Human Interactive Communication, 2009*, pages 334–339, Oct 2009. 12

[39] C. Heyer. **Human-robot interaction and future industrial robotics applications**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4749–4754, October 2010. 13

[40] A. Kochan. **Robots and operators work hand in hand**. *Industrial Robot: An International Journal*, **33**(6):422–424, 2006. 13

[41] A. D. Santis, B. Siciliano, A. D. Luca, and A. Bicchi. **An Atlas of physical Human-Robot Interaction**. *Mechanism and Machine Theory*, **43**(3):253–270, 2008. 14

[42] S. Haddadin, A. Albu-Schäffer, M. Frommberger, J. Rossmann, and G. Hirzinger. **The "DLR Crash Report": Towards a Standard Crash-Testing Protocol for Robot Safety-Part I: Results**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 272–279, Kobe, Japan, 2009. 14

[43] S. Haddadin, A. Albu-Schäffer, M. Frommberger, J. Rossmann, and G. Hirzinger. **The "DLR Crash Report": Towards a Standard Crash-Testing Protocol**

for Robot Safety-Part II: Discussions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 280–287, Kobe, Japan, 2009. 14

[44] M. Zinn, O. Khatib, B. Roth, and J. Salisbury. **Playing it safe [human-friendly robots]**. *Robotics & Automation Magazine, IEEE*, **11**(2):12–21, 2004. 14

[45] O. Ogorodnikova. **How Safe the Human-Robot Coexistence Is? Theoretical Presentation**. *Acta Polytechnica Hungarica*, **6**(4):51–74, 2009. 14

[46] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. **Safety evaluation of physical human-robot interaction via crash-testing**. In *Robotics: Science and Systems Conference (RSS2007)*, pages 217–224, 2007. 14

[47] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. **The Role of the Robot Mass and Velocity in Physical Human-Robot Interaction-Part I: Unconstrained Blunt Impacts,"**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1331–1338, Pasadena, USA, 2008. 14

[48] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. **The Role of the Robot Mass and Velocity in Physical Human-Robot Interaction-Part II: Constrained Blunt Impacts,"**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1339–1345, Pasadena, USA, 2008. 14

[49] EN ISO 10218-2:2008. **Robots for industrial environments - Safety requirements - Part 2: Robot system and integration**, 2008. 14, 15

[50] EN ISO 10218-1:2006. **Robots for industrial environments - Safety requirements - Part 1: Robot**, 2006. 15

[51] S. Moon and G. S. Virk. **Survey on ISO standards for industrial and service robots**. In *Proceedings of the ICCAS-SICE international joint conference*, pages 1878 –1881, August 2009. 15

[52] C. Harper and G. S. Virk. **Towards the Development of International Safety Standards for Human Robot Interaction**. *International Journal of Social Robotics*, **2**(3):229–234, March 2010. 15

[53] F. Brecht, W. Bart, B. Luc, L. G. J. Ramon, and T. F. Carlos. **Robot Vision**, chapter Industrial Robot Manipulator Guarding Using Artificial Vision. InTech, 2010. 16, 17

[54] A. Bicchi and G. Tonietti. **Fast and soft arm tactics: Dealing with the Safety-Performance Tradeoff in Robot Arms Design and Control**. *Robotics & Automation Magazine, IEEE*, **11**(2):22–33, 2004. 16

[55] R. Ham, T. Sugar, B. Vanderborght, K. Hollander, and D. Lefeber. **Compliant actuator designs**. *Robotics & Automation Magazine*, **16**(3):81–94, 2009. 16

[56] M. Zinn, O. Khatib, B. Roth, and J. Salisbury. **A new actuation approach for human friendly robot design**. *The international journal of robotics research*, **23**(4–5):379, 2004. 16

[57] M. Schweitzer, C. Trommer, A. Karguth, J. Kunz, T. Lens, and O. von Stryk. **SAFE HUMAN INTERACTION WITH THE COMPLIANT ROBOT ARM BIOROB**. In *Proceedings of the 55th International Scientific Colloquium*, Ilmenau, Germany, TU Ilmenau, 2010. 16

[58] K. Salisbury, W. Townsend, B. Ebrman, and D. DiPietro. **Preliminary design of a whole-arm manipulation system (WAMS)**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, **1**, pages 254–260, April 1988. 17

[59] O. Olesya. **Human-Robot Interaction. Safety problems**. In *Proceedings of the International Workshop on Robotics in Alpe-Adria-Danube Region*, Hungary, 2006. 17

[60] E. Cheung and V. Lumelsky. **Proximity sensing in robot manipulator motion planning: system and implementation issues**. *IEEE Transactions on Robotics and Automation*, **5**(6):740–751, December 1989. 17

[61] D. Kulić and E. Croft. **Strategies for Safety in Human Robot Interaction**. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, **1**, pages 644–649, Coimbra, Portugal, 2003. 17, 22

[62] D. Kulić and E. Croft. **Pre-collision safety strategies for human-robot interaction**. *Autonomous Robots*, **22**(2):149–164, February 2007. 17, 22

[63] W. Erlhagen, A. Mukovskiy, E. Bicho, G. Panin, C. Kiss, A. Knoll, H. van Schie, and H. Bekkering. **Goal-Directed Imitation for Robots: A Bio-Inspired Approach to Action Understanding and Skill Learning**. *Robotics and Autonomous Systems*, **54**(5):353–360, 2006. 18

[64] C. Atkeson, J. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaul, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, E. Kawato, and M. Kawato. **Using humanoid robots to study human behavior**. *Intelligent Systems and their Applications, IEEE*, **15**(4):46–56, jul 2000. 18

[65] R. G. J. Meulenbroek, J. Bosga, M. Hulstijn, and S. Miedl. **Joint-Action Coordination in Transferring Objects**. *Experimental Brain Research*, **180**(2):333–343, 2007. 18

[66] M. Goodrich and D. Olsen. **Seven principles of efficient human robot interaction**. In *Proceedings on the IEEE International Conference on Systems Man and Cybernetics*, **4**, pages 3943–3948, 2003. 19, 22

[67] S. Stork, C. Stössel, H. J. Müller, M. Wiesbeck, M. Zäh, and A. Schubö. **A Neuroergonomic Approach for the Investigation of Cognitive Processes in Interactive Assembly Environments**. In *Proceedings of the 16th IEEE International Symposium on Robot and Human interactive Communication (RO-MAN)*, pages 750–755, August 2007. 19

[68] M. Buss, M. Beetz, and D. Wollherr. **CoTeSys - Cognition for Technical Systems**. In *Proceedings of the 4th COE Workshop on Human Adaptive Mechatronics (HAM)*, 2007. 19

[69] A. Bauer, D. Wollherr, and M. Buss. **Human-Robot Collaboration: A Survey**. *International Journal of Humanoid Robotics*, **5**(4):47 – 66, 2008. 19

[70] B. Grosz. **Collaborative systems**. *AI magazine*, **17**(2):67–86, 1996. 19

[71] F. Tang and L. E. Parker. **Peer-to-Peer Human-Robot Teaming through Reconfigurable Schemas**. In *AAAI Spring Symposium on "To Boldly Go Where No Human-Robot Team Has Gone Before"*, Stanford University, March 2006. 19

[72] G. Hoffman and C. Breazeal. **Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team**. In *HRI '07: Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 1–8, New York, NY, USA, 2007. ACM. 20

[73] N. Hawes, J. Wyatt, and A. Sloman. **An Architecture Schema for Embodied Cognitive Systems**. Technical Report CSR-06-12, University of Birmingham, School of Computer Science, November 2006. 20, 21

[74] M. Giuliani, C. Lenz, T. Müller, M. Rickert, and A. Knoll. **Design Principles for Safety in Human-Robot Interaction**. *International Journal of Social Robotics*, **2**(3):253–274, September 2010. 20, 22, 23, 24

[75] B. Muir and N. Moray. **Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation**. *Ergonomics*, **39**(3):429–460, 1996. 22

[76] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Michael Goodrich. **Common metrics for human-robot interaction**. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40. ACM, 2006. 23

[77] R. Pfeifer and J. Bongard. **How the body shapes the way we think: a new view of intelligence**. The MIT Press, 2007. 24

[78] A. Sloman. **Some Requirements for Human-like Robots: Why the recent over-emphasis on embodiment has held up progress**. In B. Sendhoff, E. Koerner, O. Sporns, H. Ritter, and K. Doya, editors, *Creating Brain-like Intelligence*, pages 248–277. Springer-Verlag, Berlin, 2009. 24

[79] M. Namoshe, N. S. Tlale, C. M. Kumile, and G. Bright. **Open middleware for robotics**. In *Proc. 15th International Conference on Mechatronics and Machine Vision in Practice M2VIP 2008*, pages 189–194, 2–4 Dec. 2008. 25

[80] N. Mohamed, J. Al-Jaroodi, and I. Jawhar. **Middleware for Robotics: A Survey**. In *Proc. IEEE Conference on Robotics, Automation and Mechatronics*, pages 736–742, 21–24 Sept. 2008. 25

[81] A. Shakhimardanov, J. Paulus, N. Hochgeschwender, M. Reckhaus, and G. K. Kraetzschmar. **Best Practice Assessment of Software Technologies for Robotics**. Technical report, Best Practice in Robotics, 2010. 25

[82] T. Blum, N. Padoy, H. Feussner, and N. Navab. **Modeling and Online Recognition of Surgical Phases Using Hidden Markov Models**. In *Medical Image Computing and Computer-Assisted Intervention*, **5242** of *Lecture Notes in Computer Science*, pages 627–635, Berlin / Heidelberg, 2008. Springer-Verlag. 28

[83] N. Padoy, T. Blum, H. Feussner, M.-O. Berger, and N. Navab. **On-line Recognition of Surgical Activity for Monitoring in the Operating Room**. In *Proceedings of the 20th Conference on Innovative Applications of Artificial Intelligence*, pages 1718–1724, July 2008. 28

[84] N. Padoy, T. Blum, S.-A. Ahmadi, H. Feussner, M.-O. Berger, and N. Navab. **Statistical modeling and recognition of surgical workflow**. *Medical Image Analysis*, 2010. 28

[85] A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal. **Discovering Optimal Imitation Strategies**. *robotics and autonomous systems, Special Issue: Robot Learning from Demonstration*, **47**(2-3):69–77, 2004. Sponsor: Swiss National Science Foundation. Top-3 2009 RAS most cited papers for the last five years. 28

[86] M. Huber, A. Knoll, T. Brandt, and S. Glasauer. **When to assist?-Modelling human behaviour for hybrid assembly systems**. In *Proceedings of ISR/ROBOTIK 2010*. VDE VERLAG GmbH, 2010. 28, 29, 71, 101, 102

[87] M. Huber, A. Knoll, T. Brandt, and S. Glasauer. **Handing Over a Cube**. *Annals of the New York Academy of Sciences*, **1164**(Basic and Clinical Aspects of Vertigo and Dizziness):380–382, 2009. 28

[88] E. Schneider, T. Villgrattner, J. Vockeroth, K. Bartl, S. Kohlbecher, S. Bardins, H. Ulbrich, and T. Brandt. **EyeSeeCam: An Eye Movement-Driven Head Camera for the Examination of Natural Visual Exploration**. *Annals of the New York Academy of Sciences*, **1164**(1):461–467, 2009. 29

[89] L. Rabiner. **A tutorial on hidden Markov models and selected applications in speech recognition**. *Proceedings of the IEEE*, **77**(2):257–286, 1989. 30

[90] C. Lenz, A. Sotzek, T. Röder, H. Radrich, M. Huber, S. Glasauer, and A. Knoll. **Human workflow analysis using 3D occupancy grid hand tracking in a human-robot collaboration scenario**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3375–3380. IEEE, 2011. 30, 32, 56, 57, 58, 60, 90, 91, 93

[91] G. Panin. **Model-based Visual Tracking: the OpenTL Framework**. Wiley-Blackwell, 2011. 34, 37, 38, 39, 40, 41, 42, 43, 44

[92] R. Hartley and A. Zisserman. **Multiple View Geometry in Computer Vision**. Cambridge University Press, 2004. 35, 59

[93] G. Panin, E. Roth, and A. Knoll. **Robust Contour-based Object Tracking Integrating Color and Edge Likelihoods**. In *International Workshop on Vision, Modeling and Visualization (VMV)*, Konstanz, Germany, October 2008. 36

[94] S. Baker and I. Matthews. **Lucas-Kanade 20 Years On: A Unifying Framework**. *Int. J. Comput. Vision*, **56**(3):221–255, 2004. 38

[95] T. Drummond and R. Cipolla. **Visual Tracking and Control using Lie Algebras**. In *CVPR*, **02**, pages 2652–2659, Los Alamitos, CA, USA, 1999. IEEE Computer Society. 38

[96] A. Blake and M. Isard. **Active Contours: The Application of Techniques from Graphics,Vision,Control Theory and Statistics to Visual Tracking of Shapes in Motion**. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998. 40

[97] J. Xavier and J. Manton. **On the Generalization of AR Processes To Riemannian Manifolds**. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, **5**, 14-19 2006. 40

[98] S. S. Blackman and R. Popoli. **Design and Analysis of Modern Tracking Systems**. Artech House Radar Library, 1999. 41

[99] L. D. Stone, T. L. Corwin, and C. A. Barlow. **Bayesian Multiple Target Tracking**. 1st. Artech House, Inc., 1999. 41

[100] B. Ristic, S. Arulampalm, and N. Gordon. **Beyond the Kalman filter: particle filters for tracking applications**. Artech House, Boston, Ma., 2004. 41

[101] R. E. Kalman. **A New Approach to Linear Filtering and Prediction Problems**. *Transactions of the ASME–Journal of Basic Engineering*, **82**(Series D):35–45, 1960. 42, 59

[102] G. Welch and G. Bishop. **An Introduction to the Kalman Filter**. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995. 42

[103] S. J. Julier and J. K. Uhlmann. **Unscented filtering and nonlinear estimation**. *Proceedings of the IEEE*, **92**(3):401–422, 2004. 43

[104] E. Wan and R. van der Merwe. **Chapter 7**. In S. Haykin, editor, *The Unscented Kalman Filter*. Wiley Publishing, 2001. 43

[105] B. Stenger, P. R. S. Mendonca, and R. Cipolla. **Model-Based 3D Tracking of an Articulated Hand**. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, page 310, Los Alamitos, CA, USA, 2001. IEEE Computer Society. 44, 50

[106] P. Li, T. Zhang, and B. Ma. **Unscented Kalman filter for visual curve tracking**. **22**(2):157–164, February 2004. 44

[107] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li. **Estimation with Applications to Tracking and Navigation**. John Wiley and Sons, Inc., New York, NY, USA, 2002. 44

[108] Y. Bar-Shalom and X.-R. Li. **Multitarget-Multisensor Tracking: Principles and Techniques**. YBS Publishing, 1995. 44

[109] A. G. O. MUTAMBARA. **Decentralized Estimation and Control for Multisensor Systems**. CRC Press, Inc., Boca Raton, FL, USA, 1998. 44

[110] S. THRUN, W. BURGARD, AND D. FOX. **Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)**. The MIT Press, September 2005. 44

[111] T. VERCAUTEREN AND X. WANG. **Decentralized sigma-point information filters for target tracking in collaborative sensor networks**. *IEEE Transactions on Signal Processing*, **53**(8-2):2997–3009, 2005. 44

[112] G. PANIN, C. LENZ, AND A. KNOLL. **Real-time template tracking with Bayesian information filters on Lie algebras (submitted)**. *Computer Vision and Image Understanding*, 2012. 45, 46, 47, 48

[113] N. J. EMERY. **The eyes have it: the neuroethology, function and evolution of social gaze**. pages 581–604, August 2000. 46

[114] D. DEMENTHON AND L. S. DAVIS. **Model-Based Object Pose in 25 Lines of Code**. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 335–343, London, UK, 1992. Springer-Verlag. 47

[115] F. PARK AND B. MARTIN. **Robot sensor calibration: solving AX=XB on the Euclidean group**. *Robotics and Automation, IEEE Transactions on*, **10**(5):717 –721, October 1994. 47

[116] E. BEGELFOR AND M. WERMAN. **How to Put Probabilities on Homographies**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(10):1666–1670, 2005. 47

[117] S. ARULAMPALAM, S. MASKELL, N. GORDON, AND T. CLAPP. **A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking**. *IEEE Transactions on Signal Processing*, **50**(2):174–188, February 2002. 48

[118] A. DOUCET, N. DE FREITAS, AND N. GORDON. **Sequential Monte Carlo methods in practice**. Springer, New York, 2001. 48

[119] M. ISARD AND A. BLAKE. **Condensation – conditional density propagation for visual tracking**. *International Journal of Computer Vision (IJCV)*, **29**(1):5–28, 1998. 48, 50, 54

[120] C. LENZ, G. PANIN, T. RÖDER, M. WOJTCZYK, AND A. KNOLL. **Hardware-assisted multiple object tracking for human-robot-interaction**. In F. MICHAUD, M. SCHEUTZ, P. HINDS, AND B. SCASSELLATI, editors, *HRI '09: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 283–284, La Jolla, CA, USA, March 2009. ACM. 49, 55, 62

[121] Z. KHAN. **MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(11):1805–1918, 2005. Member-Tucker Balch and Member-Frank Dellaert. 49

[122] G. PANIN, T. RÖDER, AND A. KNOLL. **Integrating Robust Likelihoods with Monte-Carlo Filters for Multi-Target Tracking**. In *Proceedings of the International Workshop on Vision, Modeling and Visualization (VMV)*, Konstanz, Germany, October 2008. 49

[123] C. Lenz, T. Röder, M. Eggers, S. Amin, T. Kisler, B. Radig, G. Panin, and A. Knoll. **A Distributed Many-Camera System for Multi-Person Tracking**. In *Proceedings of the 1st International Joint Conference on Ambient Intelligence*, Malaga, Spain, November 2010. 50

[124] D. Brščić, M. Eggers, F. Rohrmüller, O. Kourakos, S. Sosnowski, D. Althoff, M. Lawitzky, A. Mörtl, M. Rambow, V. Koropouli, J. M. Hernández, X. Zang, W. Wang, D. Wollherr, K. Kühnlenz, C. Mayer, T. Kruse, A. Kirsch, J. Blume, A. Bannat, T. Rehrl, F. Wallhoff, T. Lorenz, P. Basili, C. Lenz, T. Röder, G. Panin, W. Maier, S. Hirche, M. Buss, M. Beetz, B. Radig, A. Schubö, S. Glasauer, A. Knoll, and E. Steinbach. **Multi Joint Action in CoTeSys - Setup and Challenges**. Technical Report CoTeSys-TR-10-01, CoTeSys Cluster of Excelence: Technische Universität München & Ludwig-Maximilians-Universität München, Munich, Germany, June 2010. 50

[125] T. Starner, J. Weaver, and A. Pentland. **Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(12):1371–1375, Dezember 1998. 50

[126] M. Elmezain, A. Al-Hamadi, S. S. Pathan, and B. Michaelis. **Spatio-Temporal Feature Extraction-Based Hand Gesture Recognition for Isolated American Sign Language and Arabic Numbers**. In *Proceedings of 6th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 254–259, 2009. 50

[127] V. Pashaloudi and K. Margaritis. **Feature Extraction and Sign Recognition for Greek Sign Language**. In *Proceedings of the 11th IEEE Mediterranean Conference on Control and Automation (MED)*, 2003. 50

[128] M. Correa, J. Ruiz-del Solar, R. Verschae, J. Lee-Ferng, and N. Castillo. **Real-Time Hand Gesture Recognition for Human Robot Interaction**. *RoboCup 2009: Robot Soccer World Cup XIII*, pages 46–57, 2010. 50

[129] L. Brethes, P. Menezes, F. Lerasle, and J. Hayet. **Face tracking and hand gesture recognition for human-robot interaction**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, **2**, pages 1901–1906. IEEE, 2004. 50

[130] P. Ziaie, T. Müller, and A. Knoll. **A Novel Approach to Hand-Gesture Recognition in a Human-Robot Dialog System**. In *Proceedings of the First Intl. Workshop on Image Processing Theory, Tools and Applications*, Sousse, Tunesia, November 2008. 50

[131] C. Shan, T. Tan, and Y. Wei. **Real-time hand tracking using a mean shift embedded particle filter**. *Pattern recognition*, **40**(7):1958–1970, 2007. 50

[132] G. Panin, S. Klose, and A. Knoll. **Real-time articulated hand detection and pose estimation**. In *International Symposium on Visual Computing (ISVC)*, Las Vegas, Nevada, USA, December 2009. 50

[133] G. McAllister, S. J. McKenna, and I. W. Ricketts. **Hand tracking for behaviour understanding**. *Image and Vision Computing*, **20**(12):827 – 840, 2002. 50

[134] D. Mohr and G. Zachmann. **FAST: Fast Adaptive Silhouette Area based Template Matching**. In *Proceedings of the British Machine Vision Conference*, pages 39.1–39.12. BMVA Press, 2010. doi:10.5244/C.24.39. 50

[135] C. Lenz, G. Panin, and A. Knoll. **A GPU-accelerated Particle Filter with Pixel-level Likelihood**. In *International Workshop on Vision, Modeling and Visualization (VMV)*, Konstanz, Germany, October 2008. 50, 51, 54

[136] K. Oka, Y. Sato, and H. Koike. **Real-Time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems**. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, FGR '02, pages 429–, Washington, DC, USA, 2002. IEEE Computer Society. 50

[137] I. Laptev and T. Lindeberg. **Tracking of Multi-state Hand Models Using Particle Filtering and a Hierarchy of Multi-scale Image Features**. In *Proceedings of the Third International Conference on Scale-Space and Morphology in Computer Vision*, Scale-Space '01, pages 63–74, London, UK, 2001. Springer-Verlag. 50

[138] C. W. Ng and S. Ranganath. **Real-time gesture recognition system and application**. *Image and Vision Computing*, **20**(13-14):993–1007, 2002. 50

[139] A. A. Argyros and M. I. A. Lourakis. **Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera**. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 368–379, 2004. 50

[140] M. Isard and A. Blake. **Icondensation: Unifying low-level and high-level tracking in a stochastic framework**. In H. Burkhardt and B. Neumann, editors, *Computer Vision — ECCV'98*, **1406** of *Lecture Notes in Computer Science*, pages 893–908. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0055711. 50

[141] Q. Yuan, S. Sclaroff, and V. Athitsos. **Automatic 2D Hand Tracking in Video Sequences**. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*, **1**, pages 250–256, January 2005. 50

[142] H. Fei and I. Reid. **Probabilistic Tracking and Recognition of Non-Rigid Hand Motion**. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 60–67, Los Alamitos, CA, USA, 2003. IEEE Computer Society. 50

[143] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. **Color-Based Probabilistic Tracking**. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 661–675, London, UK, 2002. Springer-Verlag. 50, 51

[144] D. Comaniciu, V. Ramesh, and P. Meer. **Kernel-Based Object Tracking**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(5):564–575, 2003. 51

[145] K. Nummiaro, E. Koller-Meier, and L. J. V. Gool. **An adaptive color-based particle filter**. *Image Vision Comput.*, **21**(1):99–110, 2003. 51

[146] A. Diplaros, T. Gevers, and N. Vlassis. **Skin detection using the EM algorithm with spatial constraints**. *IEEE International Conference on Systems, Man and Cybernetics*, **4**, 2004. 51

[147] N. Friedman and S. Russell. **Image Segmentation in Video Sequences: A Probabilistic Approach**. In *Annual Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997. 51

[148] H. Greenspan, J. Goldberger, and I. Eshet. **Mixture model for face-color modeling and segmentation**. *Pattern Recognition Letters*, **22**(14):1525–1536, 2001. 51

[149] M. Yang and N. Ahuja. **Gaussian Mixture Model for Human Skin Color and Its Application in Image and Video Databases**. In *Proc. SPIE: Storage and Retrieval for Image and Video Databases VII*, **3656**, pages 458–466, 1999. 51, 52

[150] Z. Zivkovic. **Improved adaptive Gaussian mixture model for background subtraction**. In *Proceedings of the 17th International Conference on Pattern Recognition*, **2**, 2004. 51

[151] A. P. Dempster, N. M. Laird, and D. B. Rubin. **Maximum Likelihood from Incomplete Data via the EM Algorithm**. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**(1):1–38, 1977. 52

[152] V. Vezhnevets, V. Sazonov, and A. Andreeva. **A survey on pixel-based skin color detection techniques**. In *Proceedings of Graphicon*, pages 85–92, 2003. 52

[153] M. J. Jones and J. M. Rehg. **Statistical Color Models with Application to Skin Detection**. *International Journal of Computer Vision*, **46**:81–96, 2002. 52

[154] **Nvidia CUDA, `http://www.nvidia.com/cuda`**. 53

[155] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard. **Cg: A system for programming graphics hardware in a C-like language**. *ACM Transactions on Graphics*, **22**(3):896–907, July 2003. 53

[156] R. J. Rost. **OpenGL(R) Shading Language (2nd Edition)**. Addison-Wesley Professional, January 2006. 53

[157] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell. **A Survey of General-Purpose Computation on Graphics Hardware**. In *Eurographics 2005, State of the Art Reports*, pages 21–51, August 2005. 53

[158] D. Jauregui, P. Horain, M. Rajagopal, and S. Karri. **Real-time particle filtering with heuristics for 3D motion capture by monocular vision**. In *IEEE International Workshop on Multimedia Signal Processing*, pages 139–144. IEEE, 2010. 55

[159] J. Brown and D. Capson. **A Framework for 3D Model-Based Visual Tracking Using a GPU-Accelerated Particle Filter**. *IEEE Transactions on Visualization and Computer Graphics*, (99), 2011. 55

[160] A. Elfes. **Using occupancy grids for mobile robot perception and navigation**. *Computer*, **22**(6):46–57, June 1989. 56

[161] B. Schiele and J. Crowley. **A comparison of position estimation techniques using occupancy grids**. *Robotics and autonomous systems*, **12**(3-4):163–171, 1994. 56

[162] S. Thrun. **Learning occupancy grid maps with forward sensor models**. *Autonomous robots*, **15**(2):111–127, 2003. 56

[163] P. Stepan, M. Kulich, and L. Preucil. **Robust data fusion with occupancy grid**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **35**(1):106–115, 2005. 56

[164] C. Fulgenzi, A. Spalanzani, and C. Laugier. **Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid**. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1610–1616. IEEE, 2007. 56

[165] D. Beymer. **Person counting using stereo**. In *Proceedings of the Workshop on Human Motion*, pages 127–133, 2000. 56

[166] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. **Multi-Camera People Tracking with a Probabilistic Occupancy Map**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(2):267–282, February 2008. 56

[167] L. Chen, G. Panin, and A. Knoll. **Multi-Camera People Tracking with Hierarchical Likelihood Grids**. In *Proceedings of the 6th International Conference on Computer Vision Theory and Applications (VISAPP11)*, Algarve, Portugal, March 2011. INSTICC press. 56

[168] P. Viola and M. Jones. **Rapid object detection using a boosted cascade of simple features**. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 2001. 57

[169] L. Sentis and O. Khatib. **Task-Oriented Control of Humanoid Robots Through Prioritization**. In *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, Santa Monica, USA, November 2004. IEEE-RAS/RSJ. 62, 63

[170] Y. Yang and O. Brock. **Elastic Roadmaps: Globally Task-Consistent Motion for Autonomous Mobile Manipulation in Dynamic Environments**. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006. 62

[171] L. Sentis. **Synthesis and Control of Whole-Body Behaviors in Humanoid Systems**. PhD thesis, Standford University, 2007. 63

[172] E. Freund, M. Schluse, and J. Rossmann. **Dynamic collision avoidance for redundant multi-robot systems**. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, **3**, pages 1201–1206, 29 Oct.–3 Nov. 2001. 63, 66

[173] E. Freund and J. Rossmann. **The basic ideas of a proven dynamic collision avoidance approach for multi-robot manipulator systems**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, **2**, pages 1173–1177, 27–31 Oct. 2003. 63, 66

[174] C. Lenz, M. Rickert, G. Panin, and A. K. Knoll. **Constraint task-based control in industrial settings**. In *IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 3058–3063, Piscataway, NJ, USA, 2009. IEEE Press. 65, 94, 95, 97, 98

[175] A. De Santis, A. Albu-Schäffer, C. Ott, B. Siciliano, and G. Hirzinger. **The skeleton algorithm for self-collision avoidance of a humanoid manipulator**. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1–6, 4–7 Sept. 2007. 65

[176] G. van den Bergen. **A Fast and Robust GJK Implementation for Collision Detection of Convex Objects**. *Journal of Graphics Tools*, **4 (2)**:7–25, 1999. 66

[177] O. Khatib. **Real-time obstacle avoidance for manipulators and mobile robots**. *The international journal of robotics research*, **5**(1):90, 1986. 66

[178] R. Fletcher. **Practical Methods of Optimization Part II: Constrained Optimization**, chapter 10: Quadratic Programming, pages 229–258. John Wiley & Sons, New York, second edition, 1987. 66

[179] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. **Human-Robot Interaction in Handing-Over Tasks**. In *RO-MAN 08: Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication*, pages 107–112, München, Germany, 2008. IEEE Robotic and Automation Society. 67, 68, 70, 101

[180] M. Huber, C. Lenz, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. **Human Preferences in Industrial Human-Robot Interactions**. In *Proceedings of the International Workshop on Cognition for Technical Systems*, Munich, Germany, October 2008. 67, 68, 69, 70, 99, 100, 117

[181] S. Shibata, K. Tanaka, and A. Shimizu. **Experimental analysis of handing over**. *Proceedings of the IEEE International Workshop on Robot and Human Communication*, pages 53–58, 1995. 69

[182] T. Bhattacharjee, Y. Oh, J.-H. Bae, and S.-R. Oh. **Controlling redundant robot arm-trunk systems for human-like reaching motion**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2000–2005. IEEE, 2010. 69

[183] M. Svinin, M. Yamamoto, and I. Goncharenko. **Simple Models in Trajectory Planning of Human-Like Reaching Movements**. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1662–1667. IEEE, 2010. 69

[184] V. Gallese. **Before and below 'theory of mind': embodied simulation and the neural correlates of social cognition**. *Philosophical Transactions B*, **362**(1480):659, 2007. 70, 101

[185] M. Huber, C. Lenz, C. Wendt, B. Färber, A. Knoll, and S. Glasauer. **Efficient Robot-supported Assembly using a Predictive Concept of Assistance (submitted)**. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2012. 71, 102

[186] S. Glasauer, M. Huber, P. Basili, A. Knoll, and T. Brandt. **Basics in Joint Action: Spatial and Temporal coordination in Hand-overs**. In *Proceedings of the 28th annual conference of the Robotics Society of Japan (RSJ)*, Nagoya, Japan, 2010. 70

[187] F. Wallhoff, J. Blume, A. Bannat, W. Rösel, C. Lenz, and A. Knoll. **A skill-based approach towards hybrid assembly**. *Advanced Engineering Informatics*, **24**(3):329 – 339, August 2010. The Cognitive Factory. 79, 88, 90

[188] T. Rehrl, A. Bannat, J. Gast, G. Rigoll, and F. Wallhoff. **cfHMI: A Novel Contact-Free Human-Machine Interface**. In J. Jacko, editor, *Proc. Int. Conf. on Human-Computer Interaction HCI*, **LNCS 5611**, pages 246–254. Springer, 2009. 79, 88

[189] B. Buxbaum, R. Schwarte, T. Ringbeck, M. Grothof, and X. Luan. **MSM-PMD as correlation receiver in a new 3 D-ranging system**. In *Proceedings of SPIE the International Society for Optical Engineering*, 2002. 80

[190] T. Ringbeck and B. Hagebeuker. **A 3D Time of flight camera for object detection**. In *Proceedings of the 8th Optical 3-D Measurement Techniques*, 2007. 80

[191] B. Winkler. **Safe Space Sharing Human-Robot Cooperation Using a 3D Time-of-Flight Camera**. *International Robots and Vision Show*, 2007. 80

[192] I. Schiller, C. Beder, and R. Koch. **Calibration of a PMD camera using a planar calibration object together with a multi-camera setup**. In *Proceedings of the International Society for Photogrammetry and Remote Sensing Congress*, Beijing, China, 2008. 80

[193] C. Barber, D. Dobkin, and H. Huhdanpaa. **The quickhull algorithm for convex hulls**. *ACM Transactions on Mathematical Software (TOMS)*, **22**(4):469–483, 1996. 81

[194] M. Rickert. **Efficient Motion Planning for Intuitive Task Execution in Modular Manipulation Systems**. PhD thesis, Technische Universität München, 2011. 84

[195] P. Lukowicz, J. A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner. **Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers**. In A. Ferscha and F. Mattern, editors, *Pervasive Computing - LNCS*, **3001** of *Lecture Notes on Computer Science*, pages 18–32. Springer-Verlag, Berlin / Heidelberg, 2004. 90

[196] P. Viviani and C. Terzuolo. **Trajectory Determines Movement Dynamics**. *NEUROSCIENCE*, **7**(2):431–437, 1982. 99

[197] D. Olsen and M. Goodrich. **Metrics for Evaluating Human-Robot Interactions**. In *Performance Metrics for Intellligent Systems Workshop*, 2003. 101

[198] C. Vesper, A. Soutschek, and A. Schubö. **Motion coordination affects movement parameters in a joint pick-and-place task**. *The Quarterly Journal of Experimental Psychology*, **62**(12):2418–2432, 2009. 102