BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

HUMAN-ROBOT INTERACTION: ADVANCED TASK-CENTERED INTERFACES FOR NON-EXPERT USERS

INTERAKCE ČLOVĚKA S ROBOTEM: POKROČILÁ ÚKOLOVĚ ORIENTOVANÁ ROZHRANÍ PRO NEODBORNÉ UŽIVATELE

PH.D. THESIS
DISERTAČNÍ PRÁCE

AUTHOR — ING. ZDENĚK MATERNA
AUTOR PRÁCE

SUPERVISOR — DOC. RNDR. PAVEL SMRŽ, PH.D.
VEDOUCÍ PRÁCE

BRNO 2018

Ing. Zdeněk Materna: *Human-Robot Interaction: Advanced Task-centered Interfaces for Non-Expert Users*, doctoral thesis Brno, Brno University of Technology, Faculty of Information Technology, 2018.

ABSTRACT

Recent years brought a growing trend of deploying robots in novel applications where they are not only supposed to co-exist with and work next to humans but to actually closely collaborate with them on shared complex tasks. Capabilities of the robotic systems need to be substantially expanded in order to make the close, rich as well as natural human-robot interaction possible. Indeed, the interaction will not only happen between caged robots and highly specialized experts any more. More and more often, it will interconnect safe and interactive robots with non-expert users with various background. Consequently, the amazingly complex machines, that the current robots are, will become even more complex. This poses further challenges for the design of their user interfaces.

The objective of this thesis is to research and develop solutions for the close interaction between non-expert users and complex robots. The research was done in two different contexts: assistive service and industrial collaborative robots. Although these two domains have diverse requirements, related concepts could be used when designing the human-robot interaction. To cope with limitations of the current approaches, a novel method for task-centered interaction has been proposed. The most important aspects of the method are the utilization of mixed reality and robot-integrated capabilities, communication of the robot's inner state, context sensitivity, and usage of task-appropriate modalities. For each of the two mentioned domains, a user interface was designed and implemented. Both interfaces were successfully evaluated with non-expert users, who were able to carry out non-trivial tasks in cooperation with a robot. The reported evaluation provides an evidence that the realized method significantly improves the close human-robot interaction, which had not been entirely possible with previous approaches. The method's key characteristics provide guidelines for new designs of next user interfaces in the collaborative robotics.

KEYWORDS

Human-robot interaction; teleoperation; remote manipulation; colllaborative robots; simplified programming.

ABSTRAKT

Poslední roky přinesly rostoucí trend nasazení robotů v nových aplikacích, kde se od nich očekává nejen práce vedle lidí, ale skutečná spolupráce na společných komplexních úlohách. K umožnění blízké, bohaté a přirozené interakce člověka s robotem, bude nutné podstatně rozšířit schopnosti současných robotických systémů. Dále již nebude docházet k interakci jen mezi roboty v bezpečnostních klecích a experty na jejich programování. Stále častěji budou interagovat s bezpečnými spolupracujícími roboty uživatelé bez odborných znalostí z oblasti robotiky, s různorodým vzděláním a zkušenostmi. Úžasně složitá zařízení, kterými dnešní roboti jsou, se tak stanou ještě složitějšími, což představuje zásadní výzvu pro návrh jejich uživatelských rozhraní.

Cílem této práce je zkoumat a vyvinout řešení umožňující blízkou interakci mezi neodbornými uživateli a komplexními roboty. Výzkum byl zaměřen na dvě oblasti robotiky: asistenční servisní a průmyslové spolupracující roboty. Ačkoliv se tyto dvě oblasti vyznačují odlišnými požadavky, pro návrh interakce mezi člověkem a robotem je možné použít podobné principy. Nedostatky stávajících přístupů jsou řešeny návrhem nové metody pro úlohově zaměřenou interakci. Nejvýznamější aspekty metody jsou využití smíšené reality, autonomních funkcí robota, komunikace vnitřního stavu robota, kontextová citlivost a použití modalit vhodných pro danou úlohu. Pro obě oblasti zaměření výzkumu bylo na základě metody navrženo a implementováno uživatelské rozhraní. Obě rozhraní byla úspěšně ověřena s neodbornými uživateli, kteří díky nim byli schopni úspěšně spolupracovat s robotem na složitých úlohách. Publikovaná ověření rozhraní prokazují, že realizovaná metoda významně zlepšuje blízkou interakci mezi člověkem a robotem, která s dosavadními přístupy nebyla plně dosažitelná. Klíčové aspekty metody představují vodítko pro návrh uživatelských rozhraní v oblasti spolupracujících robotů.

KLÍČOVÁ SLOVA

Interakce člověka s robotem; teleoperace; vzdálená manipulace; spolupracující roboti; zjednodušené programování.

BIBLIOGRAPHIC CITATION

Ing. Zdeněk Materna: *Human-Robot Interaction: Advanced Task-centered Interfaces for Non-Expert Users*, doctoral thesis Brno, Brno University of Technology, Faculty of Information Technology, 2018.

DECLARATION

I declare that this dissertation thesis is my original work and that I have written it under the guidance of Doc. RNDr. Pavel Smrž, Ph.D.. All sources and literature that I have used during my work on the thesis are correctly cited with complete reference to the respective sources.

Brno, 2018

Ing. Zdeněk Materna, January 22, 2019

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

IFR	International Federation of Robotics	5
IMU	inertial measurement unit	4
SME	small and medium-sized enterprises	2
SAR	spatial augmented reality	19
ISAR	interactive spatial augmented reality	13
HRI	human-robot interaction	3
HMI	human-machine interaction	3
HCI	human-computer interaction	3
SRS	Multi-Role Shadow Robotic System for Independent Living	ix
AR	augmented reality	18
VR	virtual reality	4
AI	artificial intelligence	4
UI	user interface	4
ISO	International Organization for Standardization	4
IAD	Intelligent Assist Device	6
DoF	Degrees of Freedom	15
WoZ	Wizard of Oz	27
GUI	Graphical User Interface	18

1

MOTIVATION AND BACKGROUND

> *An automated machine that does just one thing is not a robot. It is simply automation. A robot should have the capability of handling a range of jobs at a factory.*
>
> — Joseph Engelberger

A close, face to "face" human-robot interaction has been so far more topic of research or science-fiction than something that actually happens in everyday life (with exception of robotic vacuum maintenance). Apparently, this will change sooner or later as robots for various applications are getting more affordable and human labor tends to be more expensive. First, a close interaction between humans and robots will become more frequent in industry, where caged robots are being replaced by collaborative ones. As the robots will move out of the cages, they will work alongside human workers. Then, trend towards humans and robots closely collaborating on the same task could be expected in order to increase productivity. To enable such close collaboration and maintain safety, a rich human-robot interaction will be inevitable. At the same time, service robots will more and more often come to contact with people in hospitals, institutional care facilities and prospectively also in private households. What have industrial robots in common with service robots? There must be some interface allowing human users interact with them: to check their state, give them goals, visualize robot intentions, etc. In general, in both contexts it has to be assumed that the users are general public, majority of them will not be roboticians or programmers and the future interface design has to respect this.

1.1 ORGANIZATION OF THE THESIS

The thesis in form of collection of published articles is organized as follows. This chapter provides definition of the basic framework of this thesis as well as motivation and justification for the conducted research. Chapter 2 formulates the thesis statement, the related objectives and presents the achieved contributions. A general state of the art overview is given within Chapter 3. Despite overview of the academic solutions, also commercially available ones are included. A more specific overviews of (academic) state of the art could be found within the respective sections of the included papers. Based on the current state of the art, a novel method

Figure 1.1: Elderly person being served by the SRS robot within user tests in a laboratory imitating home environment (credit: http://srs-project.eu/milan_test_may)

for human-robot task-centered interaction is proposed in Chapter 4. Chapters 5 to 8 are previously published research papers. Chapter 9 concludes this work.

1.2 SCOPE OF THE THESIS

The scope of this work follows the two main projects in which I was involved during my Ph.D. studies.

The first project was SRS[1]. It was focused on development of user interfaces for semi-autonomous personal care robot, helping elderly people to life independently at home as long as possible (see Figure 1.1). Specifically, I worked on user interface for teleoperation of the robot in cases, where it could not handle particular action autonomously.

The second project is Collaborative robot 2.0: cognition of the work environment, augmented reality-based user interface, simple deployment and reconfiguration[2]. Goal of the project is to come-up with novel solutions for collaborative robots, to simplify their deployment in small and medium-sized enterprises (SME). I'm research leader of this project and my main responsibility is a design, implementation and testing of a projected user interface (see Figure 1.2).

---

[1] EU-7FP-IST - Seventh Research Framework Programme, 7E12056, 247772, 2011-2013, http://srs-project.eu.

[2] Funded by Technology Agency of the Czech Republic, project code TJ01000352, duration from 2017-09-01 to 2019-08-31.

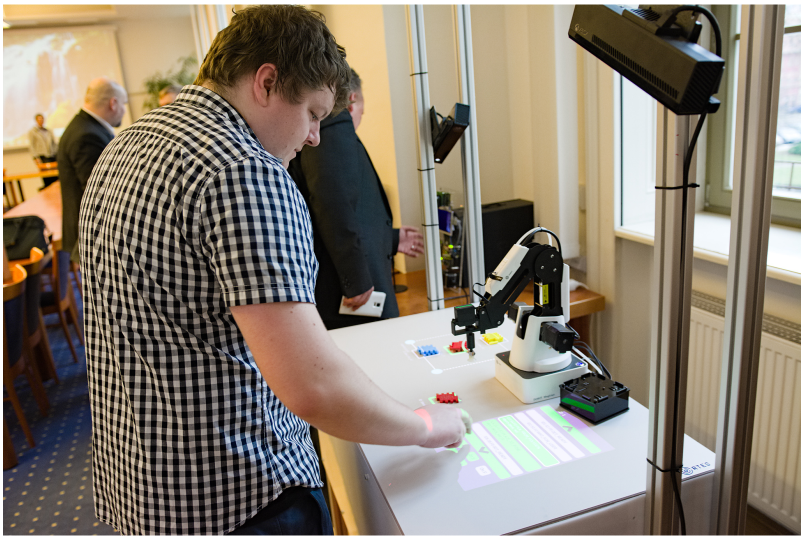


Figure 1.2: The user programs a table-top robot to perform a pick and place task.

In either case, methodologies originating from human-computer interaction (HCI) were used to design and evaluate interfaces. Moreover, both cases are linked by focus on non-expert users, where interaction was designed in order to take as much advantage as possible from robot-integrated capabilities as e.g. sensing of the environment and motion planning.

The following sections provide a brief overview of the past and present directions in the field of human-robot interaction (HRI) research (Section 1.3), introduction into specifics of personal service robots (Section 1.4), collaborative industrial robots (Section 1.5) and the chapter is closed by a summary with respect to the scope of this thesis (Section 1.6). The both Sections 1.4 and 1.5 also provide overview of the past research projects in the respective area of interest. The purpose of this overview is to provide an insight into the broader context of recent activities within the field and justification of the selected research topics.

1.3 HUMAN-ROBOT INTERACTION

The HRI is an interdisciplinary research domain originating from human-machine interaction (HMI) and HCI fields. According to the definition (based on usability research) from [45] it is *"a field of study dedicated to understanding, designing, and evaluating robotic systems for use by or with humans"*, while its problem is *"to understand and shape the interactions between one or more humans and one or more robots"*. Broader definition was stated in [29]: *"HRI is the science of studying people's behaviour and attitudes towards robots in relationship to the physical, technological and interactive features of the robots, with the goal to develop robots that facilitate the emergence of human-robot interactions that are at the same time efficient (according to the original requirements of their envisaged area of use), but are also acceptable to people, and meet the social and emo-*

tional needs of their individual users as well as respecting human values". As it turns out from definitions, inherent part of HRI is design of robots. Naturally, robots have to more or less (based on their application) interact with humans so, there has to be some user interface to mediating this interaction.

In this thesis, the scope will be limited to the design of robot interfaces, in particular for personal service robots and for collaborative industrial robots. Those types of robots are usually incredibly complex machines. Machines based on sophisticated hardware and with continuously improving and expanding capabilities. Although robots does not posses general artificial intelligence (AI) yet, anyway it might be highly difficult to understand their inner state, to predict their actions, to understand what and how they perceive [134]. Communicating robot's inner state to the user could be seen as one of the main challenges in design of user interface (UI). As robots are usually not working all the time in a fully stand-alone mode, there is also need to direct their activity, in another words, to set them goals or to coordinate the joint task between the human and the robot.

Previously, a lot of research was focused on teleoperation of robots as at the time, use cases where direct (face to face) or close HRI could occur were highly limited as service robots were non existent and in the industry, robots were strictly separated from humans[3]. With emergence of safe industrial robots (e.g. UR5 by Universal Robots in 2008) as well as research service robots (e.g. PR2 from Willow Garage in 2010), more effort was put into research of the close HRI.

For an interface to be functional (and bidirectional), there has to be at least on input and one output modality. Traditionally, the output modality used to be mainly a computer screen. Within the context of teleoperated robots, an ecological approach to UI design gained significant popularity [95] with its main benefit of improved situational awareness over "traditional" (2D video) interfaces. Input modalities tend to be mouse, keyboard or joystick. Recently, many less traditional modalities were investigated as various controllers (3D mouse, inertial measurement unit (IMU)-based devices), stereoscopic displays, virtual reality (VR), etc. To enable closer interaction within the context of collaborative robots, it is inevitable for the robot to perceive its environment and especially its human partner: his/her position, activity, or intentions.

1.4 PERSONAL SERVICE ROBOTS

The term service robot is according to the International Organization for Standardization (ISO) standard 13586 defined as *"robot that performs useful tasks for humans or equipment excluding industrial automation applications"* [62]. Current spread of service

---

[3] For good reasons, see i.e. `https://en.wikipedia.org/wiki/Robert_Williams_(robot_fatality)`

robot	Willow Garage PR2	SoftBank Robotics Pepper	Fraunhofer Institute for Manufacturing Care-O-Bot 4
introduced	2010	2014	2015
purpose	general research platform	interactive receptionist, research platform	"basis for commercial service robot solutions"
price	$400,000	$1,931	NA
DoF	20	20	29
safety	limited force, wireless e-stop	limited force, partially soft cover	safety lasers
features	inherently safe hardware design	out of the box functionality, applications, emotion recognition	modular design

Table 1.1: Overview of the selected personal service robots.

robots includes: logistics, care, telepresence, domestic usage, security, agriculture, entertainment, etc. Applications of the robots could be divided into the two main categories: professional and personal. This work focuses on the second one, more specifically on personal care robots with navigation and manipulation capabilities. However, there is currently no such robot available on the consumer market[4] although the ISO standard defining safety requirements of such robots is available since 2014 [61][5]. There exist several platforms for research and development (see Table 1.1 for comparison).

In the near future, adoption of the service robots is expected to rise – according to the International Federation of Robotics (IFR), sales in the segment are going to rise 20-25 % in the period 2018-2020[6]. It is also estimated that by the end of 2019, up to 31 million domestic household and 11 million entertainment and leisure robots will be deployed[7]. Those forecasts justify importance of research in the field of close HRI.

---

[4] Few so-called companion robots are available as e.g. Paro, or mobile robots without manipulation capabilities as e.g. KOMPAÏ.

[5] It concerns: physical assistant robots, mobile servant robots, and person carrier robots.

[6] https://ifr.org/ifr-press-releases/news/why-service-robots-are-booming-worldwide

[7] https://ifr.org/ifr-press-releases/news/31-million-robots-helping-in-households-worldwide-by-2019

1.4.1 *Recent projects*

R4H[8] (2011-present) started as the join project between Willow Garage, Healthcare Robotics Lab at Georgia Tech and Henry Evans, a stroke survivor who is now mute quadriplegic. During the project, various experimental user interfaces were developed to support daily living activities, including interface to control PR2 robot as a body surrogate [24]. The development is based on user-centered design methodology and encompasses following topics: assistance with manipulation near the user's body, assistance with manipulation of objects in the environment and assistance with social interaction.

The Accompany[9] project (2011-2014) was focused on providing support to elderly persons to enable them to live independently at home. The companion robot (Care-O-Bot 3) operated within the intelligent environment (equipped with sensors). Three user interfaces were developed: tablet-based (showing live camera stream from the robot) and two haptic devices (to attract attention of the robot by squeezing the device). The project scope resembles in some aspects the one of SRS project, which is in more detail described in Section 5.1.

The approach of the ENRICHME[10] project (2015-2018) was to support independent living of people with mild cognitive impairments within the assisted living environment equipped with sensors and RFID tagged objects. The role of the robot (Kompaï, Tiago) was among others to offer cognitive games, remind medication and help to find objects. The HRI occurred through the robot-mounted touch screen providing graphical interface, speech recognition and synthesis.

1.5 COLLABORATIVE INDUSTRIAL ROBOTS

The term industrial robot is according to [62] defined as *automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications* and collaborative robot (often also referred to as cobot) as *robot designed for direct interaction with a human*. The Intelligent Assist Device (IAD)[11] may be seen as an alternative to cobot; however, it does not fulfill the first one definition.

---

[8] Robots for humanity, http://r4h.org, https://www.ted.com/talks/henry_evans_and_chad_jenkins_meet_the_robots_for_humanity.

[9] Acceptable robotiCs COMPanions for AgeiNg Years: http://rehabilitationrobotics.net/cms2/, use case demonstration: https://youtu.be/1CD9Gxz6qBw.

[10] Enabling Robot and assisted living environment for Independent Care and Health Monitoring of the Elderly, http://www.enrichme.eu/wordpress/.

[11] Gravity compensated manipulator for material handling and assembly operations.

Traditionally, in mass production the desired state was a 100% automation in order to maximize benefits resulting from the economics of scale. Nowadays, mass production evolves into mass customization which is an inherent part of the Industry 4.0 paradigm and imposes significant improvement of flexibility. On the other hand, at SME companies, flexibility of the production was always important. At the same time, with rising availability of (collaborative) robots, they are deployed at SMEs more and more often [60]. In order to allow a higher flexibility and improve productivity, there exists a trend towards:

1. Removing the strict spatial and temporal separation of human workers and robots.
2. Hybrid assembly cells, where human and robot work in parallel on the joint task.

Both trends are possible due to collaborative robots (see Table 1.2), which are designed to be safe to work alongside humans. However, a cell fitted with a collaborative robot is not automatically fully safe and risk assessment for the particular application has to be performed to comply with ISO/TS 15066:2016. The most common use is for tasks as packaging, palletizing, automatic (bin) picking, quality control, assembly, sorting, sanding, polishing, etc. A vision is often utilized to cope with uncertainty as e.g. slightly variable position of parts. The main advantages over caged/fenced robots are:

- Easier deployment and programming.
- Reduced expenses on safety equipment (sensors, barriers).
- Better utilization of floor space - lower real estate expenses.
- Enabling to form hybrid cells.

Additionally, the hybrid cells where human and robot may work in parallel might be more expensive; however, poses several advantages:

- Increased productivity.
- More uniform quality.
- Lowering risk of health problems by offloading a repetitive or non-ergonomic parts of the task to a robot.

Four types of human-robot collaboration are defined in [38]:

- Safety-rated monitored stop

robot	Universal Robots UR5	Rethink Robotics Sawyer	ABB Yummi
introduced	2008	2015	2015
DoF	6	7	2×7
payload / radius	5kg / 850mm	4kg / 1260mm	2×0.5kg/559mm
price	$34,000	$40,535	$40,000
rated life	35,000 hours	30,000 hours	NA
safety	limiting joint position/speed, TCP position/speed/-force, momentum, power	backdriveable series elastic actuators, light/eyes signalling status / next motion	padded arms, collision detection, cartesian speed supervision
programming	touch teach pendant with 3D visualization, hand guiding	embedded controls (training cuffs / navigator) + display, hand guiding	hand guiding, tablet, ABB Rapid

Table 1.2: Overview of the selected collaborative industrial robots (note: Rethink Robotics was closed down on October 3, 2018).

- Hand guiding
- Speed and separation monitoring
- Power and force limiting

1.5.1 *Recent projects*

SMEROBOTICS[12] (2012-2016) aimed to create robots suitable for SME companies. The approach was to take advantage of knowledge base and embedded cognition of the robot to achieve high flexibility (due to frequent production changes) and robustness to uncertainties.

FACTORY-IN-A-DAY[13] (2013-2017) project was mainly focused on reduction of installation time and cost. The approach was built upon learnable robot skills (so called "apps") that allow fast setup and teaching. The safety was tackled by leveraging of proximity-sensing robot skin and online path re-planning algorithms.

---

[12] http://www.smerobotics.org/

[13] http://www.factory-in-a-day.eu/

ColRobot[14] (2016-2019) is a H2020 project, aimed on development of an autonomous, navigation-capable mobile manipulator acting as a worker's "third hand" with following envisioned functionality: delivering kits, tools, parts, and holding work pieces. The worker interacts with the robot using gestures, touch commands and demonstrations.

1.6 MOTIVATION AND BACKGROUND SUMMARY

So far, actually deployed robots tend to be moreover pre-programmed machines, exactly following the given procedure, with none to low abilities to interact with humans. Within the academic research domain, a lot of effort was made in order to allow more or less natural HRI. For instance, the perceived importance of the problem could be illustrated on a high number of associated research projects, where some selected ones were briefly introduced in the text above. However, there still remains unsolved challenges and many of the problems were solved in a rather isolated way. Within the commercial sphere, there seems to be trend of growing importance of interactive features, probably driven by demand for hybrid assembly cells, which are; however, currently still not truly widespread. In a near future, interaction-able robots will likely become reality and later they will become omnipresent. Therefore, it is necessary to develop appropriate interaction methods and derived interfaces for such robots.

---

[14] Collaborative Robotics for Assembly and Kitting in Smart Manufacturing, https://www.colrobot.eu/.

2

OBJECTIVES AND CONTRIBUTIONS

> *Research is what I'm doing when I don't know what I'm doing.*
>
> — Wernher von Braun

The area of interest of this thesis and its significance was justified in Chapter 1. In this chapter, the thesis statement is formulated together with related research objectives and overview of the main contributions is given.

The thesis statement was selected as a more appropriate for this work, instead of hypothesis or research question (which is equivalent to hypothesis, just formulated in form of a question) as the research in the field of HRI is largely of a qualitative or at the best of a mixed nature [112, 72]. Because of that, it is problematic to formally prove or disprove a hypothesis commonly used in domains where quantitative research prevails.

2.1 THESIS STATEMENT

The thesis statement directing the research efforts within this work is formulated as follows:

A specifically designed user interface may enable non-expert users to accomplish non-trivial joint tasks with highly complex robots.

For the purposes of this work, "non-expert user" is a user without specific knowledge of robots, automation, or computer science; however, potentially with domain or task specific knowledge. The robot should be understood as a personal service robot or a collaborative industrial robot. A complex robot, is a robot with at least partial autonomy and basic cognitive abilities. This work aims on an interaction where a user and a robot collaborate on the same task (spatially collocated or displaced), they interact regularly and in a non-trivial way and the interaction preferably happens within the task space. Such interaction is referred to as "close".

2.2 RESEARCH OBJECTIVES

In order to gain support for the thesis statement, following objectives were formulated.

1. *Define an integrative method for close human-robot interaction.*

As it could be seen from the state of the art overview (see Chapter 3), the existing methods are still somehow limited in various aspects. Promising partial approaches have been published; however, with limitations as e.g. suitability for only trivial tasks or on the other hand, unsuitability for non-expert users. Consequently, there is an opportunity to integrate those partial approaches into a novel method which could serve as a basis for design of next generation user interfaces allowing effective task-centered interaction.

2. *Apply the method within the contexts of interest.*

In order to allow evaluation of the method and demonstrate how it generalizes to different contexts (use cases), more than one user interface based on the method should be implemented. Naturally, specifics of the contexts have to be taken into account. The contexts of interest within this thesis are assistive service robots and collaborative industrial robots.

3. *Investigate if and how underlying autonomy could support human-robot interaction.*

Interacting with a highly complex and eventually fully or partially autonomous robot might be challenging for various reasons: automatically triggered actions of the robot might be confusing (*why the robot did that?*), natural communication cues from human-human communication are missing or are insufficiently supplemented, etc. Interaction becomes even more challenging if it happens remotely, where the user also has to build a mental model of the remote environment and track or estimate its state.

It is hypothesized that the interface enabling user to trigger and parametrize robot autonomous functions would help to keep mental workload low and thus maintain collaboration effective.

4. *Investigate what modalities are appropriate for convenient interaction.*

Input and output modalities are the essence of each interface. The modalities and their usage have to be chosen appropriately according to the robot, the user, the environment and the task at the hand. Inputs has to enable users to influence robot actions and outputs have to communicate robot's current state, task state, problems, etc. Multimodal interaction has to be designed in a way, that it provides a coherent and plausible user experience.

5. Investigate how the joint task should be presented to make it comprehensive and how to support situation awareness.

For any team to be effective, the joint task has to be known in the first place and naturally, it has to be understood by all participating members. Moreover, task progress, changes to plan and exceptions has to be tracked. The robot may perform all of this internally as well as human. However, a human short-term memory capacity is limited and high mental load might lead to an increased workload. Thus, the required information should be provided by the interface. On the other hand, overwhelming the user with too much data would be counterproductive. Information has to be shown intelligently, in a context-sensitive manner.

6. Evaluate the method-based interfaces with non-expert users.

The method can only be evaluated indirectly, through evaluation of the user interfaces based on it. Although usability or technical issues of the concrete implementation will definitely play role in the evaluation and will affect the results, if the main aim of the interface will be satisfied without major issues, it could be claimed that the objective was fulfilled.

2.3 CONTRIBUTIONS

The main contribution of this thesis is the novel method for task-centered interaction (further described in Section 4.1). Moreover, for the purpose of context-specific evaluation of the method and its particular aspects, two fully functional user interfaces based on the method have been developed, enabling non-expert users to:

- teleoperate assistive service robots and
- program industrial robots and collaborate with them.

Both interfaces are based on centering the interaction into the task context and mixed reality: a virtual 3D scene in case of teleoperation and a shared workspace with interactive spatial augmented reality (ISAR) in the case of industrial robot programming. The usage of mixed reality helps to avoid attention switches and to lower mental demands, thus improving efficiency of interaction. Low level control is avoided by using a semi-autonomous robot, with advanced sensing capabilities, able to carry out particular tasks independently. Both approaches sharing the same fundamental principles were evaluated in several user studies with promising results (for overview see Section 4.2).

2.4 PUBLICATIONS

The conducted research has been published in several papers, where those where I was the main contributor are included as chapters of this thesis (with my contribution expressed as a percentage in parentheses):

- Chapter 5: *Teleoperating Assistive Robots: A Novel User Interface for Remote Manipulation and Navigation Relying on Semi-Autonomy and Global 3D Environment Mapping* (40 %).
- Chapter 6: *Simplified Industrial Robot Programming: Effects of Errors on Multimodal Interaction in WoZ experiment* (40 %).
- Chapter 7: *Using Persona, Scenario, and Use Case to Develop a Human-Robot Augmented Reality Collaborative Workspace* (50 %).
- Chapter 8: *Interactive Spatial Augmented Reality in Collaborative Robot Programming: User Experience Evaluation* (35 %).

Other relevant publications which I significantly contributed to:

- Design of the human-robot interaction for a semi-autonomous service robot to assist elderly people [87] (10 %).
- Teleoperation of domestic service robots: Effects of global 3d environment maps in the user interface on operators' cognitive and performance metrics [86] (15 %).
- Semi-autonomous domestic service robots: Evaluation of a user interface for remote manipulation and navigation with focus on effects of stereoscopic display [88] (20 %).
- Industrial human-robot interaction: Creating personas for augmented reality supported robot control and teaching [124] (10 %).

I also contributed to the following technical report:

- Deliverable D4. 5.2–Context-aware Virtual 3D Display Final Report [121] (20 %).

3

GENERAL STATE OF THE ART

3.1 RELATED WORK

This chapter provides overview of the recent existing work within the scope of this thesis complementary to the respective sections of the included papers, which are focused more specifically according to the topic of each paper. Sections 3.1.1 to 3.1.3 corresponds to "investigative" objectives 3, 4 and 5 stated in Chapter 2. To the end, Section 3.2 offers a brief overview of current non-academic solutions already available on the market and Section 3.3 provides summary.

3.1.1 *Semi-autonomous Robots*

This section particularly focuses on ability of different approaches to cope with (non-expert[1]) user input: if and how robot (semi-)autonomous functions are parameterizable and triggered.

In order to allow robots to function efficiently and safely in a complex and highly unstructured or semi-structured environments as private households and SME, some form of a partial autonomy is often utilized. The partial autonomy in this case means, that the system is able to cope with user inputs and adjusts its function according to them or may be temporarily switched to more or less manual control mode[2]. Various approaches exists as: semi-autonomy, adjustable autonomy, mixed initiative, sliding autonomy, etc. Within these approaches, interfaces are usually specifically designed to minimize cognitive load (a concept associated with working memory in the cognitive load theory) of the users which is achieved by various means. A target user group has to be known and considered to, among others, avoid expertise reversal effect [66] which may occur when an over-simplified user interface (providing too much guidance or abstracted information) is used by individuals with more prior knowledge [64]. Within the context of assistive robots, possible cognitive or physical limitations of the end users has to be taken into account.

---

[1] https://en.oxforddictionaries.com/definition/non-expert

[2] Full manual control is often not applicable as e.g. setting each joint position during teleoperation session of high Degrees of Freedom (DoF) robot would be extremely difficult if not impossible.

An underlying autonomy of the system could assist users to simplify e.g. manipulator control. In [130], a grasp database and motion planning is used to control arm of the assistive robot. The user sets Cartesian coordinates of the end effector and may press button to initiate an autonomous grasp or place sequence. Orientation of the end effector is set automatically according to current mutual position of the effector and an object to be grasped. Another approach, usable for teleoperation over high-latency or unreliable networks, is based on user intent recognition [17]. The system classifies (delayed) user input and according to scene state provides assistance. The user is given freedom to switch system modes (manual, semi-autonomous, autonomous), synchronize local visualization with remote actual state or to plan robot motion to fit its state in the local visualization. While previously mentioned systems were limited to one functionality, the system from [46] represents an integrated environment with different tools to support daily living activities of a motor impaired user. Its video-centric web-based interface allow control with a variable level of autonomy: an object may be selected by a user and then grasped autonomously, or a user may set gripper pose and close the gripper manually. The system is also equipped with a task-level planning system to provide cognitive support during complex or long-running tasks and to enable task-relevant undo function. During operation, the interface shows steps of the current task and automatically switches its mode, according to the current step of the task. The user may decide to perform any part of the task manually or has to it if the automatic execution fails.

Another approach (used extensively for rescue robots, for instance during DARPA Robotics Challenge) is based on affordances, defining relationship between a robot and actionable objects in its environment. The interface described in [83] is based on an integrated task execution system and affordances (constituting of 3D model and metadata) for interaction with physical objects. The affordance may be detected automatically, an operator may give a hint to the perception system (e.g. by selecting a region in the image) or fit the affordance fully manually. The operator may preview the robot plan and request or decline its execution. Normally, the task is executed fully autonomously and the operator just supervises its execution. If needed, the operator may switch to a semi-autonomous operation (e.g. by providing a previously mentioned hint to the perception system) or to a low-level teleoperation.

In industrial applications, high-level robot programming based on underlying autonomous functions gain a significant attention. For instance, the mobile manipulator in [100] supports a task-level programming based on a small set of parametrizable skills (derived by the authors from existing worker instructions), where parameters are set either by a user through various modalities or by an automatic task planner. In this approach (and also generally), a skill is composed of primitive robot motions (motion primitives). The skills are object-centered – meaning that program execution depends on cognition and that execution is to some degree robust to changes in the environment. The approach from [116] is also based on a task-level programming; however, the used interface is highly unconventional: specifically designed tangible blocks are used to select objects, to assign a required action as well as to specify order of actions. A robot's program is compiled from used blocks. The advantage of the approach is clear: interaction occurs within the task context and is highly intuitive (requires no learning). On the other hand, suitability for more complex tasks seems questionable, despite recent addition of a projected overlay, providing support during robot programming [118].

Further overview of the related work related to semi-autonomy may be found in Section 5.2.1.

3.1.2 *Modalities and devices*

Any machine (e.g. a computer, a robot), in order to be usable by humans, must have an interface through which happens interaction between the machine and its user. The interface has one or more input and output channels. These channels are called modalities, where a single modality could be defined as a mode of communication according to human senses or type of computer input devices [63].

If interaction happens through more modalities, it becomes multimodal. As a human-human communication is inherently multimodal, the multimodal interaction is in general considered as a more natural than the singlemodal interaction. The most often utilized human senses are vision and hearing as they constitute a high bandwidth communication channels. Different input or output modalities could be used simultaneously as it is the case in a human-human interaction or consecutively as it is so far the case in most human-machine interactions. Each modality may be used to communicate different type of information, or more modalities may be used to communicate the same information – in this case, the interface could be considered as redundant [138].

The choice of modalities and their actual usage depends on the particular task, a robot and an end-user group. For personal robots, speech is often utilized although natural language processing is a highly complex problem. In order to cope with associated difficulties, authors of [30] evaluated an approach based on vision and speech recognition supported by a learning algorithm and a set of failover modalities (mobile phone application, external microphones, and a tablet mounted on the robot's chest) to make interaction with a social robot more robust. The results from [120] suggests, that a combination of more simple modalities (color, sound and vibration) may successfully convey emotions (happy, sad, angry, relaxed).

If the task permits, even a robot's body may be used as an interface. For instance, robot's intents may be indicated by its posture [10] or by a specifically designed motion [16]. Also the robot arms may be used for inputting information (see Section 3.1.2.2).

Similarly, a user's posture or motion may be used to communicate information to the robot. For instance, pointing in the household scenarios was shown to have a sufficient accuracy (9.6 cm) for object selection [108]. Gestures might be detected using vision, depth data, by a wearable device as e.g. Myo Armband [102] or by an IMU device such as Wii Remote [5]. However, in real-world applications, gesture-based control might not be robust enough as it has to cope with e.g. spontaneous human motions [105]. Similarly to gestures, gaze could be used to select objects e.g. to command a robot to pick them up [79]. The gaze-based input is of special importance for users with motor impairments and thus limited other possibilities of commanding a robot. Moreover, a user's physiological condition could be measured by a biofeedback sensor allowing a system to adapt dynamically to the user e.g. by estimating workload [56]. Emotional state of the user (anger, happiness) might be estimated using a far infrared camera [15].

The task sensitivity of modality selection could be demonstrated on results from [113] where three modalities (voice, gesture and tablet) were used for two tasks: training of a welding path and correction of the trained path. While the tablet performed best for the path planning (in terms of a self-reported mental workload), a voice control was better for path correction. Moreover, the importance of modality selection rises with the task complexity [126].

Probably the most common form of a human-robot interface is still a Graphical User Interface (GUI) application on a standard computer monitor accompanied by a mouse and a keyboard. Eventually, the visualization within this setup might be stereoscopic to improve depth perception [130, 88]. Various devices could be used in conjunction with the mouse and the keyboard for input as e.g. a joystick or a 3D mouse [130].

Recently, handheld devices with a touch screen gained a significant attention thanks to their portability and ability to realize augmented reality (AR) (more on AR in Section 3.1.2.1). For instance, the system from [100] uses a tablet to create sequence of skills constituting program of an industrial robot. Consequently, other methods as kinesthetic teaching and pointing gestures are used to set parameters of skills, e.g. to select a particular object for "pick object" skill. The touch-based device may be even integrated into the robot itself [30, 10, 35]. The main advantages of the touch input are that it is easy to use and widely known to the general public.

3.1.2.1 *Mixed reality*

The mixed reality could be understood as a display method based on the merging real and virtual environments. More exactly, the mixed reality lies anywhere on the "virtuality continuum", except its extrema (real and virtual environments) [92]. According to such definition, the augmented reality should be considered as a subset of mixed reality; however, in practice, the terms are commonly used interchangeably. In other words, the objective of the mixed reality is to enhance the reality with an artificial content rather than provide a purely virtual immersive environment as it is the case for virtual reality [13]. The augmented reality system is supposed to have three following characteristics [8]:

1. Combines real and virtual.
2. Interactive in real time.
3. Registered in 3D.

A mixed reality platform might be based on a handheld device [90, 123, 81], a head mounted display [54, 131] or a camera-projector solution [22, 28, 42]. When designing the interface, perceptual issues as e.g. a limited field of view, a depth ordering and occlusion introduced by the selected technology and used method has to be taken into account [74]. Despite potential problems, the mixed reality has potential to improve HRI. For instance, it could help to avoid context switches which are normally inevitable when the user has to observe the real environment and the robot as well as the video interface [54]. Another usage could be to convey the robot's intents, especially for appearance-constrained robots [131, 22, 28] not able to convey those by other means.

Nowadays, especially spatial augmented reality (SAR) seems to be a highly promising method enabling users to interact with the robot within the task-context. For instance, its use was investigated to program a mobile welding robot [5] or in a long-term study focused on projecting assembly instructions [42]. In contrast with handheld devices, SAR has following advantages: both hands are free, projection is visible by anyone, no physical load caused by need to hold the device. Although the head mounted display also frees users' hands, there is question of its long-term use suitability (possible health risks) and moreover, contemporary devices are expensive[3] and probably not robust enough for e.g. usage in industrial environments. Moreover, the head mounted displays are either tethered or with limited battery life[4] which might limit its deployment even further.

---

[3] Microsoft HoloLens Commercial Suite $5.000, Meta 2 Augmented Reality Development Kit $1.495, MagicLeap One The Creator Edition $3.000 (expected price).

[4] Microsoft HoloLens has declared battery life of 2-3 hours of active use.

As the mixed reality is relatively new (both in general as well as in the field of HRI research), there is lack of proven interface patterns, design guidelines and usability evaluation methods. As the technology is not mature, technical problems are also common (lagging interface, bad registration, etc.). All of the previous problems might contribute to sort of contradictory results of some studies. For instance, the study from [81] reports a positive effect of AR; however, the users of the AR needed more support than those using baseline solution. In the long-term study [42], the AR system projecting assembly instructions leaded to reduced learning curve of novice assemblers; however, performance for expert workers decreased. In the study [123], usage of the tablet-based AR leaded to decreased mental demands; however, to increased task completion times.

3.1.2.2 *Physical interaction*

A physical interaction of a user and a robot may refer to an unwanted contact between those two or to an intended contact in cases where the robot (arm) itself is used as an input or output modality. The intentional interaction could happen with the robot itself (if the robot arm could reduce its stiffness) or trough an additional device. Probably the most common examples of using a robot's arm as an input modality are kinesthetic teaching [135] and programming by demonstration [3]. Those methods seem relevant especially for non-expert users. For instance, in the user study [135], participants with no prior experience with industrial robots and with good spatial vision abilities rated physical interaction as easy, comfortable and self-explanatory. On the other hand, participants with prior experience rated the interaction less self-explanatory and reported a higher cognitive load. Another possible approach is to command the robot with relatively simple haptic commands as taping and pushing [44], which could potentially improve user experience and allow to better maintain physical and cognitive engagement with the task. Despite utilizing a robot arm as an input device, the arm could also communicate information to the user – acting as an output modality [16], or it could even act in a bidirectional manner [128]. Robot arms not originally designed for any form of physical interaction could be retrofitted to provide such functionality, e.g. by addition of tactile surface sensors for gesture input [94].

Further overview of the related work related to modalities and devices may be found in Sections 5.2.2, 6.3 and 8.2.

3.1.3 *Task presentation and situation awareness*

An explicit communication (usually by visualization) of the task and its current state is usually not needed for trivial tasks (as those quite often used in user experiments). However, for more complex tasks as e.g. assembly of a product or long-running tasks as e.g. a remote manipulation with many required steps, the issue of a suitable task presentation arises. Knowledge of the current task state is related to situation awareness, as well as to safety. For instance, when the user knows which object is the robot going to manipulate, he or she can avoid touching it and thus avoid potential collision [82]. However, there is a challenge on how to display state of a highly complex system (e.g. a cooperative workcell) in a comprehensive form [33]. There exist several solutions for (collaborative) robot programming [4, 116, 73]; however, only a few of them also provides some task execution monitoring [5, 99, 51, 83] – usually limited to highlighting current step of the program, without any further cues for the human user. Some of the solutions uses elements of within task-space interaction, as e.g. SAR for setting welding points [5] or kinesthetic teaching for setting positions [100]; however, the major amount of interaction still happens on a monitor or a handheld device. In that case, the split attention effect [65] may occur, leading to unnecessary increase of user's cognitive load.

3.2 COMMERCIALLY AVAILABLE SOLUTIONS

When considering personal robots available on the market, the existing options are moreover limited to some form of intelligent assistants similar to Amazon Alexa or Google Home (which may be considered as smart speakers), although robotized to some extent. Typically, the functionalities include: natural language processing, facial recognition, notices, controlling smart home appliances, security features, telepresence, sharing or getting information, etc.

For instance, despite aforementioned typical functions, Jibo by the company of the same name has articulated torso and is able of smooth animated motions. In contrast to speaker-like intelligent assistants, the interaction between Jibo and the user may be potentially richer – the robot may express certain information using motion and a touchscreen face. Moreover, the robot is able to respond to touches of its body (e.g. rubbing of its head). ElliQ by Intuition Robotics (production scheduled for the end of 2018) focuses on elderly users and attempts to offer an active aging companion.

In contrast to the previous robots, Buddy by Blue Frog Robotics is mobile and has an arm equipped with a miniature projector. Another approach could be represented by KOMPAÏ-2 (KOMPAÏ robotics)[5], which is a healthcare robot able to provide standing/walking support and to carry small items, which user may put into its tray.

---

[5] The robot is currently available for evaluations and pre-deployments.

There are also various robots wholly focused on the telepresence functionality as e.g. Beam by Suitable Technologies. A humanoid-like robot by Softbank Robotics – Pepper, is currently not being used in private households; however, rather in shops or offices to invite customers, etc. It attempts to recognize an emotional state of its interlocutor and adapts his behavior accordingly. The arms are mainly used for gesturing, although also able of a basic manipulation with objects.

There also exist narrowly specialized robots as e.g. Paro by PARO Robots – a therapeutic robot with the appearance of a baby seal, capable of sensing touch, heat and sounds.

Within the field of industrial collaborative robots, the greatest attention is naturally given to the safety features of the robots. The collaborative robots are designed either in a way that they do not have enough power to harm a human co-worker (ABB Yummi, Rethink Robotics Baxter/Sawyer), or their power could be limited to allow a collaborative operation (Universal Robots URx, Kuka LBR iiwa/iisy).

The robots usually have ability of sensing collisions through measurement of joint torques and are able to stop their operation in case of an unwanted contact with an obstacle. There exist various approaches to further enhance safety of collaboration. For instance, Yummi has a soft foam padding, Rethink Robotics robots uses a special type of actuators (Series Elastic Actuators) able to absorb energy and Franka Emika uses a torque-based control (in contrast to a more common velocity or position-based control).

There also exist various third party solutions – e.g. a padded cover with tactile and capacitive sensors (MRK-Systeme SafeInteraction, Blue Danube Robotics AIRSKIN). The usage of capacitive sensor enables robot to sense the immediate proximity of a human co-worker and stop even before actual contact occurs. A similar device (Faude 3D COLLISION PROTECTION) is available also for UR5 robot.

Although some of the robots have integrated vision (Yummi, TM5) or obstacle sensors (sonar in case of Baxter), they are not able to sense its human co-worker and adapt their motions accordingly (trajectories are pre-programmed anyway). Commonly, an external safety sensor as e.g. a laser curtain is utilized whose output signal may slow down or shut down the robot if the worker disrupts the perimeter.

A physical interaction with robots is rare and mainly occurs exclusively during programming – in case of the robots which support a lead-through teaching of waypoints/trajectories. A limited number of robots are specifically design with interaction in mind, e.g. with integrated input/output interaction elements. An example could be Baxter, which posses LCD displaying an animated face able to convey a current state of the robot (where e.g. confused face means error or misconfiguration). Moreover, the display may show a robot's program. The robot also has some LEDs (Attention/Condition Ring) and controls (Training Cuff, Navigator). It is possible to program the robot solely using these controls and the integrated display without any external device. However, fine-details or complicated tasks has to be anyway programmed offline (using Intera Studio). Another example of the robot with integrated control could be Franka Emika (Franka Pilot).

There even exist accessories to retrofit non-interactive robots, e.g. a light and sound devices mountable on robot's flange (Alumotion YOUring, Faude ProLight). The flange adapter could be also equipped with buttons to simplify some common tasks during programming (switch to a zero gravity mode, store current position, etc.).

3.3 GENERAL STATE OF THE ART SUMMARY

The previous sections provided an overview of the current state of the art solutions on the field of HRI. From this overview, it seems that HRI is still quite limited and there is a great potential for improvements enabling a closer teamwork between human users and robots. The chosen solution within this thesis is to combine existing approaches in a novel way, in order to realize task-centered interaction suitable for non-expert users. The resulting method is presented in the next chapter.

4

METHOD FOR TASK-CENTERED INTERACTION

The following chapter introduces a novel method for task-centered interaction, which has been applied and evaluated within two different use cases.

4.1 PROPOSED METHOD

The aim of the method is to integrate already existing approaches with a high potential to improve HRI within the intended use cases in order to benefit from the resulting synergic effect. The essential idea of the method is that interaction should happen within the task space (whether it is a real or a virtual one), with the highest possible utilization of already available modalities. The another important aspect of the method is lowering the user's cognitive load by e.g. transferring interaction onto a higher level of abstraction (task-level interaction) and providing just enough information in order to allow the user to fully focus on the task at the hand. The previously stated features also contribute to the suitability of the method for non-expert users for which the method is explicitly intended. To the best of my knowledge, the method represents a novel approach to the HRI. The method is defined by its following key characteristics.

Interaction elements embedded into the scene.

Originating in ecological user interface design methodology, aimed on lowering user's cognitive load and attention switches. Could be achieved by usage of the mixed reality approach.

Utilization of robot-integrated capabilities.

Utilization of robot capabilities as a sensing of the environment or an automated motion planning enables the task-level interaction – effectively reducing demands on the user as e.g. less inputs are required. Moreover, integrated safety features as e.g. a collision avoidance or an environment-aware motion planning could reduce stress for the users and allow them to focus on the task at the hand rather then on continuous checking whether the robot performs safely. In order to achieve this, advanced perception capabilities are needed.

Communication of the robot inner state.

In order to make the robot's actions predictable and understandable by the user, it is inevitable to e.g. visualize its inner state, particularly perception (which objects or obstacles are detected), intentions (goal of the current movement) and current execution status (waiting for user input, error situation). Communication of the relevant robot inner states to the user could also lead to increased safety (user is aware of what the robot is doing at the moment and may avoid potentially dangerous situations) and spatio-temporal context awareness – lowering demands on a short-term memory and thus lowering workload.

Context-sensitive user interface.

The interface should present the right information at the right time, according to the current task and environment state instead of presenting excessive amount of information all the time. A limited amount of the context-relevant information helps to maintain a reasonable mental load.

Task-appropriate modalities.

Input and output modalities selected according to the task and its specifics. Maximize utilization of already present modalities as robot arm, or user's body (e.g. sensing pose and activity of the user). Bring as much as possible of the interaction into the task-space by making it interactive itself.

Although individual above mentioned characteristics have been already utilized in some form in the existing literature (see Chapter 3), their combination has not yet been used. The named characteristics when used jointly, allow rich and close HRI.

4.2 APPLICATION AND EVALUATION

The proposed method was used to direct design of the user interface for teleoperation of semi-autonomous service robots. The single-window interface is based on integrated 3D virtual scene. The scene consists of visualization of continuously updated 3D model of the remove environment, robot model and various interaction elements. Interactive in-scene elements serve for two main purposes: navigation and manipulation. A user may freely choose from various interaction methods with variable level of autonomy according to current needs. For instance, the user may set waypoints for the robot and it navigates there autonomously (planned trajectory is visualized) or directly teleoperate the robot (with support of the collision avoidance system). When performing a manipulation task, the interface guides the user through the process step by step. For both navigation as well as manipulation tasks, a 3D mouse is used. The control using the mouse is transformed using the non-linear formula and adjusted according to the current 3D scene viewpoint so it provides an easy to use and intuitive input modality. Optionally, a stereoscopic visualization is available in order to convey depth perception cues. The interface in full detail and its evaluation process is further described in the Chapter 5.

The method was also applied to the problem of industrial robot programming. In particular, to the use case of a worker's robotic assistant. In this case, interaction happens within the shared workspace, centered around an interactive workshop table with ISAR. The interface allows an ordinary skilled worker to parametrize the robot's program, e.g. to adapt it to changes in production. The ISAR is used to visualize robot perception, display context-relevant notifications and finally, to show explicitly the robot's program. The program visualization allows to switch between steps during learning phase and it shows a current instruction (including its context, i.e. previous and following program instruction) during an execution phase. Among the interactive table, robot arms might be used as input devices (e.g. for tasks requiring 3D data input). The interface design started with Wizard of Oz (WoZ) experiment further described in Chapter 6. The goal of the experiment was to reveal a relationship between a input error rate and a user preference for various modalities. After that, the target use case and the initial scenario were specified (see Chapter 7) and the initial prototype of the system was developed. In order to evaluate the method and uncover usability issues of the prototype, a lab experiment was carried out with six regular workshop workers. The current state of the system and the experiment are further described in Chapter 8.

5

TELEOPERATING ASSISTIVE ROBOTS: A NOVEL USER INTERFACE FOR REMOTE MANIPULATION AND NAVIGATION RELYING ON SEMI-AUTONOMY AND GLOBAL 3D ENVIRONMENT MAPPING

5.1 INTRODUCTION

Autonomous systems cannot yet be programmed to handle all possible situations. A remote human operator may help the robot to solve many difficult situations. The collaboration between humans and robots, often referred to as either shared autonomy or human in the loop, might be highly useful in cases where robots often fail, e.g. in object recognition and environment manipulation. On the other hand, an operator should not be bothered by repetitive low-level tasks which can be solved by the robot itself. Then, the operator is not overloaded with solving trivial issues and may concentrate on the important ones and, for example, control more robots due to the time freed. The challenging issue is to equip a potential human operator with easy-to-use but powerful interaction and control tools to act appropriately and effectively in various situations.

This paper describes a novel 3D interactive user interface and its components. The interface allows a user to assess the situation on a remote site, safely navigate in environments with obstacles and with narrow passages where autonomous navigation is likely to fail and to grasp previously untrained objects in cluttered scenes, in various poses and on non-flat surfaces. It is based on common low-cost hardware and can be optionally used with a 3D mouse for intuitive robot navigation and arm control. Additionally, stereoscopic display may be used for improved depth perception. It also includes a module for building a memory-efficient 3D map of the environment, which is used for both visualization purposes and for the planning of collision-free arm trajectories.

The interface has been developed as part of a larger system within the SRS project[1]. The goal of the SRS project [107, 103] was to develop a personal robot able to support elderly people in independent living at their residence. Based on the results of a survey of user-demanded features and on considering what is

---

[1] Multi-Role Shadow Robotic System for Independent Living, http://srs-project.eu (accessed 12/07/2015), technical documentation available at http://wiki.ros.org/srs_public (accessed 12/07/2015)

realistic to implement on current hardware [85], when designing the remote user interface, our primary objectives were navigation and manipulation capabilities.

The SRS project adopts a semi-autonomous paradigm, where under normal circumstances the robot is controlled by its autonomous system, which follows instructions given by the elderly person. Local control is based on a mobile device, which allows the user to initiate autonomous actions such as "bring an object". So most of the time, the robot is controlled by its autonomous system without any remote intervention. In case a problem occurs with task execution, there is a second, more advanced interface, which is typically used by a family member who lives separately. The family member can, through a tablet-based interface, control the robot to help the elderly person physically with their daily living tasks. If there is a problem unsolvable by the previous two interfaces, a professional operator is called who can remotely control the robot through the most advanced interface (the one described in this work) and use semi-autonomous functionality to guide the robot, e.g. to bring an object unknown to its autonomous system. The autonomous system and its connection to various interfaces is further described in [107].

The Care-O-bot 3[2] service robot [110] was used as a project demonstration platform. It is based on an omnidirectional platform with positionable torso and a sensor head, a Kuka LBR dexterous manipulator (7 DOF) equipped with a Schunk SDH three-finger hand (7 DOF) and tactile sensors. The robot uses three 2D laser scanners for obstacle avoidance and a Microsoft Kinect RGB-D camera for 3D perception.

To create the interface, we have combined various existing components with newly designed and developed ones in a novel way, enabling semi-autonomous operation of the robot. The results of two experiments with novice users [86, 88] have suggested high effectiveness and suitability of the approaches incorporated in our user interface. Even a short simulation-based training of 60 minutes (including introduction to the robot) was sufficient for achieving high success rates in navigational, search, and manipulation tasks in a home-like environment. In previous publications we have described the overall usage concept underlying the present user interface [85, 87], the framework enabling its semi-autonomy [107], and results of experiments on user interface components [86, 88]. The present paper describes the latest iteration of the user interface, iteratively improved based on the results of several evaluations.

This paper describes a user interface for a semi-autonomous robot. Section 5.2 presents related work. Section 5.3 gives an overview of the goals that motivated development. Section 5.4 describes the development and evaluation procedure. The

---

[2] http://www.care-o-bot.de/en/care-o-bot-3.html (accessed 12/07/2015)

interface architecture and its basic functionality are detailed in Section 5.5. Sections 5.6 and 5.7 describe two main use cases for our interface: remote navigation and manipulation. Section 5.8 draws conclusions.

5.2 RELATED WORK

In this section, we will give a brief overview of the previous work related to remotely operated robots from different perspectives.

5.2.1 *Robot Control Architecture*

Various approaches exist for assistive robot control architecture. For instance, the robot presented by Michaud et al. is fully teleoperated and focuses mainly on establishing communication between teleoperator and elderly person [91]. When a teleoperator is not available, the robot is not able to perform any task. To overcome the lack of true autonomy, some approaches introduce nearly full autonomy with the possibility of human intervention when necessary. These approaches are referred to as semi-autonomy, shared autonomy, adjustable autonomy, or human in the loop. Such systems may provide to the operator tools with various levels of autonomy. For instance, the system proposed by Muszynski et al. based on egoperspective visualization offers three levels of autonomy [93]. A similar approach was designed by Bruemmer et. al. where the robot also offers different levels of autonomy [21]. Their user study has shown that users performed better when using tools with more autonomy. Similar results suggesting that more autonomy leads to an improved teleoperator performance were obtained in [24, 77]. The recent efforts utilize human semantic knowledge to help robots perform better [133], which might lead to less operator intervention and thus to decreased workload. Using a robot's motion planner instead of low-level joint control can be also considered a semi-autonomous approach and according to [132] it is also more effective. Using high-level arm control including a cartesian planner and collision avoidance according to [77] allows users to focus fully on the cognitive part of the task, which is usually the most challenging for the robot.

5.2.2 *Visualization And User Interaction*

Traditional video-based interfaces transmitting images from a camera mounted on a robot provide low situational and spatial awareness and increase the risk of collisions [7]. The lack of human-robot awareness, e.g. knowledge of the robot's state and the state of the environment are the primary causes of incidents during teleoperation [31]. The main problem of video-based teleoperation lies in the limited field of view and the absence of depth data [140]. Traditionally, additional information is shown to the user in a separate window or overlaid over the video on the sides. Individual information on the state of the robot and the environment must be mentally correlated, which increases cognitive load. The ecological interface paradigm [95], on the other hand, fuses as much information as possible into a one coherent virtual scene and acts as a form of a mixed-reality. Interfaces based on this paradigm appear to provide better situation awareness and require less mental load [7]. A virtual scene presented to the operator can be based on a manually created 3D model [75], an extruded 2D map [21], or a continuously updated 3D model based on sensor measurements. Results of a study by Mast et al. [86] have suggested the usefulness of an automatically built and updated 3D environment model for navigating a robot remotely.

In case of video-based egocentric interfaces aimed at robot navigation, a joystick used to be a frequent choice. New ways of control were introduced for virtual reality-based interfaces, which are using exocentric display perspective such as "point and click" [24], when a goal position for the robot is specified by clicking a place in the virtual environment. Most recent interfaces tend to use virtual widgets, also called interactive markers [24, 77]. The advantage of these markers is that they are an integral part of the virtual scene and no special device is required as opposed to control using e.g. the Phantom device [37], motion capture [70], data gloves [58], or brain-computer interfaces [11]. A crucial issue associated with the difference between the input devices and the visualization is the potential problem of display-control misalignments introduced by using different coordinate systems. Thus, the remote operator has to keep switching mentally between the coordinate systems. This issue has been addressed by either using artificial cues [25] or by choosing an appropriate coordinate system.

5.2.3 *Imaging Equipment*

A conventional 2D display can only convey depth perception based on monocular depth cues, consisting of perspective, occlusion, lighting and shadows, relative object size, surface textures, etc. Stereoscopic displays on the other hand enable users to naturally judge relations between objects, based on provided binocular cues [34]. Potential advantages of stereoscopy have been investigated in several studies. For instance[115] suggested that there was no significant difference in completion times between stereo and mono display in a navigation task. On the other hand, there was a substantial difference in the number of collisions against the environment, which were lower for the stereo condition. Utility of stereo display for dexterous manipulation has been investigated in [40]. In their comparison of an interface based on multiple 2D views of the scene versus stereoscopic display, the stereoscopic mode resulted in a 60 % decrease of task completion time. Influence of mono and stereo visualization of 3D scan data on users' ability to understand the environment has been investigated in [39]. This work points out that the stereoscopic visualization reduces the risk of misunderstanding the environment. Various technologies for stereoscopic display have been compared in [80] and it was found that shutter glasses provide depth impression comparable to much more expensive polarized walls or CAVE.

5.2.4 *Conclusion*

Until fully autonomous assistive robots will be available, some form of teleoperation will likely be necessary. Using a semi-autonomous approach a robot remote operator's workload can be lowered and at the same time performance increased. The degree of the underlying autonomy plays a crucial role in operators' performance. Another important factor is the user interface, its design, capabilities and ability to convey rich information. There are approaches focused on particular aspects however there is currently none utilizing a synergy of these aspects, moreover using affordable hardware for user interaction.

5.3 USER INTERFACE DESIGN GOALS

The vision underlying our user interface is a robot that acts autonomously as much as possible. Only when it fails to accomplish a task by itself, a human operator takes over remotely and intervenes with navigation or manipulation. During the intervention it is up to the operator to select appropriate tool with given level of autonomy leading to the lowest workload and safe operation. To be able to solve a wide range of problems, users were to have a high degree of control over the robot. The user interface further had to be easy to use as it was primarily aimed at teleassistants, i.e. non-roboticists who were only to receive basic training [85]. Our goals were thus to maintain a high degree of robot autonomy while allowing a high degree of controllability, in a system that would still be easy to use. We identified a number of interesting approaches for achieving these goals:

- Techniques for assisted, semi-autonomous remote manipulation and navigation, aiming to take away load from the operator and allow safe operation over unstable network connections e.g. [91, 21, 77]

- The ecological interface paradigm that enables an operator to directly infer possible actions from the visualized environment and thereby aims to reduce cognitive load and improve situation awareness and user interface usability e.g. [95, 7]
- 3D visualization of the large-scale environment outside the robot's current field of view for better spatial orientation e.g. [75, 21]
- Utilization of contemporary 3D sensors able to generate live colored 3D point clouds for a high degree of realism and detail e.g. [7, 77]
- 3D environment mapping based on 3D sensor data for realistic large-scale representation of the environment, aiming to improve spatial orientation and situational awareness e.g. [57, 139]

While each of these approaches is promising on its own, they had so far been used in a rather isolated way. For example, ecological interfaces were restricted to either navigation [95] or manipulation [7] or did not employ semi-autonomy. Some previous interfaces relying on 3D environment visualization were based on manually created 3D models [75, 21] rather than on sensor-based environment models that can be generated and kept up to date automatically. Applications of 3D environment mapping using 3D sensors were not used for visualization in the user interface [139]. We thus aimed to create a holistic solution for both semi-autonomous remote manipulation and navigation, using modern technology and integrating the above-mentioned approaches into a consistent user experience. We relied on commonly available low-cost hardware and, where possible, on software components already available. We developed own components or extensions to existing ones where necessary.

5.4 ITERATIVE DEVELOPMENT AND EVALUATION

The user interface was developed following a human-centered design process [36] in several iterations of development and testing, evolving from a conceptual prototype into a fully functional user interface. A total of 430 prospective users were involved in studies directly and indirectly related to this user interface, carried out in the SRS project [107]. Early studies focused on eliciting user requirements [85, 84] and on the development of an overall usage concept also including two reduced-functionality mobile user interfaces not described here [85, 87]. The present user interface was tested five times at different stages of development with a total of 81 users. All evaluations were carried out with non-expert users. As the focus of the present paper is the description of the user interface, we just give a brief overview of the evaluations here and, where available, refer to the publications describing them for more detail.

The first evaluation was a usability test carried out in Germany at Stuttgart Media University's User Experience Research Lab employing a horizontal prototype of the user interface (static screens simulating interaction) [85]. Seven teleassistants from home telesupport centers were recruited for this study. We determined 18 usability problems that lead to 10 design changes in the horizontal prototype.

In the second evaluation an early implementation of the user interface was tested. This evaluation was carried out by project partner Don Gnocchi Foundation in Milan. Five users remotely navigated the robot through a realistic model apartment purpose-built for evaluations. This study gave insight into the strengths and weaknesses of various control modes for remote robot navigation. Also, numerous technical and usability issues were uncovered and addressed in subsequent development.

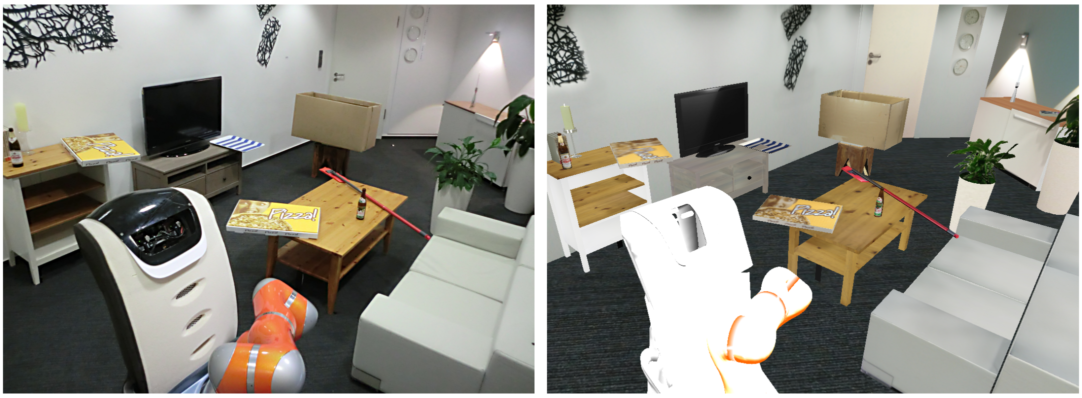
The third evaluation was again carried out in the lab in Stuttgart and employed the Gazebo robot simulator [69]. We created a detailed apartment model for carrying out evaluations in simulation under realistic conditions (Figure 5.1). It consists of three rooms, connected by corridors, and contains 80 household and furniture items with realistic physical properties such as weights and friction resistances. The apartment was precisely modeled after the site used in our later experiments. We have made this model freely available so it can be used by other researchers[3]. 14 users participated in this evaluation. The evaluation focused on strengths and weaknesses of various approaches for visualizing the remote environment in the user interface. It also served as a comprehensive pilot study for the experiments carried out subsequently in reality.

When the user interface had reached a fully functional and stable state, we carried out two experiments with more narrowly specified questions and larger numbers of participants in a purpose-built model apartment on Fraunhofer IPA's premises in Stuttgart. The first experiment, i.e. the fourth evaluation, with 27 participants investigated the utility of two different types of global 3D environment maps (voxel-based and geometric) visualized in the user interface for remotely resolving navigational problems the robot cannot handle autonomously. Results are briefly summarized in Section 5.5.3 and described in detail in [86].

The second experiment and fifth evaluation [88] was carried out with 28 participants at the Fraunhofer site. Its first purpose was to investigate potential advantages of stereoscopic presentation of the user interface for remotely resolving problematic situations with object manipulation and robot navigation. These results are briefly summarized in Section 5.5.5 and described in detail in [88]. The

---

[3] http://wiki.ros.org/srs_user_tests (accessed 12/07/2015)



(a) living room in reality (b) modeled living room

Figure 5.1: Realistic apartment model designed for evaluating the user interface; includes living room, bedroom, kitchen, corridors, and 80 household and furniture items.

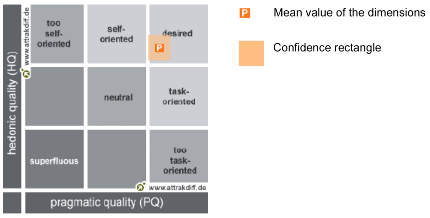
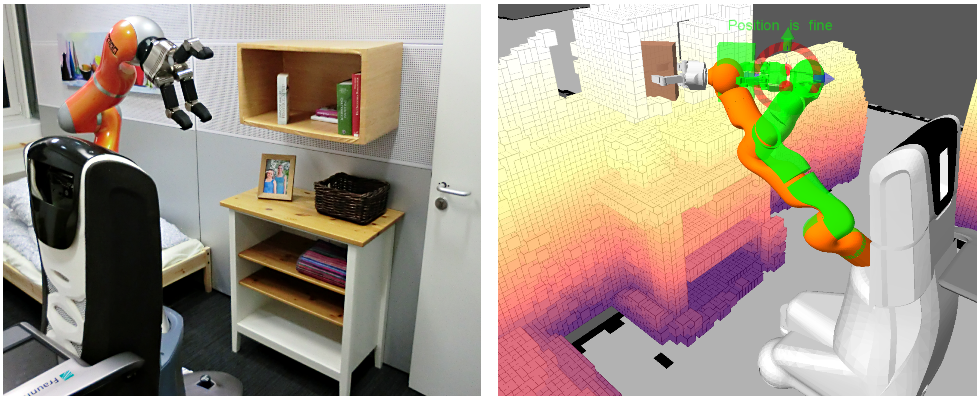


Figure 5.2: Results of the most recent user experience assessment, based on the user interface's stereo mode: mean user ratings for pragmatic quality (usability) and hedonic quality.

second purpose was to obtain an assessment of the quality of users' experience of interacting with the interface. This included ratings of usability and hedonic quality, measured with the AttrakDiff instrument [53]. The main user experience results are visualized in Figure 5.2 (based on stereo mode, which scored higher). The user interface overall falls just into the range of "desired", which is a highly encouraging result but there is also still some room for improvement. More details on these results can be found in [88].

5.5 VISUALIZATION AND INTERACTION APPROACH

The interface consists of many components, the main ones being depicted in Figure 5.4. It runs on two computers – one on the robot in a Wi-Fi network, and a remote user station. The front-end user interface is based on a visualization tool combining the interactive 3D scene showing most of the information and the side-panels with conventional elements like buttons etc. The user is provided with a 2D mouse, a 3D mouse and a conventional or a stereoscopic screen. The user station also hosts an arm motion planning component providing, among others features,



(a) view from the interface with the object already segmented and grasped

(b) robot having a problem in autonomous mode (cannot recognize object

Figure 5.3: The interface allows the user to manipulate an untrained object which cannot be handled autonomously.

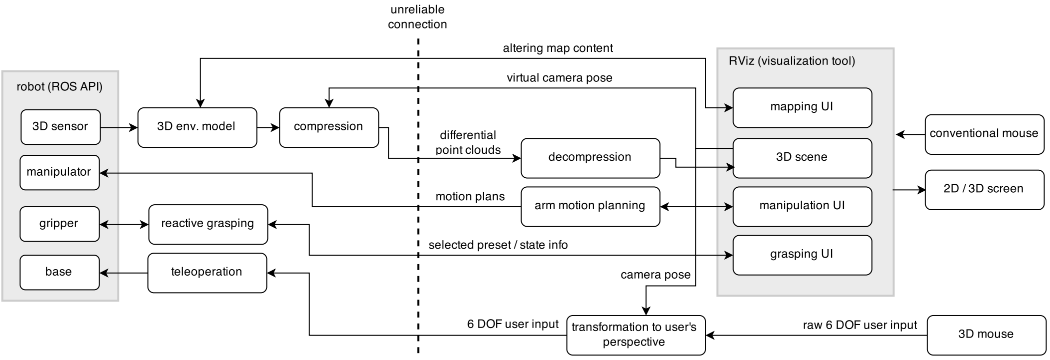


Figure 5.4: Simplified diagram showing interactions between main components of the user interface, their connection to the robot and input and output devices.

inverse kinematics which is used for visualization. The robot's computer hosts, apart from low-level drivers etc., components for mapping, grasping and teleoperation. All components communicate using the ROS middleware and thus can be easily reused.

The interface specific feature is an API which can be used by the autonomous system to ask the user for help if a problem arises. Normally, the interface is disabled. When the robot's autonomous system cannot complete some task (see Figure 5.3a), it sends a request to the interface. The interface then leads the user through the task giving text instructions for completing respective sub-tasks and automatically enabling necessary components such as an interactive virtual arm (see Figure 5.3b). When dealing with a task, the user may at some point (sub-task) decide that the main problem is solved and hand back control to the autonomous system. Alternatively, he or she may decide that the task would be too difficult to complete for the robot and finish it manually. With this approach, the operator's time is conserved as much as possible.

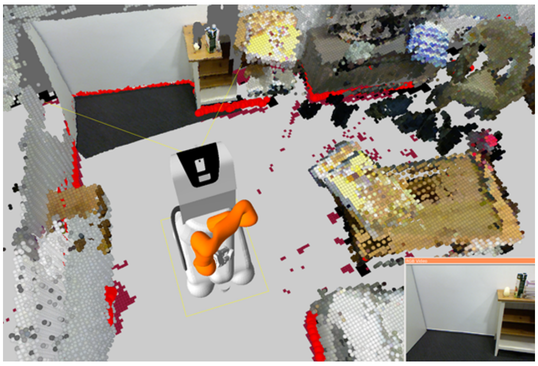


Figure 5.5: The 3D mixed reality environment consisting of a robot model, 2D laser data, a 2D map, a combination of the live RGB-D data in current field of view of the robot (visualized using yellow lines) and the 3D voxel-based map outside it and a video stream.

5.5.1 *3D Mixed Reality Environment*

The user interface is based on RViz[4], a modular 3D visualization tool, for which we developed several custom plugins and an extension for stereoscopy. The largest portion of the user interface is dedicated to a rendered view of a 3D environment. The mixed-reality environment consists of a 2D map relevant for localization and navigation, a continuously updated 3D map, a robot model in proper scale and configuration according to the robot's proprioception. Moreover, there is in-scene visualization of data from three 2D laser scanners and the RGB-D camera. The 3D scene also contains interactive markers for robot control, object representation, etc. Elements of the user interface are automatically switched on and off based on the current context.

5.5.2 *User Interaction*

The user interface can be controlled exclusively by a common 2D pointing device. Optionally, a 3D mouse may be used for some tasks. During our pre-tests, 3D mouse-based control proved to be comfortable, easy to learn, and sufficiently precise even for manipulation in complex scenes. The 2D mouse is used to set the scene view to any angle and distance, to interact with the in-scene 3D widgets, and to control the conventional part of the interface. The 3D mouse may be used to teleoperate the robot's base and to control the end effector goal pose.

The 3D mouse we used, SpaceNavigator[5], is a low-cost device with six degrees of freedom. When using the 3D mouse, all cursor movements are encoded as a

---

[4] http://wiki.ros.org/rviz (accessed 12/07/2015)

[5] http://www.3dconnexion.com/products/spacenavigator.html (accessed 12/07/2015)

vector $(t_x, t_y, t_z, r_y, r_p, r_r)$ where (t_x, t_y, t_z) represents the translational part and (r_y, r_p, r_r) the rotational part in the form of yaw, pitch, and roll angles. We consider the pose of the camera observing the mixed-reality scene and transform control inputs from the 3D mouse coordinate system to the camera perspective. This leads to controlling robot movement in the user's rather than in the robot's coordinate system. The transformation is rather simple – the translation vector (t_x, t_y, t_z) introduced by the 3D mouse is rotated along the z–axis, i.e. the one perpendicular to the floor plane in the scene, according to the current camera pose so that the translation along the z–axis t_z remains unchanged:

$$(t'_x, t'_y) = (t_x, t_y) \cdot \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \tag{5.1}$$

Here α is the current yaw angle of the camera pose in the scene coordinate system. As this transformation of the control commands to the user perspective requires much less mental rotations it should help to lower cognitive load on a user.

To enable the user to control the robot's base during teleoperation or end effector during telemanipulation very precisely at low velocities and at the same time to move fast across longer distances we have introduced a non-linearity into the SpaceNavigator outputs. The following equation is applied to each component of the 6DoF vector $(v_1, v_2, v_3, v_4, v_5, v_6)$ resulting from the 3D mouse:

$$n_i = \left(\frac{v_i}{v_i^{max}}\right)^2 \cdot v_i^{max}, \tag{5.2}$$

where v_i is the original value, n_i is the transformed value and v_i^{max} is the maximal allowed value of the i–th component.

5.5.3 *3D Voxel-Based Environment Model*

The robot's Kinect camera provides standard RGB images as well as colored point clouds at 30 Hz. The sensor has a limited field of view (57° horizontally and 43° vertically), a considerable level of noise and depth resolution decreasing quadratically with increasing distance from the sensor [67]. Mainly due to the limited field of view, using only live point clouds from the sensor for situation assessment or finding obstacles or objects to fetch would be complicated for a remote operator.

To overcome this limitation, we have introduced an environment model which combines point clouds into a consistent global map as the robot travels around the environment (see Figure 5.6). Our solution is based on the Octomap library [57], which models the environment as a grid of cubic volumes of varying size. This grid is hierarchically organized in an octree structure where each node represents

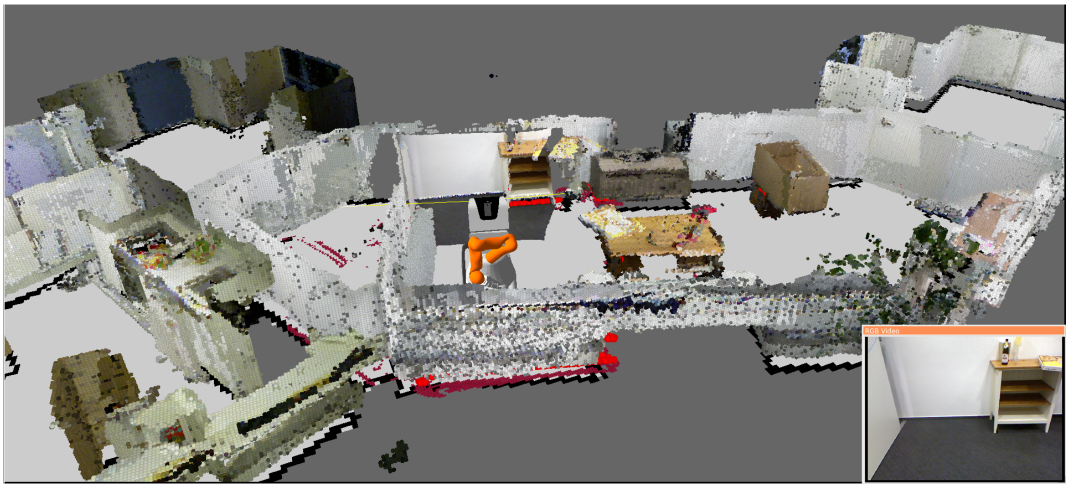


Figure 5.6: Automatically generated and updated 3D model of home-like environment covering an area of 100 m^2.

a space contained in the cubic volume, and this volume is recursively subdivided into eight subvolumes until a preset minimum voxel size is reached. The OctoMap library uses probabilistic occupancy mapping to fuse input sensor data suffering from errors and uncertainty into robust estimation of the true state of the environment. The continuously updated global map is displayed to the user and used for collision-free arm trajectory planning. The approach allows the user to see and consider the whole environment around the robot. See Figure 5.5 for an example of a visualization of a room from a home-like environment using a voxel resolution of 0.025 m. This resolution seems to be sufficient for the model to serve as a clue for spatial awareness and for obstacle avoidance. For high-precision tasks, users can rely on more detailed live sensor data (see 5.5.4).

To cope with limited network bandwidth, especially over unreliable wireless networks, we have developed modules for compressed transfer of differential frames representing the modified parts of the whole global map. They consider the position of the robot's 3D camera in the environment and its field of view and then compute and send to the user's PC the corresponding point cloud in a compressed form. At the user's PC, the point cloud is decompressed and the respective part of the global map updated. Once per 5 to 10 differential frames, the whole map is sent to be able to recover from failures. Figure 5.7 shows the network bandwidth we measured during a test run around the evaluation apartment. Results show that the differential approach can save 65 % of the network bandwidth for the resulting global map of 1,056,575 points. Memory requirements of the internal Octomap representation were growing up to 1.015 GB in this case. To further save network capacity, RGB camera images are transfered using the Theora codec. There are many other possibilities to cope with network issues but these remain to future work.

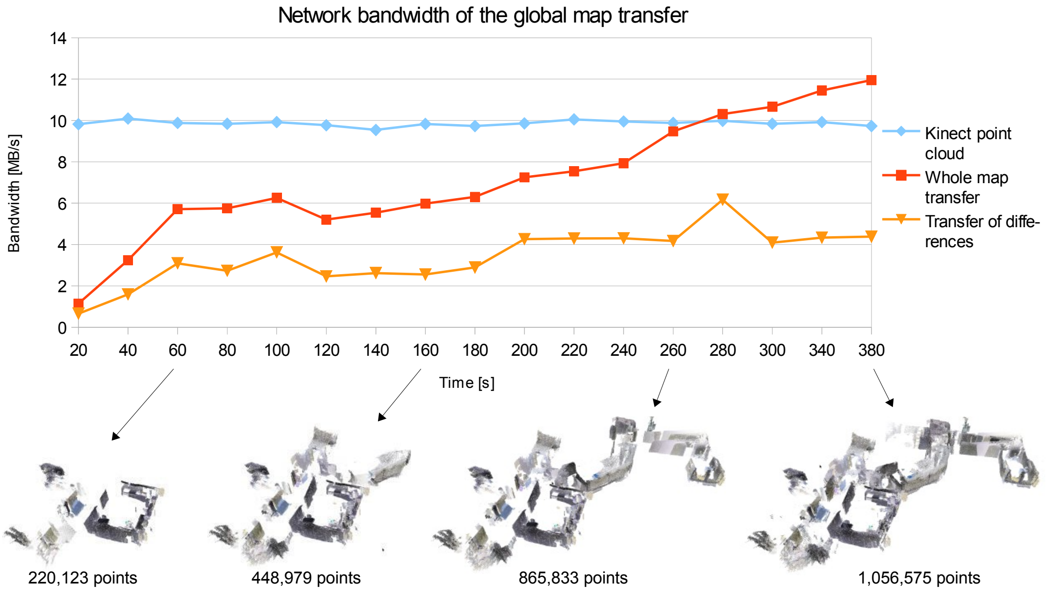


Figure 5.7: Network bandwidth for whole global map transfer is compared to sending of map differences. Input RGB-D data and environment mapping were throttled to process 1 frame per second. The whole map was sent after each 5 differential frames.

We have further extended the functionality of the standard Octomap library by:

- Allowing the user to manually modify a part of the map – either by clearing out a region of the map hindering arm trajectory planning, or by adding an artificial object to prevent the robot from going there
- Filtering incoming point clouds for ground parts and speckles so that they do not obstruct the view and the 2D map
- Removing noise and outdated parts of the 3D map using a ray-cast technique that clears out outdated parts of the environment when they are newly observed by the robot

We investigated the usefulness of visualizing global 3D environment maps in the user interface in an experiment [86]. We compared the voxel-based mapping approach described above with an alternative geometric mapping approach, optimized for low network bandwidth consumption [6], and further with a condition without any global 3D mapping. Participants accomplished various object search and obstacle navigation tasks with the robot in a home-like environment. Global 3D environment mapping showed to have substantial temporal advantages when users were searching for objects in the apartment and it lead to fewer collisions when navigating the robot around elevated obstacles. During one navigation task where all obstacles were located on the floor, 3D mapping did not show temporal advantages – presumably because all relevant environment information was already contained in the 2D laser range data. User performance with the voxel-based technique tended to be better than with the simplified geometric visualization, presumably due to higher visual detail and realism [86].

5.5.4 *Combining 3D Environment Visualizations*

An important question is how to combine the "historical" data stored in the 3D map of the environment with the live RGB-D data. It is obviously important to show the remote operator the latest data and to not obstruct the view with any artifacts stored in the 3D environment map - e.g. the previous yet outdated recordings, noise, and speckles. Moreover, the resolution of the 3D map is lower than the resolution of the live data especially for close objects.

Our approach uses the information about the current position and orientation of the robot's torso to cut out the part of the 3D map inside the current field of view and show the live RGB-D data there. We limit the maximum distance from the camera at which the points are filtered because the effective range of the sensor is limited too. To communicate the difference between live and historical data to the user, the current field of view of the sensor is visualized using two thin yellow lines, which do not obstruct the view (see Figure 5.5).

5.5.5 *Stereoscopic Display*

Stereoscopic display can improve user performance [32] and user experience [20]. It has the potential to simplify tasks that depend on the operator's depth judgments, for example reaching and grasping of objects, robot navigation in the room including obstacle avoidance, judging the robot's arm position, or the relative positions and distances of objects in the scene. Without stereo visualization the operator may be less accurate and may need to adjust the viewpoint more often to see the scene from different perspectives.

There are several commercial solutions for stereo display in computer graphics. To achieve the stereoscopic effect, we use the Nvidia 3D Vision 2 stereoscopic kit[6]. This kit consists of LC shutter glasses and driver software. The glasses use a wireless IR protocol to communicate with the emitter providing a timing signal. The stereo driver software performs the stereoscopic conversion by using 3D models transmitted by the application and rendering two separate views from two slightly different points. A fast stereo LCD monitor (120 Hz) shows these two images alternately and the shutter glasses controlled by the emitter present the image intended for the left eye while blocking the right eye's view and vice versa. The scene in RViz

---

[6] http://www.nvidia.com/object/3d-vision-main.html (accessed 12/07/2015)

is generated using the Ogre library[7], which, however, is not ready for the stereoscopic display on Linux in the version included in ROS Electric (1.7.3). Thus it was necessary to modify the Ogre library as well as RViz itself.

To assess the usefulness of stereoscopic display for this user interface, we carried out an experiment [88]. 28 participants accomplished remote manipulation and robot navigation tasks – half of the participants under stereoscopic and the other half under monoscopic display. For the task of specifying the gripper's target position for grasping an object in the remote environment (see Section 5.7.3 and Figure 5.10c and 5.10d), there was a clear temporal advantage of using stereoscopic display. Participants also reached the goals faster under stereo display for the two other types of task, i.e. defining the shape of an object to be grasped (see Section 5.7.2 and Figure 5.10a and 5.10b) and navigating the robot around obstacles (see Section 5.6.4). However, the differences were not as pronounced here and not statistically significant after multiplicity correction. We thus concluded that stereoscopic display seems to be a useful additional display mode for this kind of user interface but that its utility may vary depending on the task [88].

5.6 ASSISTED NAVIGATION

Safe and reasonably fast movement of an assistive robot can be considered an essential functionality. Contemporary robot navigation systems are quite mature and able to assure 2D navigation even in complex and dynamic environments. However, because of safety concerns, these systems are usually tuned to be conservative, to use wide safety margins, etc. This leads to improved safety but it limits the robot's abilities on the other hand. In our semi-autonomous solution, a remote operator can be contacted if there is a problem with navigation, for instance if the robot cannot move to a desired location.

To solve navigation issues, the operator may use tools with different levels of autonomy depending on the current situation and personal preferences:

- Autonomous waypoint navigation
- In-scene teleoperation
- 3D mouse teleoperation (with the option to switch off collision avoidance)

Ecological approaches for teleoperation have typically used a non-interactive 3D scene with rather simplistic visualization of an environment and a joystick to control robot movement [95, 91]. Our approach is similar to previous ones in terms of visualization using a common reference frame and the ability to freely

---

[7] http://www.ogre3d.org (accessed 12/07/2015)

adjust the viewpoint. Beyond this, it provides rich visual information and enables the user to choose an appropriate tool for teleoperating the robot suitable for the particular situation. The 3D scene in our approach is interactive so two of the available navigation tools are integrated into it.

5.6.1 *Scenarios*

Under normal circumstances, the robot navigates autonomously using path planning based on the ROS Navigation Stack[8]. While the autonomous navigation is capable of coping with most situations it fails in some cases. A typical example is a very narrow passage where the robot physically fits but, because of safety settings, is not able to pass autonomously. Autonomous navigation also cannot reach its goal if there is an obstacle blocking the path. In semi-autonomous mode, the obstacle can be removed using the manipulator or pushed away with the robot's base.

5.6.2 *Autonomous Waypoint Navigation*

The teleoperation tool with most autonomy enables the operator to send intermediate waypoints to the robot's navigation system. This can be useful for moving the robot over a longer distance or when an optimal trajectory, which would normally be chosen by the navigation system, is for some reason not feasible, e.g. when there is a risk of collision. The operator sets waypoints by clicking at a desired position and also specifies the robot's target orientation by rotating the arrow before releasing the left mouse button. After that the trajectory is planned and the plan is visualized to the operator so he or she can easily predict the robot's movement.

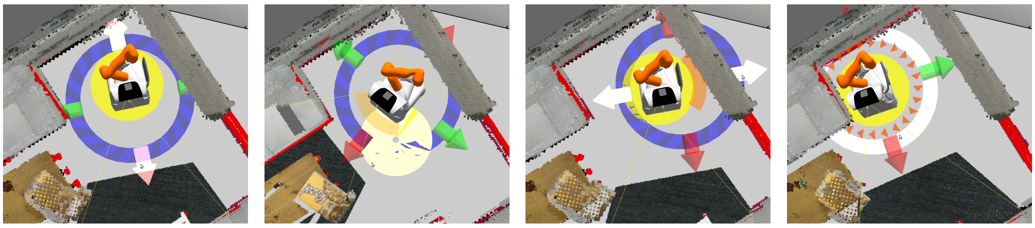
5.6.3 *In-Scene Teleoperation*

In order to provide an intuitive way to drive the robot directly within the 3D scene, we have designed a special in-scene teleoperation control that is based on ROS Interactive Markers[9]. The robot can be teleoperated for translational movement in two axes using the red and green arrows, and for rotation on the spot using the blue circle (Figure 5.8). This type of control is suitable for small and precise movements in a tight space. A more comfortable and faster way of teleoperation is realized by a yellow disk in the middle - when grabbed, the robot follows it.

---

[8] http://wiki.ros.org/navigation (accessed 12/07/2015)

[9] http://wiki.ros.org/interactive_markers (accessed 12/07/2015)



(a) translation in free space (b) following the yellow disc (c) translation towards the obstacle (d) rotation next to the obstacle

Figure 5.8: Driving the robot using the inscene teleop. Driving forward and backward is achieved using red arrows and sidewards using green arrows (a). Rotation is performed using the blue circle. The robot can be driven to a specified position by moving the yellow disc ((b)). Velocity limited marker shown when the robot cannot move in a particular direction ((c)) and rotate in place ((d)).

This type of control is more suitable for traversing larger distances in free space. However, while it allows control of more degrees of freedom at the same time, it does not provide precise control for navigation in tight environments. The in-scene control, especially the disc-following concept, was designed as an easy tool to manually drive the robot. When using the disc-following concept, the robot motion is derived from the current disc position (p_x, p_y) relative to the robot base:

$$L_M(x) = \mathrm{sign}(x) * \min(M, |x|), \tag{5.3}$$

$$v_{fwd} = L_M(C_x * p_x), \tag{5.4}$$

$$v_{rot} = \mathrm{sign}(p_x) * L_M(C_y * p_y). \tag{5.5}$$

Function $L_M(x)$ limits the maximum robot speed, C_x and C_y are constant scaling factors, v_{fwd} is the forward motion velocity and v_{rot} is the robot rotation velocity. Until the user grabs and moves the disc the position (p_x, p_y) is zero. These equations result in a smooth motion of the robot when the robot simultaneously turns to face the disc and moves towards the disc.

In many real-world situations, the robot's collision avoidance system based on two 2D laser scanners prevents moving or rotating the platform in some directions because the platform or the arm is very close to either moving or static obstacles. When the robot is close to an obstacle, it automatically reduces its velocity until zero in this particular direction to avoid a collision. In these situations it may be frustrating if the remote operator cannot easily decide in which directions movement is allowed and in which direction the robot cannot be moved. Therefore, we designed a velocity limited indicator to help the remote operator decide in which directions he or she can manually drive the robot. Indicators are shown around the robot in the 3D scene to illustrate in which directions the velocity of the robot is limited (Figure 5.8c) or if the rotational velocity is limited (Figure 5.8d). This helps the remote operator to quickly decide what is the problematic obstacle and how to drive the robot around it.

5.6.4 *3D Mouse Teleoperation*

As an alternative to the in-scene robot control that uses a conventional 2D mouse we have developed a 3D mouse control. It is up to the user's preferences and the problem at hand which way of control will be used. When using a 3D mouse, the indicators for velocity limitations due to imminent collision are available too. Compared to in-scene control using arrows and the blue ring, the 3D mouse allows the user to perform translational and rotational movements simultaneously.

5.7 ASSISTED MANIPULATION

When problems occur, fully autonomous manipulation can be substituted by a semi-autonomous solution, which has been developed as a part of the user interface. Assisted manipulation can be used in cases where automated planning of the arm trajectory fails or is not applicable. It offers a complete pipeline for manipulation tasks consisting of object detection, arm trajectory planning, and grasping.

The approach uses a collision-aware trajectory planner and offline execution. It allows the user to set a desired target position and orientation of the end effector by adjusting its virtual representation in the 3D scene. The scene includes visualization of the whole arm with proper joint positions computed by inverse kinematics. The user may visualize the trajectory animation and eventually let the robot execute it. In case of an emergency, the user can stop its execution. Due to the absence of low-level telemanipulation, latency-related problems are eliminated and thus our approach is also highly usable through unreliable wireless networks and through the Internet.

Previous approaches for remote manipulation were restricted to stationary manipulators [7], used only a video stream for user interaction [25] or used one or more joysticks for robot control [7, 25, 132]. More advanced semi-autonomous approaches often use humans' cognitive skills for selecting objects in cluttered scenes [104] or choosing appropriate grasp points on already detected objects [133] but they do not give users full manual control for cases when a particular automated procedure fails. Our approach allows the user to carry out all steps for object manipulation manually, if necessary. Decoupled motion planning and execution makes the interface highly suitable for remote operation when compared to direct telemanipulation [25, 132]. Moreover, usage of a global 3D map updated in real-time provides the user better spatial and situational awareness when compared to interfaces using single 3D snapshots [7, 24, 77].

5.7.1 *Scenarios*

The SRS autonomous system [107] offers object recognition and grasping, however its functionality is not available under certain circumstances. First, the object to be grasped must be learnt in advance. This is unproblematic for most of the objects of daily use, however there might be a need to handle an unknown object. Further, detection of a known object may fail because of occlusion in a cluttered scene, low illumination levels, or due to inappropriate robot position. Finally, even in case of a known and detected object, it might be impossible for the autonomous system to reach any of the precomputed grasping positions for various reasons. In all of these cases, a remote operator is called for providing assistance.

When there is a request for remote intervention, for instance when an unknown object shall be fetched, appropriate tools in the user interface are enabled and an operator is instructed with text messages to perform the following steps:

1. Drive the robot to a proper position (the robot is then prepared automatically for the task - the torso is tilted forward, the camera is flipped to the right direction, the arm prepared in the appropriate position, and the tray lifted up)
2. Correct 3D map (i.e. remove noise) if necessary
3. Manually segment the object from the 3D scene
4. Navigate the arm to the proper grasp position
5. Select an appropriate grasp strategy (see Section 5.7.4) and execute it
6. Navigate the arm to place the object above the tray (the gripper opens automatically)
7. Check if the object is on the tray and navigate the arm to a safe position
8. Hand back control to the autonomous system

From this sequence, some steps can be repeated and at some points it is also possible to give the autonomous system the next try after the operator fixed the problem as shown in Figure 5.9.

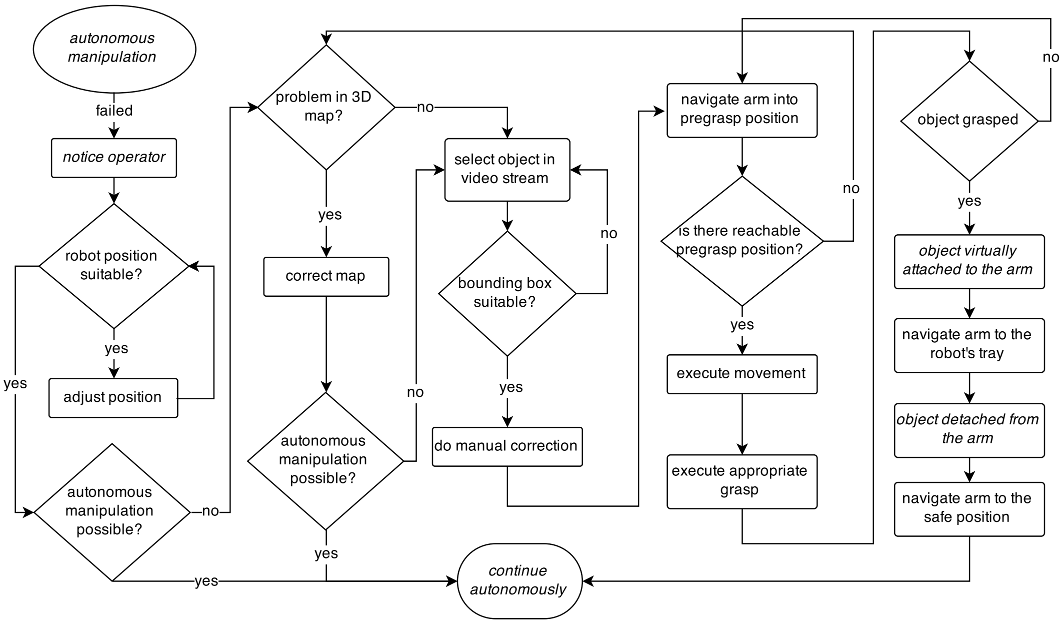


Figure 5.9: Manipulation workflow diagram. Each motion planning/execution step can be repeated or divided into more subsequent steps.

5.7.2 *Object Segmentation*

In order to use semi-autonomous manipulation for unknown or unrecognized objects, the dimensions of the object to be grasped need to be defined first. We implemented a tool which accelerates this process. When there is a need for specifying an object shape, an operator is asked to draw a box over the object in the video stream (Figure 5.10a). Based on this region of interest, we fit a bounding box to the corresponding 3D points. The estimated bounding box is shown in the 3D scene (Figure 5.10b), and the user can then adjust its pose and size according to either live 3D data or the 3D voxel-based model as there might be a lot of noise or occlusion in the original sensor data. This bounding box is then considered when planning the collision-free arm trajectory.

5.7.3 *Interactive Arm Navigation*

The visualization for arm navigation in the user interface consists of a 3D scene containing a robot representation with manipulator, a 3D collision map, the bounding box of a detected or user-specified object, and of an RViz plugin providing several functions via buttons, which are hidden by default. When a remote operation session is initiated, the operator is notified by a pop-up window and the appropriate controls become active.

For arm navigation (Figure 5.10), the operator is required to set a goal position for the end effector in the 3D scene. It can be done using interactive 3D widgets or more intuitively by a 3D mouse. While adjusting the virtual end effector position, the real manipulator does not move. Through color coding of the arm as well as a text overlay in the 3D scene, the interface indicates if the desired position is reachable by the arm and whether there are collisions with the environment model or objects. A collision-free trajectory from the start position to the goal position is planned on the user's request. If the planner cannot find a trajectory, the user may try planning with a different goal position or even with a revised robot position. Before executing the planned trajectory, the operator can run its visualization (Figure 5.10d) several times and decide if it is safe. The operator may decide to plan several trajectories for one task. When finished, the operator marks the task as completed and hands back control to the robot.

The solution for trajectory planning is based on functionality provided by the arm_navigation stack. It contains components for generating a robot-specific configuration, maintaining representation of the environment and recognized objects for collision checking, trajectory planning and filtering, inverse kinematics computation, visualization tools, etc. Our main contributions lie in making the user interface adequate for non-expert users, in providing the ability to use a 3D mouse as an input device, and in an API for integration with the autonomous system.

5.7.4 *User-Assisted Reactive Grasping*

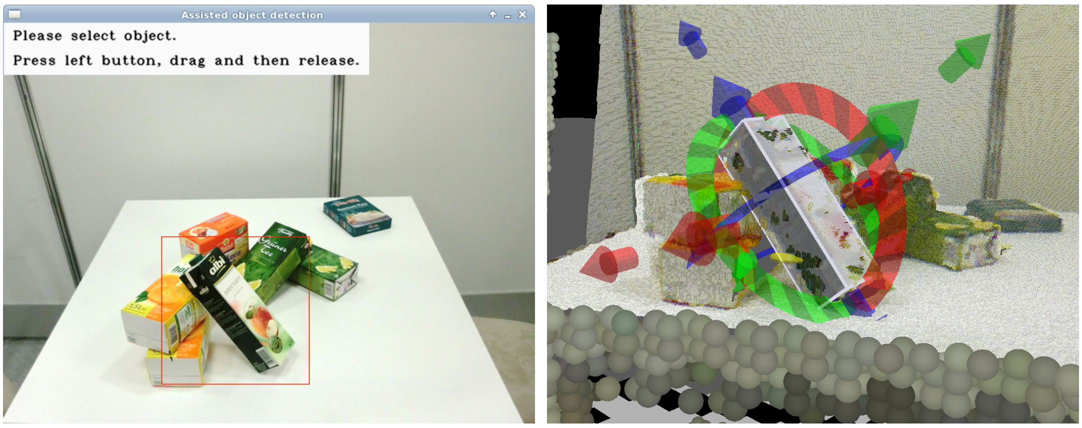
Our approach for grasping was designed to work for objects unknown to the robot, meaning that there is no known model of an object. This precludes grasping approaches based on prior shape knowledge [23].

We have developed software for the SDH[10] gripper equipped with tactile sensors, which allow easy to use, safe, and robust remote grasping. There is a predefined list of empirically determined target joint configurations with associated maximum forces for each tactile pad. The user selects an appropriate preset according to the object (e.g. “full beverage carton”). Then velocities for the joints are calculated so all joints will reach the target configuration at the same time including acceleration and deceleration ramps of configurable lengths. Any joint is stopped during the process of grasping if the maximum force from its tactile array exceeds a value defined in the chosen preset.

During informal experiments using this approach, we have been able to grasp various objects of daily use and of different shapes. However, results to a certain extent depend on previous steps and experience of the operator.

---

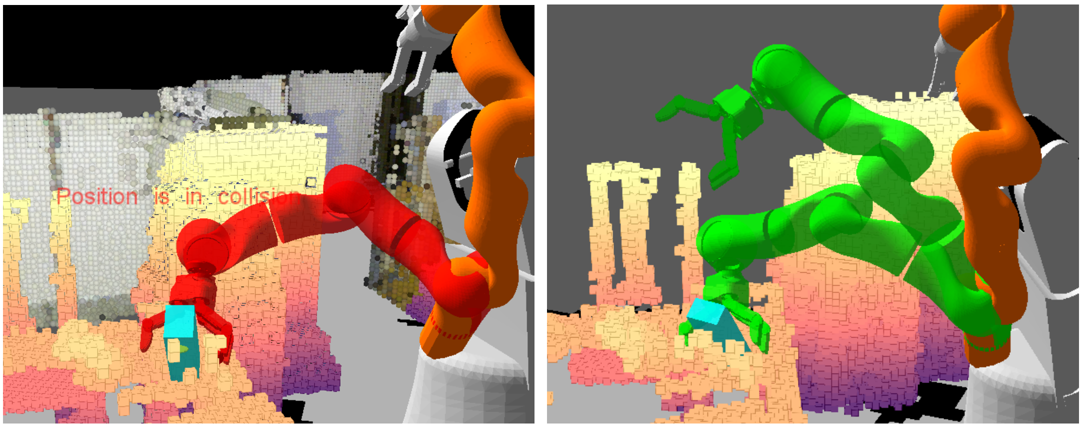
[10] Schunk Dexterous Hand



Please select object.

Press left button, drag and then release.

(a) selecting an object in the video stream (b) adjusting bounding box in the 3D scene



Position is in collision

(c) goal position not reacheable due to collision (d) visualization of planned trajectory

Figure 5.10: Assisted arm navigation used to perform a pick-and-place task.

5.8 CONCLUSIONS AND FUTURE WORK

The interface presented in this article enables intervention of a remote operator who may navigate the robot and perform manipulation of objects which cannot be handled autonomously. The interface's central features are a 3D scene display, global 3D mapping with interactive features, tools for teleoperation and telemanipulation, stereoscopic display, and control relying on a 3D mouse. The solution is built on already available and widely used components from ROS and newly designed and developed ones, such as an intuitive user interface for manipulation and a component for the efficient transport of 3D maps. Usage of the 3D interface with fused visualization of all relevant data requires only short training, shown by the fact that novice users in our experiments were all able to complete all tasks we asked them to solve. We believe that the concept of a semi-autonomous robot is promising as even remote manipulation tasks can be accomplished within reasonable time and with reasonable effort.

In order to improve user interaction, we are experimenting with head tracking to introduce motion parallax, which might be useful especially for manipulation. Another option we investigate is to allow a user to change the viewpoint with a 3D mouse. Regarding global 3D mapping, we envision a solution that avoids the influence of imprecise robot localization on a created map. For limited-bandwidth connections, user experience could be improved by using techniques like adaptive frame rates for images and point clouds.

5.9 ACKNOWLEDGMENTS

This research was supported by the European Commission, FP7 project SRS, Grant Agreement No. 247772; by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070); and by the Technology Agency of the Czech Republic project V3C, TE01020415.

6

SIMPLIFIED INDUSTRIAL ROBOT PROGRAMMING: EFFECTS OF ERRORS ON MULTIMODAL INTERACTION IN WOZ EXPERIMENT

6.1 abstract

This paper presents results of an exploratory study comparing various modalities employed in an industrial-like robot-human shared workplace. Experiments involved 39 participants who used a touch table, a touch display, hand gestures, a 6D pointing device, and a robot arm to show the robot how to assemble a simple product. To rule out a potential dependence of results on the number of misrecognized actions (resulting, e.g., from unreliable gesture recognition), a controlled amount of interaction errors was introduced. A Wizard-of-Oz setting with three user groups differing in the amount of simulated recognition errors helped us to show that hand gestures and 6D pointing are the fastest modalities that are also generally preferred by users for setting parameters of certain robot operations.

6.2 introduction

Industrial robots were traditionally used mainly in a large-scale production. This was primarily due to the large price of the automation and low flexibility requiring long and costly adaptation for new products. Recently, EU-supported projects as SMErobotics[1] and EuRoC[2] emerged to support development of easily reconfigurable cognitive robots able to achieve flexibility required for small to medium scale manufacturing. Such flexibility must be supported by easy to use and effective human-robot interaction substituting traditional ways of programming industrial robots requiring expert-level knowledge.

Our long-term goal is to create a shared-space environment similar to the experimental setup shown in Figure 6.1 where a human operator can cooperate with a semi-autonomous cognitive robot using multi-modal interaction and augmented reality: ARTable. The robot within the envisioned solution could be programmed once and then perform independently or it may continuously provide assistance to the operator. There was a research on what modalities are appropriate for what

---

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

[2] http://www.euroc-project.eu/index.php?id=challenge_1

Figure 6.1: Prototype of the human-robot shared-space environment with augmented reality user interface (image edited).

most common operations [106] in such a system. As a first step towards ARTable we were interested in how various modalities would perform in a similar experiment however under realistic conditions. Therefore we designed a WoZ experiment where input modalities were not always working perfectly and participants had to face interaction errors. The aim of the experiment was to uncover whether there is dependence between preference for using particular modality for setting particular parameter and amount of experienced interaction errors. Secondarily, we were interested in how task completion times will be influenced by used modality and amount of errors as a time-effective human-robot interaction will be of paramount importance for a practical usage of such system. Video summary of the experiment can be seen at https://youtu.be/LtiDc3pGjug.

6.3 RELATED WORK

Robot manipulators used to be programmed by experts at a low level making them less flexible to production changes. Recently, approaches allowing high-level programming by end users appeared. One of these approaches is programming by demonstration [12] also referred to as kinesthetic teaching [114], where an operator programs a robot by positioning its end-effector while learning poses [4] and/or forces [1]. Existing solutions can be divided into those allowing so called offline programming where a robot is programmed once [101, 76], those allowing a continuous human-robot collaboration [96] and those allowing both [47] modes. The interface may be for instance projected [43] or integrated into a hand-held device with augmented reality [101, 76]. Interaction also may happen in a virtual reality [47]. Alternatively to positioning a robot's end-effector, a human operator may demonstrate the task by actually performing it [71] or by giving high-level instructions using one [96] or more modalities [101].

Errors in interaction can be according to [55] divided into following types: misunderstandings, non-understandings and misconceptions. For our experiment, we choose to simulate misunderstandings with third-turn repair of the errors. Dealing with errors is often limited to resolving problems during program execution [9]. The experiment with social robot programming [18] where gesture and speech-based interfaces and even the robot's software were not perfectly reliable has shown importance of the provided feedback. However, those errors were not simulated and thus their amount was not controllable. The framework to support WoZ studies from [68] allows to insert given amount of random misrecognition errors, however it is limited to the speech-based interfaces.

Misunderstandings may be caused by a non-perfect input. For instance the pointed object estimation from [98] is reported to have 83% success rate despite usage of a prior information about location of the objects. Another approach to detection of pointing directions [119] achieved $\pm 10°$ angular and 93% distance error. The speech recognition system from [49] achieved 16% error in a noisy environment with background TV or radio. It can be speculated that amount of errors would be higher in an industrial environment.

6.4 USER STUDY DESIGN

The main goal of this study was to find out how errors affect user preference of input modality while programming a robot. We were interested in three industrial use cases: assembly, pick&place and welding of points and seams. These use cases were transformed into a simple product manufacturing scenario, better fitting our laboratory settings. A Wizard-of-Oz approach was utilized to avoid implementation specific errors. Without participant's knowledge, a man in a separated room (wizard) observed the scene through a set of cameras and simulated system responses and a feedback. Moreover, WoZ allowed us to simulate certain amount of errors in interaction.

The experimental setup consisted of a table with a top-mounted Kinect v2 sensor and a projector, a robotic platform (PR2) and a touch screen computer besides the table. All sensors were used only for surveillance purposes. During the experiment, the robot was immobile but it helped to create impression of a real robotic workspace.

A simple GUI was created to give users feedback through the projector mounted above the table. There was a bounding box around each object on the table and a label with its name. The selected object was highlighted and points and lines on the objects (selected by a user) were displayed in a different color. The user interface contained a back button used for stepping back, when the system made an error. The button was projected on the table as a red arrow for each modality except the touch screen (there was an on-screen one). Moreover, there was an area dedicated to projecting additional information, animations etc.

6.4.1 *Input Modalities*

Touch table (A) An object is selected by clicking on its projected description. Welding points and seams are selected on a projected image of the object. Assembly constraints are not set with this modality.

Touch screen (B) An object is selected by clicking on it on a screen. Welding points and seams are selected on a zoomed picture of the object. Assembly constraints are not set with this modality. Theoretically there should not be errors in determination of user intention (e.g. where user clicked), but in such a complex system, there could always raise an error, or a user can accidentally click on a wrong place.

Gesture (C) Objects and welding points are selected by pointing on them with the index finger. Welding seams are selected by hovering over a desired seam with the index finger. A gesture used to specify assembly constraint was up to the user. Hand gesture recognition and hand pointing direction recognition is widely studied problem [109, 127]. Recent research shows that 75 to 98% recognition rate is achievable [98, 119].

6D pointing device (D) Similar to C, but instead of the index finger a 6D pointing device was used. Although detection of pose and orientations of this device is more precise and robust than detection of a hand, there still may be errors caused by a user, who can point on a wrong object, or point imprecisely.

Direct robotic arm programming (E) Selecting of objects and welding points and seams was done by pointing on them with a robot's gripper. Just like the 6D pointing device, determining of pose and orientation of a robotic arm is very precise, due to reading arms actuators' internal state, but it can suffer from the same user errors.

Compared to [106], a direct robotic arm programming and a touch table were added. A speech was considered inappropriate as it is probably not sufficiently robust for noisy industrial environments. Our goal was to perform experiment under realistic conditions and we expected participants (mostly university students) to not believe speech programming without predefined vocabulary could work. Moreover, in [106] speech was the lowest rated modality.

Direct robot arm programming (kinesthetic teaching) is commonly used [114, 2], however we are using this technique in a different manner (e.g. selecting objects instead of teaching robot how to grasp them). Touch-sensitive table could be an advantageous alternative to a touchscreen in an industrial environment, as the feedback, system information and interaction with system is held in the user's working space and due to the fact, a user is not forced to divide attention between more places.

6.4.2 *Tasks*

Each participant was told to program the robot to make a simple assembly and packing in a scenario imitating the most common industrial tasks. The scenario was divided into four tasks, each consisting of ten steps (setting ten parameters) in total:

- Assembly: select two objects (e.g. plastic cover and aluminum profile) and set an assembly constraint(s) (e.g. cover orientation)
- Pick&place: select an object and select a place where to put it
- Welding point: select an object, select four points on its top side (to glue stickers in our scenario)
- Welding seam: select an object, select four edges on its top side (to seal boxes with tape)

Each task consisted of ten steps meaning that participant had to set ten parameters: i.e. five times select an object and place where to put it in the pick&place task or select and object and according of its type select one or two assembly constraints in assembly task (see Figure 6.2). According to participant's group, there were zero, one and three (i.e. 0, 10 and 30%) errors in each task. For instance, in 30% error-level group the system randomly misrecognized three parameters from ten during each of the four tasks. The errors were generated automatically by our WoZ application and were not influenced by the wizard. Order of tasks and steps was the same for all participants.

We see 0% error rate (used for experiment in [106]) as an ideal state however hardly achievable with most of the modalities. 10% seems to be a current realistic

(a) User selects plastic cap to be assembled with aluminum profile.

(b) User performs step back as a tape was selected instead of the profile.

(c) Now the intended object (profile) was selected.

(d) Animation shows how the robot understood user's assembly demonstration.

Figure 6.2: An example of a typical interaction for the assembly task using the robot arm as an input modality.

level. 30% was selected as the worst case scenario. We assume it to be the worst error ratio probably acceptable by users.

6.4.3 *Methodology*

The SUXES evaluation method for subjective evaluation of multimodal systems has been adopted [129]. It is based on collecting user's expectation and experience and provides means to analyze various interaction methods. The methodology divides experiment into following four phases:

6.4.3.1 *Background Information*

The experiment is briefly introduced to the subject by a conducter, who is with the subject during the whole experiment. Then, a background information about subject (i.e. age, technical knowledge etc.) is collected.

6.4.3.2 *User Expectation*

The conductor introduces the shared workspace, all input modalities and the feedback provided by the projector. The subject is allowed to ask questions and to try any modality. Then the subject fills in the questionnaire about his or her expectations based on the introduction.

6.4.3.3 *Experiment and User Experience*

The conducter guides the subject through four strictly defined tasks: the subject is told what is the current task and step and what to do when error occurs. The task itself is performed solely by the participant. Each subject performs those four tasks with all five modalities (with exception of assembly task, where modalities A and B are skipped). The order of modalities is random for each subject to prevent a learning effect. After that, the subject answers the same questions as in the previous step.

6.4.3.4 *Feedback*

The subject answers questions about the system using Likert scale rating (see Figures 6.3 and 6.4). Most of the subjects also filled valuable fulltext responses.

6.4.4 *Participants*

The experiment has been conducted with 39 participants assigned randomly into three groups. There were eleven males and two females in each group. Participants were mainly university students and researchers with mean age of 23.7 (CI: 22.5 to 24.9) years. Most of them (30) marked themselves as PC experts and at the same time beginners (23) or advanced (15) in robotics. Majority of participants knew what a touchless interface stands for but never used one (31), some indicated that they already used this kind of interface (7) and only one did not know something like this exists.

The whole experiment took approximately 45 minutes for each participant and the interaction itself was recorded by a video camera. Participants' answers have been collected into a spreadsheet.

6.5 RESULTS

Participants from all groups (0, 10 and 30% of interaction errors) ordered modalities according to their preference for setting a given parameter before (expectation) and after the experiment (experience). Mean of the order from expectation phase

modality	**group**	**select an object**				**select a place**				**select a point**				**select a line**				**assembly constraint**			
		r_B	r_A	p_{tp}	r_{Ao}	r_B	r_A	p_{tp}	r_{Ao}	r_B	r_A	p_{tp}	r_{Ao}	r_B	r_A	p_{tp}	r_{Ao}	r_B	r_A	p_{tp}	r_{Ao}
A	0%	3.7	3.3	-	3.3	4.3	3.2	0.015	3.5	3.0	2.9	-	3.2	3.2	3.1	-	3.3	2.8	NA	-	NA
	10%	4.4	3.4	0.012		4.6	3.9	-		3.5	3.1	-		3.8	3.3	-		2.9	NA	-	
	30%	4.1	3.3	0.0024		4.6	3.5	<0.001		2.9	3.5	-		3.0	3.4	-		2.4	NA	-	
B	0%	3.2	2.1	0.02	2.9	2.8	2.1	-	2.7	3.2	2.3	-	2.8	2.9	2.3	-	2.9	3.2	NA	-	NA
	10%	3.7	3.1	-		3.1	2.9	-		3.5	2.9	-		3.7	3.2	-		3.0	NA	-	
	30%	4.2	3.5	-		3.3	3.2	-		3.5	3.2	-		3.2	3.2	-		2.9	NA	-	
C	0%	4.2	4.2	-	3.8	3.7	3.7	-	3.6	2.7	3.7	0.021	3.6	3.0	3.8	-	3.7	3.9	4.0	-	4.1
	10%	2.9	3.6	-		2.9	3.5	-		2.4	3.9	0.0031		2.8	3.9	0.012		3.6	4.5	0.035	
	30%	2.7	3.6	0.046		2.7	3.6	0.027		2.9	3.3	-		3.4	3.4	-		3.8	3.9	-	
D	0%	2.3	3.5	<0.001	3.3	2.5	3.6	0.0045	3.4	3.7	4.0	-	3.7	3.6	3.7	-	3.5	2.3	3.5	0.011	3.4
	10%	2.1	3.4	0.0018		2.5	3.5	0.012		3.6	3.9	-		3.2	3.4	-		1.9	3.9	<0.001	
	30%	2.7	3.1	-		2.9	3.0	-		3.8	3.2	-		3.5	3.5	-		2.2	2.9	-	
E	0%	1.7	1.9	-	1.7	1.7	2.5	-	1.9	2.4	2.1	-	1.7	2.2	2.2	-	1.6	2.8	2.6	-	2.6
	10%	2.0	1.5	-		2.0	1.4	-		2.0	1.3	-		1.6	1.2	-		3.6	2.5	0.021	
	30%	1.4	1.5	-		1.5	1.7	-		1.9	1.9	-		1.9	1.5	-		3.7	2.9	-	

Table 6.1: Participants ordered modalities for each parameter separately from the most preferred (5) to the least (1) before (r_B) and after (r_A) the experiment. Where significant difference was found between r_B and r_A p-value is given. r_{Ao} stands for preference after the experiment regardless of the group (0, 10 or 30%).

is denoted as r_B and from experience phase as r_A. Statistically significant differences between r_B and r_A within one group were tested using paired t-test (p_{tp}). Differences for a particular modality across the groups were tested using Kruskal–Wallis test with Dunn's multiple comparisons test p_{Wd}. The same test was also used to compare task completion times. Confidence level of 95% was used for all tests. Experience from all participants (all groups) is denoted as r_{Ao}.

6.5.1 *Parameters*

From the Table 6.1 showing users' self-reported data it can be seen for which modality and which parameter there were significant differences between r_B and r_A. Moreover, it can be seen which modality was the most preferred for a given task regardless the amount of errors (r_{Ao}). It should be noted that r_B of C differs between 0% and 30% groups ($p_{Wd} = 0.028$).

Considering the number of significant differences between r_B and r_A from all groups, C and D were ranked significantly better six times, B and E were both worse once and A was worse four times. There are no significant differences in r_B between groups meaning that participants from different groups had similar expectations (with one exception of C in 0% group, parameter *select an object*). Moreover, there are also no differences in r_A. From these results it seems that number of errors in interaction does not have strong impact on preferred modality. In other

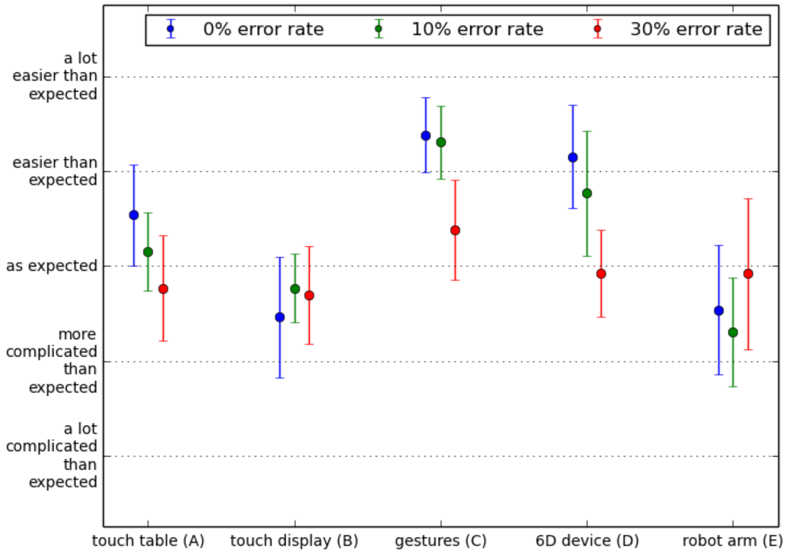


Figure 6.3: User's assessment how experience matched expectation.

words, participants from different groups had similar expectations (r_B) as well as experience (r_A). Overall, it seems that participants mostly preferred modalities C, D, followed by A, B and the least preferred was E. Figure 6.3 shows how participants evaluated expectation and experience for all modalities overall (regardless task).

6.5.2 *Task Completion Times*

Before performing a task the participants were told all relevant information. During the task, only the next step was reminded by the conducter. When beginning the task a participant pressed the "Start" button and then the "Stop" one when finished. We use time between those presses as an objective measure. The Table 6.2 shows those times as well as found significant differences between groups for each modality. Differences between modalities are noted below.

The *assembly* task (consisting of *select an object* and *assembly constraint* parameters) was performed only using C, D and E modalities. In all groups there are significant differences between C and E (0%: $p_{Wd} = 0.003$, 10%: $p_{Wd} < 0.001$, 30%: $p_{Wd} < 0.001$) and between D and E (0%: $p_{Wd} = 0.034$, 10%: $p_{Wd} < 0.001$, 30%: $p_{Wd} = 0.002$).

The *pick&place* task consisted of setting *select an object* and *select a place* parameters. In all groups there are significant differences between E and each of rest of the modalities (with max. $p_{Wd} = 0.049$).

The *welding point* task consisted of setting *select an object* and *select a point* parameters. In 0% group, time for B differs from C ($p_{Wd} = 0.0091$) and D ($p_{Wd} = 0.023$). E differs from C and D ($p_{Wd} < 0.001$). In 10% group, time for A, C and D dif-

mod.	**group**	**assembly**		**pick&place**		**welding point**		**welding seam**	
		mean time [s]	significant differences	mean time [s]	significant differences	mean time [s]	significant differences	mean time [s]	significant differences
A	0%	NA	-	34.7 (27.9, 41.5)	0/30: 0.0017 10/30: 0.038	36.8 (31.2, 42.4)	0/30: 0.003 10/30: 0.0022	33.6 (26.1, 41.0)	0/30: <0.001 10/30: 0.0056
	10%	NA		37.4 (33.5, 41.3)		35.2 (31.6, 38.9)		37.0 (33.4, 40.5)	
	30%	NA		47.5 (42.3, 52.6)		49.1 (43.8, 54.4)		53.13 (47.6, 58.6)	
B	0%	NA	-	32.8 (28.7, 36.8)	0/30: <0.001 10/30: 0.04	38.4 (34.6, 42.1)	0/30: <0.001 10/30: 0.0047	36.9 (31.7, 42.1)	0/30: <0.001
	10%	NA		41.2 (36.9, 45.4)		41.6 (37.3, 45.9)		44.2 (41.2, 47.2)	
	30%	NA		52.3 (47.1, 57.4)		54.5 (50.1, 58.9)		53.6 (48.0, 59.3)	
C	0%	54.8 (46.2, 63.5)	0/30: 0.03	28.0 (25.7, 30.3)	0/30: <0.001	28.3 (24.8, 31.8)	0/30: <0.001 10/30: 0.033	27.6 (24.6, 30.6)	0/30: <0.001 10/30: 0.019
	10%	60.4 (47.9, 73.0)		33.7 (29.4, 38.0)		31.9 (27.6, 36.2)		34.8 (30.9, 38.7)	
	30%	70.5 (61.8, 79.2)		40.9 (37.0, 44.8)		41.2 (36.3, 46.1)		47.4 (41.1, 53.7)	
D	0%	61.5 (45.4, 77.6)	0/30: 0.03 10/30: 0.044	28.8 (25.1, 32.5)	0/30: <0.001 10/30: 0.0037	29.3 (25.1, 33.4)	0/30: <0.001 10/30: 0.002	31.0 (26.6, 35.4)	0/30: <0.001 10/30: 0.0023
	10%	61.0 (50.8, 71.2)		32.3 (29.8, 34.9)		31.9 (28.6, 35.3)		36.7 (32.2, 41.1)	
	30%	88.0 (69.3, 106.6)		43.9 (39.1, 48.6)		44.5 (40.6, 48.3)		52.8 (47.9, 57.7)	
E	0%	90.2 (71.2, 109.2)	0/10: 0.013 0/30: <0.001	43.7 (40.5, 46.9)	0/10: 0.0059 0/30: <0.001	42.9 (38.1, 47.7)	0/10: 0.014 0/30: <0.001	42.6 (35.7, 49.4)	0/30: <0.001 10/30: 0.044
	10%	129.6 (112.2, 146.9)		60.6 (54.9, 66.2)		58.9 (54.1, 63.6)		58.4 (52.7, 64.1)	
	30%	156.7 (127.6, 185.8)		75.3 (69.2, 81.4)		83.0 (68.2, 97.8)		85.1 (68.9, 101.3)	

Table 6.2: Task completion mean times (with 95 % confidence intervals) for all modalities, groups and tasks. For each modality, significant differences between times are noted where found in form of $\text{group}_x/\text{group}_y : p_{Wd}$.

fers from E ($p_{Wd} < 0.001$). The 30% group shows differences between E and A ($p_{Wd} = 0.0018$) and C, D ($p_{Wd} < 0.001$).

The *welding seam* task consisted of setting *select an object* and *select a line* parameters. In 0% group, there is significant difference only between C and E ($p_{Wd} = 0.0029$). 10% group shows difference between E and A, C, D ($p_{Wd} < 0.001$) and 30% group between E and A ($p_{Wd} = 0.0105$), B ($p_{Wd} = 0.014$), C, D ($p_{Wd} < 0.001$).

For most of the tasks C and D were the fastest modalities followed by A and B. E seems to be unsuitable to the sort of tasks as those in this experiment as even 10% of errors affects performance in three of four tasks. It seems that for other modalities a little amount of errors does not play crucial role.

6.5.3 *System Opinion*

The last phase of the SUXES evaluation contains opinion questions. We used the same questions as in [106], with addition of those related to the erroneous behavior (see Figure 6.4).

Regardless of the group, participants were satisfied with ease of completing the tasks and with time needed to do so. Participants also claimed it was not difficult to understand how to use different modalities. The results are highly similar to those of [106].

Most of the subjects rated modalities C and D similar, however had a stronger believe in 6D pointing device as they expect it to be more precise than gesture, de-

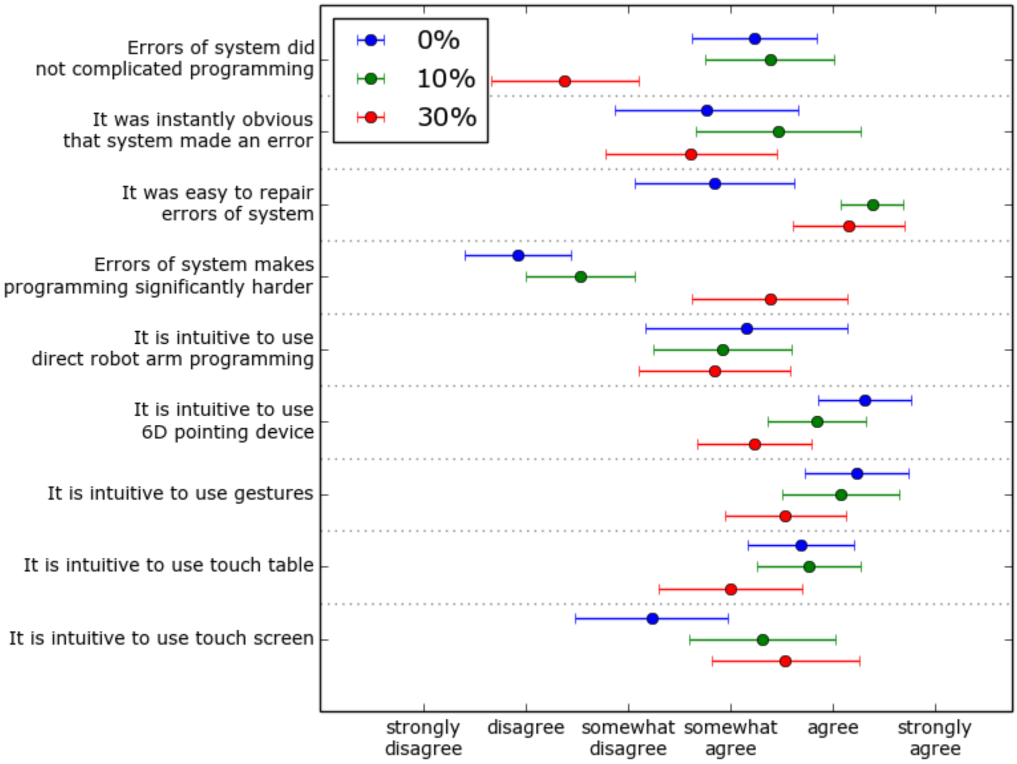


Figure 6.4: System opinion

spite there was the same amount of errors. Participants were also often distracted by the fact, that feedback was always projected on the real objects on the table and not on the place they were working with. Especially, for B most of them would prefer feedback (e.g. selected object) to be shown on the screen and not only on the table. This was however done by purpose, to ensure each modality has exactly the same feedback and participants were noticed about this in advance.

In questions related to erroneous behavior a difference can be seen between error groups. With a growing amount of the errors, perceived intuitiveness of the modalities decreases, except for the touch screen, where it grows (see Figure 6.4). This could be caused by the fact, that the touch screen is the only control commonly used by the participants. Moreover, the back button was on the screen, so the participants were not forced to think about how to press projected button as for other modalities. Modalities B and E were in general evaluated as the least intuitive. Participants stated that with growing amount of errors, programming was significantly harder and that errors in communication complicated programming.

A few of the participants found out that errors were made by purpose or that some parts of system were simulated. However, according to feedback and discussion with participants, none of them found out the experiment was WoZ.

6.6 CONCLUSIONS

The aim of the conducted experiment was to explore how different modalities used for setting common parameters when programming a robot cope with interaction errors. Participants were divided into three groups according to amount of simulated errors. Their ranking of the modalities before and after the experiment as well as answers from feedback phase were analyzed as subjective measures. Moreover, task completion times were recorded and analyzed as an objective measure.

The gesture and 6D pointing device modalities were the most preferred and fastest modalities in all groups. Touch-sensitive table and display were in general preferred similarly and similar task completion times were obtained. With respect to the task completion times as well as feedback from participants (system opinion) the robot arm seems to be inappropriate as a pointing device for tasks as those in this study and its usage should be reconsidered. It seems that order of preferred input modalities for a given task is not affected by amount of interaction errors. Obtained results support our prior speculation of 10% to be an acceptable level of errors and 30% to be a worst case scenario as especially task completion times grow dramatically.

According to the results, multi-modal interaction based on gestures with complementary usage of a 6D pointing device seems to be promising. We also see touch-sensitive table as a perspective modality however it will be necessary to improve interaction and solve setting more complicated parameters as the *assembly constraint*. The robot arm has advantage of no additional cost however, its usage is physically more demanding than other modalities and for our use-case with relatively simple tasks it had no added value. However, for different types of tasks, e.g. requiring high precision, it could be more useful.

It should be noted that our study simulated the same amount of errors for all modalities. In practice, it can be expected that for instance robot arm modality will be less error-prone than gesture recognition.

As a future work, we will extend the ARTable prototype. The projected interface will provide more information and be fully interactive in conjunction with a touch-sensitive table. Instead of a touch display, a hand-held device or a see-through video glasses with augmented reality will be used. We will also experiment further with robot arm as it could be useful for complex tasks.

ACKNOWLEDGMENTS

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602.

7

USING PERSONA, SCENARIO, AND USE CASE TO DEVELOP A HUMAN-ROBOT AUGMENTED REALITY COLLABORATIVE WORKSPACE

Up to date, methods from Human-Computer Interaction (HCI) have not been widely adopted in the development of Human-Robot Interaction systems (HRI). In this paper, we describe a system prototype and a use case. The prototype is an augmented reality-based collaborative workspace. The envisioned solution is focused on small and medium enterprises (SMEs) where it should enable ordinary-skilled workers to program a robot on a high level of abstraction and perform collaborative tasks effectively and safely. The use case consists of a scenario and a persona, two methods from the field of HCI. We outline how we are going to use these methods in the near future to refine the task of the collaborating robot and human and the interface elements of the collaborative workspace.

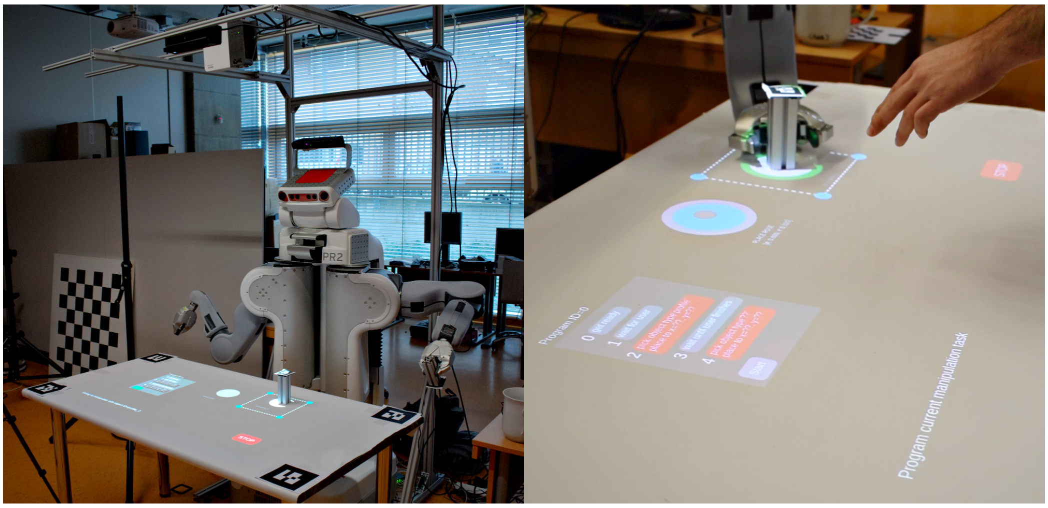
7.1 INTRODUCTION

With the emergence of affordable industrial collaborative robots it seems likely that SMEs soon will widely adopt such robots in order to achieve higher precision for specific tasks, free experienced employees from monotonous tasks, and increase productivity.

In a large-scale production, robots are usually programmed by an expert. For SMEs, batches are smaller and products may even be customized for a particular contract. Due to this, it would be beneficial to enable ordinary-skilled workers to program robots easily, without robot-specific knowledge. In this work, we present a new approach for simple robot reprogramming. The approach uses augmented reality (AR) to visualize the current program and the state of the robot's learning or execution, detected objects, instructions to a user etc. We describe an existing prototype[1], a use case of aircraft trolleys assembly and how we will apply HCI methods, in particular narrative scenarios and personas, in further development.

---

[1] The source code and technical documentation is available at `https://github.com/robofit/ar-table-itable`.



(a) (b)

Figure 7.1: Experimental setup (a): PR2 robot, top-mounted Kinect 2 and projector, table with AR markers. User interaction (b): adjusting place pose for the grasped object.

7.2 BACKGROUND

There exist various approaches to the problem of making robot programming viable for non-expert users, e.g., kinesthetic teaching [114] or visual programming [4]. Part of this problem is also the selection of suitable input modalities [89] and modalities for providing feedback to the user. One of the output modalities may be AR based on a hand-held device [122] or projected onto the workplace [43].

Scenarios are narrative stories about specific people and their activities in a specific work situation and context [52]. They describe key usage situations and they cover a multitude of aspects such as involved agents, user goals and background, work practices, system responses, tasks, context, and difficulties. Cooper et al. [27] developed the concept of personas to represent the hypothetical archetypes of users. Personas are not actual users but they represent specific users with their characteristics and work role [52]. They are given a name, a life, and a personality to make them concrete and appear *real*. Personas are an ideal instrument to design for the most relevant and common user classes.

Up to now, there are only few instances, where HCI methods were used in the field of HRI. For instance [14] uses scenarios and personas in the context of industrial robot programming.

7.3 AUGMENTED REALITY COLLABORATIVE WORKSPACE

The open source experimental setup uses the intrinsically safe PR2 robot as a demonstrator of a near-future collaborative robot and is centered around a table (see Figure 7.1a) where the HRI occurs. The interaction consists of programming a robot and collaboration on a programmed task. It happens through an interface projected onto the table using pointing gestures as an input modality (see Figure 7.1b or video[2]). The user is tracked by a Kinect sensor on the robot's head. Skeleton tracking is used to extract information about the user's position and pointing direction. Gestural control was chosen based on results of our previous experiment [89], where it was the fastest and highest ranked modality. We deal with uncertainty of pointing by highlighting pointed area on the table (circle of given radius) which serves as a visual feedback to the user. When this area visually collides with e.g. a highlight area of an object, the object is preselected. If the object is preselected for a certain time, it is selected. Objects in the scene are tracked using a top-mounted camera and AR codes on them. AR codes are also used for calibration of the whole system.

The interface contains various elements to visualize state of the robot and task as e.g. the currently loaded program. A robot's program is displayed to the user during both learning and task execution phases. Currently, the system supports basic instructions as *get ready* (move robot arms to a default pose) or *pick and place* (pick concrete object or object of given type from specified polygon and place it on given pose). The program structure is so far coded separately while program parameters (e.g. object type and place pose for *pick and place* instruction) are set by the user - the interface allows the user to select a program, set or adjust its parameters and then to collaborate on a programmed task with the robot. During program execution, the current program item is highlighted as well as e.g. objects to be manipulated by the robot.

7.4 USER-CENTERED DESIGN: USE CASE, SCENARIO AND PERSONAS

Based on our experiences from previous projects and discussions with industrial partners, we have defined our scenario as follows: *The user will teach the collaborative robot to assist him in the task of assembling aircraft service trolleys. He needs to show to the robot which parts are needed in every step of assembling, where holes must be drilled, and what parts should be glued together.*

We also defined a persona, who will act as a user in our use case: *Jan, a 22 year old man, recently graduated at technical-based high school. He works as an assembly worker at Clever Aero, a company focused on aircraft equipment. He has no experience with robots, but he loves new technologies and he is really keen into working with robots.*

These tools needs to be refined according to the demographic data, which has to be collected by observing and interviewing actual workers in real factories. Those

---

[2] https://youtu.be/yYNpKEClclA

data will then be transformed into well-defined persona(s), scenario and a use case, in order to update our current setup according to our personas' needs.

7.5 CONCLUSION AND FUTURE WORK

In our opinion, methods from HCI provide valuable tools to inform and improve HRI. With our paper, we recommend using methods such as scenarios, use cases and personas. Such instruments enable HRI solutions to better integrate user needs such as methods for simplified programming.

In the next step, we will include the results from using these methods (scenario, use case, persona) on our collaborative workspace.

In order to fulfill the defined use case and the corresponding scenario, it is now necessary to implement new robot instructions based on kinesthetic teaching as gluing and drilling. As the task is quite complex, it is inevitable to display the robot's program in addition to showing work instructions for users. The design elements as well as input methods of the user interface are adapted according to the needs of the refined personas. E.g. as our preliminary persona *Jan* often works with touch-based interfaces (phone, tablet) we will add a touch-sensitive layer on the worktable as an alternative input modality. We focus on making the system easily deployable, with multiple sensors and projectors. The user is enabled to switch between various interfaces based on the current task.

These system improvements result directly from our deployment of HCI methods in HRI. Having said this, we encourage other research groups to take a similar approach.

7.6 ACKNOWLEDGMENTS

This research was supported by the IT4IXS IT4Innovations Excellence in Science project (LQ1602).

8

INTERACTIVE SPATIAL AUGMENTED REALITY IN COLLABORATIVE ROBOT PROGRAMMING: USER EXPERIENCE EVALUATION

8.1 INTRODUCTION

Contemporary collaborative robots are collaborative in the sense that for human workers, it is safe to work alongside them. However, human-robot interaction is very limited if it exists at all: The behavior of the robot is pre-programmed without cognition of an environment, a user, tools, or the parts necessary for a given task. The robots are programmed by domain experts using specialized devices and an expert is needed even for small changes in the program. It is expected that, in the near future, collaborative robots will be cheaper and thus more affordable for small and medium-sized enterprises (SMEs). In such companies, all of the aforementioned issues will be even more prominent. As robots in SMEs will have to deal with higher product variability (smaller batches, customization) it would be beneficial to allow workers with no specific skills to make changes in a robot's program. At the same time, it will be necessary to support a close human-robot collaboration, as with rising cost of human labor, it might be expected that a trend will occur to offload non-ergonomic or repetitive parts of the workflow to robots. In order to allow this, robots will have to perceive and interact.

In this work, we present a novel approach to programming collaborative robots based on cognition, spatial augmented reality (SAR) and multimodal input and output. In order to make programming as simple as possible, programming takes place on a high level of abstraction where no robot-specific knowledge is necessary. Our intention was to make interaction with robots easy, fun, safe and effective.

In order to evaluate the approach, we developed a proof of concept system (see fig. 8.1)[1] and carried out initial user experience testing. The purpose of the testing was to discover whether there are some fundamental usability issues related to the approach as well as to find out issues related to the current implementation. In the experiment, the robot played the role of a worker's assistant, preparing parts for assembly in a fictional SME.

---

[1] The code is available at https://github.com/robofit/artable.

Figure 8.1: Setup of the novel interactive system concept where all the interaction elements (visualization and control) are gathered in a shared workspace (example of setting program parameters using a robotic arm and gestures; image edited).

8.2 RELATED WORK

Various approaches exist aimed at the simplification of robot programming or to support human-robot collaboration on a joint task. One of the techniques used to make programming robots more suitable for non-expert users is programming by demonstration. For instance, the approach proposed in [97] was rated by non-expert users as highly intuitive. However, the tasks are quite simple and there is no feedback for the user. In [125], kinesthetic teaching is used in conjunction with an iconic based programming to enable users to create and edit non-trivial programs. While the usage of a graphical user interface (GUI) on a standard monitor adds more control over the program and provides feedback, it also leads to attention switches.

The system described in [48] uses behavior trees to represent the program and was successfully deployed at an SME. The program itself is created on the monitor. The parameters of the program could be set using GUI, object recognition or kinesthetic teaching. The usage of behavior trees leads to high flexibility and the creation of reusable pieces of programs; however, it also inevitably leads to a more complicated GUI. Similarly, the system described in [59] enables users to create complex programs using kinesthetic teaching and object recognition. However, three different GUIs and voice input are involved. Moreover, its target user group consists of general programmers.

The previous approaches share a common disadvantage: The inability to show information within a task context. On the other hand, [117] uses physical blocks to create a program which is highly intuitive (requires no training), although it is limited to trivial tasks. Recently, augmented reality (AR) has been used to show important information within a task context. Probably the most common approach is to use a hand-held device. In [123], the authors recruited robot programmers and evaluated a tablet-based AR interface for programming abstracted industrial tasks. From the results, it seems that the usage of an AR may lead to a decrease in the workload and higher motivation to perform accurately. However, the usage of a tablet prevents the usage of both hands. A head-mounted display frees the user's hands and according to [111] might lead to faster task completion times and higher accuracy. Unfortunately, the currently available devices have a limited field of view. Also, a head-mounted display probably would not be suitable for long-time usage. On the other hand, SAR is able to show information in context, does not require any hand-held devices, is suitable for long-term usage, and is visible to anyone. It was recently used to implement an interactive work desk [137], show instructions to workers [41], or to show robotic data and learn trajectories [78].

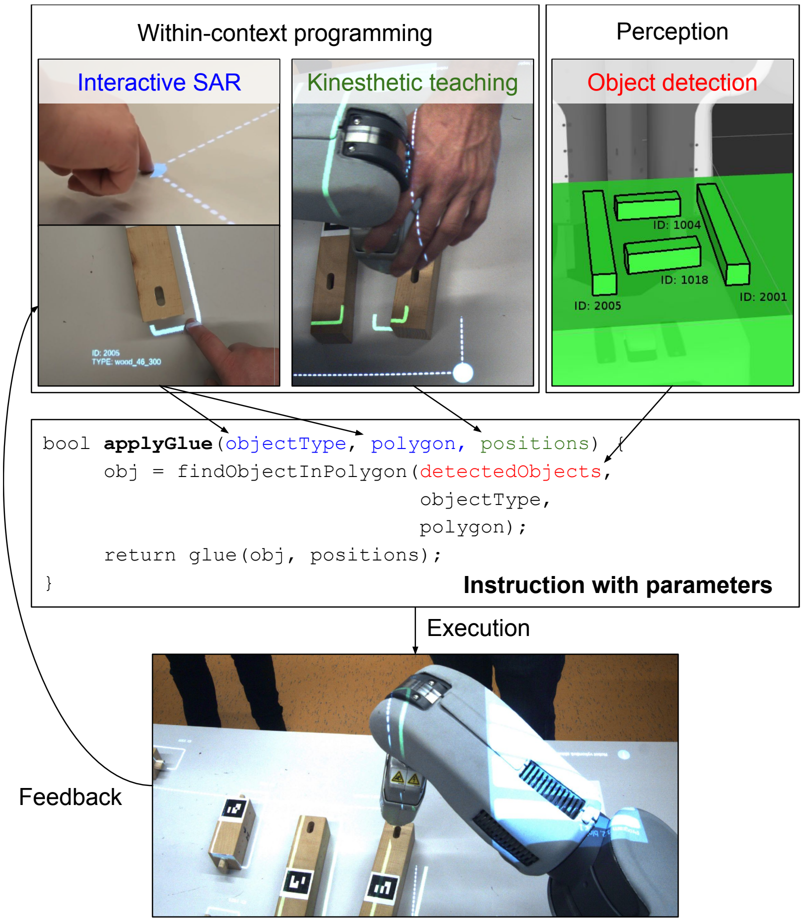
To the best of our knowledge, there is currently no existing interactive system targeting all of the following important issues:

- problems with attention switching when a monitor or a hand-held device is used to visualize the programming interface and system status during operation,
- too much information is presented to the user, leading to a higher mental workload,
- external devices are needed to fully interact with the robotic system (during both the programming and processing phases),
- low level of abstraction allowing only medium-expert users to program the robot.

8.3 PROPOSED APPROACH

We propose and initially evaluate a novel approach to collaborative robot programming with the following attributes (see also fig. 8.2):

- avoiding switching of the user's attention during programming and cooperation by placing all the interaction elements in a shared workspace,



```
bool applyGlue(objectType, polygon, positions) {
     obj = findObjectInPolygon(detectedObjects,
                                objectType,
                                polygon);
     return glue(obj, positions);
}
```

Figure 8.2: Illustration of program parameters' definition (combination of manually set parameters by the user with perceived information by the system) and its execution with visual feedback.

- decreasing the mental demands on the users by presenting the relevant information according to the current context,
- avoiding the usage of further external devices to interact with the system by making the shared workspace itself interactive,
- allowing non-expert users to work with the system by utilizing a high level of abstraction to program a robot.

Based on literature review and the current state of the technology, we see SAR as the most suitable instrument to visualize a user interface within a task context. While previous research has shown that gesture control is the preferred input modality for setting the parameters of common industrial tasks, we decided to use a touch-enabled table, which was also rated highly [89], and which is much more reliable. Moreover, together with SAR, it creates a similar user experience to tablets and smart phones, the usage of which is well-known to the general public. For tasks requiring 3D data input, the robot's arms could be used.

The user interface should be minimalistic, as the interface elements have to share space with real-world objects in the workspace: Tools, parts, etc. However, the design of the elements should allow convenient touch control. Depending on the state of the task, only the relevant information should be shown to lower the cognitive load [136]. The interface should clearly indicate the current state of the system, including an explicit representation of the robot's program and the context of the current program instruction (what happened before it and what is going to happen after it). Additional modalities, such as sound or light, could be used to for instance attract attention in special cases.

In order to make programming as well as the user interface as simple as possible, we decided to use complex instructions with a high amount of underlaying autonomy, at the price of lowering expressivity (see fig. 8.2). While theoretically, with the system from [48] one can create complex instructions from basic ones, it also makes the user interface complex and the program representation complicated. For instance, one has to set several poses, specify open and close gripper commands, etc. We believe that, for the sake of simplicity, the user should be abstracted from such low-level commands and the robot should perform them automatically.

To achieve a high level of abstraction and effective collaboration, the robot needs to perceive its surroundings as well as track its human coworker(s) and plan motions according to the current situation.

8.4 PROOF OF CONCEPT SYSTEM

To evaluate the proposed approach, a proof of concept system has been developed. The system allows end-user programming of selected industrial tasks.

8.4.1 *Setup*

The experimental setup (see fig. 8.1) was designed to be easy to deploy and modular. It is centered around a standard workshop table equipped with a capacitive touch foil. On the sides, two speaker stands are placed, connected by a truss. The truss is equipped with an Acer P6600 projector. There is a Microsoft Kinect V2 camera on each stand for object detection and calibration of the system. On one stand, there is an additional Kinect for user tracking. Each stand has its own processing unit (Intel NUC) where the projector and sensors are connected (in the study, only one projector was utilized). The unit is connected to the central computer using a wired network. The system is designed to be modular in a way so that it supports 1..n stands.

As a demonstrator of a near-future collaborative robot, we use the intrinsically safe PR2. The robot provides an additional set of sensors (Kinect and cameras on the head, cameras in the forearms). There is also a physical stop button under the table which shuts down the robot's motors.

8.4.2 *System design*

The system's state and behavior are defined and controlled by the central node and it can be manipulated by an arbitrary number of user interfaces. For instance, we currently use two interfaces: GUI projected on the table and a sound interface, providing audio feedback (e.g. confirmation of action, errors, etc.).

All parts of the system must be mutually calibrated first. Calibration of the Kinects utilizes an AR tracking library[2] to detect three markers placed on the table. One marker serves as an origin of the coordination system; the two others determine the X and Y axes. The PR2 robot is calibrated in the same way, using a head-mounted Kinect. To calibrate the projectors, a checkerboard pattern is displayed by each projector, and its corners are detected using already calibrated Kinects. In order to calibrate the touch-enabled surface, the points are projected on the table and the user has to click them. Then, homography is computed and used to convert the internal coordinates of the touch device into the common coordinate system.

Each of the objects used in our study has a set of two AR tags printed on the body, and multimarker detection is used to gain a unique ID of the object and its pose. Each object has an object type and a bounding box defined.

The manipulation pipeline is based on MoveIt! [26] and a library for grasp planning[3].

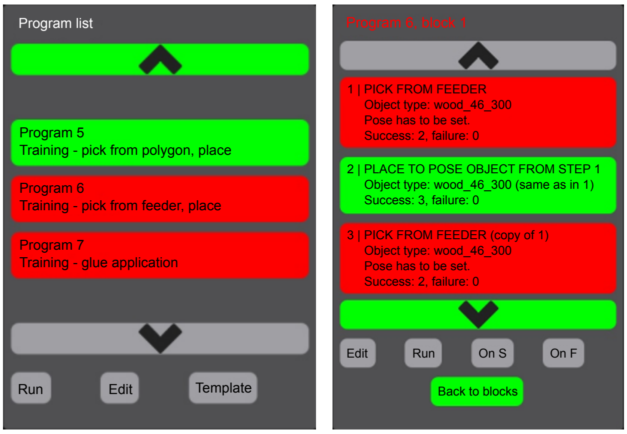
8.4.3 *Program representation*

The program in our system is a set of instructions, collected into blocks. Each program contains $1..n$ blocks; each block contains $1..n$ instructions. Every instruction execution can result in success (e.g. a successfully picked up object) or failure (e.g. failed to apply glue). Based on this result, the next instruction is determined. With this approach, simple branching and cycling of the program are possible (e.g. picking up objects from a feeder until the picking up failed, i.e. until there are no

---

[2] `http://wiki.ros.org/ar_track_alvar`

[3] `https://github.com/davetcoleman/moveit_simple_grasps`



Program list

Program 5
Training - pick from polygon, place

Program 6
Training - pick from feeder, place

Program 7
Training - glue application

Program 6, block 1

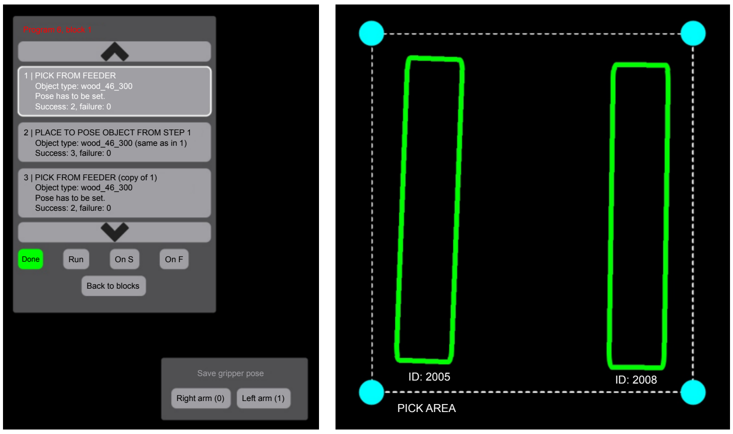
1 | PICK FROM FEEDER
Object type: wood_46_300
Pose has to be set.
Success: 2, failure: 0

2 | PLACE TO POSE OBJECT FROM STEP 1
Object type: wood_46_300 (same as in 1)
Success: 3, failure: 0

3 | PICK FROM FEEDER (copy of 1)
Object type: wood_46_300
Pose has to be set.
Success: 2, failure: 0

(a) List of programs. Green ones are ready to run, red ones need to set parameters.

(b) List of instructions. Green ones are ready to run, red ones need to set parameters.



Program 6, block 1

1 | PICK FROM FEEDER
Object type: wood_46_300
Pose has to be set.
Success: 2, failure: 0

2 | PLACE TO POSE OBJECT FROM STEP 1
Object type: wood_46_300 (same as in 1)
Success: 3, failure: 0

3 | PICK FROM FEEDER (copy of 1)
Object type: wood_46_300
Pose has to be set.
Success: 2, failure: 0

(c) A small dialog shows if the robot is able to detect an object in the feeder and allows the user to save the arm pose.

(d) Polygon defining the area on the table from which the objects will be picked up. The green outlines correspond to detected objects.

Figure 8.3: Examples of different widgets from proof of concept system.

objects left). For an example of a program structure in the form of a graph, see fig. 8.5.

Contrary to the conventional methods of programming robots, no precomputed joint configurations or arm paths are stored. By combining the perception capabilities of the system and on-the-fly motion planning, we do not rely on e.g. storing exact object positions.

It can be expected that the parameters of the program will be changed more often than the structure of the program. For this reason, we have divided the programming process into two parts. First, an empty template is created offline. This template can be seen as a description of an industrial technological process. It contains a set of instructions with defined transitions; however, without parameters. Thus, the template can be created once and later be adapted to conform to different products by setting instruction parameters.

8.4.4 *Supported instructions*

The system currently supports the following parametric instructions: *pick from polygon* (to pick up an object from a table), *pick from feeder* (to pick up parts from a gravity feeder), *place to pose* (to place a previously picked-up object on a selected place on the table) and *apply glue* (simulated gluing). Each of these instructions has certain parameters to be set by the user.

The object type must be set for all of these instructions. For the *pick from polygon* and *apply glue*, a polygon defining the area of interest on the table has to be set, so that the user can limit objects of the given type affected by the instructions.

For the *pick from feeder*, a pre-picking pose (see fig. 8.4c), used for object detection, has to be set using the robot's arm. While executing this instruction, the robot moves to the stored pose, observes the objects with its forearm camera and picks up the closest object in the direction of the gripper. For *apply glue*, the poses where the glue is supposed to be applied have to be set using an arbitrary arm of the robot.

There are also a couple of non-parametric instructions: *get ready*, *wait for user*, and *wait until user finishes*. The first one moves the robot's arms to their default position. The other instructions allow the synchronization of the system and the user. The *wait for user* instruction will pause the program execution until the user is in front of the table, while *wait until user finishes* will pause the program until the user finishes current interaction with the objects on the table. In our experiments, the behavior of these two instructions was simulated and controlled by the Wizard of Oz approach.

8.4.5 *User Interaction*

The interaction between the user and the system is currently achieved using three modalities: GUI projected on the touch-enabled surface (which serves as an input for the system and feedback for the user), kinesthetic teaching (input to the system only), and sound (feedback for the user only).

The GUI is composed of various widgets. The list of programs (see fig. 8.3a) shows all the programs stored in the system. The color of each entry suggests whether the program has set all the parameters (green; only these can be started) or some of them are not set (red). Any program can be templated (it is duplicated as a new program, with no parameters set) or edited (the user may set or adjust its parameters). During the program editation, the user can see a list of blocks of the selected program and can edit a selected block or get back to the list of programs.

When editing a block of a program, the list of instructions is shown (see fig. 8.3b). The selected instruction is always in the middle (with exception for the first and the last one) so the user can see its context. Similarly to the program list, each instruction has either a red or a green background, indicating whether it has all the parameters set. When all the parameters have been set, the selected instruction can be executed. Moreover, a gray instruction background suggests a non-parametric instruction. There are also buttons to navigate through the program, to select an instruction following either the successful or failed execution of the current instruction.

When a program has been executed, the list of instructions differs slightly. All the instructions are grayed out and are not interactive, and the buttons for pausing and stopping the program are displayed. The instruction detail shows: The type of the instruction (e.g. *pick from feeder*), the parameters (e.g. object type) and transitions for success and failure.

The user is notified about the state of the system and the errors, as well as the currently available actions, using a notification bar shown next to the front edge of the table.

It is important for the user to know the state of the system, so for every detected object an outline and ID are displayed (see fig. 8.3d). The type of the object is displayed upon clicking on the outline. For the purpose of setting the parameters, more information is shown, such as a polygon defining the area on the table, the outline of the object showing the position for object placement, etc. The same is also shown during the program execution, so the user knows in advance what object the robot will work with.

Various dialogs exist which allows the user to specify additional information. For instance, while programming an *pick from feeder* instruction, the user has to

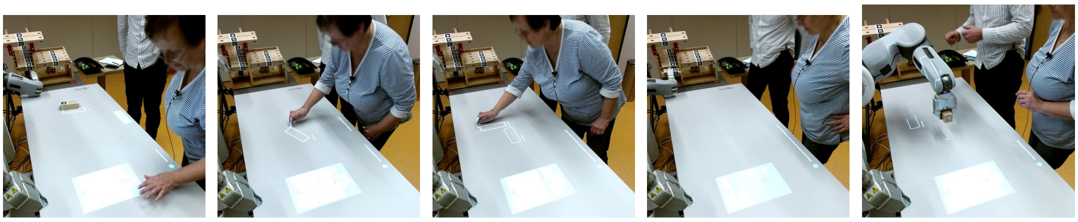
(a) User selects instruction to be set from list (pick).

(b) Object type is set by touching its outline.

(c) Robot arm is used to teach detection position.

(d) Dialog shows if robot is able to detect object in feeder.

(e) User saves position (confirmation sound is played).



(f) User selects follow-up instruction (place).

(g) User adjusts place pose by dragging it on the table.

(h) Another pose, first one also shown for convenience.

(i) User tests *pick from feeder* instruction.

(j) Test of place to pose instruction.

Figure 8.4: An example of human-robot interaction during the experiment. In this case, the user sets parameters for two *pick from feeder* instructions (one shown) and consequent *place to pose* instructions (both shown). Then, instructions are tested. Two input modalities are used: touch table and robot arm.

specify a pre-pose for object detection by manipulating the robot's arm and then confirming the position using a dialog. The pose is saved after pressing a button corresponding to the arm used (see fig. 8.3c). The whole procedure is shown in fig. 8.4 (a-e).

8.4.6 *Known Limitations*

The main input modality – touch foil – is prone to false readings when metal objects are placed on it, which makes it unsuitable for certain industrial settings. In the future, it might be replaced with or complemented by a vision-based approach (e.g. one from [137]). 3D interaction is currently limited to the kinesthetic teaching of positions, with no means for their later visualization.

8.5 EVALUATION

In order to evaluate the proposed approach and to discover the main usability issues of the early prototype, a user experience testing was carried out[4]. Prior to the experiment itself, a pilot experiment with three subjects (faculty staff) took place, which helped us to verify the functionality of the prototype and to create the final experiment design.

As measures, we choose a combination of qualitative and quantitative data. Self-reported data were obtained using a questionnaire consisting of the System Usability Scale (SUS) [19], NASA Task Load Index (TLX) [50] in its raw form (simplified, with a scale in the range [1..7]) and a custom questionnaire focusing on the specifics of the system. We recorded the task completion times and the corresponding number of moderator interventions as quantitative data.

8.5.1 *Experiment protocol*

The experiment protocol consisted of four phases. None of the phases of the experiment was time-limited. There were one moderator and one operator in a separate room in charge of system monitoring, data recording, and WoZ (used solely to simulate user activity recognition).

8.5.1.1 *Introduction*

At the beginning of the experiment, the participants signed an informed consent form. They were told a story about a fictional SME producing wooden furniture: *"The company cannot afford a dedicated robot programmer, so it bought a collaborative robot programmable by any ordinary skilled worker. The robot will serve as an assistant preparing the parts for the workers who will do the assembly."* They were given information about safety, the parts of the workspace (interactive table, robot, feeders with furniture parts), and basic usage of the interface.

8.5.1.2 *Training*

The training phase consisted of three simple programs demonstrating the supported instructions. No specific product was assembled in this phase. The parameters of each program were first set by the participant and then the program was executed. During the execution, errors (e.g. a missing object) were intentionally invoked in order to gain familiarity with the error resolution dialog. In this phase, the moderator proactively helped the participants to complete the tasks and an-

---

[4] Overview of the experiment: https://youtu.be/cQqNLy6mE8w.

swered all the questions. A short practice of the think-aloud protocol followed. After that, the participants were told to set the parameters of those three programs independently while thinking aloud.

8.5.1.3 *Main task*

The assembly process of a target product (a small stool) was explained and the participants assembled it manually. Next, the structure of the corresponding program and the expected workflow were explained.

After the questions were answered, the participants started working. When finished, they started the program and collaborated with the robot on the task of producing a stool. Two stools were produced and the participants were told that there was a demand to adapt a product - to produce a higher stool. After the parameters of the program had been adapted, they produced one more.

8.5.1.4 *Feedback*

After finishing the tasks, an open discussion took place. The participants were asked for their impressions, additional questions, etc. Then, they were asked to fill in the questionnaire.

8.5.2 *Stool assembly*

The intended workflow of the main task is that the user does the assembly while the robot prepares the parts needed in the next steps "on background". The program is divided into three blocks (see fig. 8.5). Blocks 1 and 2 have the same structure and serve to prepare the parts for the sides of the stool (two legs, two connecting parts, application of glue). The purpose of two blocks is that the user might set parts within one block to be supplied from e.g. the left feeder and in the other block from the right feeder. Block 3 serves to prepare the connecting parts for the final assembly of the sides of the stool.

8.5.3 *Participants*

In cooperation with an industrial partner (ABB Brno), six regular shop-floor workers of various ages, genders and technical backgrounds were selected (out of 27 volunteers) to take part in our study. These participants will be labeled as Participants A, B, C, D, E and F. Five of them work in quality control; one (E) works as a mechanic. The demographic data of the participants can be seen in table 8.1.

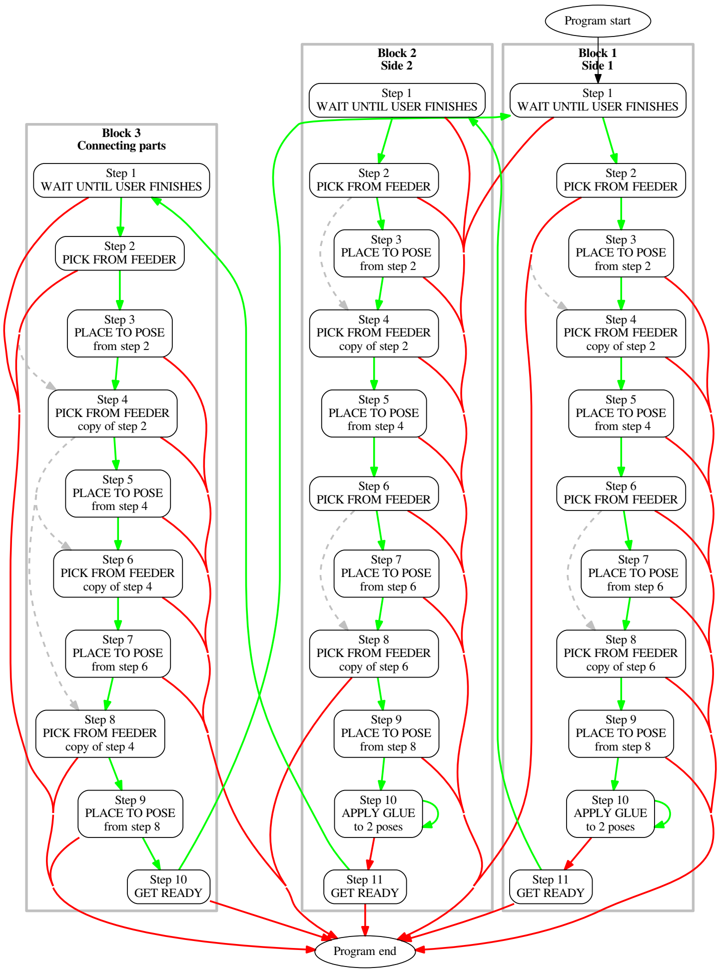


Figure 8.5: Stool production program. The green edges represent *on_success* transition, while the red ones represent *on_failure*. The grey edges show dependencies. In the case of *apply glue*, there is a loop. The robot applies glue to one object in a specified area. If an object is found, the program flow continues to the *on_success* instruction - it tries to apply glue to another object. If there is no object without glue applied, the flow continues to *on_failure* (next instruction).

8.6 RESULTS

The section provides results of the experiment.

8.6.1 *Qualitative and quantitative data*

table 8.2 shows the results per participant. The mean time to complete the main task was 2711 s (SD 620 s) with 11.7 (SD 6.7) moderator interventions. The main task consisted of setting the following instructions: 5x *pick from feeder* (2 parameters), 12x *place to pose* (1 parameter), 2x *apply glue* (4 parameters), resulting in settings of 30 parameters in total. The mean time for program adaptation task was

part.	**gender**	**age**	**education**	**experience with robots**	**attitude towards new technology**
A	F	57	vocational (technical)	none	skeptical
B	M	46	secondary (technical)	seen robot at least once	neutral
C	F	27	secondary (economics)	none	neutral
D	M	33	secondary (technical)	seen robot at least once	early adopter
E	M	24	secondary (technical)	works on workplace with robots but not next to them	neutral
F	M	34	undergraduate (technical)	none	skeptical

Table 8.1: Demographic data of the participants.

1053 s (SD 215 s). It consisted of setting: 2x *pick from feeder*, 2x *apply glue*, and optionally, adjustment of place poses (based on previously set poses), resulting in at least 12 parameters in total. These times include the delays caused by system errors (unreliable object detection, unstable manipulation pipeline, etc.). The mean SUS rating was 75.8 (SD 8.9), while for comparison, the system from [59] was scored 66.75 (SD 16.95). The mean TLX was 33.3 (SD 8.8).

From the custom questions (see table 8.3) it seems that the participants in general liked interacting with the system and felt safe; however, they were confused from time to time. However, during the experiment, in most cases it was enough to tell them to check the notification area and they were able to continue afterwards.

8.6.2 *Programming*

Observation of the users has shown that the current visualization of the robot program is probably not sufficient, as it often took considerable time to realize what was currently being programmed, especially for the case of repeating sequences of program items (e.g. *pick from feeder*, *place to pose*, *pick from feeder*, *place to pose*). Not fully consistent terminology (e.g. program instruction was sometimes referred to as item and sometimes as step) may have contributed to this. Probably because of the similar appearance, for some participants it was difficult at the beginning to distinguish between a program block and a program instruction.

Probably the most common issue during programming was the participant forgetting to press the *Edit* button in order to switch from the view-only mode to the parameter settings mode for the selected instruction. The participants often tried to adjust for example place pose and were confused as to why it was impossible. Also, it was often unclear that it is only possible to execute individual instructions.

Initially, two participants thought that the instructions (displayed in the program visualization) were for them, so they should perform e.g. *pick from feeder*. One participant asked if there are also assembly instructions for the workers.

There have been cases where the user accidentally changed the selected object type. Despite the fact that this was covered during training, some of the participants thought that the object type is selected when they put an object of that type on the table. It seems that although the objects of a selected type were highlighted differently (with a green outline), most of the participants only guessed what type was selected, or rather, checked it in the program visualization where the information was in textual form.

8.6.3 *Individual instructions*

8.6.3.1 *Pick from feeder*

Participants were often confused, as it was required to select the object type on the table and then to use a robot arm to set the pose enabling the detection of parts in the feeder. We noticed cases where the participant tried to select an object by knocking on it (instead of clicking on its outline), both on the object on the table and in the feeder. The participants commonly skipped the object selection, grabbed the robot arm and tried to set the pose, even above the object on the table, despite the fack that they were learning picking from feeder. After pressing *Edit*, dialog buttons for saving the arm pose (grayed-out at the time) were sometimes used “to select arm” before any other interaction. Most users took a new part from the feeder and put it on the table when they needed to select the object type even though there were already objects of that type that could have been used for this purpose. When adapting the program, it happened twice, that the participant by mistake set the position for the other feeder (e.g. the instruction originally used the left feeder, and they switched to the right one). This would mean that the robot would not be able later to place the object, as the following place pose (on the opposite side of the table) would be out of its reach.

8.6.3.2 *Place to pose*

Common sources of problems were unreachable place poses, or place poses too close to each other, which prevented the robot from placing parts successfully. The only possibility was to find out by trial and error. For all the participants, it was difficult initially to handle separated translation (by dragging) and rotation (using a pivot point). Some of them intuitively attempted to use multi-touch gestures (not supported by the interface thus far), including one participant who does not own any touch devices. Although the initial position of the place pose was in the middle of the table, some participants had trouble finding it, especially if there were many objects around. Some of them tried to drag the outline of a detected object or even placed an object into the outline of the place pose. Visualization of the place poses from other instructions (differentiated by a dotted line and a corresponding instruction number) were confused a few times with the current place pose and the users tried to move them.

For successful collaboration with the robot, it was necessary to organize the workspace so that the robot could prepare the parts for the next steps, while the user did the assembly. Only Participant B explicitly thought about organization of the workspace. The others had minor problems with it or required help. Participant C placed the parts in a very chaotic way. The participants were explicitly told during training that they may move widgets (e.g. program visualization) across the table; however, most of them did not use it and rather adjusted the place poses so that they did not collide with the widget.

8.6.3.3 *Glue application*

The most common issues were object type selection (attempts to select using the robot's arm) and difficulties with the number of actually stored poses (shown textually). The fact that it is necessary to store required poses only with regard to the one object and the fact that the robot will do it in the same way for other objects in a given area was also generally unclear.

8.6.4 *Program execution*

During the program execution, errors occurred relatively often, especially when the robot tried to place an object; erroneous detection prevented it from doing so. In the event of an error, a dialog appeared and sound was played. Most issues were solved just by pressing the *Try again* button. The participants were explicitly told to pay attention to errors. Some of the participants reacted immediately, others after some time and one seemed to ignore the errors and had to be told to solve them. Once in a while it was necessary to warn a participant that he or she was blocking the robot by occupying part of the table where the robot was meant to place parts.

8.6.5 *General findings*

No one complained about imperfections of the projection (shadows, inaccurate registration), low readability of the text, interface response times, etc. Each participant had an issue at least once with a non-touchable margin of the interactive table, which was not indicated by the projected interface. There were also issues

Measure	**A**	**B**	**C**	**D**	**E**	**F**
System Usability Scale	87.5	67.5	77.5	75.0	85.0	62.5
Simplified TLX	25.0	33.3	30.6	22.2	41.7	47.2
time to set program (s)	3849	3025	2618	2217	2661	1897
interventions	21	7	20	12	6	4
time to adapt program (s)	1088	1447	1118	958	738	968
interventions	11	4	12	2	2	2

Table 8.2: Qualitative measures, task completion times (stool program) and number of moderator interventions (including answering questions).

Statement	**A**	**B**	**C**	**D**	**E**	**F**
Collaboration was effective.	5	4	5	5	4	4
I felt safe.	4	5	5	5	5	5
Robot motions were uncomfortable.	2	1	1	1	1	1
It was easy to see what the robot was about to do.	4	5	5	4	4	2
The robot hindered me at work.	1	2	1	1	1	1
I watched every movement of the robot.	3	1	2	3	4	2
Learning the robot using its arm was intuitive.	4	4	5	5	5	4
Learning the robot using the interactive table was intuitive.	4	4	5	5	5	3
Interactive table shows all necessary information.	5	2	5	5	5	4
Sometimes I did not know what to do.	5	5	4	2	4	4

Table 8.3: custom questionnaire, 1 - totally disagree, 5 - totally agree

with pressing the buttons twice, where user tried, for example, to select an instruction which was immediately unselected. While inactive buttons were grayed out, most users tried to press them anyway when they thought they should work.

With many objects on the table or during the stool assembly, there was considerable visual clutter. Interestingly, no one mentioned it. Difficulties with moving interface elements (e.g. place pose) across longer distances were observed, especially if there were many objects on the table. Again, no one complained or asked if there was an alternative method to dragging.

As a complementary modality, there were sounds (confirmation, warning, error). Only Participant B explicitly appreciated it.

Regarding safety, only Participant A once noted that a particular movement was probably not safe. No one used the emergency stop button.

8.7 CONCLUSIONS

In this work, we targeted problems of the existing solutions in the area of interaction between the human workers and the industrial collaborative robots, particularly in the context of programming robots in SMEs. The proposed and tested interaction system is an attempt to reduce the mental demands and attention switching by centering all interaction elements in the shared workspace. This is achieved by the interactive SAR (combination of projection and a touch-enabled table) and kinesthetic teaching. Non-expert users program a robot on a high level of abstraction, and work within the task context, free of any additional external devices and with immediate visual feedback.

The conducted user experience tests proved the potential of our concept when all six regular shop-floor workers were able to program the robot to prepare parts for a stool assembly, to collaborate with the robot, and to adapt the program for an alternative product within a reasonable time.

During the experiment, no fundamental issues forcing us to reconsider the approach were found. However, the task state awareness in particular has to be improved as well as support for the workspace layout. The participants rated the system positively despite a number of minor usability issues and system errors caused by its experimental nature.

In addition to the revision of the interface to solve the usability issues, we plan to investigate multi-touch support, group operations, intelligent placement of user interface elements, and visualization of robot reachability.

ACKNOWLEDGMENTS

The work was supported by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

9

DISCUSSION AND CONCLUSIONS

9.1 ACHIEVEMENT OF RESEARCH OBJECTIVES

The research objectives were formulated in order to gain support for the claimed research statement. The following sections provide overview on how each research objective was fulfilled.

1. Define an integrative method for close human-robot interaction.

Based on the current state of the art, a novel method has been proposed. The method combines various already existing approaches in an original and previously unpublished way and provides a solid basis for design of advanced user interfaces. The method is specifically intended to allow non-expert users to accomplish non-trivial tasks within the use cases of remote operation of assistive robots and collaboration with industrial robots. It is sufficiently general, which allows application to other use cases; however, this remains as a challenge for the future work.

2. Apply the method within the contexts of interest.

The method was applied to the design of two user interfaces: the interface for remote operation of assistive service robots and the interface for collaborative industrial robots. For each application, it was necessary to take into account specifics of the use case, e.g. remote operation in one case and collocated interaction in the other. Despite this, all the key characteristics of the method were used.

3. Investigate if and how underlying autonomy could support human-robot interaction

Both developed interfaces heavily rely on an underlying autonomy, or other robot-integrated capabilities. For instance, the interface described in Chapter 5 uses continuously updated 3D model of the environment helping to overcome narrow field of view of the robot's main 3D sensor. Interface with visualization of global 3D mapping showed a clear temporal advantage for certain search and navigation tasks. Within the interface, both teleoperation as well as telemanipulation relies on integrated motion planning and collision avoidance in order to lower user's cognitive load.

The interface for robot programming (see Chapter 8) uses robot cognitive capabilities (ability to detect objects in its workspace) and on the fly motion planning to simplify process of programming as well as provide aid during task collaboration.

Although the influence of underlying autonomy usage was not investigated explicitly, both interfaces were successful (in the sense that users were able to solve tasks relatively easily and rated the interfaces positively) and therefore it could be concluded that utilization of underlying autonomy leads to improved HRI.

4. Investigate what modalities are appropriate for convenient interaction

The user preference of different modalities considering a variable amount of (synthetically induced) interaction errors for setting the most common parameters in industrial robot programming use case was the main focus of the research paper which is included as Chapter 6. From five input modalities, gestures and the touch sensitive table were the two most preferred ones. The gesture-based control was used for some preliminary experiments (see Chapter 7).

The touch sensitive table was later integrated into a fully functional prototype of interactive shared workspace. During the prototype evaluation, some specific usability issues related to the touch-sensitive table modality were identified (see Chapter 8) and should be taken into account for future designs. The prototype used SAR for visualization (output modality) and together with the touch-sensitive table formed the ISAR interface, which was rated as highly intuitive.

Interaction modalities were also considered when designing the teleoperation interface for semi-autonomous assistive robot (see Chapter 5), where the 3D mouse was selected as a suitable device for given tasks: teleoperation and setting the desired end effector pose. A non-linear transformation was applied to data from the 3D mouse to allow precise as well as fast movements. Further, control was adjusted according to the current 3D scene viewpoint (user's perspective) to make interaction more intuitive.

5. Investigate how the joint task should be presented to make it comprehensive and how to support situation awareness

Within the assistive robot use case, there is actually no exact procedure to be followed as the way of solving the problem depends on the operator's decision, who can use various tools according to personal preference and the problem at the hand. The interface is built upon ecological approach enabling the operator to directly infer possible actions from visualization of the environment. When the sub-task requires specific steps to be carried out, the interface provides textual guidance and automatically switches to proper visualization according to the task state. For analogical use cases, providing an operator with freedom to choose suitable approach and tools seems appropriate in order to maximize benefit from usage of operator's cognitive abilities.

On the other hand, interaction within the industrial use case could be strictly limited to the exact order of steps, e.g. given by technological process or limitations. For this case, task representation internally based on ROS messages and visualization based on ISAR has been developed. Moreover, the same interface allows both visualization of the task progress during its execution as well as setting of parameters for individual instructions. The program visualization is designed in a way that it provides context to the current instruction in a form of showing also previous and following instruction. In order to improve the situation awareness, there are short textual notifications and visualization of robot intentions where e.g. an object to be manipulated is highlighted.

6. Evaluate the method-based interfaces with non-expert users.

The interface for remote operation of assistive robots was thoroughly evaluated within the SRS project. The interface was tested out by a 81 non-expert users in total, both under simulation and within the real conditions. Some first evaluations served to figure out usability problems and to refine the interface. Later, two larger studies with more specific research questions were carried out: one was focused on comparing two modes of 3D environment visualization for solving remote navigation problems and the other one on potential utility of stereoscopic visualization for solving remote manipulation problems. In both studies, the tasks were realistic and far from trivial. Despite that, all users were able to finish all tasks and also the qualitative measures obtained were encouraging.

The interface for industrial robots allows non-expert users to program the robot and to collaborate with it on non-trivial tasks – it was evaluated on task consisting of 32 instructions with 30 parameters to be set in total. However, there are still some unsolved usability issues left for the future work. Although all users were able to solve the tasks, at least 4 moderator's interventions were required during setting program parameters and 2 during program adaptation task, which indicates potential for improvement.

9.2 CONCLUSIONS

There exist various solutions for HRI; however, many of them are not suitable for non-expert users, are constrained to basic tasks only or does not deal with close interaction. At the same time, robots are becoming more and more complex as their functionality and abilities to sense are expanding. There is a great challenge on how to utilize those features to maximize benefit for the human-robot team, whether it is a worker and its robotic collaborator in a factory or an assistive service robot helping an elderly person at home and its remote operator. Within this thesis, the central idea on how to face this challenge is formulated as the thesis statement. The specific approach supporting the statement was found by fulfilling the research objectives.

The selected direction of the research within this thesis was to realize a task-centered interaction. In other words, to embed the interface into the task-space, which is possible e.g. by using a mixed-reality approach. The method defined by several key characteristics was formulated. Two different user interface designs for two different use cases were implemented and evaluated with non-expert users, who were (without excessive training) able to achieve non-trivial tasks. Successful evaluation of the two implemented interfaces within different use cases and under different conditions (robot, environment, spatially co-located / remote interaction) indicates potential of the method as well as solid support for the thesis statement. The defined key characteristics may be seen as guidelines for design of forthcoming user interfaces.

9.3 FUTURE WORK

In the follow up research, I will mainly focus on interaction with collaborative industrial robots, as currently, this context seems to have a higher potential for real-world applications compared to the context of assistive service robots. In particular, I will focus on improved task understanding and awareness, which is of great importance for complex collaborative tasks. It would be interesting to investigate if and how the ISAR approach could be combined with another mixed-reality approaches as head-mounted displays and how it could be extended to non-flat surfaces.

Another direction of research will be to investigate if and how the proposed method could be extended in order to make HRI adaptable according to the current internal state of a user. For instance, measurement of a user's physiological state as heart rate could improve HRI by allowing the system to react on estimated cognitive workload of a user. Also, as all evaluations so far happened under highly controlled laboratory conditions (which might be seen as limitation of the conducted research), it would be desirable to carry out an out of the lab experiment, preferably a long-term one, to gain more insight into potential technical and usability issues, under the real conditions.

Naturally, the long-term goal is to bring the results of the research into a real-world applications, thereby help to accelerate adoption of collaborative robots, improve working conditions of workers and finally, to contribute to the peaceful future relationships between humans and robots in general.

BIBLIOGRAPHY

[1] Fares J Abu-Dakka, Bojan Nemec, Aljaž Kramberger, Anders Glent Buch, Norbert Krüger, and Ales Ude. Solving peg-in-hole tasks by human demonstration and exception strategies. *Industrial Robot: An Int. Journal*, 41(6):575–584, 2014.

[2] Sonya Alexandrova, Maya Cakmak, Kaijen Hsiao, and Leila Takayama. Robot programming by demonstration with interactive action visualizations. In *Procs. of Robotics: Science and Systems*, Berkeley, USA, July 2014. doi: 10.15607/RSS.2014.X.048.

[3] Sonya Alexandrova, Maya Cakmak, Kaijen Hsiao, and Leila Takayama. Robot programming by demonstration with interactive action visualizations. In *Robotics: science and systems*. Citeseer, 2014.

[4] Sonya Alexandrova, Zachary Tatlock, and Maya Cakmak. Roboflow: A flow-based visual programming language for mobile manipulation tasks. In *Robotics and Automation (ICRA), 2015 IEEE Int. Conference on*, pages 5537–5544. IEEE, 2015.

[5] Rasmus S Andersen, Simon Bøgh, Thomas B Moeslund, and Ole Madsen. Task space hri for cooperative mobile robots in fit-out operations inside ship superstructures. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 880–887. IEEE, 2016.

[6] Georg Arbeiter, Richard Bormann, Jan Fischer, Martin Hägele, and Alexander Verl. Towards geometric mapping for semi-autonomous mobile robots. In *Spatial Cognition VIII*, pages 114–127. Springer, 2012.

[7] J Alan Atherton and Michael A Goodrich. Supporting remote manipulation with an ecological augmented virtuality interface. *In Proceedings of the AISB Symposium on New Frontiers in Human-Robot Interaction, Edinburgh, Scotland*, 2009.

[8] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

[9] Anders Billesø Beck, Anders Due Schwartz, Andreas Rune Fugl, Martin Naumann, and Björn Kahl. Skill-based exception handling and error recovery for collaborative industrial robots. In *Procs. FinE-R Workshop*, pages 5–10, 2015.

[10] Gal Bejerano, Gregory LeMasurier, and Holly A Yanco. Methods for providing indications of robot intent in collaborative human-robot tasks. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 65–66. ACM, 2018.

[11] Christian J Bell, Pradeep Shenoy, Rawichote Chalodhorn, and Rajesh PN Rao. Control of a humanoid robot by a noninvasive brain–computer interface in humans. *J. of neural eng.*, 5(2):214, 2008.

[12] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Robot programming by demonstration. In *Springer Handbook of Robotics*, pages 1371–1394. Springer, 2008.

[13] Mark Billinghurst, Adrian Clark, Gun Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[14] Petra Björndal, Mikko J Rissanen, and Steve Murphy. Lessons learned from using personas and scenarios for requirements specification of next-generation industrial robots. In *HCII*, pages 378–387. Springer, 2011.

[15] Laura Boccanfuso, Quan Wang, Iolanda Leite, Beibin Li, Colette Torres, Lisa Chen, Nicole Salomons, Claire Foster, Erin Barney, Yeojin Amy Ahn, et al. A thermal emotion classifier for improved human-robot interaction. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 718–723. IEEE, 2016.

[16] Christopher Bodden, Daniel Rakita, Bilge Mutlu, and Michael Gleicher. Evaluating intent-expressive robot arm motion. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 658–663. IEEE, 2016.

[17] Jonathan Bohren, Chris Paxton, Ryan Howarth, Gregory D Hager, and Louis L Whitcomb. Semi-autonomous telerobotic assembly over high-latency networks. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 149–156. IEEE Press, 2016.

[18] Cynthia Breazeal, Cory D Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ Int. Conference on*, pages 708–713. IEEE, 2005.

[19] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[20] Nora Broy, Elisabeth André, and Albrecht Schmidt. Is stereoscopic 3d a better choice for information representation in the car? In *Proceedings of the 4th AutomotiveUI*, pages 93–100. ACM, 2012.

[21] David J Bruemmer, Douglas A Few, Ronald L Boring, Julie L Marble, Miles C Walton, and Curtis W Nielsen. Shared understanding for collaborative control. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(4):494–504, 2005.

[22] Elsa Bunz, Ravi Teja Chadalavada, Henrik Andreasson, Robert Krug, Maike Schindler, and Achim Lilienthal. Spatial augmented reality and eye tracking for evaluating human robot interaction. In *RO-MAN 2016 Workshop: Workshop on Communicating Intentions in Human-Robot Interaction, New York, USA, Aug 31, 2016*, 2016.

[23] Nutan Chen, Chee-Meng Chew, Keng Peng Tee, and Boon Siew Han. Human-aided robotic grasping. In *RO-MAN, 2012 IEEE*, pages 75–80. IEEE, 2012.

[24] Tiffany L Chen, Matei Ciocarlie, Steve Cousins, Phillip Grice, Kelsey Hawkins, Kaijen Hsiao, Charles C Kemp, Chih-Hung King, Daniel A Lazewatsky, Adam Leeper, et al. Robots for humanity: A case study in assistive mobile manipulation. *IEEE Robotics & Automation Magazine*, 2013.

[25] Keshav Chintamani, Alex Cao, R Darin Ellis, and Abhilash K Pandya. Improved telemanipulator navigation during display-control misalignments using augmented reality cues. *SMC, Part A: Systems and Humans, IEEE Transactions on*, 40(1):29–39, 2010.

[26] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.

[27] Alan Cooper et al. *The inmates are running the asylum:[Why high-tech products drive us crazy and how to restore the sanity]*. Sams Indianapolis, IN, USA:, 2004.

[28] Michael D Coovert, Tiffany Lee, Ivan Shindev, and Yu Sun. Spatial augmented reality as a method for a mobile robot to communicate intended movement. *Computers in Human Behavior*, 34:241–248, 2014.

[29] Kerstin Dautenhahn. The encyclopedia of human-computer interaction, 38. human-robot interaction.

[30] Michiel de Jong, Kevin Zhang, Aaron M Roth, Travers Rhodes, Robin Schmucker, Chenghui Zhou, Sofia Ferreira, João Cartucho, and Manuela Veloso. Towards a robust interactive and learning social robot. In *Proceedings*

of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pages 883–891. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[31] Jill L Drury, Jean Scholtz, and Holly A Yanco. Awareness in human-robot interactions. In *SMC 2003, IEEE International Conference on*, volume 1, pages 912–918. IEEE, 2003.

[32] Richard Edmondson, Kenneth Light, Andrew Bodenhamer, Paul Bosscher, and Loren Wilkinson. Enhanced operator perception through 3d vision and haptic feedback. In *SPIE Defense, Security, and Sensing*, pages 838711–838711. SPIE, 2012.

[33] Mica R Endsley. Automation and situation awareness. In *Automation and human performance*, pages 183–202. Routledge, 2018.

[34] Manuel Ferre, Rafael Aracil, and M Sanchez-Uran. Stereoscopic human interfaces. *Robotics & Automation Magazine, IEEE*, 15(4):50–57, 2008.

[35] David Fischinger, Peter Einramhof, Konstantinos Papoutsakis, Walter Wohlkinger, Peter Mayer, Paul Panek, Stefan Hofmann, Tobias Koertner, Astrid Weiss, Antonis Argyros, et al. Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems*, 75:60–78, 2016.

[36] International Organization for Standardization. *Ergonomics of Human-system Interaction: Part 210: Human-centred Design for Interactive Systems*. ISO, 2010.

[37] Juliano Franz, Anderson Maciel, and Luciana Nedel. Assessment of a user centered interface for teleoperation and 3d environments. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 953–960. ACM, 2013.

[38] Jeff Fryman and Bjoern Matthias. Safety of industrial robots: From conventional to collaborative applications. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–5. VDE, 2012.

[39] Tomofumi Fujiwara, Tetsushi Kamegawa, and Akio Gofuku. Stereoscopic presentation of 3d scan data obtained by mobile robot. In *SSRR, 2011 IEEE International Symposium on*, pages 178–183. IEEE, 2011.

[40] Wai-keung Fung, Wang-tai Lo, Yun-Hui Liu, and Ning Xi. A case study of 3d stereoscopic vs. 2d monoscopic tele-reality in real-time dexterous teleoperation. In *IROS, 2005 IEEE/RSJ International Conference on*, pages 181–186. IEEE, 2005.

[41] Markus Funk. Augmented reality at the workplace: a context-aware assistive system using in-situ projection. 2016.

[42] Markus Funk, Andreas Bächler, Liane Bächler, Thomas Kosch, Thomas Heidenreich, and Albrecht Schmidt. Working with augmented reality?: a long-term analysis of in-situ instructions at the assembly workplace. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 222–229. ACM, 2017.

[43] Andre Gaschler, Maximilian Springer, Markus Rickert, and Aaron Knoll. Intuitive robot tasks with augmented reality and virtual obstacles. In *Robotics and Automation (ICRA), 2014 IEEE Int. Conference on*, pages 6026–6031. IEEE, 2014.

[44] Brian Gleeson, Katelyn Currie, Karon MacLean, and Elizabeth Croft. Tap and push: Assessing the value of direct physical control in human-robot collaborative tasks. *Journal of Human-Robot Interaction*, 4(1):95–113, 2015.

[45] Michael A Goodrich and Alan C Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[46] Phillip M Grice and Charles C Kemp. Assistive mobile manipulation: Designing for operators with motor impairments. In *RSS 2016 Workshop on Socially and Physically Assistive Robotics for Humanity*, 2016.

[47] Kelleher R Guerin, Sebastian D Riedel, Jonathan Bohren, and Gregory D Hager. Adjutant: A framework for flexible human-machine collaborative systems. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ Int. Conference on*, pages 1392–1399. IEEE, 2014.

[48] Kelleher R Guerin, Colin Lea, Chris Paxton, and Gregory D Hager. A framework for end-user instruction of a robot assistant for manufacturing. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6167–6174. IEEE, 2015.

[49] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[50] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

[51] Stephen Hart, Paul Dinh, John D Yamokoski, Brian Wightman, and Nicolaus Radford. Robot task commander: A framework and ide for robot application development. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1547–1554. IEEE, 2014.

[52] Rex Hartson and Pardha S Pyla. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.

[53] M Hassenzahl, M Burmester, and F Koller. Attrakdiff: A questionnaire to measure perceived hedonic and pragmatic quality. In *Mensch & Computer*, pages 187–196, 2003.

[54] Hooman Hedayati, Michael Walker, and Daniel Szafir. Improving collocated robot teleoperation with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 78–86. ACM, 2018.

[55] Graeme Hirst, Susan McRoy, Peter Heeman, Philip Edmonds, and Diane Horton. Repairing conversational misunderstandings and non-understandings. *Speech communication*, 15(3):213–229, 1994.

[56] Johannes Hoecherl, Thomas Schlegl, Thomas Berlehner, Harald Kuhn, and Britta Wrede. Smartworkbench: Toward adaptive and transparent user assistance in industrial human-robot applications. In *ISR 2016: 47st International Symposium on Robotics; Proceedings of*, pages 1–8. VDE, 2016.

[57] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. doi: 10.1007/s10514-012-9321-0. URL http://octomap.github.com. Software available at http://octomap.github.com (accessed 12/07/2015).

[58] Haiying Hu, Jiawei Li, Zongwu Xie, Bin Wang, Hong Liu, and Gerd Hirzinger. A robot arm/hand teleoperation system with telepresence and shared control. In *AIM. Proceedings, 2005 IEEE/ASME International Conference on*, pages 1312–1317. IEEE, 2005.

[59] Justin Huang and Maya Cakmak. Code3: A system for end-to-end programming of mobile manipulator robots for novices and experts. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 453–462. ACM, 2017.

[60] Ifr. International federation of robotics, Mar 2018. URL https://ifr.org/papers/robots-and-the-workplace-of-the-future.

[61] ISO 13482:2014. Robots and robotic devices – Safety requirements for personal care robots. Standard, International Organization for Standardization, Geneva, CH, February 2014.

[62] ISO 8373:2012. Robots and robotic devices – Vocabulary. Standard, International Organization for Standardization, Geneva, CH, March 2012.

[63] Alejandro Jaimes and Nicu Sebe. Multimodal human computer interaction: A survey. In *International Workshop on Human-Computer Interaction*, pages 1–15. Springer, 2005.

[64] Kevin Sebastian Kain, Susanne Stadler, Manuel Giuliani, Nicole Mirnig, Gerald Stollnberger, and Manfred Tscheligi. Tablet-based augmented reality in the factory: Influence of knowledge in computer programming on robot teaching tasks. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 151–152. ACM, 2017.

[65] Slava Kalyuga, Paul Chandler, and John Sweller. Managing split-attention and redundancy in multimedia instruction. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 13 (4):351–371, 1999.

[66] Slava Kalyuga, Paul Ayres, Paul Chandler, and John Sweller. The expertise reversal effect. *Educational psychologist*, 38(1):23–31, 2003.

[67] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[68] Scott R Klemmer, Anoop K Sinha, Jack Chen, James A Landay, Nadeem Aboobaker, and Annie Wang. Suede: a wizard of oz prototyping tool for speech user interfaces. In *Procs. of the 13th annual ACM symposium on User interface software and technology*, pages 1–10. ACM, 2000.

[69] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IROS. Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[70] Jonathan Kofman, Xianghai Wu, Timothy J Luu, and Siddharth Verma. Teleoperation of a robot manipulator using a vision-based human-robot interface. *Industrial Electronics, IEEE Transactions on*, 52(5):1206–1219, 2005.

[71] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The Int. J of Robotics Research*, 32(8):951–970, 2013.

[72] C.R. Kothari. *Research Methodology: Methods and Techniques*. New Age International (P) Limited, 2004. ISBN 9788122415223. URL https://books.google.ch/books?id=hZ9wSHysQDYC.

[73] Martin Kraft and Markus Rickert. How to teach your robot in 5 minutes: Applying ux paradigms to human-robot-interaction. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017.

[74] Ernst Kruijff, J Edward Swan, and Steven Feiner. Perceptual issues in augmented reality revisited. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 3–12. IEEE, 2010.

[75] Daniel Labonte, Patrick Boissy, and François Michaud. Comparative analysis of 3-d robot teleoperation interfaces with novice users. *SMC, Part B: Cybernetics, IEEE Transactions on*, 40(5):1331–1342, 2010.

[76] Jens Lambrecht, Martin Kleinsorge, Martin Rosenstrauch, and Jörg Krüger. Spatial programming for industrial robots through task demonstration. *Int J Adv Robotic Sy*, 10(254), 2013.

[77] Adam Eric Leeper, Kaijen Hsiao, Matei Ciocarlie, Leila Takayama, and David Gossow. Strategies for human-in-the-loop robotic grasping. In *Proceedings of the seventh annual ACM/IEEE international conference on HRI*, pages 1–8. ACM, 2012.

[78] Florian Leutert, Christian Herrmann, and Klaus Schilling. A spatial augmented reality system for intuitive display of robotic data. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 179–180. IEEE Press, 2013.

[79] S Li, J Webb, X Zhang, and CA Nelson. User evaluation of a novel eye-based control modality for robot-assisted object retrieval. *Advanced Robotics*, 31(7):382–393, 2017.

[80] Salvatore Livatino and Giovanni Muscato. Robot 3d vision in teleoperation. In *World Automation Congress (WAC), 2012*, pages 1–6. IEEE, 2012.

[81] Stéphane Magnenat, Morderchai Ben-Ari, Severin Klinger, and Robert W Sumner. Enhancing robot programming with visual feedback and augmented reality. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 153–158. ACM, 2015.

[82] Sotiris Makris, Panagiotis Karagiannis, Spyridon Koukas, and Aleksandros-Stereos Matthaiakis. Augmented reality system for operator support in

human–robot collaborative assembly. *CIRP Annals-Manufacturing Technology*, 65(1):61–64, 2016.

[83] Pat Marion, Maurice Fallon, Robin Deits, Andrés Valenzuela, Claudia Pérez D'Arpino, Greg Izatt, Lucas Manuelli, Matt Antone, Hongkai Dai, Twan Koolen, et al. Director: A user interface designed for robot operation with shared autonomy. In *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*, pages 237–270. Springer, 2018.

[84] Marcus Mast, Michael Burmester, Eva Berner, David Facal, Lucia Pigini, and Lorenzo Blasi. Semi-autonomous teleoperated learning in-home service robots for elderly care: A qualitative study on needs and perceptions of elderly people, family caregivers, and professional caregivers. In *Proceedings of the twentieth ICROM*, pages 06–09, 2010.

[85] Marcus Mast, Michael Burmester, Katja Krüger, Sascha Fatikow, Georg Arbeiter, Birgit Graf, Gernot Kronreif, Lucia Pigini, David Facal, and Renxi Qiu. User-centered design of a dynamic-autonomy remote interaction concept for manipulation-capable robots to assist elderly people in the home. *J. of HRI*, 1(1), 2012.

[86] Marcus Mast, Michal Španěl, Georg Arbeiter, Vít Štancl, Zdeněk Materna, Florian Weisshardt, Michael Burmester, Pavel Smrž, and Birgit Graf. Teleoperation of domestic service robots: Effects of global 3d environment maps in the user interface on operators' cognitive and performance metrics. In *Social Robotics*, pages 392–401. Springer, 2013.

[87] Marcus Mast, Michael Burmester, Birgit Graf, Florian Weisshardt, Georg Arbeiter, Michal Španěl, Zdeněk Materna, Pavel Smrž, and Gernot Kronreif. Design of the human-robot interaction for a semi-autonomous service robot to assist elderly people. In *Ambient Assisted Living*, pages 15–29. Springer, 2015.

[88] Marcus Mast, Zdeněk Materna, Michal Španěl, Florian Weisshardt, Georg Arbeiter, Michael Burmester, Pavel Smrž, and Birgit Graf. Semi-autonomous domestic service robots: Evaluation of a user interface for remote manipulation and navigation with focus on effects of stereoscopic display. *Int. J. of Social Robotics*, 7(2):183–202, 2015. ISSN 1875-4791. doi: 10.1007/s12369-014-0266-7. URL `http://dx.doi.org/10.1007/s12369-014-0266-7`.

[89] Zdeněk Materna, Michal Kapinus, Michal Španěl, Vítězslav Beran, and Pavel Smrž. Simplified industrial robot programming: Effects of errors on multimodal interaction in woz experiment. In *RO-MAN*, pages 200–205. IEEE, 2016.

[90] George Michalos, Panagiotis Karagiannis, Sotiris Makris, Önder Tokçalar, and George Chryssolouris. Augmented reality (ar) applications for supporting human-robot interactive cooperation. *Procedia CIRP*, 41:370–375, 2016.

[91] François Michaud, Patrick Boissy, Daniel Labonté, Simon Briere, Karine Perreault, et al. Exploratory design and evaluation of a homecare teleassistive mobile robotic system. *Mechatronics*, 20(7):751–766, 2010.

[92] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[93] Sebastian Muszynski, J Stuckler, and Sven Behnke. Adjustable autonomy for mobile teleoperation of personal service robots. In *RO-MAN, 2012 IEEE*, pages 933–940. IEEE, 2012.

[94] Arne Muxfeldt, Jan Niklas Haus, Jingyuan Cheng, and Daniel Kubus. Exploring tactile surface sensors as a gesture input device for intuitive robot programming. In *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, pages 1–4. IEEE, 2016.

[95] Curtis W Nielsen, Michael A Goodrich, and Robert W Ricks. Ecological interfaces for improving mobile robot teleoperation. *Robotics, IEEE Transactions on*, 23(5):927–941, 2007.

[96] J Norberto Pires, J Norberto Pires, Germano Veiga, and Ricardo Araújo. Programming-by-demonstration in the coworker scenario for smes. *Industrial Robot: An Int. J*, 36(1):73–83, 2009.

[97] Eric M Orendt, Myriel Fichtner, and Dominik Henrich. Robot programming by non-experts: Intuitiveness and robustness of one-shot robot programming. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 192–199. IEEE, 2016.

[98] Maria Pateraki, Haris Baltzakis, and Panos Trahanias. Visual estimation of pointed targets for robot guidance via fusion of face pose and hand orientation. *Computer Vision and Image Understanding*, 120:1–13, 2014.

[99] Chris Paxton, Andrew Hundt, Felix Jonathan, Kelleher Guerin, and Gregory D Hager. Costar: Instructing collaborative robots with behavior trees and vision. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 564–571. IEEE, 2017.

[100] Mikkel Rath Pedersen, Lazaros Nalpantidis, Rasmus Skovgaard Andersen, Casper Schou, Simon Bøgh, Volker Krüger, and Ole Madsen. Robot skills

for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing*, 37:282–291, 2016.

[101] Alexander Perzylo, Nikhil Somani, Stefan Profanter, Markus Rickert, and Alois Knoll. Multimodal binding of parameters for task-based robot programming based on semantic descriptions of modalities and parameter types. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Workshop on Multimodal Semantics for Robotic Systems, Hamburg, Germany*, 2015.

[102] TXN Pham, K Hayashi, Christian Becker-Asano, S Lacher, and Ikuo Mizuuchi. Evaluating the usability and users' acceptance of a kitchen assistant robot in household environment. In *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*, pages 987–992. IEEE, 2017.

[103] Lucia Pigini, David Facal, Alvaro Garcia, Michael Burmester, and Renzo Andrich. The proof of concept of a shadow robotic system for independent living at home. In *Computers Helping People with Special Needs*, pages 634–641. Springer, 2012.

[104] Benjamin Pitzer, Michael Styer, Christian Bersch, Charles DuHadway, and Jan Becker. Towards perceptual shared autonomy for robotic mobile manipulation. In *ICRA, 2011 IEEE International Conference on*, pages 6245–6251. IEEE, 2011.

[105] Clemens Pohlt, Sebastian Hell, Thomas Schlegl, and Sven Wachsmuth. Impact of spontaneous human inputs during gesture based interaction on a real-world manufacturing scenario. In *Proceedings of the 5th International Conference on Human Agent Interaction*, pages 347–351. ACM, 2017.

[106] S. Profanter, A. Perzylo, N. Somani, M. Rickert, and A. Knoll. Analysis and semantic modeling of modality preferences in industrial human-robot interaction. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ Int. Conference on*, pages 1812–1818, Sept 2015. doi: 10.1109/IROS.2015.7353613.

[107] Renxi Qiu, Ze Ji, Alexandre Noyvirt, Anthony Soroka, Rossi Setchi, DT Pham, S Xu, N Shivarov, Lucia Pigini, Georg Arbeiter, et al. Towards robust personal assistant robots: Experience gained in the srs project. In *IROS, 2012 IEEE/RSJ International Conference on*, pages 1651–1657. IEEE, 2012.

[108] Camilo Perez Quintero, Romeo Tatsambon Fomena, Azad Shademan, Nina Wolleb, Travis Dick, and Martin Jägersand. Sepo: Selecting by pointing as an intuitive human-robot command interface. In *ICRA*, pages 1166–1171, 2013.

[109] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2012. ISSN 1573-7462. doi: 10.1007/s10462-012-9356-9. URL `http://dx.doi.org/10.1007/s10462-012-9356-9`.

[110] Ulrich Reiser, Christian Pascal Connette, Jan Fischer, Jens Kubacki, Alexander Bubeck, Florian Weisshardt, Theo Jacobs, Christopher Parlitz, Martin Hägele, and Alexander Verl. Care-o-bot® 3-creating a product vision for service robot applications by integrating design and technology. In *IROS*, volume 9, pages 1992–1998, 2009.

[111] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. Communicating robot arm motion intent through mixed reality head-mounted displays. *arXiv preprint arXiv:1708.03655*, 2017.

[112] N.J. Salkind. *Encyclopedia of Research Design*. Number sv. 1. SAGE Publications, 2010. ISBN 9781412961271. URL `https://books.google.cz/books?id=pvo1SauGirsC`.

[113] Marc Schneider and Barbara Deml. Analysis of a multimodal human-robot-interface in terms of mental workload. In *Advances in Ergonomic Design of Systems, Products and Processes*, pages 247–260. Springer, 2017.

[114] Casper Schou, Jens S Damgaard, S Bogh, and Ole Madsen. Human-robot interface for instructing industrial tasks using kinesthetic teaching. In *Robotics (ISR), 2013 44th Int. Symposium on*, pages 1–6. IEEE, 2013.

[115] David R Scribner and James W Gombash. The effect of stereoscopic and wide field of view conditions on teleoperator performance. Technical report, DTIC Document, 1998.

[116] Yasaman Sefidgar, Sarah Elliott, and Maya Cakmak. A system for situated tangible programming of robot skills. In *IROS Workshop on Learning for Collaborative Robotics: Enabling Flexible, Redeployable and Agile Industrial Applications*, 2017.

[117] Yasaman S Sefidgar, Prerna Agarwal, and Maya Cakmak. Situated tangible robot programming. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 473–482. ACM, 2017.

[118] Yasaman S Sefidgar, Thomas Weng, Heather Harvey, Sarah Elliott, and Maya Cakmak. Robotist: Interactive situated tangible robot programming. In *Proceedings of the Symposium on Spatial User Interaction*, pages 141–149. ACM, 2018.

[119] Dadhichi Shukla, Ozgur Erkent, and Justus Piater. Probabilistic detection of pointing directions for human-robot interaction. In *Digital Image Computing: Techniques and Applications (DICTA), 2015 Int. Conference on*, pages 1–8. IEEE, 2015.

[120] Sichao Song and Seiji Yamada. Expressing emotions through color, sound, and vibration with an appearance-constrained social robot. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 2–11. ACM, 2017.

[121] Michal Spanel, Zdenek Materna, Vit Stancl, Tomas Lokaj, Pavel Smrz, and Pavel Zemcik. Deliverable d4. 5.2–context-aware virtual 3d display final report. 2013.

[122] Susanne Stadler, Kevin Kain, Manuel Giuliani, Nicole Mirnig, Gerald Stollnberger, and Manfred Tscheligi. Augmented reality for industrial robot programmers: Workload analysis for task-based, augmented reality-supported robot control. In *RO-MAN*, New York, USA, August 2016. IEEE.

[123] Susanne Stadler, Kevin Kain, Manuel Giuliani, Nicole Mirnig, Gerald Stollnberger, and Manfred Tscheligi. Augmented reality for industrial robot programmers: Workload analysis for task-based, augmented reality-supported robot control. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 179–184. IEEE, 2016.

[124] Susanne Stadler, Nicole Mirnig, Manuel Giuliani, Manfred Tscheligi, Zdenek Materna, and Michal Kapinus. Industrial human-robot interaction: Creating personas for augmented reality supported robot control and teaching. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 291–292. ACM, 2017.

[125] Maj Stenmark, Mathias Haage, and Elin Anna Topp. Simplified programming of re-usable skills on a safe industrial robot: Prototype and evaluation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 463–472. ACM, 2017.

[126] Gerald Stollnberger, Astrid Weiss, and Manfred Tscheligi. " the harder it gets" exploring the interdependency of input modalities and task complexity in human-robot collaboration. In *RO-MAN, 2013 IEEE*, pages 264–269. IEEE, 2013.

[127] J. Suarez and R. R. Murphy. Hand gesture recognition with depth images: A review. In *RO-MAN, 2012 IEEE*, pages 411–417, Sept 2012. doi: 10.1109/ROMAN.2012.6343787.

[128] Xiang Zhi Tan, Aaron Steinfeld, and M Bernardine Dias. Communication through movement: An alternative method of interaction for hri, 2016.

[129] Markku Turunen, Jaakko Hakulinen, Aleksi Melto, Tomi Heimonen, Tuuli Laivo, and Juho Hella. Suxes-user experience evaluation method for spoken and multimodal interaction. In *INTERSPEECH*, pages 2567–2570, 2009.

[130] Jörn Vogel, Katharina Hertkorn, Rohit U Menon, and Máximo A Roa. Flexible, semi-autonomous grasping for assistive robotics. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 4872–4879. IEEE, 2016.

[131] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 316–324. ACM, 2018.

[132] Jeffrey D Will, Kevin L Moore, and Ian K Lynn. Optimizing human-robot teleoperation interfaces for mobile manipulators. *Industrial Robot: An Int. J.*, 40(2):173–184, 2013.

[133] Thomas Witzig, J Marius Zollner, Dejan Pangercic, Sarah Osentoski, Rainer Jakel, and Rudiger Dillmann. Context aware shared autonomy for robotic manipulation tasks. In *IROS, 2013 IEEE/RSJ International Conference on*, pages 5686–5693. IEEE, 2013.

[134] Robert H Wortham, Andreas Theodorou, and Joanna J Bryson. Improving robot transparency: real-time visualisation of robot ai substantially improves understanding in naive observers. *IEEE RO-MAN 2017*, 2017.

[135] Sebastian Wrede, Christian Emmerich, Ricarda Grünberg, Arne Nordmann, Agnes Swadzba, and Jochen Steil. A user study on kinesthetic teaching of redundant robots in task and configuration space. *Journal of Human-Robot Interaction*, 2(1):56–81, 2013.

[136] Daniela Wurhofer, Verena Fuchsberger, Thomas Meneweger, Christiane Moser, and Manfred Tscheligi. Insights from user experience research in the factory: What to consider in interaction design. In *Human Work Interaction Design. Work Analysis and Interaction Design Methods for Pervasive and Smart Workplaces*, pages 39–56. Springer, 2015.

[137] Robert Xiao, Scott Hudson, and Chris Harrison. Supporting responsive cohabitation between virtual interfaces and physical objects on everyday surfaces. *HCI*, 1(1):12, 2017.

[138] Lihua Xue, CJ Liu, Y Lin, and WJ Zhang. On redundant human-robot interface: Concept and design principle. In *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on*, pages 287–292. IEEE, 2015.

[139] Manuel Yguel and Olivier Aycard. 3d mapping of outdoor environment using clustering techniques. In *ICTAI, 2011 23rd IEEE International Conference on*, pages 403–408. IEEE, 2011.

[140] Jie Zhu, Xiangyu Wang, and Michael Rosenman. Fusing multiple sensors information into mixed reality-based user interface for robot teleoperation. In *SMC 2009, IEEE International Conference on*, pages 876–881. IEEE, 2009.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". It is available for LaTeX via CTAN as `classicthesis`.

Final Version as of January 22, 2019 (`classicthesis` version 1.0).