Interactive Restriction of a Mobile Robot's Workspace in Traditional and Smart Home Environments

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Dennis Sprute, M.Sc.

geb. am 26.03.1991 in Minden

Gutachterinnen/Gutachter

Prof. Dr.-Ing. Klaus Tönnies
Prof. Dr. Dr.-Ing. Matthias König
Prof. Dr. Juan Augusto

Magdeburg, den 20.10.2020

Sprute, Dennis:
Interactive Restriction of a Mobile Robot's Workspace in Traditional and Smart Home Environments
Doctoral Thesis, Otto-von-Guericke University, Magdeburg, 2020
Date of submission: 17.04.2020
Date of defense: 23.09.2020

Abstract

Nowadays, mobile service robots can autonomously navigate in human-centered environments, e.g. traditional or smart home environments, and provide services to humans, such as vacuum cleaning, fetching objects or acting as social robots. Although humans appreciate these services of robots, there are scenarios in which humans want to restrict the workspaces of their mobile robots, e.g. due to privacy concerns or to avoid robots' navigation errors. For this purpose, a human has to specify restriction areas in an interaction process with a robot. This interaction is challenging due to the transfer of complex spatial information about the restriction areas and the necessity to provide feedback during the interaction process with only limited mobile robot's on-board capabilities. Moreover, the interaction process has to fulfill ambitious user requirements concerning (1) correctness, (2) flexibility, (3) completeness, (4) accuracy, (5) interaction time, (6) user experience and (7) learnability. Current solutions to this problem, i.e. the interactive restriction of a mobile robot's workspace, do not optimally address these user requirements. However, an appropriate solution to this problem is essential to foster the deployment of mobile robots in human-centered environments.

To address this problem, we propose virtual borders as a data structure to flexibly model restriction areas. These non-physical borders are incorporated into a human-aware navigation framework to enable a human the restriction of a mobile robot's workspace and change of its navigational behavior. In order to allow a human to specify the components of a virtual border in a traditional home environment, we propose two alternative interaction methods based on (1) a laser pointer and (2) augmented reality (AR). Experimental results show that the laser pointer approach mostly features an acceptable performance on the user requirements but without a significant improvement with respect to a state-of-the-art solution. This state-of-the-art solution was identified in a literature review and is based on sketching restriction areas on an occupancy grid map (OGM) of the environment. In contrast to this, the second proposed interaction method based on AR reveals a good performance on most of the user requirements outperforming the state-of-the-art solution.

The reasons for the inferiority of the proposed laser pointer approach are two drawbacks identified in the evaluation: (1) a direct line of sight between human and mobile robot is required, which leads to an increase of interaction time and negatively affects user experience aspects. (2) In addition, the limited robot's on-board feedback capabilities only allow simple feedback, which has a negative effect on the user experience. To improve the laser pointer approach, we address these drawbacks by incorporating components of a smart home environment into the interaction process. To this end, we extend the laser pointer method by leveraging a (1) smart camera network, (2) a smart display and (3) a smart speaker to enhance the mobile robot's perceptual and interaction capabilities. A particular challenge is the cooperative perception of laser spots from multiple stationary and mobile cameras during the interaction process. Therefore, we propose a multi-stage algorithm to extract a

single virtual border from multiple camera observations. The results of an experimental evaluation demonstrate that the interaction method features an improved interaction time and user experience while not negatively affecting the other user requirements. In terms of user requirements, the interaction method thus performs better on average than the state-of-the-art solution.

Finally, all interaction methods available so far (state-of-the-art as well as proposed interaction methods) are based on pure human-robot interaction (HRI), i.e. a human specifies a restriction area by explicitly defining all components of a virtual border. This is time intensive and leads to a linear interaction time with respect to the length of a virtual border. We tackle this limitation by proposing a learning and support system (LSS) on top of the smart home's camera network, which aims to reduce the interaction time. This system learns from multiple interaction processes and supports a human through appropriate recommendations for virtual borders in future interaction processes. To this end, the LSS employs a combination of semantic segmentation, frequent itemset mining and AR. An experimental evaluation of the LSS reveals a reduced interaction time to a constant level without a negative effect on the other user requirements. Hence, these learning capabilities can further improve the state-of-the-art performance.

In summary, this work provides novel solutions to interactively restrict the workspace of a mobile robot, which achieve good results on the user requirements in our evaluations and outperform the current state-of-the-art solution in traditional as well as smart home environments.

Keywords Robot Workspace Restriction · Virtual Borders · Human-Robot Interaction · Smart Home · Intelligent Environment · Network Robot System · Laser Pointer · Augmented Reality

Zusammenfassung

Heutzutage können mobile Serviceroboter autonom in menschenzentrierten Umgebungen navigieren, wie z.B. in herkömmlichen oder intelligenten Wohnumgebungen, und den Menschen Dienste anbieten, wie z.B. staubsaugen, Gegenstände holen oder als soziale Roboter agieren. Obwohl Menschen diese Dienste der Roboter schätzen, gibt es Szenarien, in denen die Menschen den Arbeitsbereich der mobilen Roboter einschränken wollen, z.B. aus Gründen der Privatsphäre oder um Navigationsfehler der Roboter zu vermeiden. Hierzu muss ein Mensch Restriktionsbereiche in einem Interaktionsprozess mit dem Roboter spezifizieren. Diese Interaktion ist aufgrund des Transfers von komplexen räumlichen Informationen über die Restriktionsbereiche und der Notwendigkeit für Feedback während des Interaktionsprozesses mit eingeschränkten Fähigkeiten des mobilen Roboters anspruchsvoll. Zudem muss der Interaktionsprozess ambitionierte Benutzeranforderungen hinsichtlich (1) Korrektheit, (2) Flexibilität, (3) Vollständigkeit, (4) Genauigkeit, (5) Interaktionszeit, (6) Nutzererlebnis und (7) Lernfähigkeit erfüllen. Aktuelle Lösungen für dieses Problem, d.h. die interaktive Beschränkung des Arbeitsbereichs eines mobilen Roboters, gehen nicht optimal auf diese Anforderungen ein. Eine angemessene Lösung dieses Problems ist jedoch wichtig, um den Einsatz mobiler Roboter in menschenzentrierten Umgebungen zu fördern.

Um dieses Problem zu adressieren, verwenden wir virtuelle Grenzen als Datenstruktur zur flexiblen Modellierung von Restriktionsbereichen. Diese nicht-physischen Grenzen werden in ein menschenfreundliches Navigations-Framework integriert, um Menschen die Beschränkung des Arbeitsbereichs eines mobilen Roboters und die Änderung dessen Navigationsverhaltens zu ermöglichen. Um einem Menschen die Spezifizierung der Komponenten einer virtuellen Grenze in einer herkömmlichen Wohnumgebung zu erlauben, schlagen wir zwei alternative Interaktionsmethoden vor, die auf einem (1) Laserpointer und (2) Augmented Reality (AR) basieren. Experimentelle Ergebnisse zeigen, dass der Laserpointer-Ansatz meistens eine akzeptable Leistung bezüglich der Benutzeranforderungen aufweist, jedoch ohne signifikante Verbesserung in Bezug auf den aktuellen Stand der Technik. Diese Methode des aktuellen Stands der Technik wurde in einer Literaturrecherche ermittelt und basiert auf dem Zeichnen von Restriktionsbereichen auf einer Belegungskarte der Umgebung. Im Gegensatz dazu offenbart der zweite Ansatz basierend auf AR eine gute Leistung in Bezug auf die meisten Benutzeranforderungen und übertrifft die Leistung des Stands der Technik.

Die Gründe für die Unterlegenheit des Laserpointer-Ansatzes sind zwei Nachteile, die in der Evaluation identifiziert wurden: (1) eine direkte Sichtverbindung zwischen Mensch und mobilem Roboter ist erforderlich. Dies führt zu einer Erhöhung der Interaktionszeit und wirkt sich negativ auf die Aspekte des Nutzererlebnisses aus. (2) Zudem erlauben die begrenzten Feedbackmöglichkeiten des mobilen Roboters nur einfaches Feedback, was sich negativ auf das Nutzererlebnis auswirkt. Um den Laserpointer-Ansatz zu verbessern, reagieren wir auf diese Nachteile, indem wir Komponenten einer intelligenten Wohnumgebung in den Interaktionsprozess integrieren. Zu diesem Zweck er-

weitern wir die Laserpointer-Methode um ein (1) Kameranetzwerk, (2) ein intelligentes Display und (3) einen intelligenten Lautsprecher, um die Wahrnehmungs- und Interaktionsmöglichkeiten des mobilen Roboters zu stärken. Eine besondere Herausforderung hierbei ist die kooperative Wahrnehmung von Laserpunkten durch mehrere stationäre und mobile Kameras während des Interaktionsprozesses. Daher entwickeln wir einen mehrstufigen Algorithmus, um eine einzige virtuelle Grenze aus mehreren Kamerabildern zu extrahieren. Die Ergebnisse einer experimentellen Auswertung zeigen, dass die Interaktionsmethode eine verbesserte Interaktionszeit und ein verbessertes Nutzererlebnis bietet, ohne die anderen Benutzeranforderungen negativ zu beeinflussen. In Bezug auf die Benutzeranforderungen erzielt die Interaktionsmethode somit im Durchschnitt eine bessere Leistung als der Stand der Technik.

Schließlich basieren alle bisher erwähnten Interaktionsmethoden (sowohl Stand der Technik als auch in der Arbeit entwickelte Interaktionsmethoden) auf reiner Interaktion zwischen Mensch und Roboter, d.h. ein Mensch spezifiziert einen Restriktionsbereich durch explizite Definition der Komponenten einer virtuellen Grenze. Dies ist zeitintensiv und führt zu einer linearen Interaktionszeit in Bezug auf die Länge einer virtuellen Grenze. Wir gehen auf diese Einschränkung ein, indem wir ein Lern- und Unterstützungssystem (LSS) entwickeln, das auf dem Kameranetzwerk der intelligenten Wohnumgebung aufbaut und auf die Reduzierung der Interaktionszeit abzielt. Dieses System lernt aus mehreren Interaktionsprozessen und unterstützt den Menschen durch geeignete Empfehlungen für virtuelle Grenzen in zukünftigen Interaktionsprozessen. Zu diesem Zweck verwendet das LSS eine Kombination aus semantischer Segmentierung, Frequent Itemset Mining und AR. Eine experimentelle Auswertung des LSS zeigt eine reduzierte Interaktionszeit auf ein konstantes Niveau, ohne die anderen Benutzeranforderungen negativ zu beeinflussen. Daher können diese Lernfähigkeiten die Leistung des Stands der Technik weiter verbessern.

Zusammenfassend bietet diese Arbeit neue Lösungen zur interaktiven Einschränkung des Arbeitsbereichs eines mobilen Roboters, die gute Ergebnisse bezüglich der Benutzeranforderungen in den Evaluationen erzielen und die aktuelle Lösung des Stands der Technik sowohl in herkömmlichen als auch intelligenten Wohnumgebungen übertreffen.

Schlüsselwörter Roboter-Arbeitsbereichseinschränkung · Virtuelle Grenzen · Mensch-Roboter Interaktion · Smart Home · Intelligente Wohnumgebung · Netzwerkrobotersystem · Laserpointer · Augmented Reality

Contents

1 Introduction

1.1 Motivation

Starting from the beginning of robots in the middle of the twentieth century, the first robots, that were deployed in large quantities, were industrial robots in the 1970s. These stemmed from the development of digital computers and miniaturized components enabling the design of computer-controlled robots. They were exclusively deployed in factories for automation purposes, e.g. in the automotive, chemical, food or electronics industry, and they became the backbone of industrial manufacturing. (SICILIANO and KHATIB, 2016) These industrial robots were typically fixed robotic arms performing tasks, such as handling, welding, assembly or painting. For this purpose, they were required to handle high payloads and achieve a high speed and precision. In combination with missing perceptual capabilities, this resulted in hazardous working conditions for humans in factories. Thus, their working environment was strictly separated from the industrial robots' workspaces to ensure a safe human-robot coexistence/collaboration. (HÄGELE *et al.*, 2016)

In the 1990s, the robotics community triggered a change in the robots' scope by focusing on new research areas, such as service robotics. These robots targeted new potential markets to enhance the quality of human life. This implied the ability to autonomously operate in weakly structured environments, which was a large contrast compared to industrial robots working in their highly structured environments. This research culminated in a new generation of robots from the beginning of the 2000s. These robots left the factories and entered the human world. Thus, they were expected to safely co-habit with humans and provide services resulting in a benefit for humans. (SICILIANO and KHATIB, 2016) Since that time, we have witnessed an increase of robot sales and an emergence of numerous new robot services and applications (HÄGELE, 2016). For example, vacuum cleaning robots clean the floor (JONES, 2006), companion robots assist elderly people (GROSS *et al.*, 2015), collaborative robots interact with humans (VELOSO *et al.*, 2012) and service robots fetch and deliver objects (KUNZE *et al.*, 2012). Other well-known examples are lawnmower robots working in the garden (SCHEPELMANN *et al.*, 2010), tour guiding robots in museums (THRUN *et al.*, 2000) or robotic butlers (BOHREN *et al.*, 2011).

All these examples have in common that the robots move and work in the environment using their locomotion system, i.e. they are *mobile robots* (CORKE, 2017). Since the environment is a human-centered environment, i.e. humans live or work there, it arises a new constellation that has not been occurred before: a direct coexistence of humans and robots in the same environment resulting in a human-robot shared space. Humans and robots now share the same physical space without any restrictions, such as active constraints (BOWYER *et al.*, 2014) or protective devices known from factories (ISO, 2010). Furthermore, the humans are typically non-experts, i.e. they are neither the programmers of the robots nor do they have deep insights into robots and their functionality.

This constellation, mobile robots in environments with non-expert humans, gives opportunities for new applications and robot services, but also raises challenges that need to be addressed. These are especially challenges that focus on the human factor in the robot's environment, such as physical human-robot collaboration (CHUY *et al.*, 2006), interaction between human and robot (GOODRICH and SCHULTZ, 2007) or robot navigation in the presence of humans (CHARALAMPOUS *et al.*, 2017). A challenge, that is on the intersection of the research fields of human-robot interaction (HRI) and human-aware robot navigation, is the interactive restriction of a mobile robot's workspace. A mobile robot's *workspace* is the space that can be reached by the robot using its locomotion system. Since a mobile robot's workspace is typically only limited by physical borders, e.g. walls or furniture, the robot can freely operate in the entire environment. Although humans appreciate the services of mobile robots, there are scenarios in which humans want to restrict the workspaces of their mobile robots, i.e. they want to specify *restriction areas*. One reason for this demand is that people want to specify certain areas for working, e.g. they want a mopping robot to perform a spot cleaning in a dirty area or they want a lawnmower robot to only cut grass without driving across a flower bed. Another reason is that people want their mobile robots, which are often equipped with cameras, to not enter certain areas due to privacy concerns (APTHORPE *et al.*, 2018). For example, the acceptance of cameras in intimate rooms, such as bed- or bathrooms, is lower compared to living rooms (ZIEFLE *et al.*, 2011). Thus, humans need to exclude these rooms from the mobile robots' workspace. Furthermore, mobile robots should circumvent carpet areas to prevent them from getting stuck and provoke navigation errors (HAWES *et al.*, 2017). Other examples for restriction areas, that should be excluded from a mobile robot's workspace, are kids' corners to prevent a vacuum cleaning robot from vacuuming toy blocks or pets' water dishes to avoid a robot tackling it and spilling water on the floor.

All these scenarios can be summarized to the problem, that we deal with in this thesis: *the restriction of a mobile robot's workspace and change of its navigational behavior according to the humans' needs*. As illustrated in the different scenarios, each restriction area is defined by spatial information consisting of a boundary and an occupancy value. The boundary describes the shape of the restriction area and its location in the environment. The occupancy value indicates if the area should be excluded from or included into the mobile robot's workspace. These restriction areas cannot be completely recognized by a mobile robot. While the boundary of a restriction area can be de-

tected in some scenarios with simple algorithms, e.g. edge detection, it is not possible in general. Especially, it is not possible for restriction areas without expressive visual characteristics, e.g. privacy zones or dirty areas. Furthermore, the occupancy value cannot be inferred by the mobile robot or derived from the environment's geometry. For example, the decision whether a certain room is treated as a privacy zone or whether a carpet should be crossed (or not) depends on the human's decision and requires explicit knowledge of a human. Besides, the semantic of a restriction area, that could give a hint concerning the occupancy value, cannot be determined due to perceptual or computational limitations of the mobile robot, e.g. a kids' corner should usually not be intruded but the semantic cannot be inferred due to the computational complexity of the problem and the robot's limited computational power. In addition, if certain objects define a restriction area, e.g. toy blocks or pets' water dishes, these are flat and lightweight so that they cannot be recognized as physical obstacles by the robot's on-board sensors, e.g. depth sensor or bumper. Thus, a human has to provide the necessary information about a restriction area. For this purpose, an *interaction process* between human and robot allowing the transfer of this information is inevitable.

In this interaction process, a human has to convey spatial information about a restriction area to the mobile robot. This is challenging because it is difficult to transfer complex spatial information between a human and robot. Considering an interactive restriction of the workspace involving a human, it is also necessary to support the human during the interaction process, e.g. current progress of the spatial information transfer or the state and result of the interaction process. Hence, an interaction process also requires a feedback channel from the robot to the human to provide immediate feedback during the interaction process. This is challenging due to the complexity of the feedback information, e.g. spatial information indicating the result of the interaction process, and the limited feedback capabilities of mobile robots. Moreover, the interaction process has to be designed to be applicable by non-expert users and not exclusively by robot programmers. Therefore, an adequate interaction design considering this aspect is essential for a successful *user interaction*, i.e. the execution of an interaction process.

Following the positive trend of autonomous mobile robot deployments in human-centered environments, this problem has a high relevance because mobile robots respecting human needs will be a key factor for the acceptance of robots in these environments. If robots do not satisfy these needs, it will result in a dissatisfaction or even rejection of robots in human-centered environments. Furthermore, there are currently no sufficient solutions to this problem as we will describe in detail in Chapter 2. A reason for this lack of sufficient solutions is the immaturity of the research field of HRI, which is relatively young compared to robotics in general. Robots find their ways into human-centered environments, and humans have to inevitably interact with robots. It is a crucial question how to allow natural and robust interaction between humans and robots. Especially, transferring spatial information during interaction is a hard challenge.

Figure 1.1: Images of exemplary home environments consisting of free space, walls and physical objects (Images from ADE20K dataset (ZHOU *et al.*, 2019)).

1.2 Scope of the Thesis

The interaction process described in the previous section comprises three different components: (1) an environment, (2) a human and (3) a mobile robot. Since we cannot deal with all combinations of these components, we define the components and their characteristics, that we deal with in this thesis, in the following paragraphs.

The first component is the **environment**, which serves as setting and an actor of the interaction process. In this work, we consider two kinds of environments: (1) a traditional home environment and (2) a smart home environment. We focus on home environments because humans live there and mobile service robots start to pervasively find their ways into human home environments (HÄGELE, 2016). Hence, they are optimal examples for human-robot shared spaces. A traditional home environment is composed of multiple rooms or separated areas, such as corridors or a bed-, bath- or living room. The areas are either open rooms or separated by doors. Rooms are surrounded by walls and can contain physical objects on the ground, such as tables, chairs or plants. These physical objects occupy up to approximately 30% of the environment. While most of the (heavy) furniture, such as sofas, has a fixed place in the environment, light-weight objects, such as plants or chairs, can be moved. Images of some exemplary home environments are visualized in Figure 1.1. Additionally, there can be decorative or functional objects on the ground, e.g. (1) pets' water dishes, (2) carpets or (3) kids' corners characterized by toys or toy blocks as illustrated in Figure 1.2. These are typically static areas whose physical location changes only minimally over time. Other areas are temporary, e.g. dirty areas are only present until cleaning. These areas have in common that they can have arbitrary shapes and sizes. However, the boundary surrounding such an area is typically not longer than approximately 10 m.

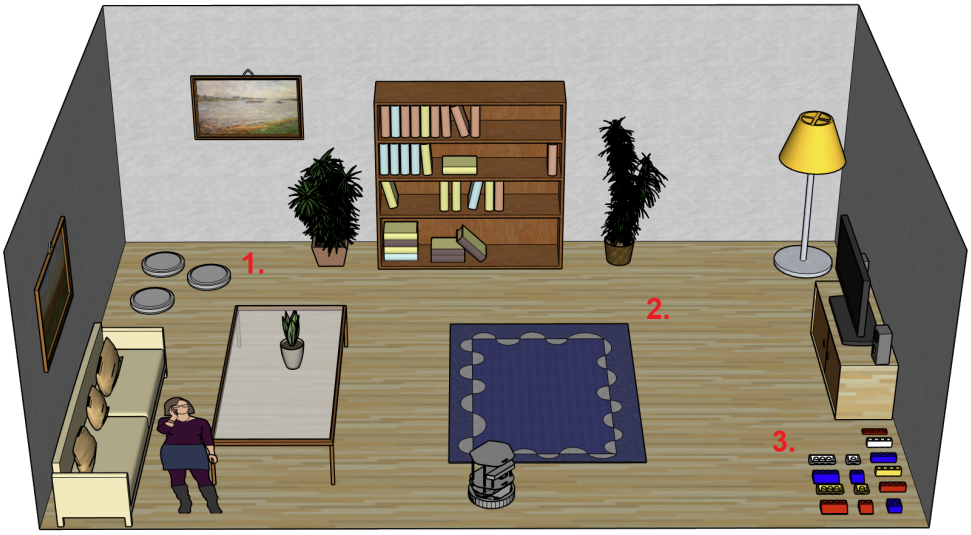


Figure 1.2: Exemplary scenarios for the problem showing (1) pets' water dishes, (2) a carpet and (3) a kids' corner.

In addition to traditional home environments, we consider smart home environments in this work because we currently witness a trend towards smart home environments (STATISTA, 2018). A smart home environment, as a certain smart environment, is a specialized form of a traditional home environment that incorporates additional embedded devices (AUGUSTO and NUGENT, 2006). These devices are connected to each other via a network, which allows the mutual exchange of data and the use of intelligent software for smart services, e.g. sensor-based reactions of the smart environment. Typical sensors are camera sensors integrated in the environment to capture images of the scene. However, only parts of the environment are typically covered by the cameras' fields of view to account for privacy concerns and due to occlusions. Thus, there is only a partial observation of the scene. Other examples for sensors are motion, sound, air quality or light sensors. In contrast to sensors, actuators interact with the environment and can change the state of it, e.g. door openers, lights or displays. A combination of actuator and sensor is a voice-controlled intelligent personal assistant that incorporates microphones, loudspeakers and software. For this purpose, the smart environment can also draw on resources, e.g. processing power or memory, from cloud services.

The second component of the interaction process is the **human**, which is the intended user of the system. A human's characteristics are derived from the specified environment in which the human lives. Home environments are not restricted to a certain user group, but the residents are typically non-experts, i.e. they only have a moderate experience with robots, and prefer an easy-to-install system correctly fulfilling its task instead of a highly complex and experimental system. In the context of this thesis, we assume a non-expert to be in the age group of 18-64 years, and we do not make restrictions regarding gender. According to STATISTA (2018), this age group covers all smart home users in major European countries with a proportion of female users ranging from 37.1% in the UK

to 43.8% in Spain. This statistic shows that we cover a large user group with our selection. Additionally, we assume a non-expert to be able to interact with common consumer devices, such as tablets or smartphones. Therefore, we restrict the user group to people with no cognitive impairments or upper limb disorders. These humans are residents of the environment and own a mobile robot to benefit from the robot's services, such as fetching objects, vacuum cleaning or social interaction.

The last component of the interaction process is a **mobile robot** as a central actor because it is the goal to restrict its workspace. According to CORKE (2017), there are different kinds of mobile robots depending on their environment, i.e. ground, air and water. In this work, we restrict our focus on mobile robots working on the ground plane, especially on wheeled robots (CHUNG and IAGNEMMA, 2016). In particular, we focus on robots that are able to rotate around their own vertical axis, such as differential-drive or omnidirectional mobile robots. We choose this category because wheeled robots represent the largest group of mobile robots used in applications (SICILIANO *et al.*, 2009). In addition to their locomotion system, mobile robots should be able to build a map of their environment and to accurately localize themselves in the environment (STACHNISS *et al.*, 2016), i.e. their pose consisting of a 2D position and orientation with respect to a map coordinate frame is known. For this reason, mobile robots are often equipped with additional sensors to perceive the environment, e.g. rotary encoders on their wheels or laser scanners. Furthermore, robots should have a front-mounted RGB-D camera to acquire color and depth images of their surroundings (HALMETSCHLAGER-FUNEK *et al.*, 2019). This can be used for mapping the environment or interaction with the human. In order to perform service tasks autonomously, the mobile robots should also have navigation capabilities including a local obstacle detection, i.e. the computation of a collision-free trajectory to a target location considering obstacles observed by the robot's sensors during motion execution (MINGUEZ *et al.*, 2016). To this end, they need on-board processing power, and we assume the robots to have computational capabilities comparable to a state-of-the-art embedded computing board or mid-range laptop without a graphics processing unit (GPU). Moreover, a mobile robot typically provides possibilities for non-speech audio sound, colored light feedback and physical interaction. These robot characteristics cover a large spectrum of today's mobile robots with representatives, such as TurtleBots (ACKERMAN, 2013), the humanoid robot "Pepper" (PANDEY and GELIN, 2018), the companion robot "Max" (SCHROETER *et al.*, 2013), the mobile manipulator "TIAGo" (PAGES *et al.*, 2016) or CoBots (VELOSO *et al.*, 2015). The mobile robots' operational status can be divided into two general modes: (1) autonomous mode, in which the robot provides services to humans in the environment, and (2) standby mode, in which the robot is inactive and typically parked at a certain position, e.g. at the charging station. During both modes, it is possible that residents are present in the environment. For example, a vacuum cleaning robot parks at a charging station, starts autonomous cleaning at a predefined time (independent of the presence of residents in the environment) and returns to its initial position for charging after finishing the cleaning service. Since a human can restrict a mobile robot's workspace at any time, a robot has to be aware of interaction intends of the human.

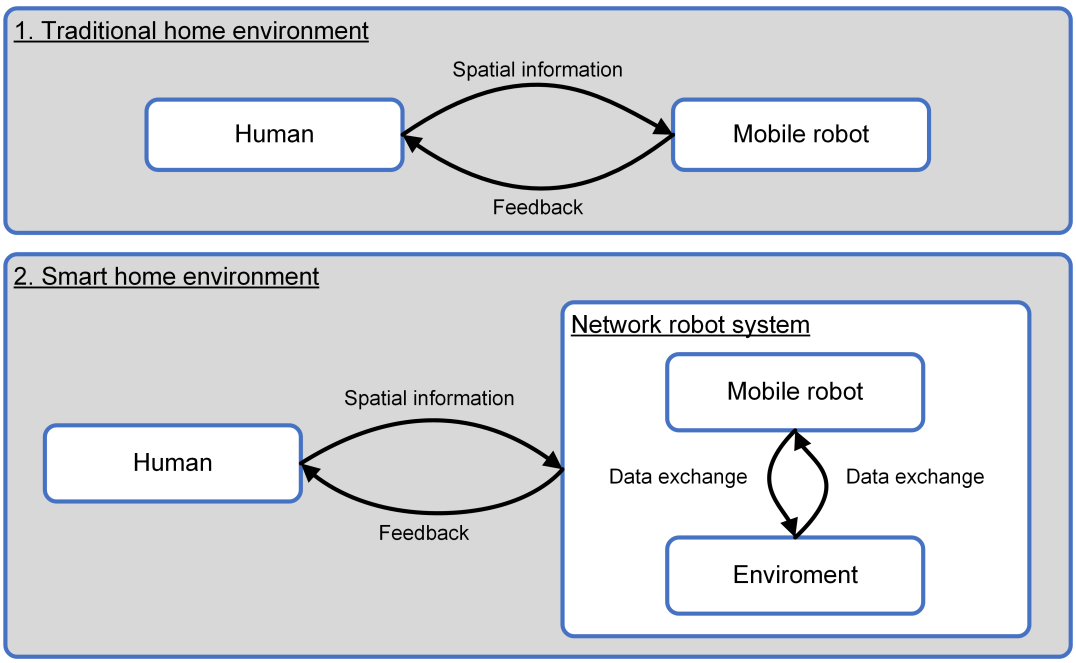


Figure 1.3: Components of an interaction process for both environments.

These three components, i.e. environment, human and mobile robot, are actors in the interaction process as shown Figure 1.3. In case of a traditional home environment, the interaction takes place between a human and a mobile robot. A human transfers spatial information to the mobile robot, that in turn provides feedback of the interaction process to the human. This setting comprises the challenges of the transfer of complex spatial information from human to robot and a feedback channel to inform the human about the state of the interaction process. In case of a smart home environment, the environment becomes an additional actor in the interaction process. Thus, the human does not directly interact with the mobile robot but instead communicates with a network robot system (NRS). This is an integrated system consisting of a mobile robot and smart home environment that share mutual data. In addition to the challenges above, this setting also includes the cooperation of mobile robot and smart environment in the interaction process as a challenge.

1.3 User Requirements and Quality Levels

In addition to the application challenges, there are also ambitious user requirements, that need to be fulfilled in an interaction process. These are derived from the problem's scope and deal with functionality, usability and user experience. Moreover, we define three quality levels for the requirements to clearly distinguish between *unacceptable* and *acceptable* solutions. A *good* solution constitutes an optimal or very hard to reach solution. If no good solution is defined for a requirement, it is the same as the acceptable solution. The requirements and quality levels are described below:

- **Correctness:** This requirement is derived from the general problem to restrict a mobile robot's workspace. In this context, correct means that the mobile robot considers the user-defined workspace and changes its navigational behavior after a successful interaction process. Thus, an acceptable solution should be correct as summarized in Table 1.1.

Table 1.1: Quality level description for the correctness.

Quality level	Description
Unacceptable	No change of navigational behavior after successful interaction
Acceptable	Change of navigational behavior after successful interaction

- **Flexibility:** Flexible means that arbitrary restriction areas can be defined by a human, i.e. the workspaces can have arbitrary shapes and sizes. Since different restriction areas with different shapes and sizes are described in the scope of this thesis, an acceptable solution should be flexible as summarized in Table 1.2.

Table 1.2: Quality level description for the flexibility.

Quality level	Description
Unacceptable	Restriction areas with certain shapes and sizes
Acceptable	Restriction areas with arbitrary shapes and sizes

- **Completeness:** The completeness is an aspect of the effectiveness, which is defined as the "accuracy and completeness with which users achieve specified goals" (ISO, 2018), i.e. how successful does the user accomplish the interaction process. We consider a solution as acceptable if an interaction process is performed successfully with a probability of at least 90% as shown Table 1.3. This is a high value underlining the challenging problem. Moreover, a solution is considered as good if 95% of the interaction processes are performed successfully.

Table 1.3: Quality level description for the completeness.

Quality level	Description
Unacceptable	$< 90\%$ success
Acceptable	$\geq 90\%$ success
Good	$\geq 95\%$ success

- **Accuracy:** The accuracy is another aspect of the effectiveness as indicated by the definition above. However, we split this requirement to distinguish between both aspects to emphasize

the importance of the accuracy. Some scenarios introduced in Section 1.1 require an accurately restricted workspace, i.e. the restriction areas should be exactly at the position where the human wants them to be. For example, a vacuum cleaning robot should move as close as possible along the boundary to a kids' corner or a mopping robot should accurately mop around a carpet. Therefore, it is important that the interaction process allows an accurate restriction of the workspace. An acceptable accuracy is defined in Table 1.4 and is achieved if the user-defined restriction area as the result of the interaction process intersects with the intended restriction area by at least 70%. A typical value is 50% as used in the PASCAL Visual Object Classes (PASCAL VOC) challenge to define correct object detections (EVERINGHAM *et al.*, 2010). However, we increase this threshold by additional 20% to emphasize the importance of the accuracy and make the problem more challenging. Furthermore, an accuracy of at least 80% is considered as good and indicates an extremely accurate solution.

Table 1.4: Quality level description for the accuracy.

Quality level	Description
Unacceptable	$< 70\%$ intersection
Acceptable	$\geq 70\%$ intersection
Good	$\geq 80\%$ intersection

- **Interaction time:** This requirement corresponds to the efficiency of the interaction process dealing with the "resources used in relation to the results achieved" (ISO, 2018). The interaction time is the time needed to restrict a mobile robot's workspace. This is an indicator for the usability of the interaction process, and thus it should be as efficient as possible. In this work, we consider an interaction time of 60 seconds as acceptable for a human as summarized in Table 1.5. This is a reasonable threshold for a challenging problem including the transfer of spatial information from human to robot. Moreover, this is a relatively short interaction time when considering static restriction areas that only change minimally over time, such as carpets or privacy zones. If the interaction time falls below a threshold of 30 seconds, we consider a good interaction time. This is half of the acceptable threshold and underlines the ambition of the requirement.

Table 1.5: Quality level description for the interaction time.

Quality level	Description
Unacceptable	> 60 seconds
Acceptable	≤ 60 seconds
Good	≤ 30 seconds

- **User experience:** Since non-expert humans play a major role in the interaction process, it should be applicable by non-experts and should account for their needs. First of all, we require the restriction of the workspace to be unobtrusive, i.e. no additional physical borders are used to restrict the workspace. This is a necessary condition because we consider an indoor home environment where it is not acceptable to install physical obstacles on the ground. In case of the user experience, there is no unified definition, but ALBEN (1996) gives a general definition of this term that summarizes several "aspects of how people use an interactive product: the way it feels in their hands, how well they understand how it works, how they feel about it while they're using it, how well it serves their purposes, and how well it fits into the entire context in which they are using it" (ALBEN, 1996). For our interaction process, we derive some specific aspects, such feedback capabilities, intuitiveness, comfort or satisfaction. We consider an acceptable user experience if a human's positive attitude outweighs the negative attitude concerning the user experience as shown in Table 1.6. Considering a bipolar scale ranging from negative (0%) to neutral in the middle (50%) to positive attitude (100%), an acceptable user experience is defined on the right side of the scale (> 50%). Additionally, a good user experience corresponds to a strong positive attitude (> 75%).

Table 1.6: Quality level description for the user experience.

Quality level	Description
Unacceptable	
	– Obtrusive or
	– Negative attitude (≤ 50%)
Acceptable	
	– Unobtrusive and
	– Positive attitude (> 50%)
Good	
	– Unobtrusive and
	– Strong positive attitude (> 75%)

- **Learnability:** The learnability refers to the ability of the interaction process to be accomplished by novice users. According to WEISS *et al.* (2009), this comprises principles like familiarity, consistency, generalizability, predictability, and simplicity. This is important because non-expert users generally do not know how to interact with a robot. The learnability is considered acceptable if the human's personal attitude concerning the learnability is positive (> 50% on a bipolar scale ranging from negative (0%) to neutral (50%) to positive (100%)) or if there is a continuous improvement with respect to the usability requirements (completeness, accuracy and interaction time) when repeating the interaction process. An exception is the case if the completeness, accuracy and interaction time reach an acceptable level when inter-

acting for the first time. In this case, we also consider the learnability as acceptable. A good learnability is reached if there is a strong positive attitude towards learnability (> 75%) and a continuous improvement with respect to the completeness, accuracy or interaction time. The descriptions of the quality levels are summarized in Table 1.7.

Table 1.7: Quality level description for the learnability.

Quality level	Description
Unacceptable	
	– Negative attitude (≤ 50%) and
	– No continuous improvement with respect to the completeness, accuracy and interaction time
Acceptable	
	– Positive attitude (> 50%) or
	– Continuous improvement with respect to the completeness, accuracy or interaction time
Good	
	– Strong positive attitude (> 75%) and
	– Continuous improvement with respect to the completeness, accuracy or interaction time

An acceptable solution to our problem fulfills all requirements with at least an acceptable quality level. If at least one of the requirements is unacceptable, the solution is also unacceptable. Concrete instruments to assess the requirements, e.g. questionnaires or time measurements, will be introduced in the corresponding evaluation sections of this thesis.

1.4 Objectives

After deriving user requirements for an interaction process and defining quality levels for them, it is the main objective to allow non-expert users the interactive restriction of a mobile robot's workspace and to change its navigational behavior. We subdivide the main objective into three objectives that build on each other:

Objective 1. This objective deals with the investigation of interaction methods and user interfaces for the restriction of a mobile robot's workspace. A *user interface* gives an opportunity for interaction between a human and robot, while an *interaction method* describes the way of how to employ the user interface in an interaction process to achieve the goal, i.e. the restriction of the mobile robot's workspace. It is the goal to identify promising user interfaces from other disciplines and to develop novel interaction methods employing the user interfaces. This interaction focuses on

the interaction between human and robot in a traditional home environment. Therefore, the interaction method does not rely on devices of a smart home environment. Moreover, there are two goals concerning the requirements with descending priority: (1) at least one of the proposed interaction methods should perform better than the current state-of-the-art solution and (2) should be an acceptable solution for our problem, i.e. achieving at least acceptable performance on all user requirements.

Objective 2. This objective deals with the investigation of the role of a smart home environment in the interaction process. The main idea is to extend at least one of the interaction methods from the previous objective to incorporate additional smart home components to improve the interaction process compared to the interaction without smart home components. It is the goal to reduce the interaction time by employing additional sensors of the smart home, which increase the mobile robot's perceptual abilities. Moreover, additional sensors and actuators should enable new interaction and feedback opportunities with the goal to improve the user experience. This interaction method should be an acceptable solution.

Objective 3. This objective deals with the investigation of learning capabilities. The goal is to learn from multiple user interactions and to support the human in future interaction processes through recommendations for interactions. For this purpose, we build on findings from the previous objectives, i.e. an adequate user interface and algorithms for the incorporation of smart home components into the interaction process. This aims to reduce the interaction time to a good quality level while preserving the quality levels of the other requirements, i.e. at least acceptable quality levels.

Finally, it is the goal to prototypically implement the interaction methods and to empirically evaluate them with non-expert users. The aim of this experimental evaluation is to test the performance of the interaction methods with regard to the user requirements and in comparison to a state-of-the-art baseline method.

1.5 Thesis Outline

Chapter 2 introduces necessary background knowledge and gives an overview of related works dealing with autonomous robot capabilities, robot motion restriction, user interfaces for HRI and intelligent environments. The chapter concludes with an assessment and a classification of existing approaches in relation to the objectives of the thesis. Based on this assessment, we identify a research gap and formulate three open research questions, that need to be answered to achieve the objectives. These research questions form the basis for the following chapters.

Chapter 3 starts with a definition of the problem setting. This is the basis for the introduction of the notation of a virtual border as a data structure to flexibly model a restriction area. This is a non-physical border, that is not directly visible to a human, but that is respected by mobile robots during

navigation. For this purpose, we develop an algorithm that integrates a virtual border into a given map of the environment. The resulting map serves as basis for a navigational costmap and enforces a mobile robot to change its navigational behavior. In order to allow non-expert humans to specify such a virtual border, we propose two interaction methods based on (1) mediator-based pointing gestures using a common laser pointer and based on (2) an augmented reality (AR) application running on a RGB-D tablet. Both interaction methods rely on pure HRI without incorporating devices from a smart home environment. Finally, our proposed interaction methods are empirically evaluated and compared with a baseline method regarding the requirements.

Based on the results of the previous chapter, **Chapter 4** investigates the role of a smart home environment in the interaction process. To this end, we propose an interaction method based on a laser pointer that incorporates the smart home environment, especially its heterogeneous sensors and actuators, as additional actor in the interaction process. It comprises an architecture that integrates a mobile robot into a smart home with the purpose of supporting the interaction process in terms of interaction time and user experience. Furthermore, the interaction method involves the cooperation of stationary and mobile cameras to perceive laser spots and an algorithm allowing the extraction of virtual borders from multiple camera observations. We experimentally evaluate the interaction method with regard to the requirements and compare the results with the interaction method from the previous chapter (without support of a smart home environment).

The developed algorithm from the previous chapter, i.e. the extraction of virtual borders from multiple cameras, is used as a part of a learning and support system (LSS), that is proposed in **Chapter 5**. This system learns from previous interaction processes and supports a user in future interaction processes through recommendations for virtual borders. It is based on semantic scene understanding performed on images acquired from cameras integrated in the smart home environment and a frequent itemset mining approach. Recommendations for virtual borders are conveyed to a human through the AR interface developed in Chapter 3. The subsequent experimental evaluation tests the benefits of the system's learning capabilities with respect to the requirements.

Finally, we conclude our work and summarize the contributions in **Chapter 6**. Additionally, we describe current limitations of our solutions and point out work for the future.

1.6 Contributions

This thesis comprises several contributions, that have been previously published. In all publications listed below, I did most of the work including conceptualization, realization, evaluation and paper/article writing. In case of patent applications, I did not write the patent. Due to acknowledgement of co-authors, the first person plural, e.g. *we* or *our*, is used in this thesis instead of the first person singular, e.g. *I* or *my*, when talking about the contributions. The relevant publications and their contributions to the overall objective are listed below:

- SPRUTE, D., R. RASCH, K. TÖNNIES, and M. KÖNIG (2017). A framework for interactive teaching of virtual borders to mobile robots. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1175–1181:
  We introduce the notation of a virtual border as a data structure to model restriction areas. In order to specify a virtual border and allow humans the restriction of their mobile robots' workspaces, we propose a framework for interactive teaching of virtual borders based on robot guidance. A reference implementation using visual markers validates the approach by showing its correctness.

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019c). This far, no further: Introducing virtual borders to mobile robots using a laser pointer. In *IEEE International Conference on Robotic Computing (IRC)*, pp. 403–408:
  We expand the previously introduced notation of a virtual border by considering different types of virtual borders to account for the flexibility requirement. Furthermore, we propose our first interaction method based on a laser pointer and the robot guidance framework to address the user requirements of an interaction process.

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2018). Virtual borders: Accurate definition of a mobile robot's workspace using augmented reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8574–8581:
  As an alternative to the mediator-based pointing gesture interaction using a laser pointer, we propose a second interaction method allowing a human to specify a virtual border using AR on a RGB-D tablet.

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019b). A study on different user interfaces for teaching virtual borders to mobile robots. *International Journal of Social Robotics 11*(3), 373–388:
  We contribute the results of a comprehensive user study on teaching virtual borders considering several interaction methods. Both proposed interaction methods are compared with a state-of-the-art interaction method regarding the requirements. The experimental results show that the AR-based interaction method outperforms the others and that the laser pointer interaction method achieves mostly acceptable results, but also has potential for improvements, e.g. in terms of interaction time and user experience.

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019a). Interactive restriction of a mobile robot's workspace in a smart home environment. *Journal of Ambient Intelligence and Smart Environments 11*(6), 475–494 and
  KÖNIG, M. and D. SPRUTE (2019). Verfahren und Robotersystem zur Eingabe eines Arbeitsbereichs. DPMA Patent DE102018125266B3:
  In order to address the shortcomings of the laser pointer approach revealed in the user study, we propose an interaction method based on a laser pointer in combination with a smart home

environment. This comprises the integration of a mobile robot into a smart home environment, the cooperation of stationary and mobile cameras to perceive laser spots and an algorithm for the extraction of virtual borders from multiple camera observations. The results demonstrate that additional components of a smart home environment can improve the interaction process in terms of interaction time and user experience.

- SPRUTE, D., P. VIERTEL, K. TÖNNIES, and M. KÖNIG (2019). Learning virtual borders through semantic scene understanding and augmented reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4607–4614 and
  KÖNIG, M., D. SPRUTE, and P. VIERTEL (2020). Verfahren und Robotersystem zur Eingabe eines Arbeitsbereichs. DPMA Patent DE102019126903B3:
  We contribute a learning and support system based on semantic scene understanding and a frequent itemset mining approach. The system learns from multiple user interactions and creates recommendations for interactions to support a human in future interaction processes. To this end, we build on the previous contributions by leveraging the cooperation of multiple smart home cameras to create recommendations from multiple camera views and by employing the AR user interface to convey the recommendations to the human. Experimental results show that learning capabilities can significantly reduce the interaction time while preserving the results of the other user requirements.

2

Background and Related Work

The main objective of this thesis is to allow non-expert humans the interactive restriction of their mobile robots' workspaces in a (smart) home environment respecting the highlighted requirements in Section 1.3. In order to clearly understand the objective and solutions to this problem, some additional background knowledge is necessary. For this reason, we first give an overview of autonomous robot capabilities covering topics from map representations, mapping and localization, robot navigation and interaction capabilities. This section is intended to explain how maps of physical environments are structured and created, how mobile robots can localize themselves in a map of the environment, how robot path planning and obstacle avoidance work and how robots can interact with humans using their on-board components. It is especially important to understand what state-of-the-art mobile robots are capable of and how to intervene in the navigational framework to change a mobile robot's navigational behavior. Subsequently, we present works that restrict a mobile robot's motion or workspace. Here we distinguish between implicit approaches known from the research field of human-aware robot navigation and explicit approaches employing methods from the field of human-robot interaction (HRI). In order to identify alternative user interfaces for our first objective, the following section covers user interfaces employed in related HRI applications, especially visual displays and gestures. The second objective of this thesis deals with the incorporation of devices from a smart home environment in the interaction process. Therefore, the next section summarizes related works from the area of intelligent environments, which comprises the fields of smart environments and ambient intelligence. As a consequence, the integration of robots into intelligent environments results in the research field of network robot systems, that we introduce subsequently. We show how sensors and actuators of a smart environment can be incorporated into a system to enhance robot applications and interaction with humans. The last objective targets to investigate learning capabilities by learning from user interactions. For this purpose, we dedicate a subsection to works related to learning in intelligent environments. Finally, based on our objectives stated in Section 1.4 and the contributions of related works to the objectives, we point out a research gap and derive open research questions as basis for the remainder of this thesis.

2.1 Autonomous Robot Capabilities

Autonomous robots are robots that perform service tasks with a high degree of autonomy. For this purpose, robots need certain capabilities, such mapping, localization and navigation capabilities (BEKEY, 2005). This section is intended to give background knowledge in these fields to understand how a robot internally stores and builds a representation of its environment, how it determines its position and orientation with respect to the environment and how robot motion planning works. This is important to understand how a mobile robot's workspace can be modelled and restricted and how to intervene in the robot's navigation framework to change its motion behavior. Moreover, important sensors necessary for these capabilities are revealed, and communication channels for the interaction with humans are discussed.

2.1.1 Map Representations

In order to allow mobile robots to autonomously navigate in an environment and provide services, they rely on an internal model of the environment. Such a model is a map representation containing a geometric representation of the physical environment including free and occupied spaces. There is a wide variety of different map representations due to different areas of application, e.g. robot pose estimation or navigation. FUENTES-PACHECO *et al.* (2015) give an overview of different map representations divided into metric and topological maps. Metric maps preserve the geometric properties of the environment, while topological maps are more abstract describing the connection between several positions in the environment.

Metric maps can be further subdivided into occupancy grid maps (OGMs) and landmark-based maps. OGMs model the environment by means of cells containing a status for the occupancy of the corresponding area, i.e. a place is modelled as free or occupied. Originally developed by MORAVEC and ELFES (1985), they are especially popular in 2D mapping and navigation due to their discretized representation of the environment preserving most of the spatial information. Moreover, the detailed spatial information allows an accurate localization of a mobile robot in the environment. An extension to this binary representation are coverage maps (STACHNISS and BURGARD, 2003). These represent an area by an occupancy probability, which makes the representation of the environment more accurate in case of low-resolution grid maps[1]. However, both can be memory consuming depending on the size of the environment and the resolution of the map. An alternative to this dense representation, are feature or landmark-based maps that only consist of coordinates of salient features in the environment (CHONG and KLEEMAN, 1999). Thus, they are memory-saving and scale well with large environments. A drawback is its dependence on adequate landmarks and their density. Furthermore, they are not optimal for path planning because a missing landmark does not

[1] Although coverage maps are an extension of OGMs, the terms are typically used as synonyms.

imply a free space (WALLGRÜN, 2010). Nonetheless, they can be used for path planning in open environments when there are no obstacles between the landmarks.

As opposed to the representation as metric maps, topological maps represent the environment in a graph-like format listing unique places as vertices and paths between places as edges (KORTENKAMP and WEYMOUTH, 1994). For example, a partially enclosed area, such as a room in an indoor environment, can be modelled as vertex, while neighboring areas are connected by edges (BLOCHLIGER *et al.*, 2018). This simple and compact abstraction of the environment yields a lower memory consumption and better scalability compared to metric maps. However, they are not suitable for tasks requiring a high accuracy due to their simplicity (GARCIA-FIDALGO and ORTIZ, 2015).

Since all these basic map representations have their strengths and weaknesses, they are also combined resulting in hybrid representations gaining advantages from the others and eliminating shortcomings (BUSCHKA and SAFFIOTTI, 2004). For example, THRUN *et al.* (1998) combine metric and topological maps to take advantage of the lower complexity of topological maps and the higher resolution of metric maps, and CHEN *et al.* (1997) combine multiple maps of the same type but with different resolutions. More details on spatial representations, that are not directly relevant for this thesis, and their organization are described by WALLGRÜN (2010).

2.1.2 Mapping and Localization

A map representation is the underlying data structure for higher-level robotic algorithms, such as mapping and localization. STACHNISS (2009) gives a concise description of both terms and how they mutually depend each other: mapping is the process of creating a map representation of the environment by integrating information from the robot's sensors, while localization is about determining the position and orientation of a robot, i.e. the pose, relative to a map coordinate frame. The author also points out the overlap of both areas which is known as simultaneous localization and mapping (SLAM), i.e. the creation of a map and the simultaneous localization of a robot inside the map given continuous sensor measurements.

STACHNISS *et al.* (2016) describe three paradigms to solve the SLAM problem: extended kalman filters (EKFs), particle filters and graph-based optimization techniques. The earliest solution to the SLAM problem was the EKF formulation by SMITH *et al.* (1990), which uses a multivariate Gaussian to represent an estimate of the robot's pose. However, its computational costs and inconsistent maps lowered the popularity of this paradigm. A more popular alternative is the use of non-parametric statistical filtering techniques known as particle filters. MONTEMERLO *et al.* (2002) proposed the FastSLAM algorithm that models a robot's pose as a set of particles where each particle represents a hypothesis for the current pose of the robot. The algorithm, especially its improved form FastSLAM 2.0 (MONTEMERLO *et al.*, 2003), allows computational efficient pose updates, which makes it the basis for several state-of-the-art SLAM algorithms in mobile robotics. The third

paradigm makes use of a graphical representation and non-linear sparse optimization techniques. LU and MILIOS (1997) developed a first working solution to the SLAM problem by building a graph of landmarks and robot poses as nodes. While the first two paradigms are online solutions, the graph-based approach is an offline solution that solves the full SLAM problem calculating the posterior probability over the entire robot's path. In contrast to this, online SLAM calculates a posterior probability for the map and the current robot's pose given the measurements and relations between the robot's pose up to a certain time. For an in-depth tutorial covering details of SLAM, that are out of this thesis' scope, we refer to the two-part tutorial (DURRANT-WHYTE and BAILEY, 2006) and (BAILEY and DURRANT-WHYTE, 2006). Moreover, the textbook by THRUN *et al.* (2005) dedicates several chapters to the problem, and CADENA *et al.* (2016) give a view into the future of SLAM.

In order to get sensor measurement updates for a SLAM algorithm, there are different kinds of sensors available which are classified by FUENTES-PACHECO *et al.* (2015) as exteroceptive and proprioceptive sensors. Exteroceptive sensors like sonars, range lasers, GPS and cameras can measure up to a certain distance and are noisy, while proprioceptive sensors comprise accelerometers, gyroscopes and wheel encoders that are used in dead reckoning to incrementally estimate the robot's pose. Since laser range sensors are expensive and proprioceptive sensors suffer from cumulative errors due to noise, there is a trend towards cameras as standalone sensors for solving the SLAM problem (YOUSIF *et al.*, 2015), which is then referred to visual simultaneous localization and mapping (VSLAM). Cameras can retrieve depth information while providing information about the environment's color and texture. Besides, cameras are more energy-efficient and less cost-intensive, which makes their deployment popular. One of the first works on VSLAM was conducted by DAVISON (2003) using a single monocular camera and extracting visual features from the images. Since then, several works have been proposed using vision as only exteroceptive sensor (TAKETOMI *et al.*, 2017). In particular, RGB-D sensors play a major role in recent developments because of the possibility to obtain direct range measurements and dense images simultaneously.

Another reason for the advent of VSLAM approaches is that cameras can provide odometry information without the use of proprioceptive sensors, which is referred to visual odometry (VO) (NISTER *et al.*, 2006). It is the process of estimating the egomotion of a robot moving through an environment and using an attached camera as the only input sensor for the estimation. The name was introduced by NISTER *et al.* (2004) following the naming of wheel odometry, that integrates the wheel turns over time to estimate the egomotion. VO tries to estimate the 3D motion of the camera in two consecutive camera frames and to calculate the new camera pose using the previously calculated transformation between the camera frames. This is a sequential process that updates the robot's pose as soon as a new camera frame arrives. Since VO only cares about a locally consistent path with respect to an initial pose, it is an essential component in a VSLAM system to obtain the robot's (camera) path. Main steps of a VO's processing pipeline comprise feature detection and matching in two consecutive camera frames, motion estimation in the presence of outliers and camera pose optimization (SCARAMUZZA and FRAUNDORFER, 2011) (FRAUNDORFER and SCARAMUZZA, 2012).

These works show that today's mobile robots can robustly build maps of their environment and localize themselves when considering structured indoor environments and accurate sensor measurements. In order to acquire accurate sensor measurements, laser range scanners and recently also cameras are employed as basis for mapping and localization. This supports our assumption concerning the equipment of a mobile robot with a camera.

2.1.3 Robot Navigation

Mobile robot navigation is a robot's capability to autonomously reach a certain goal position given knowledge about its environment and sensor measurements. The set of reachable positions is defined as the mobile robot's workspace. Navigation depends on other robot capabilities we introduced in the previous subsection, such as creating a map representation of the environment and determining the robot's pose inside the environment. The robot's trajectory to the goal position, which is the result of a navigation algorithm, should be efficient and reliable, i.e. it should be the shortest path from the current to the goal position while avoiding obstacles in the environment. Thus, navigation is an essential capability for autonomous mobile robots to provide services and enable a long-term autonomy. SIEGWART *et al.* (2011) describe two key components of a robot navigation system that complement each other: (1) path planning and (2) obstacle avoidance.

Path planning calculates a trajectory from the robot's current to a goal position given a map of the environment. Hence, it is a strategic problem-solving competence allowing the robot to decide how to achieve its goal. To this end, the first step in a path planning system is to create a discrete map representation of the environment, that may be transformed from a continuous representation. There are basically two kinds of planners that differ in how they use the discrete decomposition of the environment: (1) graph search and (2) potential field planners. Graph search planners first construct a connectivity graph that serves as basis for a subsequent graph search. A popular graph construction technique is the approximate cell decomposition due to the popularity of grid representations, such as OGMs. Other approaches are visibility graphs and Voronoi diagrams. (SIEGWART *et al.*, 2011) After constructing a connectivity graph employing a decomposition technique, the best path can be determined by a graph search algorithm, e.g. depth-/breadth-first search, Dijkstra's or A* algorithm. In contrast to graph search, potential field planners impose a mathematical function on the free space with a gradient indicating the direction to the goal. (KLANCAR *et al.*, 2017) This kind of path planning is often performed offline, i.e. a path is calculated based on the current state of the environment and not adapted to state changes during execution of the plan (BUNIYAMIN *et al.*, 2011). Such a planning algorithm is also known as global planner.

However, during the execution of the plan resulting from the global path planner, a mobile robot has to react to unforeseen obstacles in its local environment, such as moving people or objects not modelled in the map of the environment. Since a global planner only considers obstacles that are

known in advance, state-of-the-art navigation systems also incorporate an additional component, i.e. a local planner for obstacle detection. According to SIEGWART *et al.* (2011), it focuses on the changing of the robot's trajectory based on recent sensor measurements of the robot's environment. Among others, the authors describe some popular local planners, such as Bug algorithms (LUMELSKY and SKEWIS, 1990), vector field histograms (BORENSTEIN and KOREN, 1991) or dynamic window approaches (FOX *et al.*, 1997). Furthermore, they provide a comprehensive comparison of the most popular obstacle avoidance algorithms considering evaluation criteria, such as the robot's characteristics, sensors and performance metrics. These works demonstrate the maturity of this research field, which enables an efficient and reliable robot navigation in well-structured environments.

2.1.4 Interaction Capabilities

Another capability of an autonomous robot is its capability to perceive and interact with the environment. The previous subsections already revealed cameras as important sensors to perceive the state of the environment. However, mobile robots can also employ simple on-board actuators to interact with the environment or humans. Considering mobile robot's in the scope of this thesis, there is a limited set of communication channels allowing a robot to convey information to a human. As stated in Section 1.2, our considered mobile robots have only opportunities for colored light feedback and non-speech audio sound. Thus, there are basically two opportunities to convey non-verbal feedback to the user.

The first opportunity deals with the use of colored light. This is a quite simple but expressive feedback mechanism to transfer simple information, such as status information in various electronic devices (HARRISON *et al.*, 2012). For this purpose, several periodic and non-periodic light behaviors can be realized on a single light source with fixed color, such as blinking or bright flashes. Moreover, this idea can be extended to multiple light sources and multiple colors, e.g. a light signaling pattern (CHA *et al.*, 2017) or a colored light strip (BARAKA *et al.*, 2016). The main idea behind these approaches is that colored light and light behaviors can convey internal status information of a robot to a human. Therefore, PÖRTNER *et al.* (2018) conducted an online survey exploring how the light color, that is emitted using an attached colored LED strip on a mobile robot, affects the intelligibility of robot status information. The results show that colored light signals are suitable to provide simple status feedback in different HRI scenarios. Similarly, BARAKA and VELOSO (2018) introduce the use of colored lights as a non-verbal communication channel to express the dynamic robot state. Their mapping from a robot's internal state to a light animation space helps humans to better understand the robot's state and actions. This color-semantic mapping is the foundation for several deployments in HRI scenarios, e.g. SZAFIR *et al.* (2015) use an LED ring on an aerial drone to communicate the motion direction to a human, MONAJJEMI *et al.* (2016) provide feedback about the intent of an unmanned aerial vehicle (UAV) through an LED strip mounted on the front side of the

drone and COLLINS *et al.* (2015) employ pulsating patterns of colored light to express a companion robot's affective state.

The second opportunity for interaction employs non-speech audio sound, which is another simple and low-cost solution. For this purpose, speakers are used to generate sounds with varying intensity and period. These sounds are in general used as warning signals, e.g. HO *et al.* (2007) use auditory in-car warning signals to inform a driver of potential collisions, or to indicate internal states of a robot, e.g. KIM *et al.* (2009) employ simple beep tones to indicate different types of malfunctions with a focus on vacuum cleaning robots. However, due to its simplicity, non-speech audio sound is rarely used as single communication channel. Instead it is quite popular in combination with colored light, e.g. for alerting humans (CHAN and NG, 2009). Another application area is addressed by CHA and MATARIĆ (2016), who use sound in combination with color to enable a robot to request help in a human-robot collaborative task. To this end, they employ a simple beep sound and flashing colored light. Moreover, SONG and YAMADA (2017) also combine sound with color and vibration feedback to express the affective state of a social robot. Their results suggest that sound can be used as single communication channel to convey certain emotions, e.g. a falling beep sound for a sad emotion, and that combinations of these three communication channels can effectively provide feedback to a human.

These works show that colored lights and non-speech audio sound can be used to convey simple information to humans, such as a robot's internal state or emotion. The signal behaviors are designed to be understood by non-expert users, e.g. green and red colored lights are associated with a successful and an erroneous status. However, in order to give feedback of more complex information, such spatial information, these communication channels are not sufficient.

2.2 Robot Motion Restriction

The previous section showed the maturity of today's autonomous robots, e.g. the SLAM problem is solved for structured indoor environments given adequate range sensors (CADENA *et al.*, 2016) and mobile robots are able to efficiently and reliably navigate in environments for a long time (BISWAS and VELOSO, 2016). However, mobile robots are nowadays not only expected to safely and robustly navigate in the environment, but also in a human-aware way considering the presence of humans and their needs (SISBOT *et al.*, 2010). This affects the mobile robot's position, velocity and acceleration. Thus, robot navigation is restricted by further constraints, in addition to physical obstacles, which limit the workspace of a robot. For this purpose, CHIK *et al.* (2016) give an overview of common navigation frameworks and explain how to integrate social costs into a navigation framework. Additional to a global and local planner as introduced in the previous section, they suggest feeding the global planner not only with a map of the environment but also with social costs. These social costs are manifold and can include abstract concepts, such as object paddings (SVENSTRUP *et al.*,

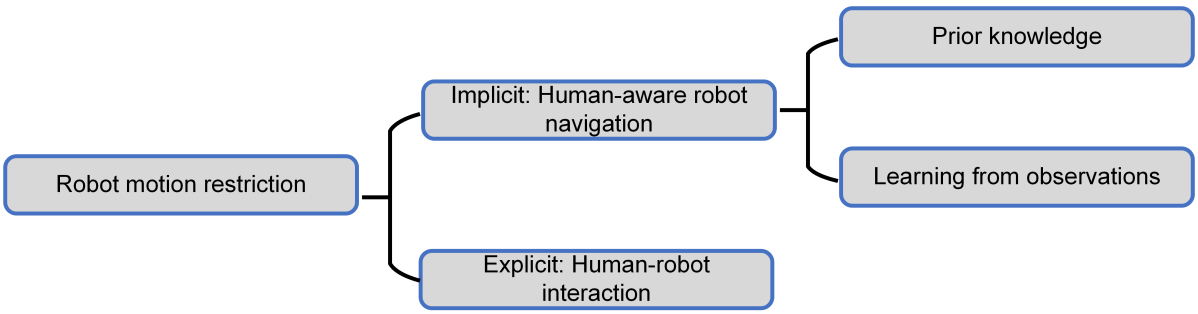


Figure 2.1: Taxonomy of robot motion restriction approaches.

2010), occlusions of objects (CHUNG *et al.*, 2009) or social conventions of how to pass a person (PACCHIEROTTI *et al.*, 2005) or approach a person (AHN *et al.*, 2018). In order to define these social costs and change the robot's navigational behavior, we distinguish between two different categories as shown in Figure 2.1: (1) the implicit adaption of the mobile robot's navigational behavior based on prior knowledge and learning from observations known from the field of human-aware robot navigation and (2) explicit methods from the field of HRI allowing a human to interactively restrict the mobile robot's motion. Implicit and explicit refers to the way how a human can affect the mobile robot's navigational behavior. While a human can only influence the navigational behavior implicitly in the former category, a human can explicitly define how the mobile robot should change its navigational behavior in the latter category. This section aims to introduce both categories and to point out relevant works that pursue the same objective, i.e. the restriction of a mobile robot's motion and change of its navigational behavior.

2.2.1 Implicit: Human-Aware Robot Navigation

The first category of works comprises approaches from the research fields of socially-aware and human-aware robot navigation. Both terms are interchangeable and focus on the same topic, i.e. the development of mobile robots that act according to identified social conventions to enable a comfortable interaction between humans and robots (RIOS-MARTINEZ *et al.*, 2015). This research is on the intersection between HRI and robot motion planning. According to an exhaustive literature search by KRUSE *et al.* (2013), human-aware robot navigation has its origins in the years around 2005 as a consequence of a change of robots' scope from industrial robots in factories to autonomous robots in human-centered environments. The authors identified three categories of properties to enable a human-aware robot navigation, i.e. comfort, naturalness and sociability.

Comfort is the ability to move in the environment without annoying or stressing surrounding humans. This does not only include to navigate safely, as known from traditional robot navigation, but also in a way that a human feels save (KRUSE *et al.*, 2013). The research in this field aims to

reduce human's discomfort, that is caused by several aspects, e.g. if a mobile robot passes a human too fast or to close. A popular concept, that addresses the issue of comfort distance, is the virtual personal space around a person introduced by HALL (1966). The author proposes proxemic interpersonal distances for different levels of comfort, e.g. intimate, personal, social or public distance. This prior knowledge can be incorporated into a robot's motion behavior to respect these distances, e.g. SISBOT *et al.* (2007) developed a human-aware motion planner incorporating the distance to a human as a criterion, LINDNER (2015) models personal space as affordance spaces attached to humans and MEAD and MATARIĆ (2017) present a computational framework of proxemics. Another technique, that goes beyond simple comfort distances, is to minimize the probability to encounter humans and avoid disturbing them (TIPALDI and ARRAS, 2011). This is accomplished by learning and modelling human activity events in a probabilistic spatio-temporal map. Others works in this category encompass an effective human comfortable safety framework (TRUONG and NGO, 2016) or the adaption of the robot's motion according to human trajectories (ALEMPIJEVIC *et al.*, 2013).

The second property of a human-aware navigation is the naturalness that deals with human-like motions, which can be achieved by adequate dynamics and velocities (KRUSE *et al.*, 2013). This research tries to reduce the difference in motion between human and robot to make the robot's motion more predictable and understandable. A typical example for this category is following a person in a natural way, i.e. following the direction instead of the path of a person (GOCKLEY *et al.*, 2007) or considering different situations during following (ZENDER *et al.*, 2007). Another branch of works in this category focuses on how to approach people, e.g. ALTHAUS *et al.* (2004) developed methods to approach a group of people without disturbing them, YAMAOKA *et al.* (2010) established a positioning model to enable robots to appropriately present objects to people and RAMÍREZ *et al.* (2016) implemented a navigation planner incorporating social costs to allow a robot to approach a person from the front. Other works in this category comprise walking side-by-side with people (FERRER *et al.*, 2017) or navigating in crowded environments (STEIN *et al.*, 2013).

Sociability is the last property mentioned by KRUSE *et al.* (2013), that adapts the robot's motion according to high-level cultural conventions. This distinction is important because a mobile robot can navigate in a natural way and can respect a person's comfort zone, but it can still violate social conventions. An example is standing in line that requires the robot to join the line at the right position, i.e. the end of the line, and adapting its position keeping the personal space to the front person (NAKAUCHI and SIMMONS, 2002). Another example for a social convention is switching to a certain side when two persons approach in the opposite direction in a narrow hallway. For this purpose, KIRBY *et al.* (2009) model one side of a person with higher social costs allowing the mobile robot to prefer a certain side when encountering a person, e.g. they implemented the social costs to pass a person on the right side. A similar example is to wait until the encountering person passed the narrow passage (TRINH *et al.*, 2015) or to pass a person on the left side when overtaking it, which is implemented as a social rule by PANDEY and ALAMI (2010). Furthermore, VEGA *et al.* (2019) model

groups of people as spatial density functions. This allows a robot to circumvent a group of people instead of crossing and disturbing the group.

These examples show that it is possible to change a mobile robot's navigational behavior with the goal to make it aware of humans. For this purpose, the costmap used for navigation, which is traditionally based on an OGM of the environment, is manipulated. In human-aware robot navigation, additional costs are integrated into this costmap to achieve the desired navigational behavior. These social costs can be either handcrafted employing prior knowledge, e.g. a human's personal space (MEAD and MATARIĆ, 2016), or can be learned from observations, e.g. human motion patterns (O'CALLAGHAN *et al.*, 2011). Although this effectively restricts a mobile robot's workspace and changes its navigational behavior, it is not possible to incorporate knowledge from a human in an interactive way, which is a challenge of the problem we address. Therefore, we can leverage the knowledge of how to integrate additional social costs into a robot's navigation framework, but we have to additionally focus on methods from HRI to allow humans the interactive restriction of mobile robots' workspaces.

2.2.2 Explicit: Human-Robot Interaction

In contrast to works in the previous category, this category focuses on approaches allowing a human to restrict a mobile robot's workspace in an explicit way. The most popular example is probably the workspace restriction of a robotic lawnmower to indicate the boundaries of the garden or to adjacent flowerbeds. For this purpose, the human places a wire around the area to be restricted (PRASSLER *et al.*, 2016). This is connected to a power source and can be sensed by the lawnmower robot's inductivity sensor if it is in the proximity of the wire. Thus, the robot turns around and does not cross the wire, which effectively restricts its workspace. A similar solution is available for vacuum cleaning robots where users place magnetic strips in the indoor environment to indicated areas to be avoided by the robot (NEATO, 2017). Beacon devices are an alternative to these wired solutions (CHIU, 2011). These devices are battery-powered and can be placed in the robot's workspace. They emit an infrared light beam, that is sensed by a mobile robot and that is treated as an obstacle. Thus, the typical obstacle avoidance behavior is evoked by these "virtual walls", and the robot does not cross the light beam. Since beacon devices emit light beams, either as a line or as a circle around the device, their flexibility is unacceptable. Moreover, wired solutions and beacon devices feature an unacceptable user experience since the physical placement of additional components in the environment is obtrusive. The state-of-the-art solution for today's home robots is to allow humans to sketch restriction areas on an OGM of the environment on a mobile device, e.g. a smartphone or tablet (ACKERMAN, 2017). Similarly, WILDE *et al.* (2018) restrict a mobile robot's workspace in an industrial environment by specifying additional spatial and temporal constraints on the robot's motion, such as avoidance areas. Although these solutions are flexible

and unobtrusive, as opposed to the previous ones in this section, we hypothesize that these solutions have drawbacks concerning accuracy and user experience. The reason for this is that it is hard for a (non-expert) user to establish correspondences between points in the environment and on the OGM shown on a display. This problem is also observed by GROMOV *et al.* (2019), who describe the transformation linking between the map's and human's reference frame as a mental rotation problem. An alternative, that addresses this problem, is proposed by SAKAMOTO *et al.* (2009) who use top-view cameras integrated into the environment to show a live video stream on a tablet. The human can then directly sketch the workspace of a robotic vacuum cleaning task on the tablet's screen. A similar work for the control of a vacuum cleaning robot's workspace is proposed by SEIFRIED *et al.* (2009). However, these solutions only work when the whole environment is covered by the cameras' fields of view and if there are no occlusions. Thus, it is not applicable for our problem.

In contrast to the implicit approaches from the previous subsection, there is only limited work on the explicit interaction to allow humans the restriction of mobile robots' workspaces. Moreover, the presented solutions have not been systematically evaluated regarding any of our identified requirements as introduced in Section 1.3. However, we conclude that the sketch interface on a graphical user interface (GUI) (ACKERMAN, 2017) is the most promising solution because the other ones are intrusive, power-consuming and/or inflexible. For these reasons, the sketch interface has prevailed in domestic robot applications over the others. Nonetheless, we argue that there are more suitable user interfaces for our problem that can outperform the current state-of-the-art solution with respect to the requirements.

2.3 User Interfaces for Human-Robot Interaction

In order to identify alternative user interfaces for our first objective, a user interface has to has two basic properties. These properties are derived from the interaction process between human and robot as described in Section 1.1:

1. **Transfer 2D:** The interaction from human to robot can be generally reduced to a transfer of spatial information, i.e. a restriction of a mobile robot's workspace operating on the ground plane. Since a restriction area is defined by 2D positions, the user interface should allow a human to convey 2D coordinates to the robot.

2. **Feedback:** Since a mobile robot only provides limited feedback capabilities as stated in Subsection 2.1.4, it is desirable that the user interface incorporates an opportunity to give information about the interaction process to the human. This could be information about the status of the interaction process, the current 2D position specified by the human, the robot's workspace or instructions for the human regarding the interaction process.

This information exchange between human and robot can be accomplished using different communication channels. GOODRICH and SCHULTZ (2007) distinguish between five media for communication, that address three of the five senses, i.e. seeing, hearing and touch:

- **Visual displays** convey visual information to a human on displays. These displays can have several forms, such as a traditional GUI shown on a monitor. More recent advances in mixed-reality technology also allow interaction through augmented reality (AR) or immersive virtual reality interfaces. A typical application is teleoperating a robotic arm using a GUI (DESAI *et al.*, 2012) or a virtual reality headset (LIPTON *et al.*, 2018).

- **Gestures** include finger, hand, arm, head, face or body movements to convey information or interact with the environment (MITRA and ACHARYA, 2007). Specific gestures can also be performed with auxiliary devices, such as wearables, laser pointers or controllers. For example, ASSAD *et al.* (2013) use an arm sleeve comprising several sensors to control robots, and GROMOV *et al.* (2018) identify human pointing gestures with a wearable inertial measurement unit (IMU) to localize a robot with respect to the human operator.

- **Speech and natural language** refers to the interaction comprising auditory speech and text-based responses, e.g. controlling a mobile robot using basic speech commands (LV *et al.*, 2008) or conducting a dialogue with a robot to support a human in acquiring data for learning (KAPADIA *et al.*, 2017).

- **Non-speech audio** is the interaction using simple audio signals instead of natural language, e.g. KIM *et al.* (2009) investigate beep tones to indicate different types of robotic malfunctions.

- **Physical interaction and haptics** can be used in remote interaction to evoke a feeling of presence in remote tasks, such as vibration in teleoperation or telemanipulation (CASQUEIRO *et al.*, 2016), or in proximate interaction to physically interact with a robot, e.g. in a nursing assistant task (CHEN and KEMP, 2010) or in kinesthetic teaching where a human physically guides a robot to teach a new skill (AKGUN *et al.*, 2012).

Considering the two basic properties of a user interface for our problem and the scope of this thesis, we assess the basic communication channels regarding their appropriateness for our problem. For this purpose, we rate their appropriateness on a 3-point scale (*high* - *medium* - *low*) and summarize the results in Table 2.1. In terms of the ability to transfer 2D coordinates from human to robot, we rate visual displays and gestures best because they have been successfully used to transfer spatial information in various applications. Furthermore, Subsection 2.2.2 has shown that GUIs are the state-of-the-art solution for our problem. Speech commands can also be used for this task, but it is not as natural as the previous ones. The other two communication channels, i.e. non-speech audio and physical interaction, cannot be used to efficiently transfer spatial information, especially with a focus on non-experts, which results in a low appropriateness. Regarding the feedback property, visual displays are also assessed best because they can provide powerful visual feedback. The other

communication channels are rated with a medium or low appropriateness because their feedback is not as expressive as the feedback of visual displays or even not available. For example, gestures performed with auxiliary devices could provide simple visual feedback concerning the current 2D position specified by the human or audio feedback could be used for simple feedback concerning the status of the interaction process.

Table 2.1: Assessment of the communication channels' appropriateness regarding basic properties of a user interface for our problem.

	Visual displays	Gestures	Speech	Non-speech audio	Haptics
Transfer 2D	High	High	Medium	Low	Low
Feedback	High	Medium	Medium	Medium	Low

Based on the assessment of the communication media, we focus on the first two, i.e. visual displays and gestures, because they achieve the best results, especially when transferring spatial information, which is the fundamental property needed to solve our problem. Thus, in the following we give an overview of how these user interfaces have been employed to solve related HRI problems.

2.3.1 Visual Displays

Visual displays can be optimally used to transfer spatial information and to provide visual feedback to the human. We further differentiate this communication channel considering traditional GUIs and more recent mixed-reality interfaces.

Graphical User Interfaces

Traditional GUIs are used to visualize information using graphical elements on displays, and interaction with the display can be performed by either using touch gestures, e.g. on smartphones and tablets, or using external control devices, such as a computer mouse. Among the examples mentioned in Subsection 2.2.2 for the restriction of a mobile robot's workspace, GUIs are often used to provide an interface on a mobile robot to assist elderly people (GROSS *et al.*, 2011). Similarly, GRANATA *et al.* (2010) designed a GUI to allow elderly people to select robot tasks by visualizing corresponding icons on the robot's display. As opposed to this social interaction, SAKAMOTO *et al.* (2016) use a GUI to instruct home robots for a human-robot collaboration task, cooking meals and folding garments. A field of application, that is closely related to our problem, is to teleoperate and navigate a robot. For example, SCHULZ *et al.* (2000) implemented a web-based interface to allow humans to enter navigation goals on a map of the environment shown on a screen, HEBERT *et al.*

(2015) use a live video stream of a robot's on-board camera shown on a tablet to control the mobile robot, and VAUGHAN *et al.* (2016) developed a GUI to specify 2D waypoints for a mobile robot.

These examples show that GUIs are exhaustively used in robotics applications ranging from social HRI, human-robot collaboration to teleoperation and navigation. Especially, the latter ones are interesting for our problem because they show a way of how to provide spatial information, i.e. 2D positions for navigation purposes. For this purpose, the displays either show video streams of the robot's on-board camera and/or a map representation of the environment (NIELSEN and GOODRICH, 2006). The advantages of GUIs are that they can provide visual information and that most of the people are familiar with GUIs due to their widespread use, e.g. computer monitors, smartphones or tablets.

Mixed-Reality Interfaces

In contrast to traditional GUIs, mixed-reality interfaces merge real with virtual information of the environment (MILGRAM and KISHINO, 1994). Virtual reality interfaces immerse humans into a fully artificial environment, that can be a model of the real environment. Thus, humans are equipped with special devices, such as virtual reality headsets, that visualize the virtual environment to the human and allow the tracking of the human's motion. Another form of mixed reality is augmented reality (AR), that extends the real environment with additional information. As opposed to virtual reality, the human stays in the real world that is augmented with virtual elements. To create such a reality, a human is supposed to wear a special device, such as a head-mounted display, or to carry a mobile screen, such as a tablet or smartphone. An alternative to these mobile devices are projectors integrated into the environment that use a flat surface, e.g. the floor, to project information onto (GANESAN *et al.*, 2018). While mixed-reality interfaces are the primary channel for communication in this section, it is noted that these approaches often combine visual information with other communication channels, such as gestures or speech.

Due to their powerful capabilities and decreasing costs, we currently witness a trend towards mobile AR in robotics applications. For example, ROSEN *et al.* (2019) employ a mixed-reality head-mounted display to communicate motion intents of a robotic arm to a human. They furthermore compared it to a traditional 2D visualization on a monitor, and their results demonstrate the overall benefit of an AR interface compared to traditional monitor visualizations. Instead of visualizing motion intends, HORIKAWA *et al.* (2017) developed a near-future perception system that uses virtual reality connected to the real world to show potential hazardous situations to the human in advance. Similarly, ZOLOTAS *et al.* (2018) integrate an AR headset into a wheelchair navigation to highlight potential sources for collisions. Among applications to visualize robot motion intends, mixed-reality is also used to control robots, e.g. FRANK *et al.* (2016) developed an AR application running on a tablet for object manipulation tasks and QUINTERO *et al.* (2018) use AR in combination with gestures and

speech to program robot trajectories. A property, that makes mixed-reality interfaces popular, is the capability to provide expressive feedback. For example, an AR headset is used for feedback in controlling a drone (HEDAYATI *et al.*, 2018), in a shared control tasks (ELSDON and DEMIRIS, 2018) or in gaining insights about internal robot knowledge (LIU *et al.*, 2018). These examples show that mixed-reality interfaces are popular in recent HRI applications. The main reasons for this trend are (1) decreasing costs for mobile AR devices allowing a pervasive deployment, (2) the ability to provide rich visual feedback and (3) the seamless combination with other communication media to enhance the interaction.

2.3.2 Gestures

In addition to visual displays, we rated the appropriateness of gestures for our problem as second-best communication channel. Since we focus on transferring spatial information, we especially report on approaches dealing with pointing gestures. This is a natural and intuitive method of non-verbal communication because gestures mimic the interaction between humans (WACHS *et al.*, 2011). Typical robot application areas include pick-and-place scenarios, pointing to navigation goals or selecting a certain robot in a group. We further subdivide this category into human pointing gestures and gestures performed using mediator devices.

Human Pointing Gestures

This category comprises works where pointing gestures are performed by a human without any additional device and perceived by a mobile robot. The human pointing vector can be composed of several joint combinations evaluated by JEVTIC *et al.* (2015), i.e. wrist-hand, elbow-hand, shoulder-hand and head-hand. These 3D joint coordinates are localized using vision, and a pointing vector is extracted from the joint coordinates, e.g. (NICKEL and STIEFELHAGEN, 2007) and (SPRUTE *et al.*, 2018). Projecting this pointing vector onto a surface in the environment allows the realization of various applications.

A popular example for human pointing gestures in HRI is to control a mobile robot, e.g. DEN BERGH *et al.* (2011) developed a real-time hand pointing gesture recognition system to specify navigation goals for a mobile robot. For this purpose, they use a RGB-D sensor mounted on the mobile robot for hand posture recognition. Similarly, TÖLGYESSY *et al.* (2017) use human pointing gestures recognized by a RGB-D sensor to navigate a mobile robot by providing 2D positions. They also evaluated different pointing vectors revealing the elbow-wrist line as most accurate with an average position error of 0.33 m (-0.2252 m and 0.2402 m error in x and y direction). These inaccuracies are also observed in the work of DROESCHEL *et al.* (2011) who present an approach for pointing gesture recognition using a time-of-flight camera mounted on a domestic service robot. They evaluated their

system in an experiment and report an average position error of 0.43 m and 0.53 m considering the eye-hand and elbow-hand pointing vectors, respectively.

Other examples in this category comprise the selection of objects by pointing (QUINTERO *et al.*, 2013), the selection of a robot for interaction from a group of robots (POURMEHR *et al.*, 2013) or the designation of objects for interactive semantic mapping (COSGUN and CHRISTENSEN, 2018). These examples demonstrate that human pointing gestures are popular in transferring 2D spatial information because of a natural interaction. However, works have shown that they lack a high accuracy which is due to intrinsic inaccuracies in visual gesture recognition and the 3D reconstruction of the pointing vector. Moreover, there is no possibility allowing a human to get feedback of the current interaction process.

Mediator-Based Gestures

In contrast to the previous category, approaches in this category comprise works that deal with pointing gestures performed by a human with an additional device. For example, NAGI *et al.* (2014) require a human operator to wear two differently colored gloves. These are used to select an individual from a group of UAVs by pointing towards a certain UAV. The gloves are used to benefit from the naturalness and intuitiveness of human gestures while allowing the robust detection of the human's hands using color-based segmentation. WOLF *et al.* (2013) use a BioSleeve, i.e. a natural interface based on electromyography (EMG) and an IMU worn at the forearm, to point towards robotic navigation or manipulation goals. Another auxiliary device is used by GROMOV *et al.* (2019), who employ a wrist-mounted wearable incorporating an IMU to point to locations, e.g. to indicate an area for vacuum cleaning or a landing zone for a UAV. The authors also evaluated the pointing accuracy which reached an average of 0.5 m distance from the actual target position. However, the authors also report that the pointing accuracy can be significantly improved if an additional laser pointer mounted on the mobile robot gives visual feedback about the pointing position. Similarly, MIKAWA *et al.* (2010) use a laser pointer on a librarian robot to guide a human and indicate 3D positions. This inherent feedback capability makes laser pointers a popular mediator device in HRI research. For example, laser pointers are used to navigate a mobile robot to a target position (PAROMTCHIK and ASAMA, 2001)(SUZUKI *et al.*, 2005) or to designate objects in the environment and request a mobile robot to pick them up (KEMP *et al.*, 2008). Moreover, NGUYEN *et al.* (2008) generalize this idea to a clickable world where a user points at different objects and a robot perceives the laser spot with its on-board camera and derives a certain behavior. Besides, TROUVAIN *et al.* (2001) combine a laser pointer with speech into an integrated multi-robot control station.

In order to compare laser pointers to other user interfaces, CHOI *et al.* (2008) conducted a user study with amyotrophic lateral sclerosis (ALS) patients, that were asked to provide 3D locations to a mobile robot. Their results show that the use of laser pointers is faster compared to the use of a

GUI on a touch screen. Another user study is conducted by ROUANET *et al.* (2013) who investigated the impact of four user interfaces (GUI on a smartphone, Wiimote controller, laser pointer and human gestures) in teaching visual objects to a robot. This includes guiding the robot and drawing attention to a specific object in the environment. The results reveal that mediator devices, such as laser pointers, are more efficient for robot learning while being equally good in terms of usability and user experience compared to human gestures. This shows that interactions performed with a laser pointer benefit from the naturalness of human pointing gestures but provide additional visual feedback. Moreover, this feedback can be used to improve the pointing accuracy.

2.4 Intelligent Environments

While HRI user interfaces based on visual displays can be directly employed to transfer spatial information, gesture interfaces often require a direct line of sight between human and robot. Although this is a typical case in HRI, this can influence the performance of the interaction process. In order to circumvent this drawback and preserve the advantages of gesture interfaces, there are basically two possibilities: (1) the mobile robot tracks the interacting person and consequently adapts its field of view, or (2) additional sensors allow the extension of the interaction space. These additional sensors can be pervasively deployed in the environment and connected via a network to cover a larger space for potential interaction, i.e. a smart environment (COOK and DAS, 2005). When intelligently integrating these sensors into the interaction process to support humans, this leads to the research field of ambient intelligence. This deals with the intelligent software and is a "digital environment that supports people in their daily lives by assisting them in a sensible way" (AUGUSTO, 2007). The combination of a smart environment with an ambient intelligence is referred to an intelligent environment (AUGUSTO *et al.*, 2013). Such an environment can be a home, hospital, school, factory or even a city. This is an interdisciplinary research field covering topics from networks, sensors, actuators, human-computer interaction, pervasive computing and artificial intelligence. Related to our second and third objective, i.e. the investigation of the role of a smart home environment in the interaction process and learning capabilities, this section gives an overview of the current state of the art in the field of intelligent environments. For this purpose, we first present works related to ambient intelligence and smart environments, especially smart homes. Subsequently, we report on works that integrate robots into smart environments as additional mobile sensors and actuators. Finally, the last subsection covers works that deal with learning capabilities of an intelligent environment.

2.4.1 Ambient Intelligence and Smart Environments

AUGUSTO *et al.* (2010) describe a typical smart environment as a composition of several heterogeneous devices unobtrusively integrated into the physical environment. These are sensors and

actuators to perceive and change the state of the environment. Moreover, they are connected via a network allowing a data exchange between the devices. The overall goal of an intelligent environment is to provide services to the user addressing the comfort, economy, safety and miscellaneous human daily living factors (AUGUSTO *et al.*, 2013). An example for a certain smart environment is a smart home equipped with computing and information technology providing services to the residents (ALDRICH, 2003). For example, predicting human movement patterns to maximize comfort by managing household appliances and temperature (DAS *et al.*, 2002), displaying reminders on a smart mirror when a person stands in front of it (HELAL *et al.*, 2005) or automatically adapting the color of lights depending on the human's current activity (SPRUTE and KÖNIG, 2016).

These are examples where an ambient intelligence acts depending on observed human activities or behavior patterns. For this purpose, the smart environment perceives the state of the environment employing a sensor network. These sensors can be generally divided into two groups: (1) wearable devices worn by humans and (2) devices integrated into the smart environment. Regarding our second objective, we focus on the latter category which comprises sensors, such as microphone arrays, pressure pads integrated into furniture or movement detectors and cameras mounted at the ceiling. Especially, vision-based sensors are highly relevant for our objective because of the rich information that can be extracted from camera streams (PRATI *et al.*, 2019). For example, BRDICZKA *et al.* (2009) use multiple cameras in the smart environment to detect and track people, ZHANG *et al.* (2015) describe vision-based fall detection approaches, FLECK and STRASSER (2008) perform human activity recognition using a smart camera network and CHAVEZ *et al.* (2012) use a laser pointer whose spot is perceived by the environment's cameras for an environment control system. These examples show that camera sensors integrated into smart environments can be employed for a comprehensive perception of the environment, which is the basis for various high-level applications.

While sensors perceive the state of the environment, actuators are used to change the environment's state. This also includes devices for visualization, e.g. projections on a wall (KIM *et al.*, 2011), projections augmenting the environment with additional information (GANESAN *et al.*, 2018) or visual displays integrated into the environment (BUTZ, 2010)(KANG *et al.*, 2018). These approaches demonstrate the powerful visual feedback capabilities of a smart environment.

2.4.2 Network Robot Systems

After giving an overview of ambient intelligence and smart environment characteristics and applications, this subsection focuses on works that integrate (mobile) robots into smart environments. This integration intends to overcome inherent limitations of mobile robots and to extend smart environments with a mobility component (MASTROGIOVANNI *et al.*, 2010). There is no unified term for this integration of robots into smart environments, but several terms describe the same idea. KIM *et al.* (2007) coined the term *ubiquitous robots*, that deals with the embedding of robots into a

ubiquitous space. This space features a high degree of connectivity between several heterogeneous components classified as software components, embedded components and mobile components like mobile robots. At the same time, SAFFIOTTI *et al.* (2008) introduced the term *physically embedded intelligent systems (PEIS)* ecology, which describes the vision of integrating different devices, e.g. smart cameras and mobile robots, into a joint space opening the opportunity for more advanced robot applications. More generally, SANFELIU *et al.* (2008) points out five elements of a *network robot system (NRS)*: (1) physical embodiment, (2) autonomous robot capabilities, (3) network-based cooperation, (4) environment sensors and actuators and (5) human-robot interaction. Another term is used by NOR and MIZUKAWA (2014), who propose an intelligent space for home-based robotic services called *Kukanchi*. Finally, PYO *et al.* (2015) use the term *informationally structured environment* to describe an environment with embedded sensors monitoring and providing information to other agents in the environment. Due to the similarity of these terms, we use the term network robot system as a representative for the integration of robots into smart environments.

This integration can benefit robot applications in terms of the realization of more complex or more efficient services. For example, RUSU *et al.* (2008) exploit ubiquitous sensing and actuation in a smart kitchen environment to develop a service robot autonomously operating in a kitchen. They follow the ubiquitous robotics paradigm in which devices of the smart environment are used to accomplish complex tasks without significantly increasing the complexity of the robot itself. Another example is the work by SPRUTE *et al.* (2017), who propose a hierarchical search system that uses smart cameras integrated into the environment to increase the efficiency of a robot object search. Moreover, they employ a visual feedback system using colored lights of the smart environment to provide feedback about the state of the search process to the user. LI *et al.* (2013) introduce a NRS for ambient assisted living to deliver sophisticated healthcare services to residents, and SAKAMOTO *et al.* (2018) use their informationally structured environment to realize robot fetch-and-deliver services. Another example, that emphasizes the benefits of a NRS, is the work by GOMEZ-DONOSO *et al.* (2019) who aim to enhance ambient assisted living capabilities with a mobile robot focusing on disabled and elderly people. Cameras integrated into the environment are used to detect dangerous zones, e.g. electric panels or tripping hazards. However, due to the small size of the objects, occlusions or dynamic objects, the cameras in the environment can fail to detect the dangerous zones. Therefore, the authors integrate a mobile robot equipped with a camera into the smart environment to overcome these limitations.

In addition to the realization of more complex and efficient services, a smart environment can also be employed to allow a human a natural interaction with the NRS and its components. PARK *et al.* (2007) developed a robotic smart house to assist people with movement disabilities. In order to interact with a mobile robot and home appliances, they use simple voice commands to specify actions and cameras integrated into the environment to recognize human pointing gestures. These pointing gestures can be used to control the position of the mobile robot. RASCH *et al.* (2019) also use cameras of a smart home environment to recognize pointing gestures, that are used to select ob-

jects in the environment. These objects are then tidied up by a mobile robot. SHIBATA *et al.* (2011) propose a laser pointer approach to allow humans to control a mobile robot. To this end, they use ceiling-mounted cameras to track laser spots on the ground and instruct the mobile robot, which is connected via a network. Similarly, ISHII *et al.* (2009) use cameras integrated into the environment for laser spot tracking allowing a human to define stroke gestures with a laser pointer to control a network-connected mobile robot. Moreover, a projector on the ceiling visualizes the path of the laser spot. Compared to the works presented in Section 2.3.2, the recognition of human gestures is no more limited to the mobile robot's field of view. Instead the gestures are exclusively recognized by camera sensors integrated in the smart environment increasing the interaction space. Although this is typically larger than the interaction space of a mobile robot's on-board camera, it strongly depends on the number of cameras in the environment and their fields of view.

All these approaches show that the cooperation between robots and smart environments enables novel and more efficient robot applications. A major reason for this are enhanced perception abilities allowing a robot to overcome its on-board perceptual limitations in terms of time, space and type of information (SIMOENS *et al.*, 2018). Moreover, mobile robots can use actuators of the smart environment to enhance their feedback capabilities, e.g. displays or projectors.

2.4.3 Learning Capabilities

In addition to a smart environment, that provides sensors and actuators to perceive and change the state of the environment, a major component of an intelligent environment is the ambient intelligence. This aims to learn habits, preferences and needs of humans resulting in a context-aware environment. This knowledge can be further used to adapt the behavior of the smart environment to the context and increase human daily living factors. A system is considered learning "if it improves its performance on future tasks after making observations about the world" (RUSSELL and NORVIG, 2009). Regarding our third objective, i.e. the investigation of learning capabilities to support an interaction process, we first give a brief overview of machine learning techniques and present how they are used in intelligent environments.

Machine Learning Forms and Techniques

The input to a machine learning algorithm in general is a set of instances, i.e. a (training) dataset, and each instance is characterized by a set of features or attributes. According to DUDA *et al.* (2000), learning can be generally divided into three categories that differ in the human's feedback:

- **Supervised learning:** In this form of learning, training instances are labelled with a certain category or numeric cost, i.e. a labelled dataset. The name is derived from the fact that a (human) supervisor has to give explicit feedback to the system in form of labels.

- **Unsupervised learning:** In contrast to supervised learning, there is no supervisor providing feedback about the system's observations. Thus, this form of learning extracts knowledge without external feedback and seeks for natural groupings, hidden structures or associations in the training set.
- **Reinforcement learning:** In reinforcement learning, the system is provided by a series of (binary) feedback by a supervisor. The supervisor does not tell the actual wanted output, and therefore it is up to the system to decide why an instance causes a negative feedback.

Additionally, RUSSELL and NORVIG (2009) introduce semi-supervised learning as a fourth form of learning. It is a combination of supervised and unsupervised learning where only parts of a training dataset are provided with feedback from a supervisor. Besides, it is not sure if every feedback provided by the supervisor is correct.

In addition to these forms of learning, WITTEN *et al.* (2011) describe four different machine learning techniques that purchase different goals. (1) Classification is a typical form of supervised learning that deals with learning from a labelled dataset. The learned model is then expected to classify unseen instances according to the provided category labels. (2) Similarly, numeric prediction or regression tries to predict a numeric quantity instead of a category. (3) A popular technique of unsupervised learning is clustering where naturally formed groups of instances are sought. (4) Finally, association learning seeks associations between different features, which is independent of a supervisor's feedback. Thus, it is another form of unsupervised learning.

Applications

These machine learning forms and techniques are employed in intelligent environments to learn from humans' habits and routines with the goal to provide context-aware services to residents. An early work in this field is the work of MOZER (1998), who use feedforward neural networks to predict occupancy patterns and expected hot water usage in the house. Another classification approach is chosen by TAPIA *et al.* (2004), who use data from simple state-change sensors ubiquitously installed in the environment to train a naive Bayes classifier for activity recognition. Moreover, BRDICZKA *et al.* (2009) learn situation models in a smart home to provide context-aware services. For this purpose, they propose a multi-layer framework consisting of supervised learning of human activities and unsupervised extraction of situations. COOK (2012) learns human activity models from different smart home datasets employing supervised and semi-supervised learning techniques.

In contrast to supervised learning, unsupervised learning does not need feedback from a human teacher and automatically learns from observations, e.g. sensor data acquired from a smart home are clustered to recognize and predict activities of daily living (LAPALU *et al.*, 2013) (BOUCHARD *et al.*, 2015) or users' behaviors are learned and modelled using unsupervised fuzzy technique to adapt the behavior of the smart environment (DOCTOR *et al.*, 2005). Another example of an unsupervised

learning technique is the work of JAKKULA *et al.* (2009), who discover temporal patterns in smart home time series data for anomaly detection. For this purpose, they mine a dataset of sensor measurements for frequent sequential patterns, that describe normal behaviors, and identify deviations form these normal behaviors. Similarly, RASHIDI and COOK (2009) automatically adapt their smart home behavior by discovering patterns in humans' daily activities and generate automation rules according to the patterns. To this end, they employ a variant of the Apriori algorithm (AGRAWAL and SRIKANT, 1994) to find frequent patterns in the sensor data of the smart home. A combination of frequent pattern mining and clustering is applied by RASHIDI *et al.* (2011) to recognize activities in a smart environment. An extension to frequent sequential pattern mining is proposed by TAX *et al.* (2018), who use local process models to gain insights to smart home data. This is a frequent pattern mining technique that models patterns not only in sequential order but also allows concurrent executions, choices and loops.

These works show that especially supervised and unsupervised learning techniques are employed in intelligent environment applications with the goal to learn from observations and provide context-sensitive services. In particular, unsupervised learning gained attention in recent years due to the ability to learn automatically without manual annotation of a training dataset by a human supervisor. A major subtask in this field is the discovery of frequent patterns in different forms, such as frequent itemsets, sequential patterns or structural patterns (HAN *et al.*, 2007). These can be used subsequently for clustering, e.g. to recognize frequent activities.

2.5 Summary and Open Research Questions

Section 2.1 gave an introduction to autonomous robot capabilities and the current state of the art in related research fields. A major capability is the modelling and mapping of the robot's environment as basis for robot navigation. For this purpose, several map representations with their advantages, limitations and application areas were presented. In case of robot navigation and robot guidance, 2D OGMs modelling the environment in terms of fixed-size cells indicating the occupancy of the corresponding area revealed to be a popular representation. Building a map of the environment and simultaneously localizing a robot with respect to the map coordinate frame is referred to the SLAM problem. This problem is solved for structured indoor environments using range scanners. Based on these capabilities, robot navigation was introduced that deals with the efficient and reliable robot motion in the environment. Works in this field already reached maturity allowing a safe and robust path planning and obstacle avoidance. Furthermore, camera sensors proved to be important components in all these robot capabilities. To convey information from robot to human, colored light and non-speech audio revealed to be the only opportunities for this task considering robots in the scope of this thesis. However, these communication channels can only convey simple information, e.g. the robot's status, and no complex information, e.g. 2D areas on the ground.

After introducing background knowledge about autonomous robot capabilities, Section 2.2 distinguished between two categories of how to change a mobile robot's navigational behavior. This directly relates to our first objective, i.e. the investigation of interaction methods and user interfaces for the restriction of a mobile robot's workspace. The first possibility is the restriction based on implicit methods from the field of human-aware robot navigation. These methods effectively change the robot's navigational behavior, but they do not allow the incorporation of knowledge from a human in an interactive way. Since this is necessary to solve our problem, the second possibility relies on explicit methods from the field of HRI. While a lot of research has been conducted in the former category, there is only limited research in the latter one. The state-of-the-art solution for current domestic robots is the workspace restriction by sketching on a 2D OGM shown on a GUI. However, this interaction method has not been evaluated with respect to the requirements of an interaction process, and there are indications that this solution does not optimally address these requirements.

Therefore, we surveyed alternative user interfaces in Section 2.3. After a preselection concerning properties of a user interface for our problem, we further investigated user interfaces based on visual displays and gestures. As a result, we identified mediator-based pointing gestures and mixed-reality interfaces as promising alternative user interfaces due to their capability to transfer spatial information and to provide feedback to the human. Hence, we hypothesize that at least one of the alternative user interfaces performs better than the state-of-the-art solution and with an acceptable quality level regarding the identified requirements in Section 1.3. Both categories of user interfaces have not yet been used to restrict a mobile robot's workspace considering the scope of this thesis, which leads us to our first research question:

Research Question 1. *How to employ alternative user interfaces to restrict a mobile robot's workspace in a traditional home environment respecting the mentioned requirements?*

This comprises questions of (1.1) how to model and integrate user-defined restriction areas into the mobile robot's navigation framework and (1.2) how to design interaction methods allowing a human to interact with the robot employing alternative user interfaces.

Related to the second objective, i.e. the investigation of the role of a smart home environment in the interaction process, Subsections 2.4.1 and 2.4.2 introduced works in the fields of ambient intelligence, smart environments and network robot systems (NRS). The reviewed works show that smart environments can enhance robot applications in terms of complexity and efficiency. The main reason for this is the extension of a robot's perceptual capabilities through sensors of the smart environment. Especially, camera networks integrated into the smart environment revealed an important possibility to perceive the state of the environment and overcome limitations of robots' on-board cameras due to a restricted field of view. A NRS can also be used to improve the interaction between human and robot. For example, a human can control a mobile robot using gestures perceived by the smart home's camera network instead of the robot's on-board camera with limited field of view.

Thus, the interaction space is enlarged enabling a more efficient interaction. However, most of the approaches exclusively used the camera network of the smart environment for perception without employing the robot's on-board camera. Hence, the visual perception was restricted by the smart environment's camera view coverage and occlusions. While NRS were used to recognize gestures, with and without mediator devices, they were not used for the restriction of robots' workspaces. Especially, a cooperative perception employing cameras from the smart environment and the mobile robot has not yet been investigated. This would be an opportunity to overcome issues concerning camera view coverage and occlusions. Moreover, smart environments provide additional visualization capabilities that could be used to provide feedback to the human. These findings lead us to a second research question:

Research Question 2. *How can a smart home environment improve the interaction process of restricting a mobile robot's workspace with respect to the requirements?*

This research question deals with questions of (2.1) which sensors and actuators of a smart home environment can be used to benefit the interaction process, (2.2) how to realize a cooperation of human, robot and smart environment in the interaction process and (2.3) how to cooperatively perceive and combine multiple sensor observations to restrict the mobile robot's workspace. According to our second objective, the improvement of the interaction process deals with a reduction of the interaction time and an increase of the user experience.

After introducing applications using a NRS, Subsection 2.4.3 summarized relevant works that allow the ambient intelligence to learn from users' behaviors, habits and routines to provide context-aware services. For this purpose, several machine learning forms were employed ranging from supervised to unsupervised learning. In case of learning from observations in a smart home, unsupervised learning methods became popular because they automatically learn without supervision of a human teacher. This shows that there are already works that allow the learning from observations in smart environments. However, none of the works addressed our third objective, which aims to learn from multiple interaction processes and support the human in future interaction processes. Therefore, we formulate our third research question as follows:

Research Question 3. *How can a network robot system learn from user interactions and apply the knowledge in future interaction processes?*

This research question comprises questions of (3.1) how to encode an interaction process for machine learning, (3.2) how to learn from user interactions and (3.3) how to apply the knowledge extracted during learning to support a human in subsequent interaction processes. With regard to our third objective, this support of a human should result in a reduction of the interaction time to a good quality level while preserving the quality levels of the other requirements.

These research questions are the basis for the remainder of this thesis. They will be answered by prototypically implementing solutions and an empirical evaluation with respect to the requirements.

3

Virtual Borders and Interaction Methods

After identifying open research questions, this chapter deals with the first research question of how to employ alternative user interfaces to restrict a mobile robot's workspace in a traditional home environment respecting the mentioned requirements. For this purpose, we first formally define the problem setting and introduce the notations we use throughout this thesis. Afterwards, we propose the concept of a virtual border to flexibly model restriction areas and an algorithm to integrate virtual borders into the mobile robot's navigation framework. Subsequently, we present two novel interaction methods based on mediator-based pointing gestures and a mixed-reality interface to allow a human the definition of virtual borders. The goal is that at least one of the proposed interaction methods performs better than the state of the art and with an acceptable quality level. Therefore, the performance of the proposed interaction methods is finally evaluated with respect to the user requirements identified in Section 1.3.

This chapter's content (in similar or identical form) is mainly based on the publications below:

- SPRUTE, D., R. RASCH, K. TÖNNIES, and M. KÖNIG (2017). A framework for interactive teaching of virtual borders to mobile robots. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1175–1181

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019c). This far, no further: Introducing virtual borders to mobile robots using a laser pointer. In *IEEE International Conference on Robotic Computing (IRC)*, pp. 403–408

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2018). Virtual borders: Accurate definition of a mobile robot's workspace using augmented reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8574–8581

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019b). A study on different user interfaces for teaching virtual borders to mobile robots. *International Journal of Social Robotics 11*(3), 373–388

3.1 Problem Setting

A major component of the interaction process is the environment, in which the interaction between human and robot takes place. In order to model this environment for the robot's internal representation, we choose a 2D occupancy grid map (OGM) representation. This is preferred to other map representations, such as landmark-based or topological maps, for several reasons:

1. The main reason is the requirement concerning an *accurate* interaction process, i.e. allowing a human to accurately restrict the mobile robot's workspace as defined in Section 1.3. Therefore, a dense representation containing metric information is essential, which inevitably leads to OGMs.

2. The drawback concerning bad scalability of OGMs can be neglected due to the considered environments in the scope of this thesis, i.e. indoor home environments. These have a fixed size and cannot be extended excessively as opposed to outdoor environments.

3. OGMs are also used in state-of-the-art solutions to this problem when allowing interaction between a human and a mobile robot's map.

In this work, we denote an OGM of the environment as M. It models the environment in terms of $m \times n$ discrete cells containing free and occupied areas, such as walls or furniture. Each cell (x, y) of a map M contains an occupancy probability $M(x, y) \in [0, 1]$ for a corresponding area in the environment[1]. The set of all cells in the map M is denoted as the domain $\Omega(M)$.

Another component of the interaction process is the mobile robot, whose reference coordinate frame is denoted as R. Since our problem and subsequent solutions involve different reference coordinate frames, we denote the pose (or transformation) of a coordinate frame B with respect to a coordinate frame A as ${}^{A}T_{B}$. The leading superscript indicates the reference coordinate frame, while the trailing subscript indicates the coordinate frame being described. Due to the assumption that the mobile robot operates on the 2D ground plane, its pose can be described as a triple (x, y, θ) with the robot's current location (x, y) and orientation θ. Since we assume the mobile robot to be localized with respect to the map coordinate frame, the pose ${}^{M}T_{R}$ is known[2]. Moreover, due to the locomotion capabilities of the mobile robot, this pose is dynamic and automatically adjusted during locomotion by a localization algorithm. Therefore, we denote a pose between two coordinate frames A and B at a certain time k as ${}^{A}T_{B}^{k}$. A set of consecutive robot poses with respect to the map coordinate frame $\{T_R^0, T_R^1, ..., T_R^k\}$, i.e. a trajectory in the environment, is summarized as the robot's pose history $T_R^{0:k}$ up to time k. Each position, that can be reached by the mobile robot on a path consisting of free cells, is part of its *workspace* $\mathcal{W}$, i.e. $\mathcal{W} \subseteq \Omega(M)$.

[1] Due to simplicity, most OGMs only contain *free* (0), *occupied* (1) and *unknown* cells (-1).

[2] For reasons of readability, we omit the leading superscript in the following if the reference coordinate frame is the map coordinate frame M. This also applies for points being described.

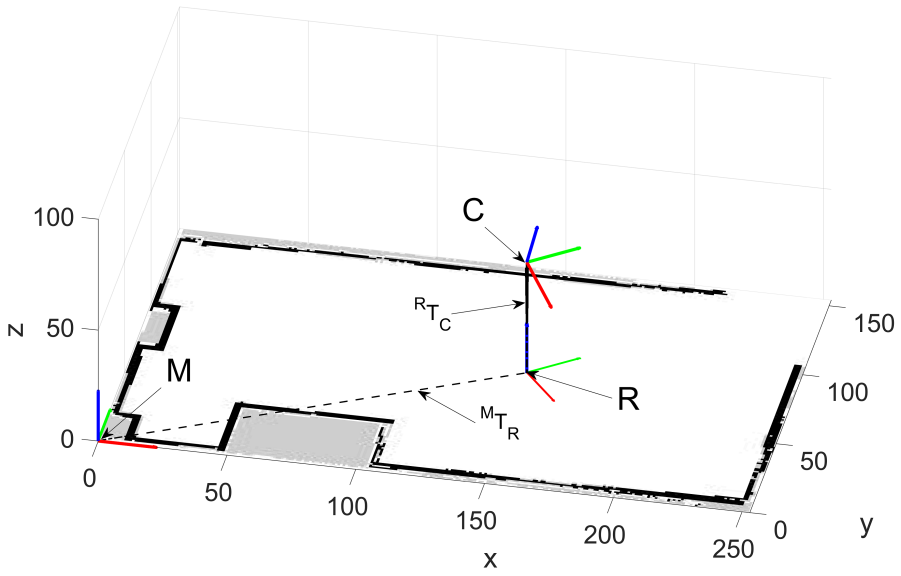


Figure 3.1: Illustration of the problem setting with occupancy grid map and relevant coordinate frames.

In addition to the mobile robot's pose, a robot in the scope of this thesis is also equipped with a front-mounted RGB-D camera C, e.g. to acquire sensor measurements for localization or interaction. This camera is attached to the mobile robot's coordinate frame R, and the transformation ${}^{R}T_{C}$ from the camera's to the robot's coordinate frame is known and static. All introduced transformations (dynamic or static) belong to the special Euclidean group $SE(3)$, which contains rigid motions in three dimensions.

Time-synchronized image streams of a color image ${}^{C}I_{RGB}$ and depth image ${}^{C}I_{Z}$ are acquired using the RGB-D camera C. The leading superscript indicates the acquiring camera, while the trailing subscript specifies the type of the image. Transformations between the different camera sensors (color and depth) are handled internally resulting in the same coordinate frame C for both camera sensors[3]. The value at a certain pixel in the color image is denoted as $I_{RGB}(x, y) \in \mathbb{R}^3_+$. Access to a pixel of the depth image of the camera is defined analogously $I_Z(x, y) \in \mathbb{R}_+$. A domain $\Omega(I)$ of an image I contains all pixels.

An illustration of the problem setting is depicted in Figure 3.1. A 2D OGM with its coordinate frame M is shown on the ground plane. White, black and gray cells indicate free, occupied and unknown areas. R indicates the mobile robot's coordinate frame and C the RGB-D camera attached to the robot. Solid and dotted black lines indicate static and dynamic transformations between coordinate frames.

[3]For reasons of readability, we omit the leading superscript in the following if the image acquiring camera is clearly determined from the context.

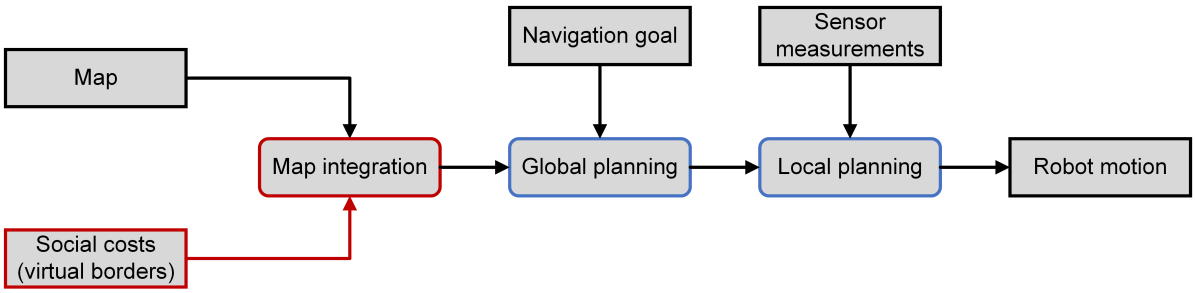


Figure 3.2: Human-aware navigation framework consisting of global and local planner. Our modifications and extensions are highlighted in red.

3.2 Workspace Restriction

After formalizing the problem setting, the goal is to restrict the mobile robot's workspace $\mathcal{W}$ and change its navigational behavior. Focusing on Research Question 1.1, i.e. how to model and integrate user-defined restriction areas into the mobile robot's navigation framework, we adapt and extend the state-of-the-art human-aware navigation framework shown in Figure 3.2. As already mentioned in Subsection 2.1.3, such a robot navigation framework consists of a global and local path planning module. The former one calculates a path from the mobile robot's current pose to the given navigation goal by considering the map of the environment and additional social costs. To this end, a costmap is created based on the given map, e.g. occupied and free cells correspond to high and low costs, and a path with minimal costs is determined. In contrast to the global planner, the local planner adapts the given path according to local obstacles perceived by on-board sensor measurements. The result is a collision-free and human-aware robot motion to the navigation goal.

In order to incorporate user-defined restriction areas into the navigation framework, there are two possibilities to change the mobile robot's navigational behavior. (1) The sensor measurements as input to the local planner could be employed to detect restriction areas and modify the robot motion accordingly. However, these cannot be detected using the mobile robot's on-board sensors as described in Section 1.1. Moreover, this possibility does not allow the explicit incorporation of knowledge from a human, which is essential for our problem. Thus, this possibility is not a solution for our problem. (2) The other possibility is the manipulation of the map, which is the input for the global planner. This could be modified to explicitly incorporate user-defined restriction areas as a result of an interaction process. Therefore, we prefer the second possibility in this work.

To this end, we first propose *virtual borders* as a data structure to model social costs, i.e. restriction areas in the case of our problem. Since these can have arbitrary shapes and sizes, a virtual border is designed to be flexible to account for this requirement. Afterwards, we present an algorithm that integrates a virtual border into a given OGM of the environment. The resulting map can be used

subsequently by the global path planner, which then generates paths respecting the user-defined restriction areas. This addresses the requirement concerning the correctness of the interaction process. Finally, the local planner is used to circumvent humans or other physical objects during the interaction process by treating them as obstacles. Both novel aspects, virtual borders and map integration algorithm (colored in red in Figure 3.2), are described in the following subsections.

3.2.1 Virtual Borders

In this work, we employ virtual borders as data structure to flexibly model restriction areas. These are non-physical borders, that are not directly visible to the human, but that are respected by the mobile robot during navigation if integrated into the environment's map. Thus, a virtual border can be used to interactively specify restriction areas and change the mobile robot's navigational behavior. To this end, a virtual border comprises spatial information necessary to specify a restriction area in an interaction process. As already described in Section 1.1, a restriction area's spatial information consists of a boundary and an occupancy value. To model this information, we compose a virtual border of the following three components:

- The **virtual border points** $\mathcal{P}$ specify the boundary of a restriction area and are structured as a polygonal chain, which should not be self-intersecting. The polygonal chain consists of $n > 1$ points $\boldsymbol{p}_i \in \mathbb{R}^2$ corresponding to coordinates on the ground plane of the environment. We distinguish between (1) closed and (2) simple polygonal chains to address the requirement concerning flexibility. In case of a simple polygonal chain, that does not partition the map into two areas, we apply a linear extension to the first and last line segment. Thus, the beginning and ending of the polygonal chain are automatically extended to the borders of the map. This allows a human to specify restriction areas with arbitrary shapes and sizes.
- A **seed point** $\boldsymbol{s} \in \mathbb{R}^2$ is the user-defined component of a virtual border that indicates the area to be modified during the interaction process. This is necessary because only one area, that is separated from the rest of the environment by the virtual border points $\mathcal{P}$, is modified in an interaction process.
- $\delta \in [0,1]$ is the user-defined component which indicates the **occupancy probability** of the area to be modified (as indicated by $\boldsymbol{s}$). Since we focus on restriction areas in this work, i.e. areas that should not be entered, we mainly focus on the occupancy values *free* (0) and *occupied* (1).

3.2.2 Map Integration

In order to enforce a mobile robot to change its navigational behavior according to a user-defined virtual border, this needs to be integrated into the map of the environment. Thus, we propose a map

integration algorithm that combines a map of the environment and a virtual border. This algorithm requires a 2D OGM of the physical environment M_{prior} and a virtual border $V = (\mathcal{P}, \boldsymbol{s}, \delta)$ as input and outputs a 2D OGM $M_{posterior}$ containing physical as well as virtual borders.

The virtual border points $\mathcal{P}$ specify the boundary of a virtual border on the ground plane and are structured as a polygonal chain:

$$\mathcal{P} = \bigcup_{i=1}^{n-1} [\boldsymbol{p}_i \boldsymbol{p}_{i+1}] \tag{3.1}$$

Since an OGM is a discrete map representation, we denote the corresponding cells of the polygonal chain in the map as $\mathcal{P}^* \subset \Omega(M_{prior})$. This polygonal chain, which is typically a single cell thick to address the requirement concerning accuracy, is first integrated into the posterior map:

$$M_{posterior}(x,y) = \begin{cases} \delta & if\ (x,y) \in \mathcal{P}^* \\ M_{prior}(x,y) & if\ (x,y) \notin \mathcal{P}^* \end{cases} \tag{3.2}$$

$\delta \in [0,1]$ is the user-defined component which indicates the occupancy probability of the area to be modified. If the virtual border points $\mathcal{P}$ constitute a closed polygonal chain, this partitions the map into two areas. In case of a simple polygonal chain, that does not partition the map, we apply a linear extension to the first $[\boldsymbol{p_1 p_2}]$ and last $[\boldsymbol{p_{n-1} p_n}]$ line segment. Thus, the beginning and ending of the polygonal chain are automatically extended to the borders of the prior map M_{prior} leading to a partitioning of the map. The posterior map now consists of two disjunct areas:

$$A_c = \{\boldsymbol{c} \in \Omega(M_{posterior}) \mid connected(\boldsymbol{c}, \boldsymbol{s}^*, M_{posterior})\} \tag{3.3}$$

and

$$A_{nc} = \Omega(M_{posterior}) \setminus A_c \tag{3.4}$$

The former one is the area directly connected to the cell corresponding to the seed point $\boldsymbol{s}^*$, while the latter one is the complementary area. We consider two cells $\boldsymbol{a} \in \Omega(M)$ and $\boldsymbol{b} \in \Omega(M)$ in a map M as $connected(\boldsymbol{a}, \boldsymbol{b}, M)$ if:

$$\begin{aligned} \exists f : [0..1] \rightarrow \Omega(M) : f(0) = \boldsymbol{a}, f(1) = \boldsymbol{b}, \\ \forall i, j \in [0..1] : M(f(i)) = M(f(j)) \end{aligned} \tag{3.5}$$

where f is a continuous mapping. Finally, the area to be modified, i.e. the cells contained in A_c, is filled with the occupancy probability δ:

$$M_{posterior}(x,y) = \begin{cases} \delta & if\ (x,y) \in A_c \\ M_{prior}(x,y) & if\ (x,y) \in A_{nc} \end{cases} \tag{3.6}$$

In order to allow a human the definition of multiple virtual borders to address the flexibility requirement, this algorithm can be performed N times resulting in a sequence of virtual borders $\mathcal{V} = \{V_1, V_2, ..., V_N\}$. Hence, the posterior map of the i-th interaction process becomes the prior map of the $i+1$-th interaction process. This posterior map can be then used by a global path planner to change the mobile robot's navigational behavior according to the user-defined restriction areas.

3.3 Interaction Methods

After answering the question of how to model and integrate user-defined restriction areas into the mobile robot's navigation framework, the three components of a virtual border need to be defined by a human in an interaction process. This leads to Research Question 1.2, that deals with the design of interaction methods allowing a human to interact with the robot. For this purpose, a user interface offers the possibility for interaction between human and robot, while an interaction method describes how to interact with the robot employing the user interface in an interaction process to achieve the goal, i.e. the definition of virtual borders. We distinguish between two types of interaction methods:

1. **Robot-dependent:** The mobile robot is directly involved in the perception and interaction with the human. Thus, it is essential for the information transfer in the interaction process. These interaction methods are typically based on simple user interfaces that are not able to transfer spatial information or feedback without active participation of the mobile robot, e.g. human gestures or remote controllers.

2. **Robot-independent:** The mobile robot is not directly involved in the perception and interaction with the human, but instead the primary focus of the interaction is set on the user interface. However, the information transferred in the interaction process are accessible to the mobile robot. These interaction methods typically employ powerful user interfaces that can transfer spatial information and feedback with their own capabilities, e.g. tablets, smartphones or augmented reality (AR) devices.

In this section, we propose an interaction method for each of these types. To this end, we employ two alternative user interfaces identified in the literature review in Section 2.3. Especially, mediator-based pointing gestures and mixed-reality interfaces were identified as promising alternatives due to their ability to transfer spatial information and provide feedback to the human. In case of mediator-based pointing gestures, we decide to investigate the use of a laser pointer as user interface because it benefits from the naturalness of pointing gestures while providing inherent visual feedback. It is reported that this feedback helps to significantly increase the accuracy, which is an important requirement of an interaction process. Another reason is that laser pointers are everyday devices that non-expert humans are familiar with. Since the spot generated by the laser pointer has to be perceived by sensors, this inevitably leads to a robot-dependent interaction method.

In case of mixed-reality interfaces, we focus on AR because it can be used to enhance the real environment with additional information allowing a powerful feedback capability. Furthermore, an AR device can be used to directly interact with the *real* environment as opposed to virtual reality. Since the user interface, i.e. the AR device, is responsible for the perception of human interactions and the feedback, this is a robot-independent interaction method. We give details on how both user interfaces are employed in interaction methods in the following subsections.

3.3.1 Laser Pointer

Our first interaction method depends on the mobile robot and is based on a laser pointer as user interface. The idea of our robot-dependent interaction method is to leverage the accurate localization of the mobile robot in the environment to address the requirement concerning accuracy. There are two possibilities to exploit this characteristic: (1) the mobile robot's pose or (2) an accurately localized position detected by the mobile robot's on-board camera can be used to specify the spatial information of a virtual border. However, since the mobile robot has a limited field of view and a restriction area can have arbitrary shapes and sizes, the mobile robot has to move in the environment to acquire the necessary spatial information.

Robot Guidance Framework

Therefore, we propose a robot guidance framework intended to specify virtual borders. The main idea is that a human employs a user interface, e.g. a laser pointer, to guide the mobile robot. To this end, a human employs the user interface to indicate a *following point* on the ground. While following, the robot simultaneously keeps track of its pose history $T_R^{0:k}$, that can be used to define components of a virtual border. As introduced in Subsection 3.2.1, a virtual border consists of a triple $V = (\mathcal{P}, \boldsymbol{s}, \delta)$. Thus, each component has to be represented in this framework. For this purpose, the framework comprises the following three states that we will refer to throughout this work:

1. **Guide:** The human employs the user interface to guide the mobile robot to the desired restriction area. This is necessary because the mobile robot is typically not within visible range of the restriction area when it is autonomous mode. For the purpose of robot guidance, the user interface indicates a following point on the ground, that is either detected and localized by the robot's on-board camera or internally provided to the robot. The mobile robot follows this position until the following point is no more visible or available.

2. **Border:** The human guides the mobile robot with the user interface as in the *Guide* state, but in addition the mobile robot simultaneously records its pose history $T_R^{a:b}$ from the time a entering the state until leaving the state b. This history is used to specify the virtual border

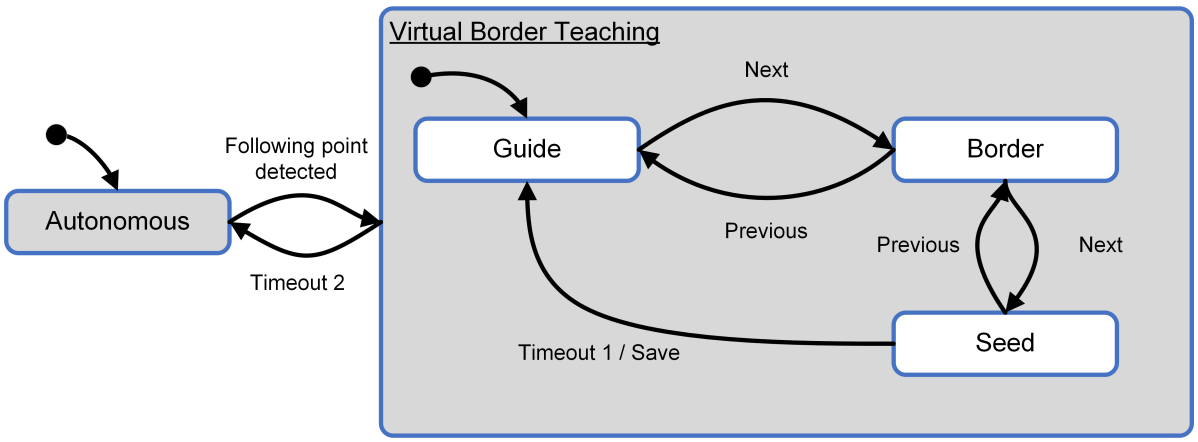


Figure 3.3: States and transitions of the robot's internal behavior including the proposed robot guidance framework.

points $\mathcal{P}$. Alternatively, instead of recording the mobile robot's pose history, the positions of the following point are stored and used as virtual border points $\mathcal{P}$.

3. **Seed:** The mobile robot stops changing its position. The human has the possibility to rotate the robot around its vertical axis employing the user interface to indicate the restriction area. After a timeout, the last following point is used to indicate the seed point $\boldsymbol{s}$, which defines the restriction area, i.e. $\delta = 1$. This should not be intruded by the mobile robot in future navigation tasks.

Figure 3.3 gives an overview of the mobile robot's internal states. The state *Autonomous* describes the default behavior of the mobile robot when providing services to humans. The new super state *Virtual Border Teaching* comprises the framework's different states and transitions, that are triggered by user interactions or timeouts. The action *Save*, that is triggered after a timeout when changing from *Seed* to *Guide* state, performs the map integration algorithm presented in Subsection 3.2.2 and thus saves the result of the interaction process.

In order to use this framework, a concrete implementation only (1) has to provide an appropriate user interface to indicate a following point for robot guidance, (2) has to implement an algorithm to recognize and localize a following point and (3) has to define the two events *Next* and *Previous*, which are used to switch between different states of the framework[4]. Thus, the implementation of the states can remain the same between different user interfaces. This makes it easy to adapt to other user interfaces, such as human gestures or remote controllers, and facilitates the portability and distribution of the interaction methods.

[4]The other events, i.e. *Timeout 1*, *Timeout 2* and *Following point detected*, are internally generated and do not need to be adapted.

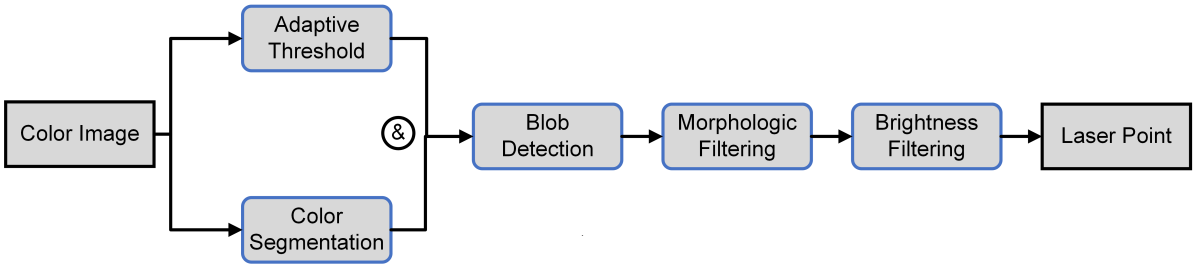


Figure 3.4: Image processing pipeline for laser point detection. The input is a color image, and the result is the 2D position of the laser point in the image plane if present.

Laser Point Detection and Localization

As already mentioned, we propose a laser pointer as user interface because we assume that a laser pointer meets the requirements of an interaction process best. To employ the robot guidance framework, the laser pointer interface has to provide a following point, which is localized by the robot's camera, and has to define both events of the framework. Therefore, a laser spot first needs to be visually detected and localized by the mobile robot's on-board camera C, which captures images of the scene. A laser spot on the ground, that is generated by a human, has several properties that will be addressed in the detection process:

1. The spot has a single color (typically green or red).

2. The spot is significantly brighter than its surrounding environment.

3. The spot has a size of approximately 5×5 mm depending on the material of the ground.

4. The spot is approximately circular.

We apply a multi-stage image processing approach to detect the laser point in the input image I_{RGB}. The processing pipeline is tailored to the characteristics of a laser point and is shown in Figure 3.4. The first steps of the image processing pipeline are processed in parallel to identify locally bright areas and areas with certain color characteristics to address Properties 1 and 2 of a laser spot. The bit-wise conjunction of both processed images results in a mask that contains pixels with both characteristics. In order to extract laser point candidates $\mathcal{C} \in \Omega(I_{RGB})$, blob detection is performed on the combined image to find connected pixels. Afterwards, blobs are discarded that do not match the morphological characteristics of a laser point, i.e. the size (Property 3) and the circularity (Property 4) of the blob. Finally, the brightness of a blob center (x_c, y_c) represented by the V-value

of $I_{HSV}(x_c, y_c)$ has to exceed a certain threshold to remain a laser point candidate[5]. If more than one blob fulfills all the criteria, the brightest candidate point $\boldsymbol{l}$ is chosen:

$$\boldsymbol{l} = \underset{(x_c,y_c)\in\mathcal{C}}{\arg\max}\, V(I_{HSV}(x_c, y_c)) \tag{3.7}$$

The $V(\cdot)$-operator extracts the V-value of $I_{HSV}(x_c, y_c)$.

In order to follow the laser point detected in the input image, its 2D image coordinate is transformed into 3D space. A 3D point ${}^{C}\boldsymbol{P} = (X, Y, Z)^T$ in space acquired from the camera C is projected onto a point $\boldsymbol{p} = (x, y)^T$ on the image plane by applying the central projection π of the pinhole camera model (HARTLEY and ZISSERMAN, 2003):

$$\boldsymbol{p} = \pi({}^{C}\boldsymbol{P}) = \left(\frac{X f_x}{Z} + c_x, \frac{Y f_y}{Z} + c_y\right)^T \tag{3.8}$$

f_x and f_y are the focal lengths in pixels, and (c_x, c_y) is the principal point in image coordinates. These intrinsic camera parameters are obtained during a calibration process. The inverse projection of an image point $\boldsymbol{p}$ into space additionally depends on its distance to the camera $Z = I_Z(\boldsymbol{p})$.

$${}^{C}\boldsymbol{P} = \pi^{-1}(\boldsymbol{p}, Z) = \left(\frac{x - c_x}{f_x} Z, \frac{y - c_y}{f_y} Z, Z\right)^T \tag{3.9}$$

After transforming the image coordinates of the laser point $\boldsymbol{l}$ into space ${}^{C}\boldsymbol{L}$, the mobile robot can follow the laser point by applying visual servoing technique (CHAUMETTE *et al.*, 2016)[6]. Thus, $\boldsymbol{L}$ is the following point used in the robot guidance framework. The distance information $I_Z(\boldsymbol{l})$ is used to adjust the mobile robot's velocity to ensure a smooth motion. Some exemplary images of an interaction process with a laser pointer are depicted in Figure 3.5.

State Change Interaction

In addition to providing a following point to guide the mobile robot, it is necessary to switch between the different states of the framework. For this purpose, the two events *Next* and *Previous* need to be implemented. Since the human already employs a laser pointer to provide a following point, it is reasonable to also use the same user interface for this task. Therefore, we use two simple visual codes generated by the laser pointer and recognized by the mobile robot's on-board camera[7]. As an alternative, a human can also push two different buttons on the mobile robot's platform to switch

[5] I_{HSV} is the color image of camera C transformed from RGB to HSV color space.

[6] Applying the rigid transformations introduced in Section 3.1, the position of the laser point can be easily transformed between the different coordinate frames.

[7] Currently, the visual codes are manually generated by a human by pressing the laser pointer's button, but an automatic generation could be easily realized in the future.

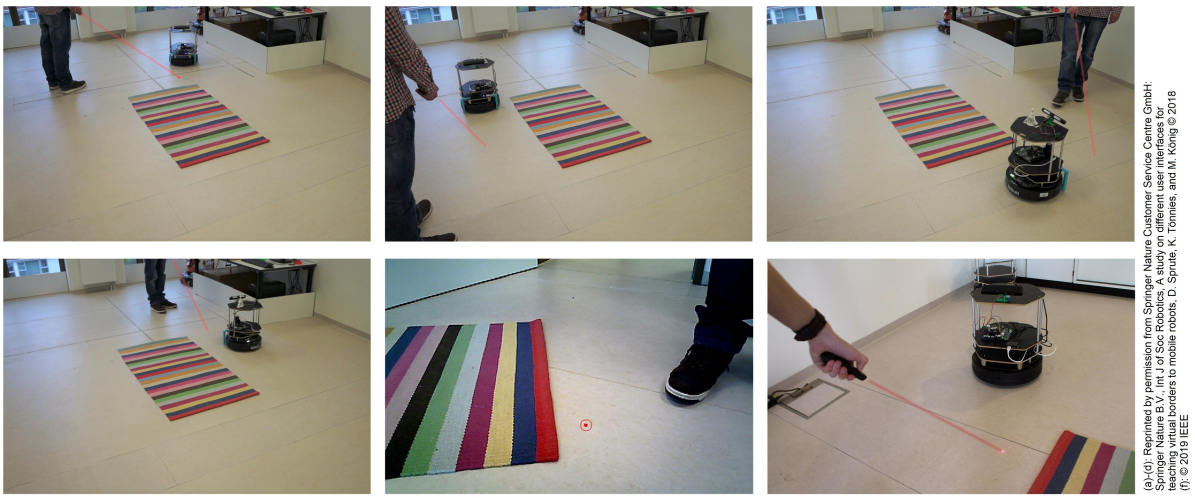


Figure 3.5: Row-wise from top left to bottom right (a-f): (a)-(d) show consecutive images of an interaction process using a laser pointer to specify a carpet as restriction area, whereas (e) and (f) illustrate the interaction process from the mobile robot's and human's perspective.

between the states. The former method is the more comfortable one, while the latter method is the more robust one. We choose this multimodal interaction because the interaction using visual codes can be error-prone under certain light conditions, e.g. certain camera angles and reflections on the ground sometimes lead to an extreme overexposure of the images making the recognition of the visual codes impossible. In this case, a human can easily use the robot's on-board push buttons to ensure the functionality of the system.

Feedback

Until now, we described our interaction method in terms of the transfer of the 2D spatial information used to specify a restriction area. But, as already identified in Section 1.1, it is also necessary to provide immediate feedback about the interaction process to the human. For this purpose, only the user interface and the mobile robot's on-board feedback capabilities are available. This additionally underlines why we choose a laser pointer as user interface to perform human pointing gestures: as opposed to other possibilities to perform pointing gestures, e.g. with a wrist-mounted inertial measurement unit (IMU) or without an additional device, a laser pointer provides an inherent visual feedback to the human. We hypothesize that this leads to a more accurate transfer of the spatial information and a better user experience. Moreover, colored LEDs on the mobile robot signalize the internal state of the interaction method, i.e. the state of the guidance framework. Each color (red, green and orange) corresponds to one of the three system's teaching states. Additional to a color change of the LED in case of a state change, the mobile robot also employs a sound feedback (beep

tones) to signalize state changes. Both communication channels are typical solutions for mobile robots with restricted feedback capabilities as revealed in Subsection 2.1.4.

3.3.2 Augmented Reality

As an alternative to this robot-dependent interaction method based on a laser pointer, we also propose a novel interaction method based on an AR application. The idea is to exploit a visual AR interface to provide a powerful feedback about the interaction process to the human. Thus, the feedback is not restricted to the limited mobile robot's feedback capabilities. In order to realize an AR application, we choose a RGB-D Google Tango tablet as user interface. In contrast to common tablets, a RGB-D tablet can perceive depth information of a scene additional to color information. We prefer a tablet solution to other AR devices, such as glasses or headsets, because we assume our intended users to be familiar with common consumer products, such as tablets or smartphones. Therefore, the use of a tablet should benefit a human and should positively affect requirements, e.g. concerning interaction time or learnability. Moreover, tablets and smartphones are widely used leading to a high number of potential users. In addition, such an AR device is not only suitable to provide feedback about the interaction process, but also allows the transfer of 2D spatial information, which is the other property of a user interface for our problem. Since an AR device combines both properties, it can be employed in a robot-independent interaction method.

AR Device Localization

In order to realize this behavior, i.e. allowing the transfer of spatial information and provide visual feedback, the AR device $Tango$ needs to be related to the map coordinate frame M. Thus, a transformation ${}^{M}T_{Tango}$ needs to be established. As opposed to the mobile robot's pose, which comprises three degrees of freedom (DoF), the pose of the AR device consists of six components (3D position and 3D orientation). Thus, it is harder to localize the 6-DoF AR device with respect to the 2D OGM of the environment M. Therefore, the AR device initially constructs a 3D map of the environment and stores it internally for localization in the environment[8]. The origin of this 3D map is the ADF (Area Description File) coordinate frame. To transform points between this 3D map ADF and the 2D OGM M, these coordinate frames need to be manually registered to each other. This manual registration has to be performed only once because both coordinate frames are static, i.e. they do not change over time. Hence, the AR device only needs to be localized with respect to the 3D map ADF. To this end, an auxiliary coordinate frame SoS (Start of Service) is introduced, which

[8]The construction of a 3D map of the environment and the localization of the AR device inside this map are performed internally by the device. There are no details of the manufacturer how this is exactly accomplished, and hence it is outside the scope of this thesis.

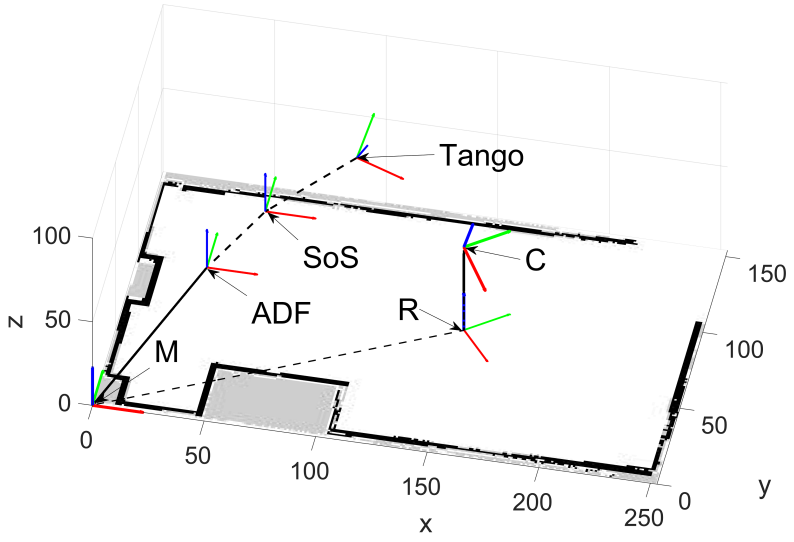


Figure 3.6: Relevant coordinate frames for the augmented reality interaction method.

marks the current pose of the AR device when starting the AR application. While moving in the environment, the AR device uses its accurate on-board visual-inertial odometry to keep track of its current pose $Tango$ with respect to SoS. Moreover, the sensor measurements are used to (re-)localize the AR device in the 3D environment, which leads to the transformation ${}^{ADF}T_{SoS}$. This localization process, i.e. the determination of the transformation ${}^{ADF}T_{Tango}$, is internally performed by the AR device and influences the accuracy of the user-defined virtual borders.

All relevant coordinate frames, which all belong to $SE(3)$, are related to each other enabling transformations of points between the coordinate frames. Figure 3.6 illustrates the relevant coordinate frames and their transformations. In addition to the coordinate frames M, R and C known from the problem setting in Section 3.1, there are three additional coordinate frames that are established by the localization process of the AR device. In summary, this leads to two requirements for the AR interaction method:

1. **3D map construction:** A 3D map of the environment ADF needs to be initially constructed using the AR device. For this purpose, it is sufficient to move the AR device through the environment.

2. **Coordinate frame registration:** After constructing a 3D map of the environment ADF, this coordinate frame needs to be manually related to the 2D OGM of the environment M. For this purpose, the transformation ${}^{M}T_{ADF}$ needs to be established.

Both requirements only need to be addressed once during an installation period and do not affect the performance of humans in an interaction process.

Spatial Information Transfer

When both maps are related to each other, i.e. 3D internal map of the AR device ADF and 2D OGM of the environment M, it is possible to transform points, that are specified with respect to the $Tango$ coordinate frame, to the map coordinate frame M. Thus, these points can be used to specify virtual borders for the mobile robot. In order to specify the three components of a virtual border, we propose an AR application for the interaction as depicted in Figure 3.7. For this purpose, a human moves around in the environment with the AR device and selects virtual border points $\mathcal{P}$ by pointing the center of the device towards the desired points on the ground plane. A green software button at the bottom right of the screen adds a new virtual border point, whose 3D position is determined employing the depth sensor of the tablet similar to the procedure in Subsection 3.3.1. The center and pointing direction is marked with a small cross at the center of the tablet's screen. The seed point $\boldsymbol{s}$ is selected analogously by touching the orange software button. Optionally, a human can remove certain (or all) points by pointing the tablet's center towards the desired position and touching the red software button at the bottom left of the tablet's screen. This button can also be used to cancel the interaction process and remove all points at the same time. The third component of a virtual border, i.e. the occupancy probability δ, is set to *occupied* by default. Thus, the seed point $\boldsymbol{s}$ indicates the restriction area. However, we also add a simple menu, that pops up by pressing the orange software button for a longer period, to change the occupancy probability δ. This enables a human to delete virtual borders, that were already integrated into the OGM of the environment, by setting the occupancy probability $\delta = 0$. Moreover, this can be used to model levels of restriction in the future, e.g. an occupancy value of $\delta = 0.75$ could mean that a mobile robot should avoid this area unless it is necessary. Finally, if all components of a virtual border are defined, a blue software button appears at the top right of the tablet's screen, which integrates the virtual border into the OGM of the environment, i.e. performing the map integration algorithm introduced in Subsection 3.2.2.

Feedback

The AR application is not only used to transfer spatial information but also to provide immediate visual feedback to the human. To this end, the tablet's screen shows an augmented live video stream of the tablet's camera. For example, the OGM of the environment is overlaid on the video stream and visualizes the workspace of the mobile robot. This includes physical but also virtual borders as a result of a successful interaction process. This makes it easy for the human to understand the workspace of the mobile robot. Moreover, the spatial components of a virtual border, i.e. virtual border points $\mathcal{P}$ and seed point $\boldsymbol{s}$, are displayed on the tablet's screen to provide feedback during the interaction process. Finally, the mobile robot's path to a navigation goal is augmented if the robot is in autonomous mode. This visualizes the change of the mobile robot's navigational behavior.

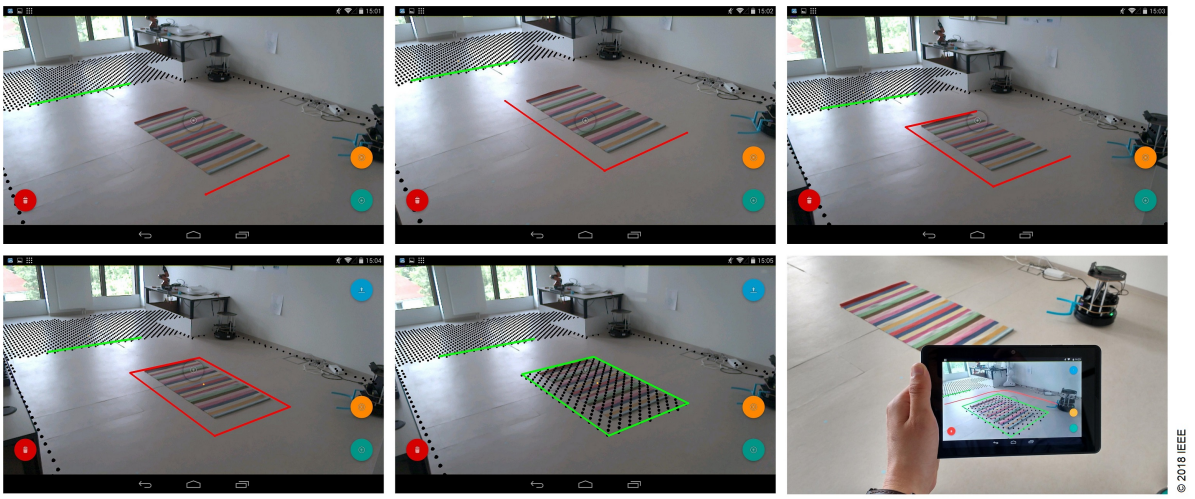


Figure 3.7: Row-wise from top left to bottom right (a-f): (a)-(e) show consecutive screenshots of an interaction process using an augmented reality application running on a RGB-D tablet to specify a carpet as restriction area. (f) shows the interaction process from the human's perspective.

In addition to this visual feedback, the AR application also includes a haptic feedback to support the human during the interaction process. For example, if the human wants to specify a closed polygonal chain as boundary $\mathcal{P}$ of a virtual border, i.e. the first and last point are the same, the tablet vibrates when pointing close to a previously defined point. Furthermore, a vibration indicates that a previously defined point can be selected for removal.

3.4 Evaluation

After introducing the interaction methods employing two alternative user interfaces for the restriction of a mobile robot's workspace, we evaluate their performance with respect to the requirements presented in Section 1.3. To this end, we conduct four different experiments, that are described in detail in the following subsections. This experimental evaluation is inspired by the USUS framework, that provides a methodological framework for evaluating aspects of a system involving the interaction of human and robot (WEISS *et al.*, 2009).

3.4.1 Baseline Method

Since it is the objective to improve the state of the art and there is no evaluation concerning the requirements in related works, we perform our experimental evaluation in comparison to a state-of-the-art solution, which was identified in the literature review in Subsection 2.2.2. This baseline method is based on a graphical user interface (GUI) shown on a common tablet, which is used to

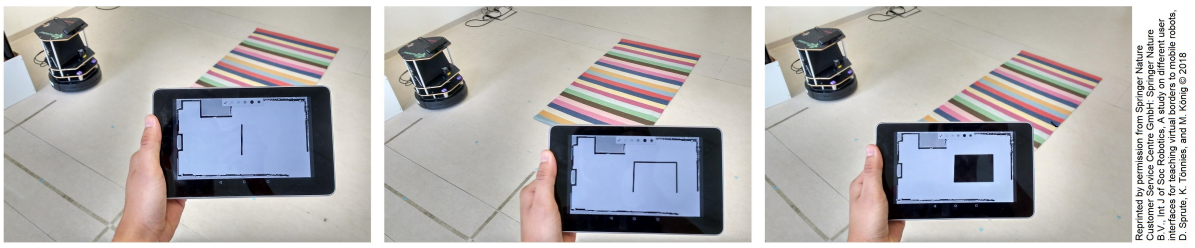


Figure 3.8: Consecutive images of the baseline interaction method using a graphical user interface displaying an occupancy grid map to specify a carpet as restriction area.

sketch virtual borders on an OGM of the environment[9]. A human holds the tablet and can freely move in the environment. In order to specify a virtual border, the human directly sketches the desired virtual border points $\mathcal{P}$ in the OGM of the environment shown on the tablet. For this purpose, the human can use the fingers to sketch a polygonal chain in the map. Additionally, a flood filling tool is available to indicate the seed point s and fill the corresponding area. The occupancy probability δ is defined by the selected filling color, which is typically black for an occupied area. As a feedback of the interaction method, the GUI directly visualizes the resulting OGM including the user-defined virtual border. Since the GUI is responsible for the spatial information transfer and feedback, this is a robot-independent interaction method, which does not require a mobile robot for perception or interaction with the human. Examples images of an interaction process employing this user interface are depicted in Figure 3.8.

3.4.2 Mobile Robot Platform

To evaluate the interaction methods, especially the robot-dependent one, a mobile robot according to the scope of this thesis is necessary. For this purpose, we use a TurtleBot v2[10] equipped with a laser scanner for localization and a front-mounted RGB-D camera C in the experiments. The camera's color images I_{RGB} are captured with a resolution of 640 × 480 pixels, and the depth images I_Z have a resolution of 160 × 120 pixels with a frame rate of approximately 25 $^{\text{frames}}/_{\text{s}}$. Additionally, the robot has a colored on-board LED, three push buttons and a non-speech audio speaker for interaction and feedback. The robot's base has differential-drive wheels allowing rotations around the vertical axis, which is similar to a typical vacuum cleaning robot used in home environments. To ensure a safe and smooth motion of the robot, its velocity is restricted to 0.2 $^{\text{m}}/_{\text{s}}$. The mobile robot's odometry data, that are needed to estimate its egomotion, are provided through motor encoders and a gyroscope. In addition to the hardware, the mobile robot can create a 2D OGM of the environment M with a resolution of 2.5 $^{\text{cm}}/_{\text{cell}}$. For this purpose, a common simultaneous localization

[9] Since there is no open-source implementation for this baseline method available, we employ a simple graphics editor to realize the functionality for evaluation.

[10] `https://www.turtlebot.com/turtlebot2` [Accessed: 26.03.2020]

and mapping (SLAM) algorithm running on the mobile robot is employed (GRISETTI *et al.*, 2007). Moreover, an adaptive Monte Carlo localization approach is used to localize the mobile robot R with respect to the environment's map M (FOX, 2003). For navigation purposes, a navigation function computed with Dijkstra's algorithm is used as global planner and a dynamic window approach is chosen for local planning (FOX *et al.*, 1997). Hence, this is a typical mobile robot platform with capabilities described in the scope of this thesis in Section 1.2.

3.4.3 Software Implementation

To allow the communication between different components of the system, e.g. tablet or mobile robot, our implementation of the interaction methods is based on the robot operating system (ROS), which is the de facto standard for robot applications (QUIGLEY *et al.*, 2009). It is a modular middleware architecture that allows communication between several components of a system, that are called nodes. These can be organized in packages to allow the easy distribution of the implementation. Therefore, we implemented all components of the interaction methods as ROS nodes. Moreover, ROS provides a large set of tools to accelerate prototyping and error diagnosis, which are used for the evaluation.

3.4.4 Experiment 1: Learnability and User Experience

The first experiment aims to evaluate our two proposed interaction methods and test hypotheses concerning learnability and user experience. It involves a typical scenario for a restriction area and multiple participants in a traditional home environment.

Independent Variables

We manipulate a single independent variable in this experiment, i.e. the interaction method, to compare the performance of the interaction methods with each other. Therefore, this variable can have one of the three values[11]:

1. **GUI:** This is the baseline interaction method based on sketching restriction areas on an OGM shown on a GUI. This method was described in detail in Subsection 3.4.1. In the experiment, we use an Asus Nexus 7 tablet with a display size of 7 inches for this interaction method.

[11] In the original experiment, there was also another interaction method based on visual markers as user interface. This was initially used to show the applicability of the robot guidance framework. However, since the focus of this interaction method was not on the improvement concerning the requirements of an interaction process, it is not considered in this experiment. For results of this interaction method, we refer the reader to (SPRUTE *et al.*, 2019b)

2. **Pointer:** This is the first proposed interaction method based on a laser pointer as user interface, that was described in detail in Subsection 3.3.1. In the experiment, we use a common (green or red) laser pointer for this interaction method.
3. **Augmented Reality (AR):** This is the second proposed interaction method based on AR and a RGB-D tablet as user interface, that was described in detail in Subsection 3.3.2. In the experiment, we use a 7-inch Google Tango tablet for this interaction method.

Hypotheses

The objective of this experimental evaluation is the test of the following hypotheses concerning learnability and user experience as defined in Section 1.3. These hypotheses are derived from Objective 1:

- **Hypothesis 1:** At least one of the proposed interaction methods achieves (1.1) a better learnability than the current state-of-the-art solution based on sketching restriction areas on a GUI and (1.2) an acceptable learnability for our problem.
- **Hypothesis 2:** At least one of the proposed interaction methods achieves (2.1) a better user experience than the current state-of-the-art solution based on sketching restriction areas on a GUI and (2.2) an acceptable user experience for our problem.

Setup

To test our hypotheses, we perform the experiment in a 6.1 × 4.0 m lab environment, that serves as indoor home environment. It comprises free space, walls, tables and chairs and is inspired by the exemplary home environments depicted in Figure 1.1. In addition, a 2.00 × 1.25 m carpet is placed on the ground to act as evaluation scenario. This is a typical example for a restriction area introduced in Section 1.1. Since the focus of this experiment is on subjective ratings of participants and not on usability criteria that could be affected by different evaluation scenarios, a single typical restriction area is sufficient. As mobile robot platform, we employ the mobile robot and its capabilities described in Subsection 3.4.2.

Procedure

Each participant of the experiment enters the lab environment and is briefly introduced to the topic by an experimenter, i.e. describing the problem and explaining restriction areas. In particular, the participant's task in the experiment is introduced, i.e. the exclusion of a carpet area from the mobile robot's workspace. For this purpose, one of the three interaction methods is randomly selected, and the experimenter explains how to use the interaction method for the given task. The explanation

includes a short introduction to the different states of the system, how to switch between the states and how to employ the user interface to specify a virtual border. Afterwards, a participant gets some time to get familiar with the user interface, e.g. handling the tablet or guiding the mobile robot, which takes a maximum of five minutes. Subsequently, a participant is asked to specify a virtual border around the fixed-placed carpet on the ground. This procedure is repeated for each interaction method. Hence, this within-subject design allows every participant to compare the different interaction methods and user interfaces. After practically evaluating the interaction methods, a participant is asked to fill a post-study questionnaire concerning the learnability and user experience. The whole experiment including the practical application of the interaction methods and answering of the questionnaire takes between 20 and 25 minutes per participant.

Participants

This experimental procedure described above is conducted by a total of 25 participants (18 male, 7 female) with a mean age of $M = 31.92$ years and a standard deviation of $SD = 11.54$ years. The age group ranges from 16 to 56 years (16-29 years: 13 participants, 30-39 years: 6 participants, 40-49 years: 2 participants, 50-59 years: 4 participants). All participants are recruited from the local environment by word of mouth. They rate their experience with robots on an 11-point Likert item with a mean of $M = 3.44$ and a standard deviation of $SD = 3.20$. The item ranges from *no experience* (0) to *highly experienced* (10). Thus, the participants represent humans with minimal to moderate experience with robots and comprise some users that own a mobile robot, such as a vacuum cleaning robot. However, these robots are only deployed in their home environments according to the manual without knowledge of how they internally work. Hence, the participants are judged as good representatives for humans in the scope of this thesis.

Measurement Instruments

To measure the learnability and user experience in this experiment, the participants of the experiment are asked to fill a post-study questionnaire[12]. In addition to general information, such as age, gender and experience with robots, the questionnaire comprises different statements concerning learnability and user experience. Each statement can be rated on a 5-point Likert item with numerical response format (LIKERT, 1932). Such an item is a bipolar scaling method expressing positive or negative response to a statement. A neutral response on a 5-point Likert item is indicated by the central value (3). Thus, this instrument is appropriate to measure a negative or positive attitude concerning a statement as defined in Section 1.3. The design of the questionnaire is inspired by the questionnaire of ROUANET *et al.* (2013), who used a similar questionnaire to measure the usability

[12] Although a continuous improvement with respect to certain usability requirements is necessary to prove a good learnability, we only measure the subjective attitude in this chapter due to practical reasons. However, this is sufficient since Objective 1 requires an acceptable learnability, which is provable with this instrument.

and user experience of different human-robot interfaces for learning visual objects. Our questionnaire consists of the following statements (translated from German)[13]:

1. It was easy to learn the handling of the user interface (1 = hard, 5 = easy).
2. I had problems to define the virtual borders (1 = big problems, 5 = no problems).
3. It was intuitive to define the virtual borders (1 = not intuitive, 5 = intuitive).
4. It was comfortable to define the virtual borders (1 = uncomfortable, 5 = comfortable).
5. I liked the feedback of the user interface (1 = bad/no feedback, 5 = good feedback).
6. Overall, it was pleasant to use the user interface (1 = unpleasant, 5 = pleasant).

The first item ($S1$) corresponds to the learnability, while the other items ($S2$-$S6$) are associated with the user experience. This is composed of multiple items to cover different aspects as mentioned in Section 1.3. Additionally, a participant is asked which interaction method he or she prefers for the given task, which is another indicator for both aspects. The participant can give multiple responses allowing the selection of none, one or multiple user interfaces. Finally, a participant has the possibility to give comments or reasons for a rating.

Analysis & Results

The answers to the questionnaire are presented in Figure 3.9a showing the mean ratings and standard deviations per statement. All interaction methods achieve a good value for the learnability ($S1$), while only the AR interaction method achieves a good level on all aspects of the user experience ($S2$-$S6$). The laser pointer approach achieves acceptable values on all aspects of the user experience, except of $S3$ dealing with the intuitiveness, where it reaches a good level ($M = 4.20$, $SD = 0.96$). The baseline is in most cases acceptable, except of an unacceptable value ($M = 2.92$, $SD = 1.57$) for the feedback ($S5$).

To evaluate the performance of the proposed interaction methods compared to the baseline, we test for statistical differences between the interaction methods[14]. For this purpose, we perform a repeated measures analysis of variance (ANOVA) on the statements $S1$-$S6$. We choose this statistical method because it allows the comparison of group means on data acquired in studies with a within-subject design. This is the case with this experiment, as one participant evaluates each

[13] For reasons of clarity, we changed the position of the first item in the following enumeration to better separate the aspects of learnability and user experience. In the questionnaire handed out to the participants, this item was at the fourth position.

[14] If we report results of statistical tests in the experiments, we always consider a significance level of $\alpha = 0.05$. Moreover, we refer the reader to the following website for an overview of statistical tests and their characteristics: `https://www.methodenberatung.uzh.ch/de.html` [Accessed: 26.03.2020]

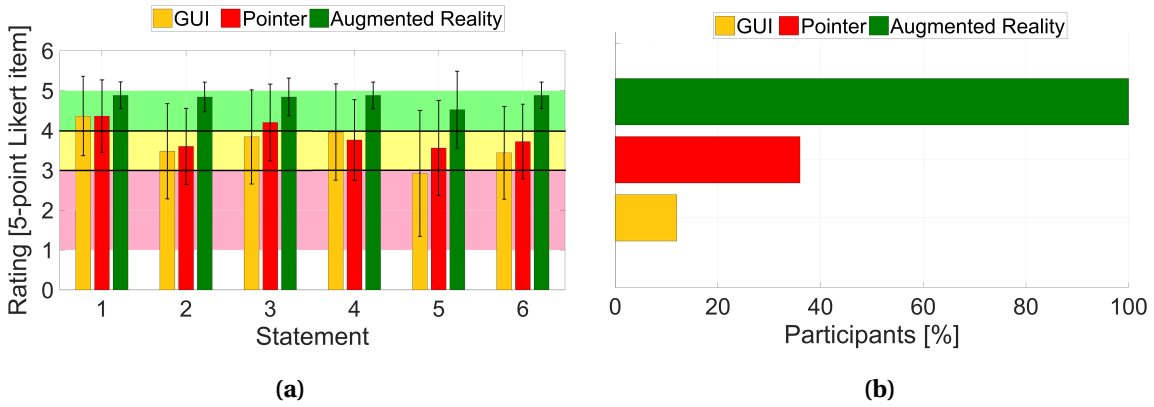


Figure 3.9: (a) Results (mean and standard deviation) of the answers to the questionnaire on a 5-point Likert item per statement. The background colors indicate the quality levels ranging from unacceptable (red) to acceptable (yellow) and good (green). (b) Overall preferences of the participants for an interaction method.

interaction method. The ANOVA is a parametric method that assumes no outliers, a normal distribution of the data and sphericity under ideal conditions. Interpreting corresponding boxplots, our data only contain a few outliers (1.5× interquartile range). However, we do not exclude them from further processing because these are no measurement errors but instead *real* outliers, i.e. participants really rated the interaction method in this way. Thus, it is a legitimate reason to not exclude them from further processing. Moreover, due to the discrete nature of Likert-item data, they violate the assumption of normality. Nonetheless, this can be neglected if there are at least 25 participants involved in the experiment, which is the case for this experiment[15]. Finally, we perform Mauchly's sphericity tests to check for the last assumption of an ANOVA. In case of a violation of sphericity, we perform a Greenhouse-Geisser adjustment to correct the violation. This applies to statements $S1$, $S2$ and $S6$.

The results of the statistical analysis are summarized in Table 3.1 where $F(df_1, df_2)$ denotes the F-distribution with its two parameters df_1 and df_2 that depend on the number of interaction methods and participants. The results show that there is no significant difference for the learnability ($S1$) but for all aspects of the user experience ($S2$-$S6$). However, the results only reveal that there is a difference between the interaction methods but not which interaction methods. Therefore, we perform Bonferroni-adjusted post-hoc tests on the significant results. Regarding statement $S2$, participants have significantly less problems ($p < 0.001$) when employing the AR interaction method ($M = 4.84$, $SD = 0.37$) compared to the baseline ($M = 3.48$, $SD = 1.19$), but there is no difference between the laser pointer ($M = 3.60$, $SD = 0.96$) and GUI interaction method[16]. Similarly, participants rate

[15] `https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/unterschiede/zentral/evarianzmessw.html` [Accessed: 26.03.2020]

[16] For reasons of readability, we mean statistically significant differences when we report a difference with a p-value.

the intuitiveness higher (p = 0.003) when using the AR (M = 4.84, SD = 0.47) compared to the GUI (M = 3.84, SD = 1.18) interaction method, but no improvement with respect to the baseline is identified when using the laser pointer (M = 4.20, SD = 0.96). Additionally, the interaction method with the AR tablet features the highest comfort (M = 4.88, SD = 0.33) with a significant difference (p = 0.002) compared to the baseline (M = 3.96, SD = 1.21). However, there is no difference between laser pointer (M = 3.76, SD = 1.01) and baseline interaction method. The AR application on the Tango tablet also achieves the highest rating for the feedback system (M = 4.52, SD = 0.96), which is significantly better ($p < 0.001$) than the GUI (M = 2.92, SD = 1.58) approach. The use of a laser pointer (M = 3.56, SD = 1.19) does not improve the feedback compared to the baseline. Finally, satisfaction is also leaded by the AR tablet (M = 4.88, SD = 0.33) followed by the laser pointer (M = 3.72, SD = 0.94) and GUI (M = 3.44, SD = 1.16). This results in a significant difference between the AR and GUI interaction method ($p < 0.001$).

Table 3.1: Statistical results of the answers to the questionnaire. A * indicates a significant result.

Statement	Aspect	F-statistic	p-value
S1	Learnability	$F(1.46, 35.12) = 3.43$	$p = 0.057$
S2	Problems	$F(1.49, 35.64) = 14.90$	$p < 0.001$*
S3	Intuitiveness	$F(2, 48) = 7.24$	$p = 0.002$*
S4	Comfort	$F(2, 48) = 11.03$	$p < 0.001$*
S5	Feedback	$F(2, 48) = 10.58$	$p < 0.001$*
S6	Satisfaction	$F(1.60, 38.47) = 20.07$	$p < 0.001$*

These results are consistent with the user preferences for an interaction method as shown in Figure 3.9b. All participants prefer the AR interaction method for the given task followed by the laser pointer approach (9 out of 25). The baseline approach was only selected by three participants.

Discussion

Regarding the learnability ($S1$), the results show that there is no significant difference between the interaction methods, but that all interaction methods feature at least an acceptable learnability, i.e. a positive personal attitude (> 3 (50%)). Thus, we conclude that the results support Hypothesis 1.2 but not Hypothesis 1.1. A reason for this result could be that all user interfaces are familiar to the participants, i.e. tablets and laser pointer. Hence, humans typically know how to interact with these devices, which leads to the good ratings (> 4 (75%)) for all interaction methods. Therefore, there is only minimal room for improvement, so that there are no significant differences between the approaches.

In case of the user experience ($S2$-$S6$), the interaction method based on AR is best rated on all aspects. There is always a significant improvement of the AR interaction method with respect to the

baseline. Moreover, the ratings are always above 4 (75%) indicating a good user experience. In contrast to this, the baseline's ratings mostly show an acceptable user experience (> 3 (50%)) except of the statement concerning the feedback ($S5$) where an unacceptable rating is achieved ($M = 2.92$, $SD = 1.58$). A reason for this could be that participants do not have the possibility with this interaction method to see if their user-defined virtual border is at the position where they want it to be. This is caused by a correspondence problem between points on the OGM and in the physical environment. Using the AR interaction method, a participant does not have this correspondence problem and thus no lack of feedback. This correspondence problem also has a negative effect on other aspects, such as problems ($S2$) or satisfaction ($S6$).

Focusing on the other proposed interaction method based on a laser pointer, this achieves at least acceptable ratings on all user experience aspects, but there are no significant differences compared to the baseline. However, the positive ratings show that the idea using a laser pointer for a gesture-based interaction is positively accepted by the participants. Nonetheless, two major drawbacks of this interaction method are revealed during the experiment, which negatively affect the user experience. (1) Since the interaction requires a direct line of sight between human and robot to transfer spatial information and due to the mobile robot's limited field of view, the mobile robot has to move to follow the laser spot on the ground. As opposed to the robot-independent interaction methods, this takes additional time, which negatively affects the user experience. (2) Moreover, the limited mobile robot's on-board feedback capabilities have a negative effect on the feedback ($S5$). Nonetheless, the acceptable rating on this aspect underlines the appropriateness of the feedback provided using the laser pointer and the robot's on-board capabilities. In summary, we conclude that Hypothesis 2.1 and 2.2 are supported by the results of the experiment since both proposed interaction methods feature an at least acceptable user experience and the AR approach is significantly better rated than the baseline method.

3.4.5 Experiment 2: Usability

After evaluating the learnability and user experience, this experiment is intended to evaluate the usability criteria, i.e. completeness, accuracy and interaction time. For this purpose, we conduct another experiment with multiple participants, but also with multiple evaluation scenarios. Compared to Experiment 1, this experiment involves less participants but an increased number and complexity of evaluation scenarios.

Independent Variables

We manipulate the same independent variable as in the previous experiment, i.e. the interaction method. Thus, the value of this variable can be one of the three interaction methods.

Hypotheses

The objective of this experimental evaluation is the test of the following hypotheses concerning usability criteria defined in Section 1.3. These hypotheses are derived from Objective 1:

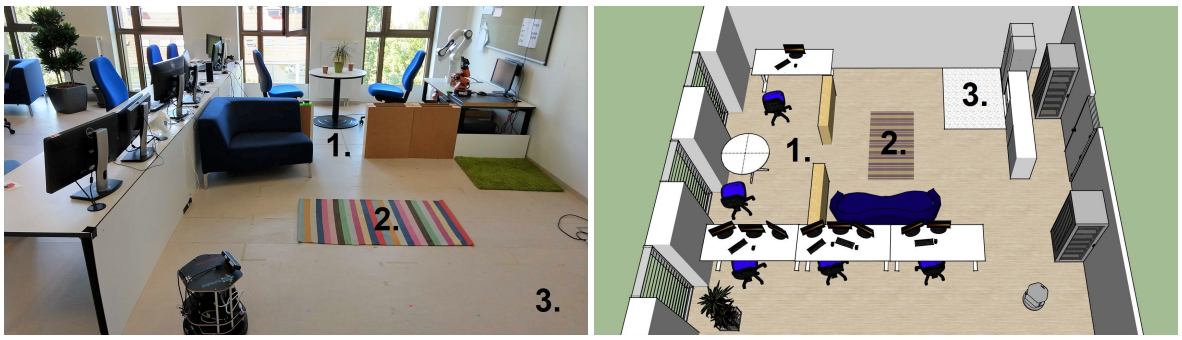
- **Hypothesis 1:** At least one of the proposed interaction methods achieves (1.1) a better completeness than the current state-of-the-art solution based on sketching restriction areas on a GUI and (1.2) an acceptable completeness for our problem.
- **Hypothesis 2:** At least one of the proposed interaction methods achieves (2.1) a better accuracy than the current state-of-the-art solution based on sketching restriction areas on a GUI and (2.2) an acceptable accuracy for our problem.
- **Hypothesis 3:** At least one of the proposed interaction methods achieves (3.1) a better interaction time than the current state-of-the-art solution based on sketching restriction areas on a GUI and (3.2) an acceptable interaction time for our problem.

Setup

To test the three hypotheses, we extend our experimental setup compared to the previous experiment to cover more scenarios for restriction areas and a larger environment. This is necessary because the usability criteria can change with different complexities of the evaluation scenarios, e.g. length and shape of the virtual border. Therefore, we perform the experiment in our 10 × 8 m lab environment, which is furnished like a realistic indoor home environment including components, such as free space, walls, plants, sofas, tables and chairs. The choice of restriction areas is motivated by the examples introduced in Section 1.1, i.e. (1) privacy zones, (2) carpets and (3) dirty areas. These scenarios cover different complexities of a restriction area. To simulate different rooms for a privacy zone, we integrate adjustable walls with a height of 0.5 m. These walls are high enough so that the robot cannot overlook them. Besides, we place a carpet on the ground and some dirt (paper snippets) in one area of the environment as basis for the evaluation scenarios. An image and a 3D sketch of a part of the environment covering the restriction areas are depicted in Figure 3.10. As mobile robot platform, we employ the mobile robot and its capabilities described in Subsection 3.4.2.

Procedure

After setting up the experimental environment, each participant of the experiment is introduced to the following three evaluation scenarios. They cover both types of virtual borders, i.e. closed and simple polygonal chains, and are good representatives for restriction areas in the scope of this thesis:



(a) **(b)**

Figure 3.10: (a) Image and (b) 3D sketch of a part of the lab environment. The three evaluation scenarios are numbered, and the mobile robot's initial pose is depicted in the bottom right of the sketch.

1. **Room exclusion:** A human wants the mobile robot to not enter a certain room due to privacy concerns. For this purpose, the human has to specify a virtual border separating the room from the rest of the environment. The length of the (simple) polygonal chain $\mathcal{P}$ is 0.70 m, and the restriction area has a size of approximately 8.00 m^2. Due to this simplicity, we consider it as a simple restriction area.

2. **Carpet exclusion:** A human wants the mobile robot to circumvent a carpet (2.00 × 1.25 m) while working, which is similar to the setup in Experiment 1. To this end, the participant has to specify a polygon with at least four corner points around the carpet and has to define the inner area as restriction area. Due to the length and shape of the virtual border, this restriction area is more complex compared to the previous restriction area.

3. **Spot cleaning:** A human wants his or her vacuum cleaning robot to perform a spot cleaning in a corner of a room. Hence, he or she specifies a polygonal chain around the area and assigns the rest of the room as restriction area. This dirty area is indicated by paper snippets on the ground. The polygonal chain has a length of 3.60 m and encompasses an area of 3.20 m^2. Since this is a restriction area with moderate length that can be specified with a simple polygonal chain, this restriction area has a moderate complexity.

After introducing a participant to the three evaluation scenarios, an interaction method is randomly selected and explained by an experimenter. The explanation includes a short introduction to the different states of the system, how to switch between states and how to employ the user interface for the given evaluation scenarios. Afterwards, a participant has approximately five minutes to get familiar with the interaction method, e.g. handling the user interface or guiding the mobile robot. Following this introductory phase, a participant performs a run for each scenario, i.e. specifying a virtual border for each scenario. The order of the scenarios is randomized. At the beginning of each

run, the mobile robot's initial pose is set to a predefined pose to allow the comparison between the results, especially in case of the robot-dependent interaction method. The shortest paths from the initial pose to the restriction areas in the three scenarios are between 2.50 and 5.40 m (Scenario 1: 5.40 m, Scenario 2: 2.50 m and Scenario 3: 3.00 m). This experimental procedure is performed for all interaction methods. Afterwards, general information about a participant is collected including age, gender and experience with robots and tablets. An experiment with a single participant takes between 15 and 20 minutes in total.

Participants

This experimental procedure is performed by two different user groups. The first user group performs the experiment employing the robot-independent interaction methods, i.e. GUI and AR, while the second user group performs the experiment with two robot-dependent interaction methods using a laser pointer[17]. This distinction between the user groups is made because there were originally two different experiments with the same setup and procedure performed, and we want to compare their results with each other. Both user groups represent humans with a moderate experience with robots ($M = 2.87$ and $M = 3.20$) rated on a 5-point Likert item, and the participants' ages match the intended age of users in the scope of this thesis. In case of the first user group evaluating the robot-independent methods, participants additionally rate their experience with tablets on a 5-point Likert item with $M = 3.93$. Thus, they have an extended knowledge indicating a familiarity with this common consumer device. In summary, both user groups represent humans in the scope of this thesis as described in Section 1.2 and are thus comparable to each other. The characteristics of both user groups are summarized in Table 3.2.

Table 3.2: Characteristics of the experiment's user groups.

	User group 1	User group 2
Evaluated methods	Robot-independent	Robot-dependent
Number	15	15
Gender	10 male, 5 female	11 male, 4 female
Age	$M = 30.33$, $SD = 11.24$, 19-59 years	$M = 28.80$, $SD = 11.44$, 17-55 years
Robot experience	$M = 2.87$, $SD = 1.19$	$M = 3.20$, $SD = 1.37$
Tablet experience	$M = 3.93$, $SD = 0.59$	-

[17] Here we only report the results of the interaction method based on a laser pointer as described in Subsection 3.3.1. The results of the other laser pointer-based interaction method (incorporating a smart home environment) are presented in the next chapter.

Measurement Instruments

The hypotheses of this experimental evaluation deal with the completeness, accuracy and interaction time. Thus, these criteria have to be quantified. Regarding the completeness, an experimenter documents during the experiment if a participant can successfully specify a virtual border for an evaluation scenario. An interaction process is successful if a participant can correctly specify the virtual border points $\mathcal{P}$ (independent of the accuracy) and a seed point $\boldsymbol{s}$ for a restriction area. The number of successful runs is used to assess the completeness of an interaction method. To this end, we calculate the ratio between the number of successful runs and the total number of runs for each interaction method and scenario.

While the completeness indicates how successful an interaction process is accomplished, the accuracy additionally measures how accurate a virtual border is defined. For this purpose, we consider the overlap between a user-defined $UD \subset \Omega(M)$ and a ground truth $GT \subset \Omega(M)$ virtual border[18]. A user-defined virtual border UD results from an interaction process and contains all cells of the map M that have been modified during the interaction process. In contrast to this, a ground truth virtual border GT is manually created before the interaction process and contains all cells of the map M that should be modified by a human in the interaction process. To determine the overlap, we calculate the Jaccard similarity index (JSI) between two virtual borders GT and UD. This calculates the ratio between the intersection and union of both virtual borders:

$$JSI(GT, UD) = \frac{|GT \cap UD|}{|GT \cup UD|} \in [0, 1] \tag{3.10}$$

The third criterion to be evaluated is the interaction time, which is the time needed to restrict a mobile robot's workspace. The time measurement starts with a sign of the experimenter and ends with the integration of the virtual border into the prior map of the environment M.

Analysis & Results

Completeness The results of the completeness are summarized in Table 3.3. All interaction methods achieve an at least acceptable completeness on average, i.e. completeness > 90%. While the laser pointer method features the same completeness as the baseline (91.1%), the proposed AR method even achieves a good value of 97.8%. Moreover, the completeness is worst in Scenario 3 for all interaction methods, whereas there is no difference between Scenarios 1 and 2.

[18] In the context of the accuracy evaluation, we consider a virtual border V as a set of cells in the domain of the environment's map $V \subset \Omega(M)$ instead of a triple $V = (\mathcal{P}, \boldsymbol{s}, \delta)$.

Table 3.3: Results of the completeness of the interaction methods.

Method	Scenario 1	Scenario 2	Scenario 3	Mean
GUI	93.3%	93.3%	86.7%	91.1%
Pointer	93.3%	93.3%	86.7%	91.1%
AR	100.0%	100.0%	93.3%	97.8%

Accuracy The accuracy results of the experiment are visualized in Figure 3.11a as colored boxplots. The AR method achieves a good accuracy in Scenario 1 (M = 98.2%, SD = 1.4%) and Scenario 3 (M = 86.8%, SD = 6.9%) and an acceptable accuracy in Scenario 2 (M = 73.7%, SD = 5.1%). The same applies for the laser pointer approach (Scenario 1: M = 97.1%, SD = 0.9%, Scenario 2: M = 77.3%, SD = 3.0% and Scenario 3: M = 88.8%, SD = 8.5%). Moreover, the accuracy decreases for all interaction methods with an increase of the complexity of the restriction areas. Especially, the baseline's accuracy drops to an unacceptable level in Scenario 2 (M = 57.3%, SD = 16.9%). Furthermore, the proposed interaction methods feature a smaller deviation compared to the baseline.

Figure 3.12 visualizes some qualitative accuracy results for the different interaction methods and evaluation scenarios. The first row depicts the three different ground truth virtual borders GT colored in yellow, while the physical environment remains in black, white and gray. A red arrow indicates the mobile robot's initial pose during each run of the experiment. The other rows show the overlapping of ground truth GT and user-defined UD virtual borders for the three interaction methods. Red and green cells visualize virtual borders specified by a human in an interaction process UD. The intersection of ground truth and user-defined areas $GT \cap UD$ is colored in green. Thus, these areas are correctly specified by the human. In contrast to these, red areas indicate user-defined areas that are not part of the ground truth virtual border $UD \setminus GT$, and yellow areas are ground truth areas not covered by the human in the interaction process $GT \setminus UD$. Hence, these are missed by the human in the interaction process. The union area $GT \cup UD$ is enclosed by a blue contour. Thus, the JSI can be visually interpreted as the ratio between the green area and the area enclosed by the blue contour.

In order to identify statistical differences between the proposed interaction methods and the baseline, we first visually inspect the boxplots for outliers, which is an assumption of parametric statistical tests. Since there is at most a single outlier per scenario and interaction method, this assumption is met. Afterwards, we perform Shapiro–Wilk tests to check for normality of the data. Due to significant results of the tests, except of Scenario 2, the data violate this assumption. Therefore, we prefer non-parametric statistical tests to compare the interaction methods since these do not assume normality of the data. However, since we have two different user groups and only one user group evaluates the baseline method, we perform a Wilcoxon signed-rank test to compare the AR with the baseline method and a Mann-Whitney U test to compare the laser pointer with the base-

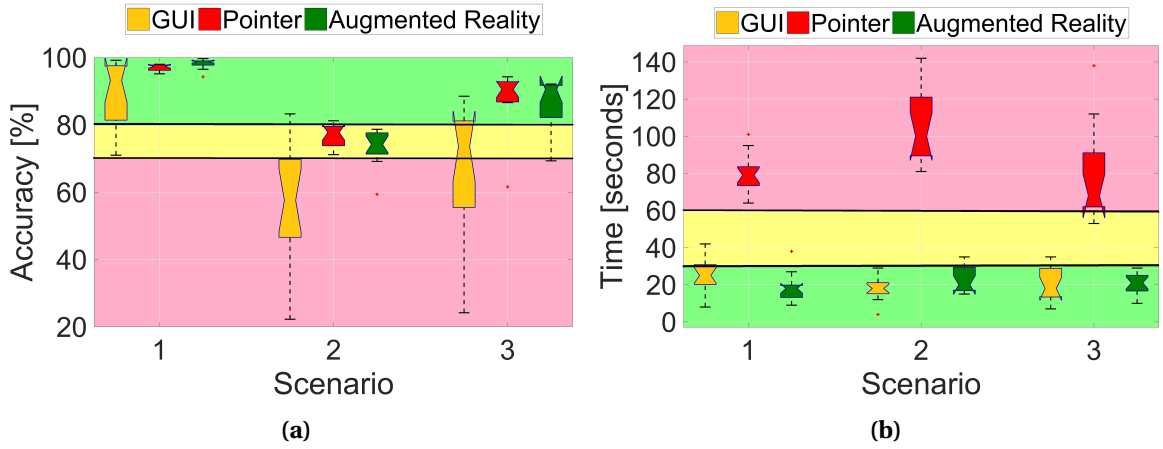


Figure 3.11: Results of the (a) accuracy and (b) interaction time of the interaction methods. The background colors indicate the quality levels ranging from unacceptable (red) to acceptable (yellow) and good (green).

line method. Both tests differ in the assumption concerning the design of the data acquisition, i.e. within-subject or between-subject design. The statistical results of the tests are summarized in Table 3.4. There is a significant improvement in accuracy when using the proposed interaction methods compared to the baseline, except for the laser pointer approach in Scenario 1.

Table 3.4: Statistical results concerning the accuracy of a method compared to the baseline. A * indicates a significant result.

	Scenario 1		Scenario 2		Scenario 3	
Method	Statistic	p-value	Statistic	p-value	Statistic	p-value
Pointer	$U = 56$	$p = 0.056$	$U = 16$	$p < 0.001$*	$U = 13$	$p < 0.001$*
AR	$Z = -2.67$	$p = 0.005$*	$Z = -2.42$	$p = 0.013$*	$Z = -3.30$	$p < 0.001$*

Interaction Time In case of the interaction time, Figure 3.11b depicts the results of the experiment. Both robot-independent interaction methods, i.e. GUI and AR, feature a good interaction time with a small deviation in all scenarios, whereas the proposed laser pointer method only achieves an unacceptable performance in all scenarios.

To statistically verify this difference between the interaction methods and select an appropriate statistical test, we check the data for outliers by visual inspection of the boxplots and for normality running Shapiro–Wilk tests. There is at most a single outlier per scenario and interaction method, and the Shapiro–Wilk tests do not become significant. Therefore, we choose paired t-tests for the

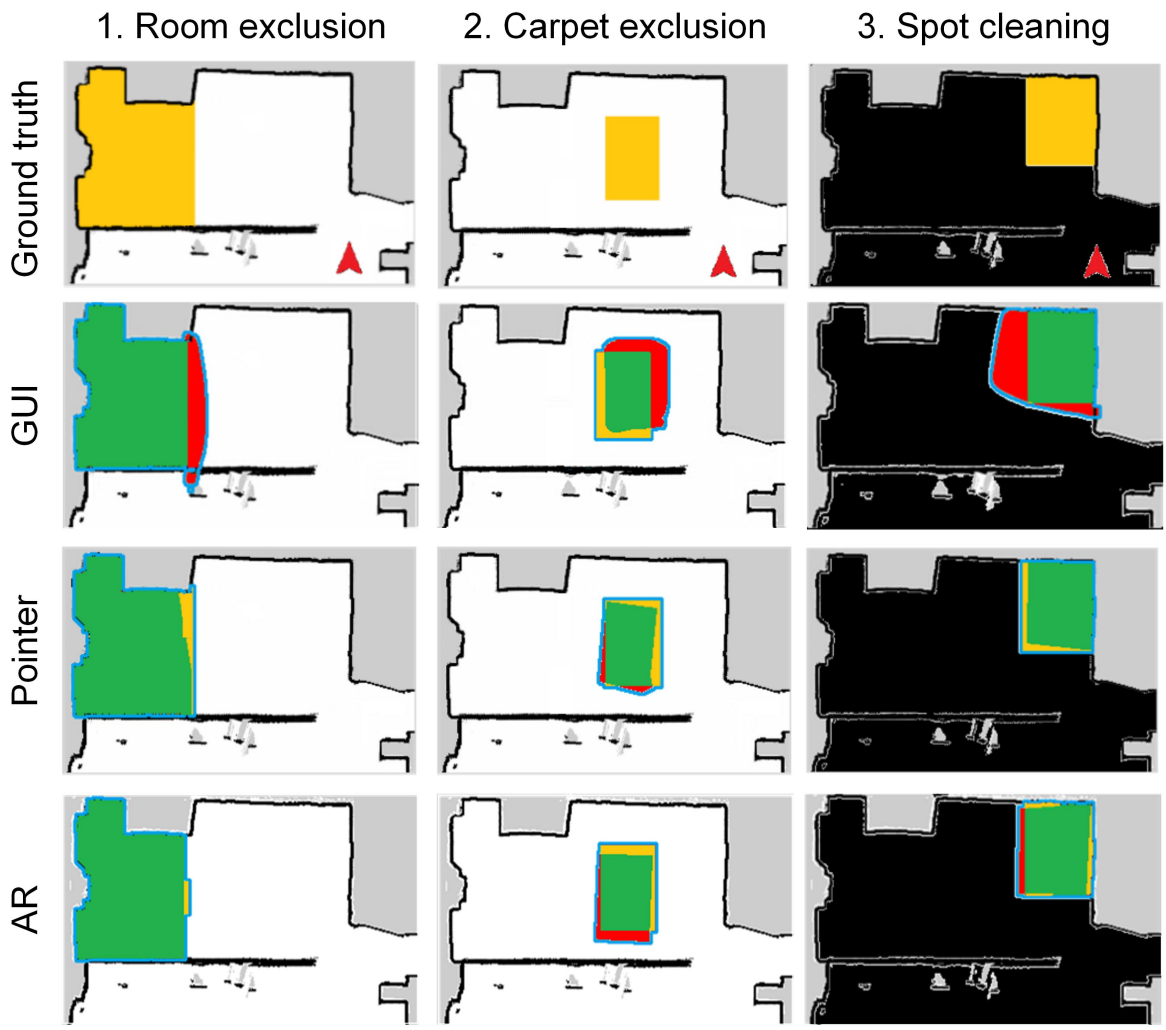


Figure 3.12: Visualization of the accuracy results. The first row shows the three different ground truth maps of the evaluation scenarios, while the following rows visualize the overlap between the ground truth and user-defined virtual borders depending on the interaction method. The maps are only colored due to visualization purposes.

comparison between the AR and baseline method and unpaired t-tests for the comparison between the laser pointer and baseline method. The distinction between paired and unpaired t-test is necessary due to the different user groups. Finally, an unpaired t-test assumes homogeneity of variances. Thus, a Levene's test is performed to compare the variances of the data. If these differ and the test becomes significant, a Welch's t-test is performed instead of an unpaired t-test, which is similar but adjusts the degrees of freedom of the t-distribution. This is the case for Scenarios 2 and 3 when comparing with the laser pointer method.

The results of the statistical tests are summarized in Table 3.5. As already observed in the boxplots, the statistical tests become significant for all scenarios when employing the laser pointer interac-

tion method. While the interaction time of the laser pointer method is longer than the baseline's interaction time in all scenarios, the AR performs better than the GUI method in Scenario 1 and worse in Scenario 2. The paired t-test does not become significant for Scenario 3. Thus, there is no difference between GUI and AR approach in this case.

Table 3.5: Statistical results concerning the interaction time of a method compared to the baseline. A * indicates a significant result.

	Scenario 1		Scenario 2		Scenario 3	
Method	Statistic	p-value	Statistic	p-value	Statistic	p-value
Pointer	$t(28) = -16.24$	$p < 0.001$*	$t(17.30) = -17.46$	$p < 0.001$*	$t(17.71) = -8.57$	$p < 0.001$*
AR	$t(14) = 3.08$	$p = 0.008$*	$t(14) = -2.78$	$p = 0.015$*	$t(14) = -0.37$	$p = 0.714$

Discussion

The results of the completeness show that all three interaction methods feature an at least acceptable completeness. Moreover, the AR approach achieves a good value on average and outperforms the other methods in each scenario. Thus, the results support Hypothesis 1.1 and 1.2 of this experiment. The incorrect runs were mainly caused by a wrong definition of the seed point s, especially in Scenario 3 where participants placed the seed point s inside the spot cleaning area. As opposed to the other scenarios, this is incorrect since the mobile robot should work inside the area. However, most of the participants noticed the mistake on their own after performing the experiment. In addition, there was a single participant who was not able to correctly specify the virtual border points $\mathcal{P}$ with the GUI approach due to a lack of orientation in the environment and the OGM displayed on the tablet's GUI.

Regarding the accuracy, both proposed interaction methods feature a good accuracy in two of the three evaluation scenarios and an acceptable level in one scenario, which leads to a good accuracy on average. Furthermore, they are significantly more accurate than the baseline method, that does not achieve an acceptable accuracy in all scenarios. Moreover, a small deviation for both proposed interaction methods indicates a constant high accuracy independent of a participant. In summary, we conclude that the results support Hypothesis 2.1 and 2.2. Additionally, the results suggest that the accuracy decreases when the scenarios become more complex, i.e. longer polygonal chain $\mathcal{P}$ or more complex shape.

In case of the interaction time, the robot-independent interaction methods achieve a good performance, whereas the laser pointer method is unacceptable with respect to this criterion. It does not fall below the threshold of 60 seconds for an acceptable interaction time. This is due to the fact that the interaction method depends on the velocity of the mobile robot and its field of view. Thus, a

human first has to guide the mobile robot to the restriction area and then has to specify the virtual border by guiding the robot along the virtual border polygon $\mathcal{P}$. This takes additional time, that is not necessary for the robot-independent interaction methods. Therefore, the longer the virtual border polygon $\mathcal{P}$ and the larger the distance between the mobile robot's initial pose and the restriction area, the longer the interaction takes between human and robot. However, since the AR method achieves a good interaction time, but not significantly better than the baseline, we conclude that the results support Hypothesis 3.2, but not Hypothesis 3.1.

3.4.6 Experiment 3: Advanced Usability

Since the results of the previous experiment suggest that there could be a relationship between the virtual border length, i.e. the length of the virtual border polygon $\mathcal{P}$, and the accuracy and interaction time, this experiment aims to further investigate this aspect. To this end, we conduct an experiment with multiple virtual borders with different shapes and sizes. This advanced usability evaluation is important to comprehensively assess these criteria and determine an appropriate quality level.

Independent Variables

We manipulate the same independent variable as in the previous experiment, i.e. the interaction method. Thus, the value of this variable can be one of the three interaction methods.

Hypotheses

The objective of this experimental evaluation is the test of the following hypotheses concerning the accuracy and interaction time:

- **Hypothesis 1:** There is a relationship between the length of a virtual border and the accuracy.
- **Hypothesis 2:** There is a relationship between the length of a virtual border and the interaction time.

Setup

To test both hypotheses, we perform the experiment in the same physical environment as described in Experiment 1. However, we further increase the number of evaluation scenarios covering restriction areas with different shapes and sizes. Therefore, we create a dataset containing ten different ground truth maps of our lab environment with manually integrated virtual borders. The lengths of the virtual borders range from 4 to 13 m, and their shapes are convex and non-convex. Thus, this

dataset covers a large range of restriction areas with different sizes as described in the scope of this thesis. The details of the dataset are summarized in Table 3.6. These evaluation scenarios are conveyed to the human in form of small markers placed on the ground to indicate the restriction areas. For reasons of clarity, only a single restriction area is marked on the ground at the same time.

Table 3.6: Characteristics of the dataset. Length indicates the length of the virtual border polygon $\mathcal{P}$ and the area indicates the size enclosed by the polygon. Corners indicate the minimal number of points necessary to specify the polygon.

	Scenario									
	1	2	3	4	5	6	7	8	9	10
Length [m]	4	5	6	7	8	9	10	11	12	13
Area [m^2]	1.00	1.50	2.25	3.00	3.75	3.50	5.25	5.50	5.75	7.00
Corners	4	4	4	4	4	4	4	6	8	8

Procedure

After setting up the first of ten evaluation scenarios on the ground and an introductory phase including an explanation of the experiment's objective, a participant is asked to specify the restriction area employing one of the three interaction methods. For each evaluation scenario, a participant performs five runs to introduce some variation into the interaction process. Different starting positions of the mobile robot and the participant's initial position are considered as variations. After performing five runs for the first scenario, this procedure is performed for the other nine scenarios resulting in 50 runs per interaction method. This procedure again is repeated for the other interaction methods resulting in a total time of approximately 240 minutes per participant.

Participants

Due to the extreme expenditure of time for a participant, this experiment is only conducted by a single participant. However, this is not crucial since we want to investigate the effect of the virtual border length on the accuracy and interaction time. The relative strength of an effect could depend on a participant, but the identification of a general effect is independent of a participant. Moreover, the participant gains experience during the experiment, which compensates individual differences between humans. Therefore, the experiment is conducted by a single participant, who is male and 26 years old. He rates his experience with robots and tablets with 4 on a 5-point Likert item. Thus, the participant's experience with robots is slightly above a typical participant in the previous experiment, while the experience with tablets is similar. This corresponds to a human with extended experience with robots and tablets, i.e. a human able to program parts of a robot and tablet.

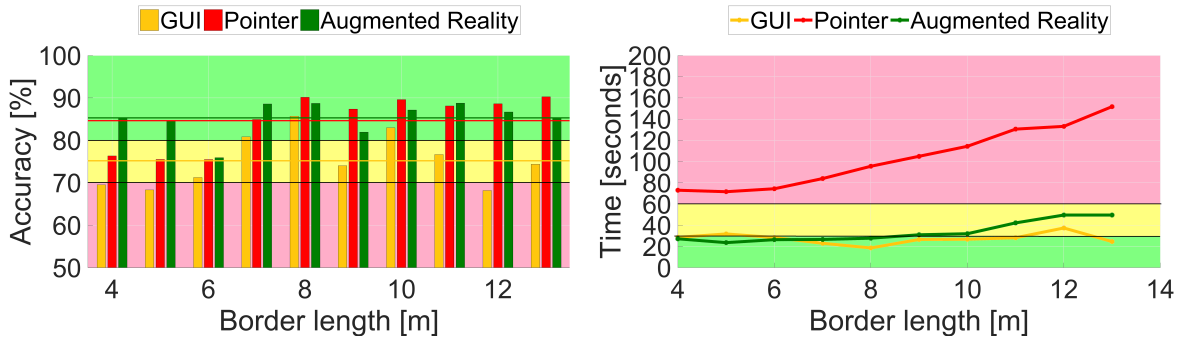


Figure 3.13: Results of the (a) accuracy and (b) interaction time depending on the virtual border length. The background colors indicate the quality levels ranging from unacceptable (red) to acceptable (yellow) and good (green).

Measurement Instruments

After conducting the experimental procedure, we quantify the accuracy of the resulting user-defined with respect to the ground truth virtual borders. To this end, we calculate the JSI as described in Experiment 2. Moreover, we measure the interaction time during each run of the experiment. However, as opposed to Experiment 2, we do not start the time measurement with a sign of an experimenter. Instead, we measure the time between the specification of the first point of the virtual border polygon $\mathcal{P}$ and the final integration of the virtual border into the OGM of the environment M. Thus, we do not include the time needed to guide the mobile robot to the restriction area in case of a robot-dependent interaction method. This time depends on the distance between the mobile robot's initial position and the restriction area and would thus corrupt measurements concerning Hypothesis 2 of this experiment. With this change of the definition of the interaction time, we only measure the actual time needed to specify a certain virtual border.

Analysis & Results

Accuracy The accuracy results of this experiment depending on the length of a virtual border are visualized as bars in Figure 3.13a. A horizontal line indicates the overall mean per interaction method. By visually inspecting the graphic, there seems to be an approximately constant relationship between the accuracy and length of the virtual border. However, in case of the laser pointer approach, there is an increase of accuracy between virtual border lengths of 6 and 8 m. Besides, the accuracies of the proposed interaction methods achieve a good level on average (Pointer: $M = 84.6\%$, AR: $M = 85.3\%$), while the baseline approach reaches an acceptable level ($M = 75.2\%$).

To verify this visual observations, we calculate the Pearson correlation coefficient $\rho \in [-1, 1]$, which measures the linear correlation between two variables, i.e. virtual border length and accuracy. Coefficients of $\rho = -1$ and $\rho = 1$ correspond to negative and positive linear correlations, while a coefficient of $\rho = 0$ indicates linearly uncorrelated data. In addition, a linear regression is performed to calculate the linear slope of the data. The correlation coefficients and the linear slopes resulting from the linear regression are summarized in Table 3.7. In case of the robot-independent interaction methods, there is only a weak linear correlation (GUI: $\rho = 0.158$ and AR: $\rho = 0.270$), whereas a strong positive linear correlation is identified for the laser pointer approach ($\rho = 0.736$)[19]. In conjunction with the low linear slope (GUI: 0.39 %/m and AR: 0.40 %/m), we conclude that the robot-independent interaction methods' accuracy is constant and thus independent of the virtual border length. The slightly higher linear slope of 1.77 %/m for the laser pointer approach is a result of the increase of accuracy between short (< 6.5 m) and long (> 6.5 m) virtual borders.

Table 3.7: Results concerning the linear relationship between the accuracy and virtual border length.

Method	Pearson's ρ	Linear slope [%/m]
GUI	0.158	0.39
Pointer	0.736	1.77
AR	0.270	0.40

Interaction Time In addition to the accuracy, the results of the interaction time depending on the length of a virtual border are depicted in Figure 3.13b. By visual inspection, a linear relationship is revealed for the proposed interaction methods, while the GUI approach seems to be approximately constant. Moreover, the interaction time of the laser pointer method is unacceptable, whereas the interaction time of the robot-independent methods is on the borderline between an acceptable and a good level.

These visual findings are verified when calculating the Pearson correlation coefficient and the linear slope using linear regression. The results are presented in Table 3.8 and show a strong linear correlation (Pointer: $\rho = 0.952$ and AR: $\rho = 0.839$) for the proposed interaction methods. The baseline's interaction time indicates linearly uncorrelated data ($\rho = 0.057$). In addition, the baseline's linear slope is extremely low (0.13 s/m) indicating a constant interaction time and independence of the virtual border length. In contrast to this, the laser pointer and AR interaction methods feature a linear slope of 9.22 s/m and 2.92 s/m. Thus, they are linearly dependent on the virtual border length.

[19] Although there are variations in the description of the strength of a coefficient, there is a general consent that $0 < \rho \leq 0.3$ is a weak, $0.3 < \rho \leq 0.7$ is a moderate and $0.7 < \rho \leq 1.0$ is a strong linear correlation (RATNER, 2009).

Table 3.8: Results concerning the linear relationship between the interaction time and virtual border length.

Method	Pearson's ρ	Linear slope [s/m]
GUI	0.057	0.13
Pointer	0.952	9.22
AR	0.839	2.92

Discussion

The experimental results show that the accuracy is approximately constant and linearly independent of the virtual border length. However, the accuracy of short virtual borders, i.e. with a length up to 6.5 m, is below the accuracy of long virtual borders when considering the robot-dependent interaction method employing a laser pointer. Inside these groups of different lengths, the accuracy is also constant. The reason for this difference is that it is hard for a human to guide the mobile robot on such a small area, which negatively affects the accuracy. This problem does not apply to the robot-independent interaction methods and thus there is no difference between short and long virtual borders. Therefore, we conclude that the accuracy is constant in general, which rejects Hypothesis 1. In the overall context, this supports the results of the previous experiment that demonstrate a good accuracy for the proposed interaction methods and an acceptable accuracy for the baseline approach.

In case of the interaction time, the results reveal a linear relationship with respect to the virtual border length for both proposed interaction methods and an approximately constant interaction time for the baseline, i.e. featuring a minimal linear slope. This constant interaction time is caused by the fact that a human can easily sketch virtual borders on the tablet's screen without moving in the environment. Employing the AR interaction method, a human has to minimally move in the environment to specify a virtual border. Hence, it is slightly slower than the baseline method. In case of the laser pointer approach, a human has to guide the mobile robot, which is restricted by its velocity constraints. Therefore, the interaction time of this method is relatively long compared to the robot-independent methods. However, since a relationship could be identified, the results support Hypothesis 2. In the overall context, these results show that the AR approach features an acceptable to good interaction time up to 13-m-long virtual borders. Since virtual borders are typically no more than approximately 10 m long as described in the scope of this thesis, the good interaction time revealed in Experiment 2 is supported by the results. Similarly, the unacceptable interaction time of the laser pointer approach is also confirmed in this experiment.

3.4.7 Experiment 4: Correctness and Flexibility

This final experiment aims to assess the approaches concerning the remaining requirements, i.e. flexibility and correctness. To this end, we demonstrate the flexibility by arguing from a design point of view and by qualitatively assessing the resulting maps from the previous experiments. Moreover, the correctness is shown in robot navigation scenarios before and after an interaction process.

Hypotheses

The objective of this evaluation is the test of the following hypotheses concerning the flexibility and correctness as defined in Section 1.3:

- **Hypothesis 1:** At least one of the proposed interaction methods achieves an acceptable flexibility for our problem.
- **Hypothesis 2:** At least one of the proposed interaction methods achieves an acceptable correctness for our problem.

Setup & Procedure

To test the first hypothesis, we assess the flexibility of an interaction method in two ways. (1) First, we prove the flexibility of an interaction method by arguing from a design point of view, i.e. we show that the design of a virtual border and the map integration algorithm inevitably lead to a flexible interaction method. We consider an interaction method to be flexible if different and multiple restriction areas can be specified employing the interaction methods. (2) Moreover, we qualitatively assess the resulting map of an interaction process concerning the ability to specify arbitrary virtual borders. For this purpose, we consider the resulting maps from the previous experiments as basis for evaluation. Furthermore, these maps are used to demonstrate the correctness of the interaction methods, i.e. the user-defined virtual borders are correctly integrated into the OGM of the environment M. In addition to this correctness of the map integration algorithm, we show the change of the mobile robot's navigational behavior in simple navigation scenarios. Such a navigation task is a typical subtask when the mobile robot is in autonomous mode and provides services to humans as described in the scope of this thesis. For this purpose, a resulting map of an interaction process is passed to the global path planner of the human-aware navigation framework presented in Figure 3.2. Subsequently, a starting and goal position for the mobile robot described in Subsection 3.4.2 are randomly chosen from the free cells in the OGM. The resulting path between starting and goal position is then analyzed according to the compliance with the user-defined virtual borders. The same navigation scenario is also performed before an interaction process and thus before integrating virtual borders to show the differences between both robot paths.

(a) **(b)**

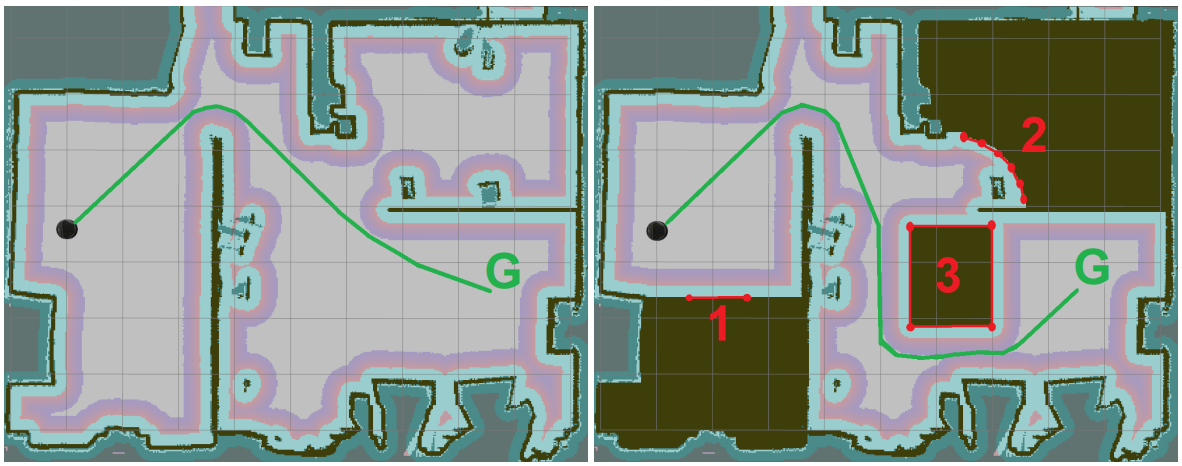
Figure 3.14: (a) Environment with three restriction areas and (b) occupancy grid map of the environment after an interaction process. Numbers are only for visualization purposes.

Analysis & Results

First of all, we prove the flexibility of all interaction methods by considering the underlying data structure of a restriction area, i.e. a virtual border consisting of three components as presented in Subsection 3.2.1. The virtual border points $\mathcal{P}$ are organized as a simple or closed polygonal chain without a restriction of shape or size. Moreover, the seed point s and occupancy probability δ allow the modification of an area of the environment, e.g. exclusion from the mobile robot's workspace. Hence, the components of a virtual border allow the definition of virtual borders with arbitrary shapes and sizes. In addition to this flexibility of a virtual border, the map integration algorithm supports the iterative incorporation of multiple virtual borders into a map of the environment. Thus, multiple virtual borders with different shapes and sizes can be specified demonstrating the flexibility of an interaction method which is based on these components.

In addition to this "flexibility by design", a certain interaction method has to allow the definition of the virtual border components. Since all interaction methods build on the same model of a restriction area, i.e. a virtual border, and employ the same map integration algorithm, there is no difference in terms of flexibility between the interaction methods. Moreover, the qualitative analysis of the resulting maps of the previous experiments reveals that all interaction methods allow the flexible definition of arbitrary virtual borders. An exemplary OGM as a result of an interaction process and the corresponding restriction areas are depicted in Figure 3.14. It shows three virtual borders with arbitrary shapes and sizes, that flexibly restrict the workspace of a mobile robot. Thus, we conclude that the interaction methods are flexible.

The qualitative analysis of the resulting maps of the previous experiments not only demonstrates the flexibility of the interaction methods, but also the correctness, i.e. the user-defined virtual bor-



(a) **(b)**

Figure 3.15: Navigation scenario with corresponding costmaps (a) before and (b) after an interaction process. The right costmap is based on the occupancy grid map shown in Figure 3.14b. A green line indicates the mobile robot's path to the given navigation goal *G*. Numbers and letters are only for visualization purposes.

ders are all correctly integrated into the OGM of the environment *M*. Thus, only the correctness of the employed global planner in the human-aware navigation framework has to be proven, i.e. the mobile robot changes its navigational behavior. As described in Subsection 3.4.2, we employ a navigation function computed with Dijkstra's algorithm as global path planner, which is a typical path planner in mobile robotics and is proven to generate correct paths. However, every other correct global path planner could be employed. An illustration of this correctness is visualized in the costmaps of a navigation scenario shown in Figure 3.15. In contrast to the navigation path before specifying virtual borders, the mobile robot changes its navigational behavior and circumvents the user-defined virtual borders after a successful interaction process. This demonstrates the correctness of the interaction methods.

Discussion

The results of this experiment demonstrate the flexibility and correctness of the interaction methods. Thus, we conclude that both hypotheses are supported by the results. Moreover, there is no difference between the interaction methods in terms of flexibility because all interaction methods are based on the flexible design of a virtual border and the map integration algorithm. Furthermore, the interaction methods are designed to allow the definition of all components of a virtual border and thus they are all flexible. Similarly, since the map integration algorithm correctly incorporates virtual borders and well-established global path planners determine the mobile robot's

navigational behavior, all interaction methods feature an acceptable correctness. In summary, both requirements depend on the overall design of the human-aware navigation framework, especially the modelling of restriction areas and their integration into the environment's map, and not on a certain interaction method, which builds on this framework.

3.5 Summary

In this chapter, we addressed Research Question 1, i.e. how to employ alternative user interfaces to restrict a mobile robot's workspace in a traditional home environment. To this end, we first defined the problem setting and introduced the data structure of a virtual border to model a restriction area. Subsequently, we adapted a state-of-the-art human-aware robot navigation framework by incorporating a map integration algorithm. This algorithm integrates a virtual border into an OGM of the environment and enables the change of the mobile robot's navigational behavior. Afterwards, we proposed two novel interaction methods to allow humans the definition of a virtual border and its components. These interaction methods are based on (1) mediator-based pointing gestures with a laser pointer and (2) an AR application running on a RGB-D tablet. In order to achieve Objective 1, we evaluated our proposed interaction methods with respect to the user requirements of an interaction process in four experiments and in comparison with a baseline interaction method. The experimental results showed that the learnability of all interaction methods is acceptable and does not differ significantly between the approaches. Moreover, the user experience of the AR method achieves a good value and is rated better than the baseline method. There is no difference in the user experience between the laser pointer and the baseline method, which both reach an acceptable quality level. In case of the completeness of an interaction process, all interaction methods feature an acceptable (GUI and Pointer) or good quality level (AR). Furthermore, both proposed interaction methods outperform the baseline in terms of accuracy by reaching a good level on average. Regarding the interaction time, the AR and baseline method feature an equally good performance, but the laser pointer approach is significantly slower. Finally, the results demonstrate that a virtual border is a flexible data structure to model restriction areas and that virtual borders are correctly integrated into the mobile robot's navigation framework.

The results of this chapter concerning the user requirements are summarized in Table 3.9. Since the AR interaction method performs with a good quality level on most of the requirements and significantly better than the state-of-the-art interaction method on average, Objective 1 could be achieved. However, the other proposed interaction method based on a laser pointer revealed two drawbacks during the experiments: (1) a direct line of sight between human and mobile robot is necessary to follow a laser spot on the ground. This leads to an increase of interaction time compared to the robot-independent methods and negatively affects user experience aspects. (2) The limited mobile robot's on-board feedback capabilities only allow simple feedback, but no complex

feedback including spatial information about the result of the interaction process. This has a negative effect on the user experience, especially in terms of the feedback aspect.

Table 3.9: Summary of the interaction methods' performance regarding the user requirements. The symbols indicate an unacceptable (−), acceptable (∘) and good (+) quality level. The ⊕ is used for an acceptable quality level if there is no good quality level defined for a certain requirement. Arrows indicate the change with respect to the baseline method.

Method	Correctness	Flexibility	Completeness	Accuracy	Time	User exp.	Learnability
GUI	⊕	⊕	∘	∘	+	∘	∘
Pointer	⊕ (→)	⊕ (→)	∘ (→)	+ (↗)	− (↘)	∘ (→)	∘ (→)
AR	⊕ (→)	⊕ (→)	+ (↗)	+ (↗)	+ (→)	+ (↗)	∘ (→)

4

Workspace Restriction in a Smart Home

After presenting virtual borders to flexibly model restriction areas and two interaction methods allowing humans to define virtual borders in a traditional home environment, this chapter deals with the second research question, i.e. how can a smart home environment improve the interaction process with respect to the requirements. To this end, we propose a novel interaction method based on the laser pointer approach from the previous chapter, but we also incorporate additional sensors and actuators of a smart home environment into the interaction method. For this purpose, we first describe our smart home design, that is intended to support the interaction process in terms of perceptual and interaction capabilities. Afterwards, we explain how we integrate the smart home components into the interaction method and give details on a cooperative perception consisting of multiple stationary and mobile cameras. This incorporation of smart home components aims to improve the interaction time and user experience while not negatively affecting the other user requirements. Therefore, we finally test these hypotheses in an experimental evaluation.

This chapter's content (in similar or identical form) is mainly based on the publications below:

- SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019a). Interactive restriction of a mobile robot's workspace in a smart home environment. *Journal of Ambient Intelligence and Smart Environments 11*(6), 475–494
- KÖNIG, M. and D. SPRUTE (2019). Verfahren und Robotersystem zur Eingabe eines Arbeitsbereichs. DPMA Patent DE102018125266B3

4.1 Interaction Method Leveraging a Smart Home

As already identified in Section 2.4, smart homes are certain smart environments, that feature additional sensors and actuators integrated into a home environment and that are connected via a network with each other. Thus, they add a perception and interaction component to a traditional home environment, which we aim to leverage in one of the interaction methods proposed in the previous chapter. We distinguish between robot-dependent, e.g. the laser pointer method, and

robot-independent interaction methods, e.g. the augmented reality (AR) application running on a tablet. While interaction methods of the latter category are typically based on powerful user interfaces capable of transferring spatial information and feedback, interaction methods of the former category require the active participation of a mobile robot in the interaction process. For example, the mobile robot's sensors are used to perceive user interactions concerning the spatial information transfer and the on-board actuators are employed to provide feedback about the status of the interaction process. This dependence on the robot's on-board capabilities comes with two major limitations compared to the robot-independent methods as summarized in Section 3.5:

1. **Direct line of sight:** The interaction requires a direct line of sight between human and robot when specifying a virtual border. Thus, due to the limited field of view of the mobile robot's camera, the robot has to move during interaction. However, the mobile robot's velocity is restricted to ensure a smooth and safe motion, which leads to a linear interaction time with respect to the border length. This can entail an unacceptable interaction time for restriction areas with typical lengths up to 10 m as described in Section 1.2.

2. **Limited feedback capabilities:** Furthermore, only limited feedback about the current state of the interaction process can be conveyed using simple on-board capabilities, such as LEDs and non-speech audio sound. Thus, no complex feedback concerning the spatial information of the specified virtual borders can be provided to the human using these communication channels, which negatively affects the user experience.

Additionally, a minor drawback of robot-dependent approaches is the interaction to change between different states of the interaction method, e.g. specifying virtual border points $\mathcal{P}$ or the seed point s. In case of the laser pointer approach, visual codes generated by the laser pointer or push buttons on the mobile robot are provided. But visual codes can be error-prone due to changing light conditions and interaction using buttons requires a human to be in the vicinity of the robot.

Hence, a solution to compensate these limitations of robot-dependent interaction methods could leverage smart home components in the interaction process as they provide additional capabilities for perception and interaction. In contrast to this, there is not directly a benefit of additional smart home components for robot-independent interaction methods because these kinds of interaction methods are based on powerful user interfaces, which already allow the transfer of spatial information and feedback. Additional sensors and actuators of a smart environment could not support the interaction process and could not increase the performance concerning the user requirements significantly in our case. Therefore, we propose a robot-dependent interaction method, which incorporates components of a smart home environment. This interaction method is an extension of the laser pointer method proposed in Subsection 3.3.1 and aims to improve the interaction time to an acceptable level and the user experience to a good level. Moreover, the performance of the other user requirements should remain on an at least acceptable quality level. Thus, turning the laser pointer method from an unacceptable to an acceptable solution for our problem.

4.1.1 Smart Environment Design

For this purpose, we first have to answer Research Question 2.1 dealing with the selection of appropriate sensors and actuators of a smart home environment, that can be used to benefit the interaction process and compensate the identified limitations. Our proposed smart home environment is illustrated in Figure 4.1 and consists of three components, which are available in typical smart home environments as highlighted in the literature review[1]:

1. **Smart camera network:** This component consisting of stationary RGB cameras is intended to increase the perceptual capabilities in addition to the mobile robot's on-board camera. Stationary cameras integrated into the environment (yellow and red fields of view) partially cover certain areas of the environment with their fields of view, while a camera mounted on the mobile robot (blue field of view) can observe areas that are not covered by the stationary cameras due to their installation or occlusions, e.g. under the table. Hence, this combination allows perception even if the stationary cameras' fields of view do not cover all areas of the environment, which is typically the case in smart home environments as described in Section 1.2.

2. **Smart display:** This component is intended to provide expressive and complex feedback to the human by visualizing the progress and result of the interaction process.

3. **Smart speaker:** This component allows the processing of voice commands and aims to facilitate the change of different states of the interaction method. To this end, an intelligent personal assistant, such as Amazon's Alexa[2] or Google Assistant[3], is employed in the background drawing on resources from a cloud infrastructure. We hypothesize that this alternative communication channel facilitates the change of different states of the interaction method because a human does not need to be in the vicinity of the mobile robot.

4.1.2 Human-Robot-Environment Interaction

After presenting the design of a smart home environment, this smart environment needs to be integrated into the laser pointer interaction method with the goal to support the interaction process. For this purpose, we answer Research Question 2.2, which deals with the question of how to realize a cooperation of human, robot and smart environment in the interaction process. In contrast to a traditional home environment, a human now interacts with a network robot system (NRS) consisting of a mobile robot and smart home environment and not exclusively with a mobile robot. Thus, we denote this interaction method as NRS solution.

[1] In this context, the word "smart", e.g. smart camera or smart display, often means that a component is connected via a network with other components and/or that the component possesses some computational power to perform simple algorithms. However, a component itself is not intelligent as in the human sense and cannot make complex decisions.

[2] `https://developer.amazon.com/alexa` [Accessed: 26.03.2020]

[3] `https://assistant.google.com` [Accessed: 26.03.2020]

Figure 4.1: A human defines a restriction area in the environment using a laser pointer. The spot is observed by stationary cameras in the environment (yellow and red field of view) and a mobile camera on a robot (blue field of view). A smart display (top right) provides visual feedback of the complex spatial information, and a smart speaker (not shown here) facilitates interaction to switch between different states of the interaction method.

Spatial Information Transfer

As already pointed out in the introduction, the interaction process for the restriction of a mobile robot's workspace (1) has to allow a transfer of spatial information from human to NRS and (2) has to provide feedback about the interaction process from NRS to human. We realize the first property by allowing a human to specify a virtual border by "drawing" directly in the environment using a common laser pointer similar to the proposed interaction method presented in Subsection 3.3.1. However, instead of directly interacting with the mobile robot employing the laser pointer, a human now interacts with the NRS to compensate the limitations of the initial laser pointer method. To this end, a laser spot is not only perceived by the mobile robot's on-board camera but also by the stationary cameras integrated into the smart home environment. Therefore, we modify the states of the robot guidance framework described in Subsection 3.3.1 to incorporate smart home components. Thus, the robot guidance framework is no more limited to a traditional home environment but can also exploit components of a smart home environment. The three states of the framework are modified and defined as follows:

- **Border:** The NRS detects and localizes laser spots, that are used to specify virtual border points $\mathcal{P}$. If the stationary cameras perceive a human's laser spot on the ground, the NRS automatically sends the mobile robot to this area, i.e. sending a navigation goal to the mobile robot. Thus, the mobile robot autonomously navigates to this area until it reaches the navigation goal or perceives a laser spot with its on-board camera. The robot can then act as mobile camera if the stationary cameras lose track of the laser spot.

- **Seed:** The NRS detects and localizes laser spots and calculates the seed point $\boldsymbol{s}$. Similar to the *Border* state, the mobile robot simultaneously moves to the laser spot position if a stationary camera perceives a laser spot. The seed point $\boldsymbol{s}$ indicates the restriction area, i.e. $\delta = 1$.

- **Guide:** A human can guide the mobile robot to a desired restriction area using the laser pointer, which is identical to the *Guide* state of the initial robot guidance framework. This state should be never reached if at least one of the stationary cameras' fields of view covers a part of the restriction area so that the mobile robot can be automatically sent to this area. Nonetheless, we incorporate this state into our interaction method to ensure its functionality in case of an absence of stationary cameras.

Especially, the modifications in the states *Border* and *Seed* are intended to significantly reduce the interaction time compared to the laser pointer method. For example, if a human starts specifying a virtual border in the field of view of at least one stationary camera, the NRS automatically sends the mobile robot to this area. Thus, if the stationary cameras lose track of the laser spot, e.g. leaving field of view or due to an occlusion, the mobile robot is already on its way to this area to act as mobile camera. This behavior should minimize the time in the *Guide* state and thus reduce the overall interaction time.

In addition to the modified states of the robot guidance framework, we also add new events to facilitate switching between the different states of the interaction method. Instead of only sequentially switching between states using the events *Next* and *Previous* as described in Subsection 3.3.1, a human can now directly switch to certain states and perform certain actions as depicted in Figure 4.2. The small number of events in the initial robot guidance framework was due to the restriction to the mobile robot's on-board capabilities. However, since the NRS provides additional interaction capabilities, we extend the set of events to the following five events, that are mapped to speech commands perceived by the smart speaker:

- **Define border:** This command is used to start the definition of virtual border points $\mathcal{P}$, thus switching to state *Border*.

- **Define seed:** This command is used to start the specification of a seed point $\boldsymbol{s}$, thus switching to state *Seed*.

- **Guide robot:** The interaction method's internal state switches to the *Guide* state so that a human can guide the mobile robot using the laser pointer without storing its positions.

- **Save:** This command is employed when a human wants to integrate and save the user-defined virtual border into the map of the environment M, i.e. performing the map integration algorithm. This command replaces the *Timeout 1* in the initial robot guidance framework, which triggers a transition to the *Guide* state.

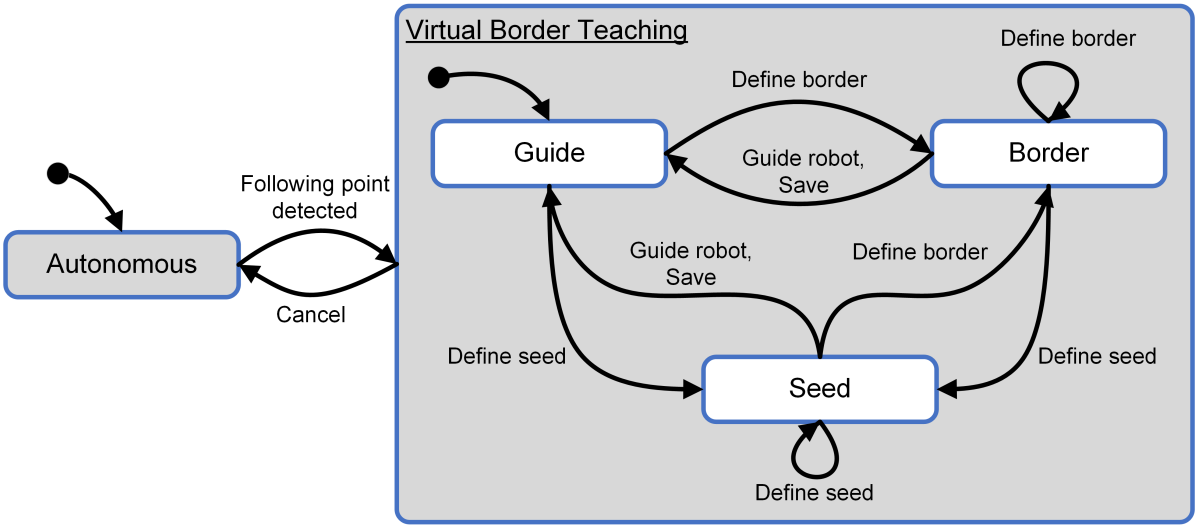


Figure 4.2: States and transitions of the adapted robot guidance framework.

- **Cancel:** If a human does not want to save the user-defined virtual border or wants to cancel the interaction process, this event triggers a transition to the state *Autonomous*. This command replaces the *Timeout 2* in the initial robot guidance framework.

Since these are only five commands, this should not mentally overload a human, who is able to hold approximately seven commands in short-term memory (MILLER, 1956). Moreover, due to the interaction employing speech commands, a human does not need to be in the vicinity of the mobile robot, e.g. to press an on-board button or provide visual codes. Thus, this state change interaction is intended to increase the user experience.

Feedback

In addition to the transfer of spatial information, we realize the second property of an interaction process, i.e. a feedback channel from NRS to human, by extending the feedback system of the laser pointer approach. This is based on mobile robot's non-speech audio sound and colored light feedback to indicate internal state changes and the detection of a laser spot. Since these communication channels cannot provide complex feedback, we additionally leverage the smart display integrated in the environment to provide more complex feedback. This includes the visualization of the 2D occupancy grid map (OGM) of the environment M, the mobile robot's current pose in the environment T_R and the progress of the spatial information transfer. After successfully accomplishing an interaction process, the mobile robot's workspace containing the user-defined virtual borders is also visualized on the display. We hypothesize that this extended feedback will support a human in the interaction process leading to an increase of user experience.

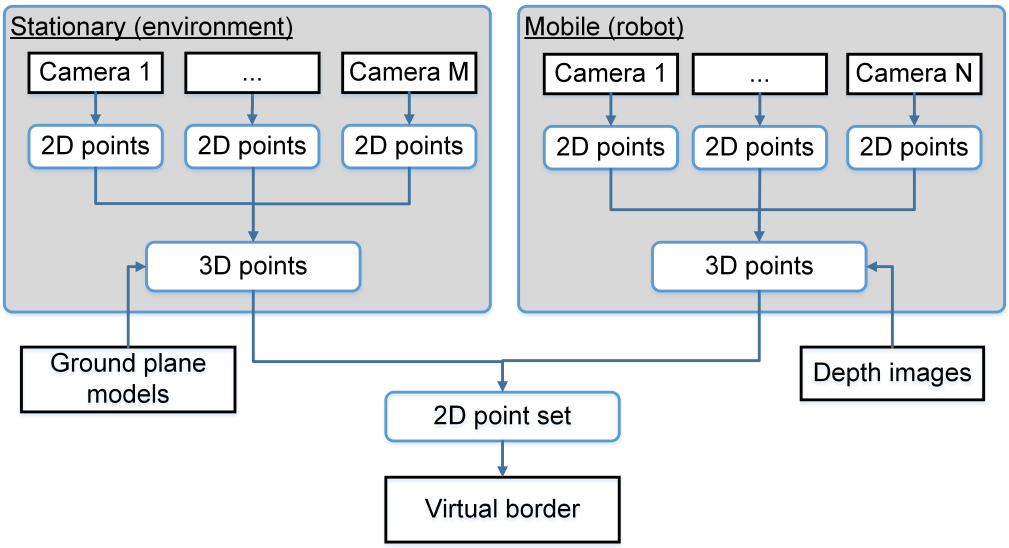


Figure 4.3: Architecture of the cooperative perception for specifying virtual borders based on multiple camera views.

4.1.3 Cooperative Perception

While the integration of the smart display and smart speaker in the interaction method is straightforward, the incorporation of the smart camera network, which is used to increase the perceptual space, is more challenging. There are mainly two reasons: (1) multiple cameras, stationary and mobile, have to be integrated into an architecture that supports the interaction process and (2) a single virtual border has to be extracted from multiple camera observations including noisy data. Our solution to these challenges is the answer to Research Question 2.3, i.e. how to cooperatively perceive and combine multiple sensor observations to restrict the mobile robot's workspace.

Architecture

Addressing the first challenge, we propose the architecture, that is illustrated in Figure 4.3. This architecture consists of M stationary cameras integrated in the environment and N mobile cameras on mobile robots[4]. Each camera independently performs laser point detection in image space resulting in a 2D point $\boldsymbol{p} \in \mathbb{R}^2$ for each detected laser spot. We apply the laser point detection algorithm presented in Subsection 3.3.1, that is based on illumination and morphological properties of a laser spot, i.e. circular shape, specific size and extreme brightness compared to its local environment. In order to combine multiple camera observations, these have to be described with respect to the same reference coordinate frame. For this purpose, the map coordinate frame M is optimal because

[4]Although we consider a single mobile robot in this work, we design the architecture with multiple mobile cameras due to scalability options in the future. However, this requires additional effort in the field of multi-robot cooperation.

it is also the reference coordinate frame for the mobile robot's localization. Thus, each observation of a point $\boldsymbol{p}$ is projected into 3D world space ${}^{M}\boldsymbol{P} \in \mathbb{R}^3$ using either a ground plane model in case of the stationary cameras or a depth image in case of the mobile cameras. This makes the laser point observations independent of the cameras, i.e. all laser spot positions are described with respect to the same reference coordinate frame M. Since the resulting points are points on the ground plane, they degenerate to 2D positions with respect to the map coordinate frame M. Finally, a polygonal chain $\mathcal{P}$ or seed point $\boldsymbol{s}$ is extracted from this point set depending on the current state of the interaction method.

To ensure transformations between the coordinate frames of the cameras and the map M, all camera transformations with respect to the map coordinate frame M have to be determined. Since the smart camera network only consists of stationary cameras, this extrinsic calibration only has to be performed once during installation of the cameras. All transformations belong to the Special Euclidean group $SE(3)$ containing rigid motions in three dimensions. Thus, our initial problem setting defined in Section 3.1 is extended by additional transformations between the stationary cameras and the map coordinate frame M.

Virtual Border Extraction

While the extraction of the seed point $\boldsymbol{s}$ from a point set is not challenging, the extraction of the polygonal chain $\mathcal{P}$ includes several challenges that need to be adequately addressed:

1. **Irrelevant clusters:** A cluster is a group of spatially nearby data points, and an irrelevant cluster is characterized by certain expansion characteristics describing the spatial expansion of the cluster. This is measured as Euclidean distance between the diagonal points of the cluster's minimum bounding box. Irrelevant clusters can occur in the point set due to the presence of other areas in the environment, that have the same characteristics as a laser point.

2. **Noise:** The 2D point set acquired from multiple camera observations contains data points of a single user-defined polygonal chain $\mathcal{P}$ but also noisy data points. These are data points that do not belong to a cluster and that can occur due to errors in the laser point detection algorithm.

3. **Spatial redundancy:** The data points can be spatially redundant because the points are obtained from different cameras that may have an overlap of their fields of view, e.g. two overlapping stationary cameras of the environment or an overlap between a stationary camera and the mobile robot's camera.

4. **Inaccuracies:** Calibration inaccuracies of the cameras and localization errors of the mobile robot can lead to inaccurate user-defined points.

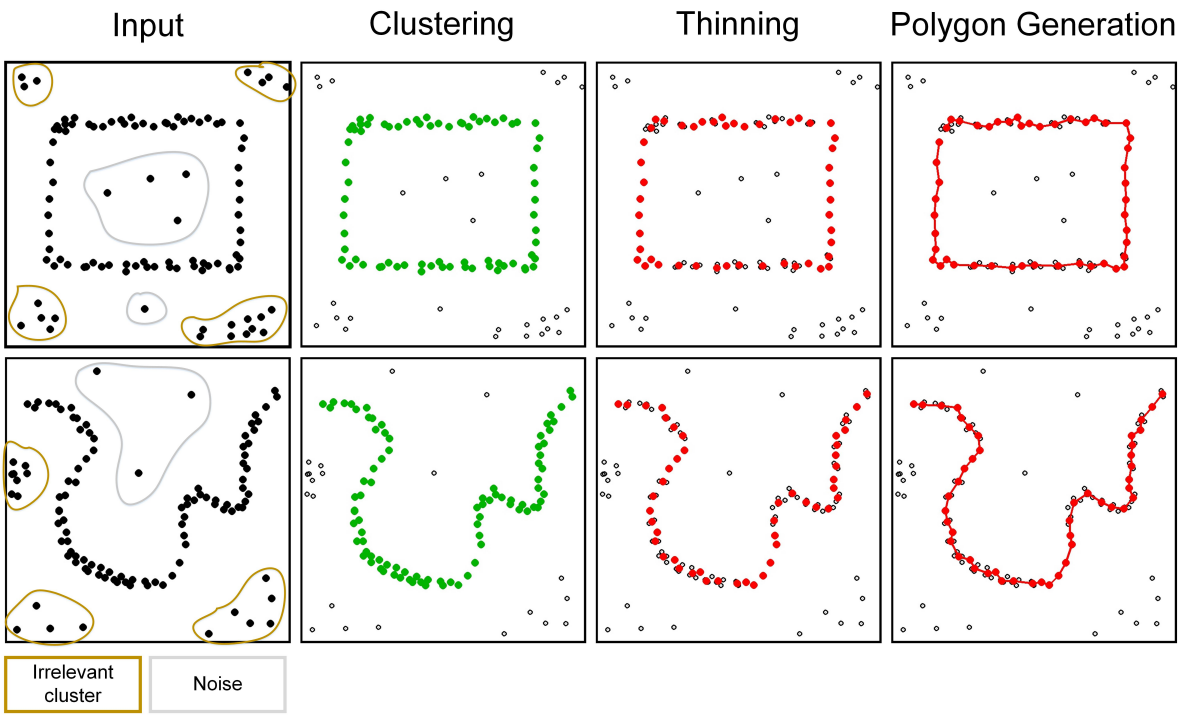


Figure 4.4: Processing stages extracting a polygonal chain from a point set including noise and irrelevant clusters. Each row corresponds to a user-defined point set.

5. **Polygon generation:** The generation of a polygonal chain $\mathcal{P}$ from the point set is challenging because the polygonal chain can have an arbitrary shape and size.

We address these challenges with a novel multi-stage virtual border extraction algorithm, which is illustrated in Figure 4.4. The figure depicts the stages of the algorithm with two exemplary polygonal chains in the first and second row. The first column visualizes the input point set containing virtual border points but also noise and irrelevant clusters. The points assigned to the virtual border cluster are colored green in the second column. Thinning the virtual border cluster yields the red point set in column three. This is used to generate a polygonal chain as shown in the last column. We reference this figure throughout this subsection to explain the multi-stage algorithm.

Clustering The first stage of the extraction algorithm is the clustering stage as denoted in Algorithm 4.1. The input is a 2D point set $pointsetIn$ as shown in the first column of Figure 4.4, and the data points belonging to the polygonal chain $pointsetOut$ are the result. This stage is designed to address the first two challenges of the virtual border extraction step, i.e. extracting a cluster of points belonging to the user-defined polygonal chain $\mathcal{P}$ and discarding noisy data points and irrelevant clusters. To this end, we first apply the DBSCAN algorithm (ESTER *et al.*, 1996) for clustering the data points (l. 3). This is a density-based clustering algorithm that can find clusters with different

shapes and sizes. It is parameterized by eps to define a distance threshold for a neighboring point and $minPts$ to define a core point. This is a point that has at least $minPts$ points within its distance eps. The result of the DBSCAN algorithm is a set of $clusters$ where each point of $pointsetIn$ is assigned to a cluster. Noisy data points, that do not belong to a cluster, are discarded. Afterwards, the algorithm selects the largest cluster with certain expansion characteristics defined by $minExp$ and $maxExp$. To this end, we order the clusters by their sizes (number of points) in descending order (l. 4) to iterate over the clusters beginning with the largest cluster (l. 5ff.). The additional parameter $minSize$ is a lower threshold for the size of a cluster to exclude small clusters due to noise. In each iteration, it is checked if the expansion of the current cluster c lies within the expansion thresholds $minExp$ and $maxExp$ to ignore irrelevant clusters (l. 6). The first cluster, that fulfills this condition, is returned as cluster of the polygonal chain. The result is visualized in the second column of Figure 4.4 as green points. The black points are either noise or irrelevant clusters.

Algorithm 4.1: Clustering stage of the virtual border extraction algorithm.

```
   Input: pointsetIn
   Output: pointsetOut
   Params: eps, minPts, minExp, maxExp, minSize
1  Function clustering(Input, Output, Params)
2      pointsetOut = ∅;
3      clusters = DBSCAN (pointsetIn, eps, minPts);
4      clusters = orderClustersBySizeDesc (clusters, minSize);
5      foreach c in clusters do
6          if minExp < expansion (c) < maxExp then
7              pointsetOut = c;
8              break;
9      return pointsetOut;
```

Thinning The second stage of the algorithm is the thinning stage (Algorithm 4.2), that reduces the number of points in the cluster resulting from the previous stage. This algorithm is designed to remove spatially redundant data points and to smooth data points due to localization errors and calibration inaccuracies. Thus, it addresses the third and fourth challenge of the virtual border extraction step. For this purpose, the thinning algorithm identifies spatially nearby data points and replaces them by their mean value. To this end, a point p with most neighbors within a distance $maxNeighborDist$ is selected (l. 4) and its neighboring points n are determined (l. 5). If there is at least one neighboring point (l. 6), the mean point is calculated for these points $p \cup n$ (l. 7). Afterwards, these points are removed from the initial $pointsetIn$ (l. 8) and the mean point is added to $pointsetOut$ (l. 9). In case that no data point has at least one neighboring point (l. 10), the iterative procedure terminates. Finally, all remaining points contained in $pointsetIn$, i.e. points

without neighbors, are added to $pointsetOut$ (l. 12), i.e. the set containing the thinned points. Thus, the thinned cluster includes the initial points, that do not have neighboring points, and mean points representing subsets of the initial points. The result is shown in the third column of Figure 4.4. Compared to the second column containing the cluster of the polygonal chain, there are fewer points due to the reduction of data points.

Algorithm 4.2: Thinning stage of the virtual border extraction algorithm.

```
   Input: pointsetIn
   Output: pointsetOut
   Parameters: maxNeighborDist
 1 Function thinning(Input, Output, Params)
 2     pointsetOut = ∅;
 3     while true do
 4         p = getPointWithMostNeighbors (pointsetIn, maxNeighborDist);
 5         n = getNeighbors (p, maxNeighborDist);
 6         if n ≠ ∅ then
 7             mean = getMean (p ∪ n);
 8             pointsetIn = pointsetIn \ {p ∪ n};
 9             pointsetOut = pointsetOut ∪ mean;
10         else
11             break;
12     pointsetOut = pointsetOut ∪ pointsetIn;
13     return pointsetOut;
```

Polygon Generation Finally, the thinned point set is the input $pointsetIn$ for the last stage, in which the polygonal chain $polygon$ is generated (Algorithm 4.3). This algorithm consists of two phases, i.e. forward and backward, and addresses the fifth challenge of the virtual border extraction step. Since a polygonal chain has a starting and ending point, we first select an arbitrary point of $pointsetIn$ as starting point and collect neighboring points in one direction. If there is no more neighboring point available, we again select the starting point and collect neighboring points in the other direction. Afterwards, the selected points are concatenated. In this context, direction corresponds to the sequence of the points of the polygonal chain. For example, considering an arbitrary point of a polygonal chain, which is not the starting or ending point, there are two directions from this point, i.e. the direction to the starting and ending point. To realize this behavior, we first initialize two empty polygonal chains $dir1$ and $dir2$ for each direction (l. 2), and we set the variable $forward$, that indicates the phase of the algorithm (l. 3). We then select an arbitrary point p (here at index 0) of $pointsetIn$, mark it and append it to $dir1$ (l. 4ff.). Afterwards, the nearest neighboring point n within a distance $maxNeighborDist$, that it not already marked, is selected (l. 8). If

there is a neighboring point n available (l. 9), we append n to one of the temporary polygonal chains depending on the variable $forward$, mark n and select n as the current point p (l. 10ff.). This procedure is repeated until there is no neighboring point n available for the current point p (l. 16), i.e. the neighboring points for the first direction are collected. In this case, we select our initial point again as current point p and switch the variable $forward$ to collect neighboring points along the other direction (l. 17ff.). Subsequently, the same procedure is performed until there is again no neighboring point n available (l. 21) or all points of $pointsetIn$ are marked (l. 7). As a last step, the order of the temporary polygonal chain $dir1$ is reversed (l. 22), and $dir2$ is appended resulting in the final polygonal chain (l. 23). This reversal of the order is necessary to create a single polygonal chain with a single direction. The result of this stage is visualized in the last column of Figure 4.4.

Algorithm 4.3: Polygon generation stage of the virtual border extraction algorithm.

```
   Input: pointsetIn
   Output: polygon
   Parameters: maxNeighborDist
 1 Function generatePoly(Input, Output, Params)
 2     dir1, dir2 = ∅;
 3     forward = true;
 4     p = pointsetIn(0);
 5     setMarked (p);
 6     dir1 = concat (dir1, p);
 7     while not allMarked (pointsetIn) do
 8         n = getNearestUnmarkedNeighbor (p, maxNeighborDist);
 9         if n ≠ ∅ then
10             if forward then
11                 dir1 = concat (dir1, n);
12             else
13                 dir2 = concat (dir2, n);
14             setMarked (n);
15             p = n;
16         else
17             if forward then
18                 p = pointsetIn(0);
19                 forward = false;
20             else
21                 break;
22     reverse (dir1);
23     polygon = concat (dir1, dir2);
24     return polygon;
```

We use the parameter values reported in Table 4.1 for the algorithms. These were determined experimentally and are independent of the environment allowing an easy portability of the algorithms. While the parameters of the clustering and polygon generation stage are essential for the correct extraction of a polygonal chain $\mathcal{P}$, the parameter of the thinning stage affects the accuracy of the interaction method. Therefore, we choose a small value of 0.1 m, which is derived from the maximal localization and calibration error we assume for the mobile robot and the stationary cameras.

Table 4.1: Parameter values for the virtual border extraction algorithm.

Stage	Parameter	Value
Clustering	eps	0.5 m
	minPts	1
	minExp	0.3 m
	maxExp	$+\infty$ m
	minSize	10
Thinning	maxNeighborDist	0.1 m
Polygon generation	maxNeighborDist	0.5 m

4.2 Experimental Evaluation

The goal of the proposed interaction method based on a NRS is to improve the interaction time and user experience compared to the laser pointer approach while not decreasing the performance of the other user requirements. Since the interaction method builds on the flexibility of a virtual border and the correctness of the map integration algorithm, we do not have to evaluate these requirements as these have already been proven in Experiment 4 of the previous chapter. However, the other user requirements could be affected by the design of the NRS interaction method. Thus, these are evaluated in an experiment with three evaluation scenarios and multiple participants similar to Experiment 1 and 2 of the previous chapter.

4.2.1 Independent Variables

In our experiment, we manipulate the interaction method as single independent variable, which can have one of the two values:

1. **Pointer:** This is the laser pointer approach proposed in Subsection 3.3.1, that is not supported by a smart home environment.

2. **Network Robot System (NRS):** This is our proposed interaction method based on a NRS described in Section 4.1. Additional to the mobile robot, the NRS features stationary cameras in the environment as additional sensors to perceive laser points. A voice control allows switching between the interaction method's states using voice commands. Among colored LEDs and non-speech audio sound on board the mobile robot, a smart display integrated into the environment acts as additional feedback device. Although this interaction method also employs a laser pointer as user interface, we denote it as NRS approach to distinguish it from the other laser pointer approach without smart home support.

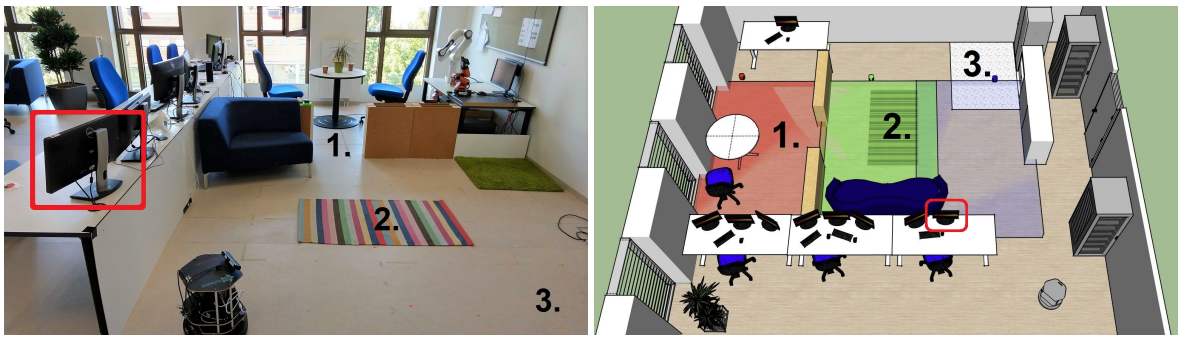
4.2.2 Hypotheses

The objective of this experimental evaluation is the test of the following hypotheses:

- **Hypothesis 1:** The NRS interaction method achieves a better interaction time than the laser pointer interaction method, which leads to an acceptable interaction time.
- **Hypothesis 2:** The NRS interaction method achieves a better user experience than the laser pointer interaction method, which leads to a good user experience.
- **Hypothesis 3:** The learnability does not decrease when employing the NRS compared to the laser pointer interaction method. Thus, the learnability should achieve an acceptable quality level.
- **Hypothesis 4:** The completeness does not decrease when employing the NRS compared to the laser pointer interaction method. Thus, the completeness should achieve an acceptable quality level.
- **Hypothesis 5:** The accuracy does not decrease when employing the NRS compared to the laser pointer interaction method. Thus, the accuracy should achieve a good quality level.

4.2.3 Setup

To test these hypotheses, we set up the same experimental environment and evaluation scenarios as described in the previous chapter in Experiment 2, i.e. three different evaluation scenarios in our 10×8 m lab environment with typical restriction areas. Moreover, the same mobile robot platform based on a TurtleBot v2 is deployed as described in detail in Subsection 3.4.2. However, to leverage a smart home environment according to our approach, we extend the traditional home environment as shown in Figure 4.5. For this purpose, we mount three RGB cameras with an image resolution of 1920×1080 pixels on the ceiling (2.95 m height, pitch angle of 90°). Thus, they provide top views of the environment. We denote these stationary cameras representing the smart camera network as red, green and blue camera. Their fields of view partly overlap as illustrated in Figure 4.5b, but



(a) **(b)**

Figure 4.5: (a) Image and (b) 3D sketch of a part of the lab environment. The three evaluation scenarios are numbered, and the three cameras' fields of view are visualized as red, green and blue rectangles. The position of the smart display for feedback is encircled in red, and the mobile robot's initial pose is depicted in the bottom right of the sketch.

they do not cover the entire environment. Hence, there is only a partial observation of the environment, which is typical for smart home environments. The restriction areas of the three evaluation scenarios are covered by the stationary cameras as follows:

1. **Room exclusion:** This area is in the fields of view of the red and green camera.
2. **Carpet exclusion:** This area is in the fields of view of the green and blue camera.
3. **Spot cleaning:** This area is partly covered by the blue camera.

In addition to the camera view coverage of the restriction areas, it is possible that participants temporarily occlude restriction areas with their bodies depending on their positions during interaction. The initial pose of the mobile robot is not covered by a stationary camera's field of view and is between 2.50 and 5.40 m (Scenario 1: 5.40 m, Scenario 2: 2.50 m and Scenario 3: 3.00 m) away from the restriction areas. Moreover, all RGB cameras are calibrated, i.e. their intrinsic camera parameters are known and their relative transformations with respect to the map coordinate frame M are determined in advance. The interaction using speech commands relies on a Wizard-of-Oz method, in which a human operator reacts on the speech commands of participants, i.e. switching between interaction method's states per remote control[5]. Furthermore, we place a 22-inch smart display on a table near the restriction areas to provide visual feedback to the participants. This display is network-connected to the NRS and shows the feedback about the interaction process, i.e. the OGM of the environment M, the robot's current pose T_R and virtual borders if specified by a participant.

[5]We do not use a cloud-based intelligent personal assistant due to network restrictions in the university's network. However, the implementation would be straightforward, and a state-of-the-art voice control would achieve a similar quality as a human operator. This method is not recognized by the participants and does not change the way in which a participant interacts with the NRS.

4.2.4 Procedure

We apply the same experimental procedure as described in Experiment 2 of the previous chapter. This is a within-subject design where a participant performs the experimental procedure with both interaction methods, i.e. with and without support of a smart home environment.

4.2.5 Participants

The participants of this experiment correspond to the second user group of Experiment 2 of the previous chapter. Thus, the experimental procedure is performed by a total of 15 participants (11 male, 4 female) with a mean age of $M = 28.80$ years and standard deviation of $SD = 11.44$ years. Their ages range between 17 and 55 years, and they are recruited from the local environment by word of mouth. Participants rate their experience with robots on a 5-point Likert item ranging from *no experience* (1) to *highly experienced* (5) with a mean of $M = 3.20$ and standard deviation of $SD = 1.37$. This corresponds to a moderate experience with robots and comprises users owning a mobile robot in their household, e.g. a vacuum cleaning robot. However, they only deploy the mobile robots in their home environments according to the manual and do not know how they internally work. Hence, we assume the participants to be good representatives for the intended users of the interaction method.

4.2.6 Measurement Instruments

In order to measure the dependent variables of this experiment, we mainly apply the measurement instruments employed in the experiments of the previous chapter. Hence, to assess the usability criteria concerning completeness, accuracy and interaction time, we use the same instruments as introduced in Subsection 3.4.5, i.e. success rate for the completeness, Jaccard similarity index (JSI) for the accuracy, and duration between start and end of an interaction process for the interaction time. In case of the learnability and user experience, we employ the following questionnaire containing statements which can be rated on 5-point Likert items with numerical response format. The questionnaire is similar to the questionnaire employed in Experiment 1 of the previous chapter, but slightly improved in some formulations. The statements are as follows (translated from German):

1. It was easy to learn the handling of the interaction method (1 = hard, 5 = easy).
2. I had problems to define the virtual borders (1 = big problems, 5 = no problems).
3. It was intuitive to define the virtual borders (1 = not intuitive, 5 = intuitive).
4. It was physically or mentally demanding to define the virtual borders (1 = hard, 5 = easy).
5. I liked the feedback of the system (1 = bad/no feedback, 5 = good feedback).

The first statement ($S1$) is used to measure the learnability, whereas the other statements ($S2$-$S5$) measure different aspects of the user experience. Furthermore, the participants are asked if the smart environment supports the interaction process. They can answer the question in a binary response including *yes* or *no*. Although this is not directly related to a certain requirement, it is interesting to gather participants' opinions concerning the overall objective. Finally, the questionnaire provides a field for free responses, e.g. to give feedback or reasons for a certain decision.

4.2.7 Analysis & Results

Interaction Time

Regarding the first hypothesis dealing with the interaction time, the results of the experimental evaluation are visualized in Figure 4.6a. Each bar comprises the measurements of all participants for an interaction method and scenario. As already revealed in the previous chapter, the laser pointer method without smart home support only achieves an unacceptable interaction time. In contrast to this, the proposed NRS approach features a significantly reduced interaction time with an at least acceptable level in all scenarios. Moreover, the results demonstrate that the NRS approach features a shorter time in the interaction method's states *Border* and *Seed* and no time in the state *Guide*.

To statistically verify this visual difference between both interaction methods, we first run Shapiro-Wilk tests to test for normality of the data (differences in the interaction times between the two interaction methods), which is an assumption of parametric statistical hypothesis tests, e.g. a paired t-test. These tests only become significant for the third scenario ($p = 0.016$). Thus, we assume the data of the first two scenarios to be approximately normally distributed, while the data of the third scenario are not normally distributed. Moreover, we interpret the corresponding boxplots for outliers (1.5× interquartile range). The boxplots reveal that there are some outliers in the data. Due to the presence of outliers and the violation of normality in the third scenario, we perform a non-parametric Wilcoxon signed-rank test to compare both interaction methods. The statistical results shown in Table 4.2 reveal a significant difference between the interaction methods in all evaluation scenarios. Hence, our proposed NRS approach is significantly faster compared to the laser pointer approach. This results in speedups of 2.8, 2.2 and 2.2 for Scenarios 1, 2 and 3.

Table 4.2: Statistical results concerning the interaction time comparing both interaction methods. A * indicates a significant result.

Scenario	Statistic	p-value
1. Room exclusion	$Z = -3.411$	$p < 0.001$*
2. Carpet exclusion	$Z = -3.408$	$p < 0.001$*
3. Spot cleaning	$Z = -3.409$	$p < 0.001$*

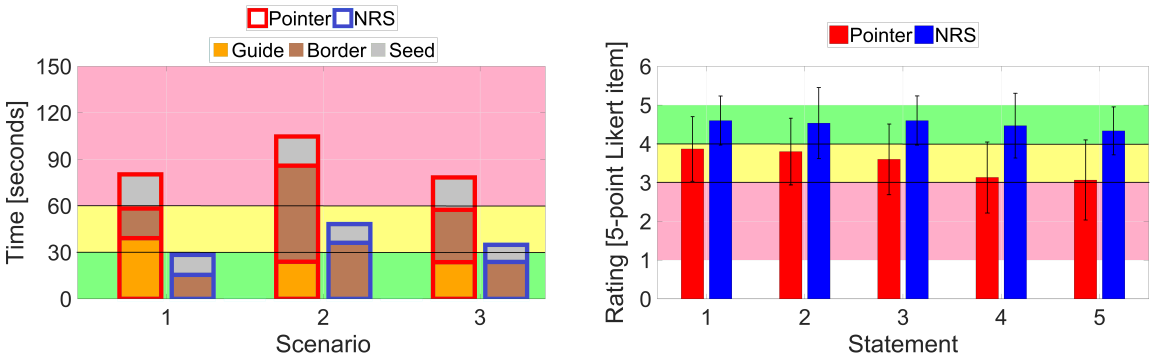


Figure 4.6: Results of the (a) interaction time and (b) answers to the questionnaire (mean and standard deviation) for both interaction methods. The background colors indicate the quality levels ranging from unacceptable (red) to acceptable (yellow) and good (green).

Learnability and User Experience

In order to test the second and third hypothesis dealing with the learnability and user experience, we consider the participants' answers of the questionnaire introduced in Subsection 4.2.6. These are visualized in Figure 4.6b with their mean and standard deviation per statement and interaction method. The results show that the NRS approach features a good quality level for the learnability and all aspects of the user experience. Moreover, the NRS approach outperforms the laser pointer approach, which reaches acceptable values on all statements.

We verify this visual observation by running Wilcoxon signed-rank tests on the statements. This non-parametric statistical test is chosen because Likert-item data violate the assumption of normality and the number of participants does not exceed the value of 25, which would justify neglecting this violation. The tests reveal statistically significant differences for all statements in the questionnaire as summarized in Table 4.3.

Table 4.3: Statistical results of the answers to the questionnaire comparing both interaction methods. A * indicates a significant result.

Statement	Aspect	Statistic	p-value
S1	Learnability	$Z = -2.373$	$p = 0.023$*
S2	Problems	$Z = -3.051$	$p = 0.002$*
S3	Intuitiveness	$Z = -2.830$	$p = 0.004$*
S4	Effort	$Z = -3.126$	$p < 0.001$*
S5	Feedback	$Z = -2.745$	$p = 0.005$*

Regarding the learnability ($S1$), there is a significant difference between the proposed NRS (M = 4.60, SD = 0.63) and the laser pointer (M = 3.87, SD = 0.83) method[6]. Furthermore, the NRS approach achieves a significantly better performance on all aspects of the user experience ($S2$-$S5$) compared to the laser pointer approach. For example, participants have significantly less problems defining the virtual borders with a NRS (M = 4.53, SD = 0.92) than without support of a smart environment (M = 3.80, SD = 0.86). Participants also find the proposed approach (M = 4.60, SD = 0.63) more intuitive than the interaction method without smart home support (M = 3.60, SD = 0.91). The strongest effect is measured for $S4$ that shows that the NRS approach is less physically or mentally demanding (M = 4.47, SD = 0.83) compared to the laser pointer approach (M = 3.13, SD = 0.92). This is consistent with the results of the interaction time as reported in the previous paragraph. Finally, there is a significant difference for the feedback of the interaction method, i.e. NRS (M = 4.33, SD = 0.62) and laser pointer (M = 3.07, SD = 1.03). In addition to the 5-point Likert item statements, participants are asked if the smart environment supports the interaction process. A large majority (14 out of 15 participants) agrees that the smart environment supports the interaction process.

Completeness

After presenting the results dealing with the first three hypotheses, the experimental results related to Hypothesis 4 dealing with the completeness are summarized in Figure 4.7a. Both interaction methods feature the same acceptable completeness, i.e. 91.1% on average. Furthermore, the completeness for the three scenarios is the same (93.3% for Scenario 1 and 2; 86.7% for Scenario 3). There are nine participants who performed all their runs successfully, four participants who failed for one of their six runs, and two participants incorrectly defined a virtual border in two of six runs.

Accuracy

Finally, the last hypothesis states that the accuracy does not decrease when employing the NRS compared to the laser pointer interaction method. To verify this hypothesis, the JSI values of the experimental evaluation are visualized in Figure 4.7b. The results demonstrate that the NRS method reaches a good accuracy in all scenarios similar to the laser pointer method. However, the laser pointer approach only achieves an acceptable accuracy in Scenario 2.

In order to statistically test the hypothesis, we again explore the data for normality and outliers using Shapiro-Wilk tests and boxplot interpretation. The Shapiro-Wilk tests do not become significant for

[6]Although the median should be used to describe the central tendency in non-parametric tests, we report the mean value in this work to better reveal the differences between the interaction methods. This is valid since we consider an interval-level of measurement (HARPE, 2015). Moreover, since other studies often report mean values for Likert items, we also want to make our results better comparable.

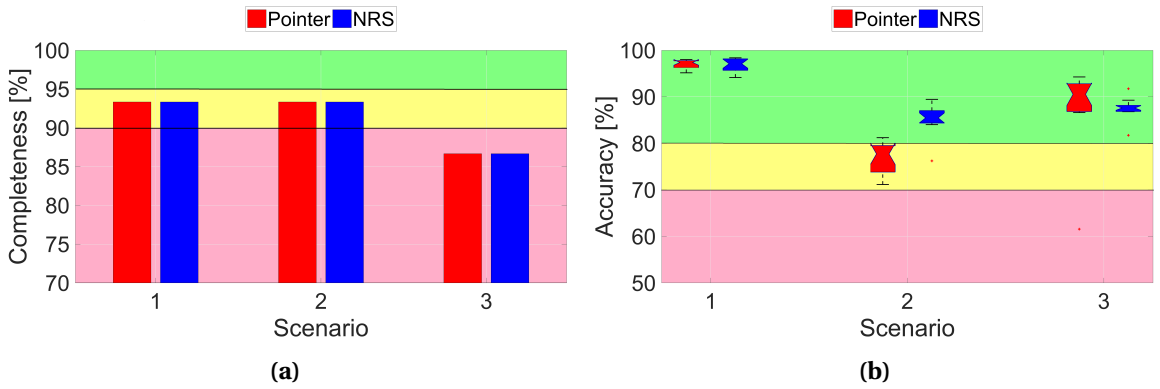


Figure 4.7: Results of the (a) completeness and (b) accuracy of both interaction methods. The background colors indicate the quality levels ranging from unacceptable (red) to acceptable (yellow) and good (green).

any scenario indicating a normal distribution of the data. However, since the data for Scenario 2 and 3 contain some outliers, we prefer a Wilcoxon signed-rank test over a paired t-test to respond to this violation of an assumption of a parametric statistical test. The statistical results are different for the three scenarios as summarized in Table 4.4.

Table 4.4: Statistical results concerning the accuracy comparing both interaction methods. A * indicates a significant result.

Scenario	Statistic	p-value
1. Room exclusion	$Z = -0.594$	$p = 0.588$
2. Carpet exclusion	$Z = -3.110$	$p < 0.001$*
3. Spot cleaning	$Z = -2.497$	$p = 0.010$*

There is no significant difference for Scenario 1 where the NRS achieves a value of (M = 97.0%, SD = 1.4%) and the laser pointer a value of (M = 97.1%, SD = 0.8%). Regarding Scenario 2, the NRS (M = 85.4%, SD = 3.0%) performs significantly better than the laser pointer approach (M = 77.3%, SD = 2.9%). This difference is reversed in Scenario 3 where the laser pointer reaches an accuracy of (M = 88.8%, SD = 8.2%), which is better than the NRS approach (M = 87.7%, SD = 2.1%). Although the result of the test is significant for Scenario 3, the difference between the means is only 1.1%, which is not notable in practice and does not lead to a different quality level.

4.2.8 Discussion

Interaction Time

The experimental results reveal a significant improvement of the interaction time between the NRS and laser pointer approach. Moreover, the NRS approach achieves an acceptable interaction time in all scenarios. Thus, the results support Hypothesis 1 of this experiment.

The reason for this significant difference is revealed by the decomposition of the time measurements into the interaction method's states. While the time for the laser pointer approach is composed of all states of the interaction method, the NRS approach does not include the *Guide* state. This is a consequence of the human-robot-environment interaction where the NRS automatically sends the mobile robot to the intended restriction area when a laser spot is detected by a stationary camera. Therefore, a human does not have to manually guide the mobile robot to the intended restriction area. Thus, the NRS approach can avoid the time in the *Guide* state.

Another reason for the time difference between the interaction methods is the time in the state *Border*, which is linear with respect to the border length as demonstrated in Experiment 3 of the previous chapter. Thus, if the user-defined virtual border is short, e.g. 0.70 m for Scenario 1, our NRS approach is only slightly faster in this state, i.e. 4 seconds difference. But if we consider a longer virtual border, e.g. the 6.50 m long border around the carpet (Scenario 2), the NRS approach is even 26 seconds faster on average. The reason is the mobile robot's velocity limitation (0.2 $^{m}/_{s}$) to ensure a safe and smooth motion of the robot. By using the NRS approach, this speed limitation can be compensated if the laser spot is in the field of view of one of the stationary cameras. Our interaction method is then only limited by the frame rate of the cameras (25 $^{frames}/_{s}$). Hence, it also features a linear interaction time but with a smaller gradient.

Another speedup is achieved when specifying the seed point $\boldsymbol{s}$ in state *Seed* because a human can directly indicate the seed point $\boldsymbol{s}$ with laser points, which are detected by a stationary camera. In case of the laser pointer approach, a human additionally has to rotate the mobile robot around its vertical axis to adjust the camera's field of view. This rotation takes additional interaction time.

Although achieving a significant speedup, there are two important aspects that affect the speedup in the interaction time: (1) the stationary cameras' coverage of the environment and (2) the distance between the mobile robot's initial pose and the restriction area. If we would decrease the number of cameras in the environment and thus the camera coverage, this would result in a smaller reduction of the interaction time. This would finally degenerate to the laser pointer approach without support of a smart environment. Moreover, the speedup strongly depends on the distance between the mobile robot's initial pose and the restriction area, which influences the performance of the laser pointer approach. This is due to the fact that the laser pointer approach requires a direct line of

sight between human and robot, and thus a human first has to guide the mobile robot to the restriction area. As described in the experimental setup, the distances in our scenarios range from 2.50 to 5.40 m. If the distances would be smaller, the time in the *Guide* state would also decrease leading to a smaller speedup. For these reasons, it is not possible to report a specific speedup value. Nonetheless, we chose a typical camera coverage in the evaluation scenarios, that allows the partial observation of the environment. Moreover, we chose the distances to the restriction areas quite liberally since much larger distances would be even realistic, e.g. in typical home environments with a single charging station for the mobile robot. Hence, we conclude that the interaction time improves with the support of a smart environment, but we cannot report a specific speedup value. Therefore, the reported speedups in Subsection 4.2.7 are intended to give an estimate and are only valid for this specific experimental evaluation.

Learnability and User Experience

The analysis of the answers to the questionnaire shows that the user experience of the NRS method achieves a good quality level and outperforms the laser pointer approach, which reaches an acceptable user experience. Hence, we conclude that Hypothesis 2 is supported by the results. A reason for the difference could be that some participants had problems to rotate the mobile robot around its vertical axis without smart home support, e.g. to specify the seed point $\boldsymbol{s}$. In this case, they moved the laser spot too fast so that the robot's on-board camera could not follow the spot on the ground. In contrast to this, the NRS approach avoids this problem by additionally perceiving the laser spot through the stationary cameras in the environment. Furthermore, the speech commands provided a more intuitive communication channel to change the interaction method's internal state than pushing on the mobile robot's on-board buttons or generating visual codes. Since the feedback system of the NRS is a superset of the laser pointer's feedback system, a major reason for this difference is the additional smart display that visualizes the OGM with the user-defined virtual borders. This complex feedback is missing for the laser pointer approach, that only features simple non-speech audio and colored LED feedback. However, a participant also wished an even stronger feedback system after the experiment. The participant did not like the change of attention between specifying the virtual border on the ground and the view on the smart display, which was positioned aside on a table.

Regarding the learnability, the proposed method based on a NRS is better rated than the laser pointer approach. Thus, the incorporation of a smart home does not negatively affect the learnability. Therefore, Hypothesis 3 is supported by the results. Since both interaction methods are based on a laser pointer as user interface, the handling of a laser pointer does not influence the learnability. However, it could be easier to learn the speech commands than the assignments of the buttons to change between states of the interaction method. Furthermore, the guiding of the mobile robot could affect the rating.

Completeness

Since both interaction methods feature an acceptable completeness, we consider Hypothesis 4 to be supported by the results. The reason for the incorrect runs was always the definition of the seed point s. While some participants were confused where to specify the seed point s, especially in Scenario 3, other participants were unfocused and noticed their mistake on their own after performing the experiment. There were no problems with the definition of the virtual border points $\mathcal{P}$.

Accuracy

The experimental results demonstrate that the accuracy of the NRS is better in Scenario 2 and worse (although not notable in practice) in Scenario 3 than the laser pointer approach. There is no difference between both interaction methods in Scenario 1. Since there is no general decrease in accuracy, we conclude that the accuracy does not decrease when employing the NRS method and that Hypothesis 5 is supported by the results. A reason for the strong difference in Scenario 2 could be that it is harder to specify the carpet's corners when guiding the robot compared to the NRS approach. The highest accuracy value is achieved in Scenario 1 because the restriction area's complexity is relatively simple, i.e. the length of the virtual border is relatively short (0.70 m) and can be described by a simple polygonal chain. Thus, there is only minimal room for errors.

4.3 Summary

Motivated by Research Question 2 and the limitations of robot-dependent interaction methods revealed in Chapter 3, we proposed a novel interaction method based on a laser pointer, that leverages a smart home environment in the interaction process. This interaction method incorporates additional sensors and actuators of a smart home environment into the interaction process to compensate the limitations of robot-dependent interaction methods. To this end, we selected appropriate smart home components with the intention to support the interaction process by enhancing the mobile robot's perceptual and interaction capabilities. These smart home components, i.e. smart camera network, smart display and smart speaker, were integrated into the interaction method by modifying the previously developed robot guidance framework. A special challenge was the development of a cooperative perception including stationary and mobile cameras to perceive laser spots and an algorithm to extract virtual borders from multiple camera observations. The results of an experimental evaluation supported our hypotheses that the proposed NRS interaction method features a significantly shorter interaction time and a better user experience compared to the laser pointer approach without support of a smart environment. This implies an acceptable interaction time and good user experience. This improvement was also confirmed by the participants' answers to the question if the smart environment supports the interaction process. Moreover, the user study

showed that the NRS interaction method does not negatively affect other user requirements concerning learnability, completeness and accuracy. Therefore, we conclude that Objective 2 of this thesis could be achieved.

Table 4.5 summarizes the experimental results of this chapter in comparison to the interaction methods without smart home support. In addition to outperforming the laser pointer method, the NRS also performs better in the overall evaluation than the state-of-the-art method employing a graphical user interface (GUI). Nonetheless, the performance of the robot-independent interaction method based on AR cannot be reached, especially in terms of completeness and interaction time.

Table 4.5: Summary of the performance of the network robot system regarding the user requirements. The symbols indicate an unacceptable (−), acceptable (∘) and good (+) quality level. The ⊕ is used for an acceptable quality level if there is no good quality level defined for a certain requirement. Arrows indicate the change with respect to the laser pointer method.

Method	Correctness	Flexibility	Completeness	Accuracy	Time	User exp.	Learnability
GUI	⊕	⊕	∘	∘	+	∘	∘
Pointer	⊕	⊕	∘	+	−	∘	∘
AR	⊕	⊕	+	+	+	+	∘
NRS	⊕ (→)	⊕ (→)	∘ (→)	+ (→)	∘ (↗)	+ (↗)	∘ (→)

5

Learning From User Interactions

After successfully incorporating additional sensors and actuators of a smart home environment into an interaction method based on a network robot system (NRS), we address the third research question of this thesis in this chapter, i.e. how can a NRS learn from user interactions and apply the knowledge in future interaction processes. Thus, in addition to hardware components of a smart environment, we also investigate learning capabilities of the NRS. To this end, we propose a novel learning and support system (LSS), that learns from multiple user interactions and supports a human in future interaction processes through appropriate recommendations for interactions. This aims to reduce the interaction time to a constant level while not negatively affecting the other user requirements. To test these hypotheses, we close the chapter with an experimental evaluation comprising multiple scenarios and participants.

This chapter's content (in similar or identical form) is mainly based on the publications below:

- SPRUTE, D., P. VIERTEL, K. TÖNNIES, and M. KÖNIG (2019). Learning virtual borders through semantic scene understanding and augmented reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4607–4614
- KÖNIG, M., D. SPRUTE, and P. VIERTEL (2020). Verfahren und Robotersystem zur Eingabe eines Arbeitsbereichs. DPMA Patent DE102019126903B3

5.1 Learning and Support System

All interaction methods available so far (state-of-the-art as well as proposed interaction methods) are based on pure human-robot interaction (HRI), i.e. a human specifies a virtual border by explicitly defining all components of a virtual border. As already identified in Experiment 3 of Chapter 3, this leads to a linear interaction time with respect to the virtual border length[1]. However, if we could learn from user interactions, i.e. the execution of multiple interaction processes, and could support

[1] Although the baseline interaction method features an almost constant interaction time with a minimal slope, its interaction time is still linear.

the human in subsequent interaction processes through appropriate recommendations for interactions, a human would not need to explicitly define all virtual border components, especially the virtual border points $\mathcal{P}$. Instead, a human could only select a recommendation of the system without the effort necessary to define all virtual border components explicitly. Thus, we hypothesize that we can further reduce the interaction time to a constant level independent of the virtual border length.

For this purpose, we first have to learn from user interactions, which leads to Research Question 3.1 of how to encode an interaction process for machine learning. These machine learning algorithms require an input in form of a data vector, e.g. raw sensor data or a feature vector. Therefore, we need to map an interaction process to such a data vector, that adequately represents the underlying interaction process. Since an interaction process deals with the restriction of a mobile robot's workspace, a restriction area represents the result of an interaction process. As described in the scope of this thesis in Section 1.2, such a restriction area usually possesses a certain semantic, e.g. privacy zone, dirty area or carpet. Therefore, the idea is to extract the semantic of a restriction area to encode an interaction process. This semantic can be often derived from the visual appearance of the restriction area when certain objects are covered, e.g. in case of carpets, pets' water dishes or kids' corners. Hence, visual semantic scene understanding can be employed to extract the semantic of a restriction area (GARCIA-GARCIA *et al.*, 2018). However, due to the mobile robot's limited perceptual and computational limitations, we build on the camera network of the smart environment introduced in the previous chapter for scene understanding. In particular, we focus on semantic segmentation with a fine-grained accuracy to consider the accuracy requirement, i.e. algorithms assign a semantic to each pixel of an image. Other forms of scene understanding, e.g. (sub-)image classification or object detection, only extract a semantic with a coarse-grained accuracy, e.g. bounding boxes around objects. This would be sufficient for extracting the semantic of a restriction area but not for giving accurate recommendations for interactions.

After encoding an interaction process, Research Question 3.2 deals with the question of how to learn from user interactions, that are encoded as semantics. The idea is that humans, who specify multiple restriction areas with a certain semantic, also want to specify other restriction areas with the same semantic. For example, if a human specifies multiple carpets as restriction areas, other carpets in the environment could be suggested as restriction areas to the human[2]. This leads to the task of frequent itemset mining as introduced in Subsection 2.4.3, which is an unsupervised learning technique dealing with the discovery of frequent itemsets in a database.

Once a frequent semantic is identified, this knowledge needs to be leveraged to support a human in subsequent interaction processes, which is the content of Research Question 3.3. For this pur-

[2]Although learning does not play a major role in small home environments with only a limited set of restriction areas, a distributed system covering multiple households with multiple user groups could benefit from this approach. However, we restrict our scope in this work to a single household, but the approach can be easily scaled to multiple environments by maintaining a central database.

pose, we first need to identify restriction areas with an identical semantic in the environment. To this end, we again perform semantic scene understanding employing the smart cameras integrated in the environment to localize the potential restriction areas. Based on these areas, the idea is to create recommendations for potential restriction areas and to convey these recommendations to the human using the feedback channel of the interaction method's user interface. However, since a recommendation comprises complex spatial information, the user interface should be able to provide complex feedback, i.e. conveying 2D spatial information. Thus, simple colored light feedback or non-speech audio sound on board the mobile robot is not sufficient. Moreover, in order to allow a human the selection of a recommendation, the user interface needs the ability to specify a 2D position, which indicates a recommendation. Since this capability is a subset of the transfer of 2D spatial information as described in the previous chapters, we can build on this capability. Hence, the properties of an interaction method's user interface do not need to be extended when incorporating learning capabilities.

5.1.1 System Architecture

These ideas lead to the development of a learning and support system (LSS) as visualized in Figure 5.1, whose overall objective is the reduction of the interaction time to a constant level. It mainly consists of two workflows, i.e. a standard and novel workflow depicted as red and green arrows. The standard workflow is the augmented reality (AR) interaction method proposed in Subsection 3.3.2, that is based on pure HRI without learning capabilities. A human interacts with the system through an *Augmented Reality* module enabling the specification of virtual border components and transmission of visual feedback concerning the interaction process. A user-defined virtual border V is passed to the *Virtual Border Integration* module, which incorporates the virtual border into the occupancy grid map (OGM) of the environment M, i.e. performing the map integration algorithm presented in Subsection 3.2.2. This map M is used in the *Human-Aware Navigation* module as basis for a global costmap. Since the resulting map contains physical as well as virtual borders, a mobile robot respects the user-defined workspace and changes its navigational behavior.

This standard workflow is extended by a novel workflow, that incorporates learning capabilities. To this end, a *Scene Understanding* module is integrated, that extracts semantic knowledge about the scene. It depends on a *Scene Perception* module, that captures images of the environment from different viewpoints. For this purpose, we consider a set of RGB cameras $\mathcal{C} = \{C_1, C_2, ..., C_m\}$ integrated in the environment as basis for scene perception. These cameras correspond to the smart camera network presented in the previous chapter. In order to learn from user interactions and create recommendations, a *Learning and Support* module, that combines the resulting map M of a previous interaction process and semantically segmented images $\mathcal{I}_{Sem} = \{{}^{C_1}I_{Sem}, {}^{C_2}I_{Sem}, ..., {}^{C_m}I_{Sem}\}$, is integrated into the workflow. This allows the system to semantically describe the areas defined by the human and learn from these interactions. Furthermore, areas with certain semantics can be

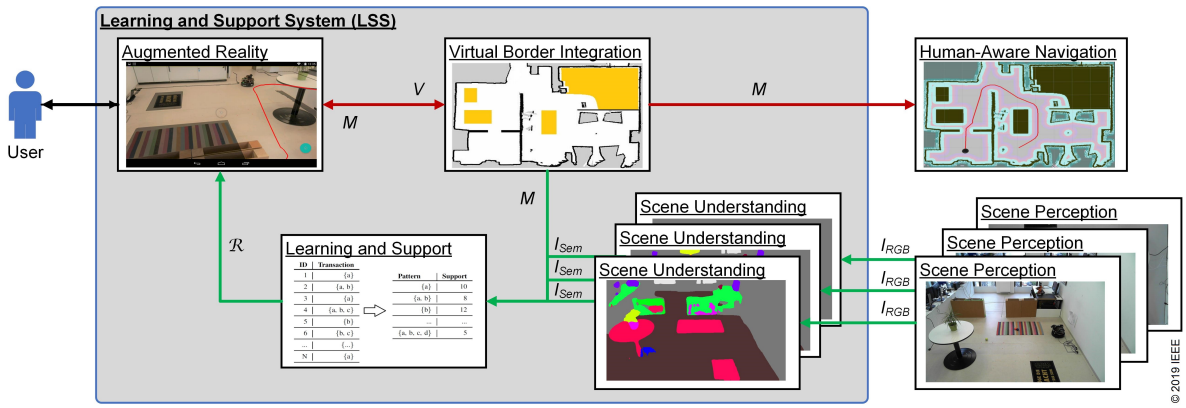


Figure 5.1: System architecture of the learning and support system consisting of several modules and a standard (red arrows) and novel (green arrows) workflow.

identified in the environment. Based on this knowledge, the module can recommend potential virtual borders $\mathcal{R} = \{V_1, V_2, ..., V_n\}$ and support the human in future interaction processes. To this end, support is finally conveyed to the human through the AR interface. Since the learning capabilities are independent of a user interface, we select the AR interface because it provides a more direct feedback channel than the laser pointer in combination with the NRS as proposed in the previous chapter. For example, to get feedback about the interaction process, visual feedback is directly augmented into the video stream of the AR device, while a human has to look at a display integrated into the environment when employing the NRS approach. However, all user interfaces allowing the transfer of spatial information and providing complex visual feedback would be possible for this task. Details of the relevant modules of the LSS are given in the following subsections.

5.1.2 Scene Understanding Module

The first module is the *Scene Understanding* module, which is employed to extract semantic information about the scene from different camera views. Therefore, there is one instance of this module for each camera $C_i \in \mathcal{C}$ integrated in the environment. The input of this module is a color image I_{RGB} provided by the *Scene Perception* module, and the output is a semantically segmented image I_{Sem} assigning a semantic class to each pixel, e.g. plant, ground or wall. This research field of semantic segmentation has made tremendous progress in recent years with the advent of deep learning techniques. These employ deep network architectures incorporating multiple layers, which are trained on huge databases with dedicated graphics processing units (GPUs). This was not possible before 2010 due to the lack of computational capacity and large annotated databases. A recent overview of deep learning techniques applied to semantic segmentation is given by GARCIA-GARCIA *et al.* (2018). Their article highlights important deep network architectures, such as AlexNet (KRIZHEVSKY *et al.*, 2012), VGG (SIMONYAN and ZISSERMAN, 2014), GoogLeNet (SZEGEDY *et al.*, 2015) or ResNet (HE

et al., 2016), and their performances in the important ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in the last years underlining the strong progress in this field (RUSSAKOVSKY *et al.*, 2015). Therefore, we build on an existing state-of-the-art deep network architecture in this module. To this end, we employ an encoder-decoder architecture consisting of a ResNet101 (HE *et al.*, 2016) with dilated convolutions as encoder and a pyramid pooling module (ZHAO *et al.*, 2017) as decoder. The model is pre-trained on the MIT Scene Parsing Benchmark (SceneParse150), which is a standard training and evaluation platform (20K/2K/3K images for training, validation and testing) based on the ADE20K dataset (ZHOU *et al.*, 2019). This dataset contains more than 20K scene-centric images annotated with semantic categories on pixel-level. The benchmark comprises 150 semantic categories from outdoor as well as indoor scenes. We choose this dataset because it contains indoor semantic categories, that are relevant for our scenarios. Furthermore, since the images are scene-centric, they can be successfully applied to similar scenes. The deep network architecture is chosen due to its state-of-the-art performance on the benchmark.

5.1.3 Learning and Support Module

This is the main module of the LSS, which is intended (1) to learn from user interactions, i.e. the semantics of previously user-defined virtual borders, and (2) to support subsequent interaction processes through the creation of appropriate recommendations for virtual borders $\mathcal{R}$. For this purpose, this module depends on the output of the *Virtual Border Integration* and *Scene Understanding* modules. The idea of learning is to identify frequent user interactions, e.g. a human often specifies restriction areas of a certain semantic, and to create recommendations for virtual borders with an identical semantic. Therefore, we formulate the problem as a frequent itemset mining task (FOURNIER-VIGER *et al.*, 2017). In this context, we denote the set of all items as $\mathcal{I} = \{i_1, i_2, \cdots, i_m\}$ and a transaction T as a subset of the itemset $\mathcal{I}$, thus $T \subseteq \mathcal{I}$. The input for this task is a transactions database $D = \{T_1, T_2, \cdots, T_n\}$ consisting of n transactions. The support of an itemset $\mathcal{X}$ is the number of transactions in the database D containing the itemset $\mathcal{X}$, i.e. $support(\mathcal{X}) = |\{T | T \in D \wedge \mathcal{X} \subseteq T\}|$. It is the objective of this task to determine frequent itemsets $\mathcal{F}$ with $support(\mathcal{F}) \geq minSupport$ where $minSupport$ is a threshold parameter specifying a minimum support value. To adapt this problem definition to our problem, we consider a semantic of a user-defined virtual border as an item $\alpha \in \mathcal{I}$, and a transaction T is a session in which a human specifies multiple virtual borders. Thus, frequent user interactions are identified by solving this task.

In order to seamlessly incorporate this problem formulation into this module, there are three main steps necessary:

1. The extraction of a restriction area's semantic from multiple camera views.
2. The identification of frequent user interactions, i.e. semantics.
3. The creation of appropriate recommendations for user interactions.

Semantic Extraction

The goal of this first step is the extraction of a restriction area's semantic from multiple camera views. This step is performed whenever a new map M_t is generated by the *Virtual Border Integration* module at index t, i.e. a human has specified a new virtual border. Algorithm 5.1 gives details on the realization, that additionally depends on the images $\mathcal{I}_{Sem}$ of the *Scene Understanding* module. These are always the most recent images of the cameras. At the beginning, a $mask$ is created that indicates map coordinates that belong to the last user-defined virtual border (l. 3). Subsequently, for each point $p \in mask$, we determine the cameras whose field of view cover this position (l. 4f.). If a field of view of a camera c covers the point p, this point is projected into the image space of camera c and the corresponding semantic value s is extracted (l. 7f.). Afterwards, the corresponding value in $histogram$ is incremented, which stores the number of occurrences per semantic (l. 9). Hence, observations from multiple cameras are combined. After iterating over all points $p \in mask$, the majority semantic α is determined and assigned to the last user-defined virtual border (l. 10).

Algorithm 5.1: Semantic extraction step of the learning and support module.

```
   Input: M_t: map at timestamp t
   Input: I_Sem: set of semantic images
   Output: α: semantic
1  Function semanticExtraction(Input, Output)
2      histogram = ∅;
3      mask = M_t - M_{t-1};
4      foreach p in mask and p ≠ 0 do
5          cams = getCorrespondingCameras (p);
6          foreach c in cams do
7              p' = transformIntoImageSpace (p, c);
8              s = getSemantic (p', ^cI_Sem);
9              histogram[s] = histogram[s] + 1;
10     α = getMajorityKey (histogram);
11     return α;
```

Frequent User Interaction Mining

The extracted semantic is subsequently added as an item α to a new transaction T_{new} and stored in the transactions database D. Additionally, we store some morphological characteristics of the virtual border concerning its area and shape to validate recommendations according to their semantic-specific characteristics later. If the database already contains a transaction in the same user session,

i.e. within a certain time interval, the item α is added to an existing transaction T_{old}. Hence, transactions with a cardinality $|T| > 1$ emerge. If a user defines multiple virtual borders with the same semantic α in a user session, we also insert multiple transactions into the database D. In order to learn from the data and extract frequent itemsets, we apply the FP-Growth algorithm proposed by HAN *et al.* (2000) on the transactions database D. It is an efficient method for mining the complete set of frequent patterns by pattern fragment growth. As a result, we obtain frequent itemsets, that we use to identify frequent semantics $\mathcal{S}$.

Creation of Recommendations

Finally, the identified frequent semantics $\mathcal{S}$ are leveraged to create appropriate recommendations for virtual borders $\mathcal{R}$. For this purpose, restriction areas with the same semantics need to be localized in the environment employing the semantic images $\mathcal{I}_{Sem}$ of the *Scene Understanding* module. This procedure is described in Algorithm 5.2, that is triggered whenever a new semantic image ${}^{C_i}I_{Sem}$ of a camera $C_i \in \mathcal{C}$ is available. The output of the algorithm is a recommendation for a set of virtual borders $\mathcal{R} = \{V_1, V_2, ..., V_n\}$. Since multiple camera observations need to be combined and virtual border recommendations need to be extracted, this task is similar to the cooperative perception in Subsection 4.1.3. Instead of combining observations of laser points, boundaries of potential virtual borders are combined. Thus, we build on the multi-stage algorithm for virtual border extraction introduced in the previous chapter. However, before combining boundaries of potential virtual borders, these first need to be localized. To this end, the algorithm creates a binary $mask$ for each frequent semantic s as basis for blob detection, which extracts the contour of connected pixels (l. 4f.). The separation concerning the frequent semantics enables the creation of hierarchical recommendations, e.g. a water dish is placed on a carpet. Afterwards, the contour of each blob is transformed into the map's coordinate frame M and stored along with its semantic value and corresponding camera (l. 6f.). To combine this point set with observations of the other cameras, the points from the other cameras are retrieved and added to the $points$ (l. 8f.). Hence, we obtain a set of potential virtual border points of a certain semantic s, which is independent of a camera. It can contain multiple virtual borders with the same semantic, e.g. multiple carpets, but also noisy points due to errors in the semantic segmentation. Furthermore, inaccurate data points can occur due to inaccuracies in the extrinsic camera calibration and the segmentation results. Thus, at this point we face the same challenges as in the virtual border extraction step in Subsection 4.1.3. Therefore, we adapt our multi-stage algorithm from the previous chapter. To this end, we first perform a density-based clustering as described in Algorithm 4.1 to extract clusters with certain characteristics, e.g. appropriate size or expansion (l. 10). The validation concerning morphological characteristics is aimed to reduce false recommendations. Afterwards, each cluster d is thinned to account for inaccuracies employing Algorithm 4.2 (l. 12), and a virtual border V is extracted employing Algorithm 4.3 (l. 13). This virtual border V is then added to the recommendations $\mathcal{R}$ if it does not already

exist in the OGM (l. 14). These recommendations are subsequently sent to the *Augmented Reality* module for visualization, which closes the human-centered interaction loop.

Algorithm 5.2: Recommendation step of the learning and support module.

```
   Input: C_i I_Sem: semantic image of camera C_i ∈ C
   Input: S: frequent semantics
   Output: R: set of virtual borders for recommendation
 1 Function createRecommendation(Input, Output)
 2     R = ∅;
 3     foreach s in S do
 4         mask = getMask (C_i I_Sem, s);
 5         blobs = blobDetection (mask);
 6         points = transformIntoMapSpace (blobs);
 7         store (s, C_i, points);
 8         foreach c in C do
 9             points = points ∪ get (s, c);
10         clusters = clustering (points);
11         foreach d in clusters do
12             d = thinning (d);
13             V = extractBorder (d);
14             R = R ∪ V;
15     return R;
```

5.1.4 Augmented Reality Module

This module acts as bidirectional interface between the human and system. It is an extension of the AR user interface proposed in Subsection 3.3.2, which allows a human to explicitly define virtual borders using a RGB-D tablet. Our extension includes the support for recommended virtual borders $\mathcal{R}$, i.e. restriction areas that are suggested by the *Learning and Support* module. In order to interact with the system, a human freely moves with the tablet in the environment while the tablet's screen shows an augmented video stream containing information, such as physical areas or virtual borders as shown in Figure 5.2. The explicit definition of a virtual border V with its boundary points $\mathcal{P}$ and seed point $\boldsymbol{s}$ is realized by pointing the tablet's center towards the desired physical locations in the environment and selecting these points using software buttons. The occupancy probability δ can be changed in a pop-up menu. In addition to this interaction based on pure HRI, recommendations for virtual borders $\mathcal{R}$ are visualized on the tablet's screen if the *Learning and Support* module identifies appropriate restriction areas (see Figure 5.2f). In this case, a human does not need to explicitly define all virtual border components $(\mathcal{P}, \boldsymbol{s}, \delta)$ explicitly, but can rather select

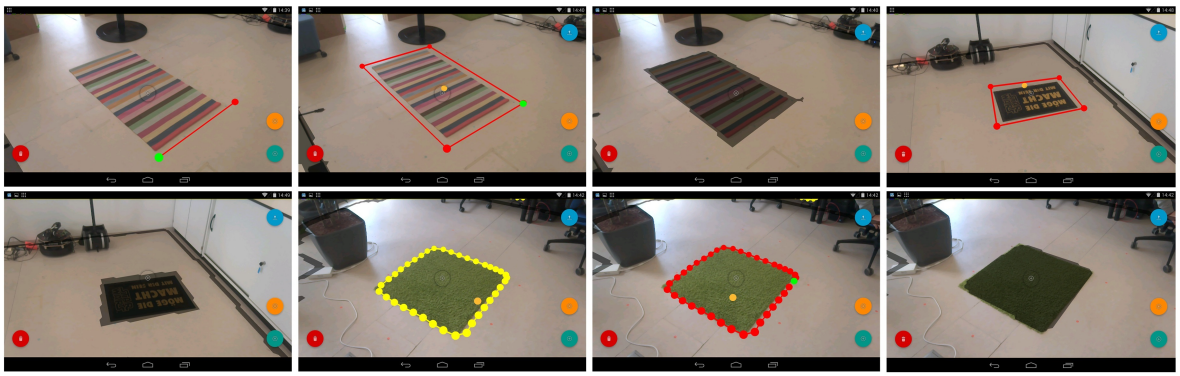


Figure 5.2: Row-wise from top left to bottom right (a-h): a human defines virtual borders around two different carpets by explicitly specifying all components of a virtual border (a-c and d-e). After defining these virtual borders, the corresponding areas are visualized as occupied (black) areas (c and e). Based on these user interactions, the system suggests a virtual border for another carpet shown as yellow boundary (f). The human can simply select the recommendation without explicit definition of corner points (g) and integrate the virtual border into the map (h).

the recommendation by pointing towards the corresponding boundary (see Figure 5.2g). Since a recommendation already comprises all components of a virtual border, a recommendation can be directly selected by the human and sent to the *Virtual Border Integration* module. This process aims to avoid the linear interaction time with respect to the length of a virtual border, that goes along with the explicit definition of the virtual border components employing the standard workflow. Thus, we hypothesize that this novel workflow including learning capabilities can reduce the interaction time compared to the standard workflow.

5.2 Evaluation

After presenting details of the LSS, we experimentally evaluate the system concerning our objective of a reduction of interaction time. To this end, we conduct two different experiments covering different requirements. While the first experiment validates the LSS concerning its recognition rate and accuracy, the second experiment evaluates the user requirements concerning interaction time, completeness, learnability and user experience. An evaluation concerning correctness and flexibility is not necessary because the LSS is an extension of the previous interaction methods, that have already been proven to be correct and flexible in Subsection 3.4.7.

5.2.1 Experiment 1: Recognition Rate and Accuracy

The first experiment deals with the recognition rate of the LSS and the accuracy of the recommendations for virtual borders. The recognition rate indicates if potential restriction areas in the envi-

ronment are correctly recognized by the LSS, while the accuracy measures the overlap of the recommendations with ground truth data. This is necessary to validate if the recommendations of the LSS achieve acceptable recognition rates and if they preserve the high accuracy of the standard workflow based on pure HRI. Thus, this is a validation step that needs to be passed before evaluating the other user requirements.

Independent Variables

While the evaluation of the recognition rate only applies to the LSS, the accuracy is also compared to the standard workflow. Thus, we manipulate the interaction method as single independent variable, which can have one of the two values below:

1. **Augmented Reality (AR):** This is the interaction method based on AR as presented in Subsection 3.3.2. It is based on pure HRI where a human explicitly defines all virtual border components. Hence, this represents the standard workflow of the LSS without learning capabilities.
2. **Learning and Support System (LSS):** This is the proposed LSS, which extends the standard by a novel workflow incorporating learning capabilities.

Hypotheses

The objective of the experimental evaluation is the test of the following hypotheses:

- **Hypothesis 1:** The recommendations of the LSS achieve acceptable recognition rates ($\geq 85\%$)[3].
- **Hypothesis 2:** The accuracy does not decrease when employing the LSS compared to the standard AR approach.

Setup & Procedure

To test both hypotheses, we first create typical restriction areas in the environment based on three categories, i.e. (1) carpets, (2) pets' water dishes and (3) kids' corners indicated by toy blocks (boxes). These categories are inspired by the scenarios mentioned in the introduction, and their semantics can be visually derived from the appearance. While an area of the former category consists of a single object, i.e. a carpet, the latter categories can be composed of multiple objects, e.g. a kids' corner can be composed of multiple boxes. We also choose different object instances, e.g. dishes with different colors and sizes, to make the scenes more realistic. In a learning phase, a human defines multiple virtual borders for these restriction areas by explicit interaction through the AR

[3] We choose the value of 85% because it is a high value indicating a robust system while at the same time allowing a small percentage of misclassifications.

interface without recommendations. Thus, the semantics of the user-defined virtual borders are stored in the transactions database D, but no recommendations $\mathcal{R}$ for the human are created. This learning phase is performed to add items to the transactions database D.

Afterwards, we validate our system based on the current content of the transactions database D in a supporting phase, i.e. the creation of recommendations for virtual borders. For this purpose, we create a dataset containing images with a resolution of 1920 × 1080 pixels from five different perspectives of a 10 × 8 m indoor environment. For each of these perspectives, we create a set of 20 different scenes inspired by typical home environments in the scope of this thesis. Each scene is unique and contains one or multiple categories for restriction areas. We also vary the setting of each scene by adding or removing additional furniture, e.g. chairs or tables, or by changing the positions of the restriction areas or furniture. Some exemplary scenes for different perspectives are depicted in Figure 5.3, and the characteristics of our dataset are summarized in Table 5.1. There is an unbalanced number of objects in the dataset because a restriction area can be composed of multiple objects in case of pets' water dishes and kids' corners. This is caused by the different sizes of the objects, i.e. dishes and boxes are smaller than carpets.

Table 5.1: Characteristics of the dataset including number of objects in the dataset and number of scenes containing at least one object of the category. The scenes are subdivided according to the five perspectives.

		Number of scenes containing object					
Category	Number of objects	Persp. 1	Persp. 2	Persp. 3	Persp. 4	Persp. 5	Sum
Carpet	72	18	12	11	12	11	64
Water dishes	152	7	10	10	11	11	49
Kids' corner	220	7	9	10	12	12	50

For each scene, we also create a ground truth OGM with a resolution of 2.5 $^{\text{cm}}/_{\text{cell}}$ containing the positions of the virtual borders. This is necessary to allow an assessment of the recommendations concerning the recognition rate and accuracy. A special case is the ground truth annotation of small objects, e.g. boxes or dishes, where even a small deviation of a few pixels would result in an unacceptable accuracy. Therefore, we additionally add a tolerance area of 1-2 cells in the OGM (2.5-5 cm) around small objects with a diameter smaller than 12 cells (30 cm)[4]. Furthermore, this tolerance area is extended between objects of the same category if they are less than 30 cm away from each other, e.g. in case of kids' corners consisting of multiple boxes. Overall, we acquire a total of 100 unique scenes containing 4.4 objects of restriction areas on average. The minimum and maximum number of restriction area objects per scene range between 1 and 14.

[4]According to RUSSAKOVSKY *et al.* (2015), the average human annotation error is 5 pixels on each dimension. Thus, this is a legitimate procedure.

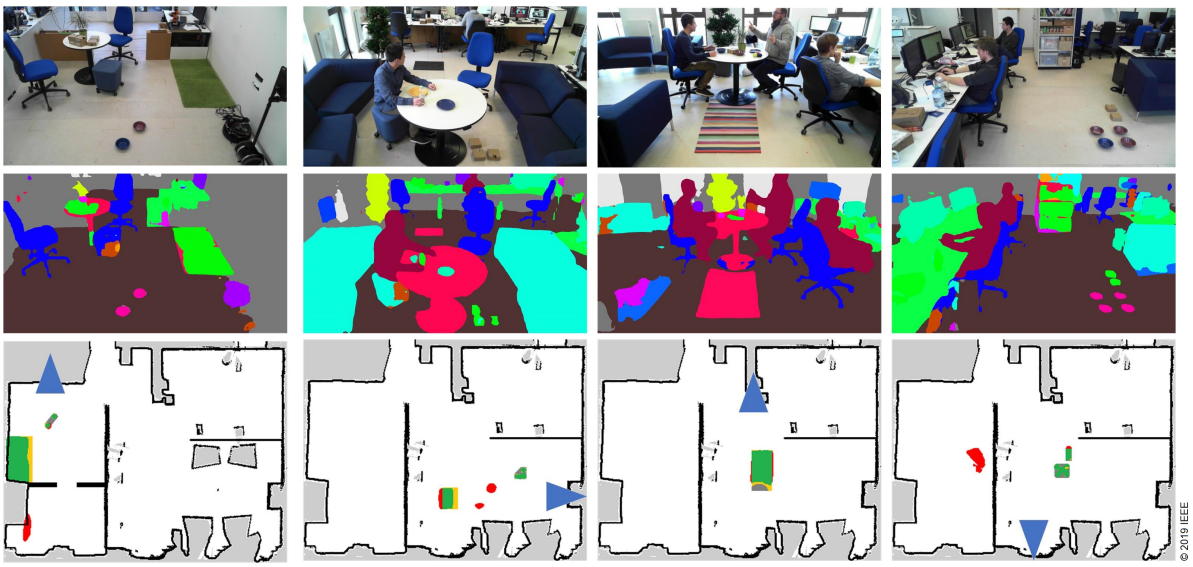


Figure 5.3: First row shows exemplary RGB images from four different perspectives containing typical restriction areas. Second row visualizes inference results of the *Scene Understanding* module where each color corresponds to a certain semantic. The last row contains maps with ground truth areas (green and yellow) and recommendations of the system (green and red). Thus, green cells indicate a correct overlap between ground truth and recommendation areas, while red areas indicate false recommendations. A blue triangle indicates the position of a camera in the scene.

Measurement Instruments

In order to measure the accuracy of a recommendation, we consider the Jaccard similarity index (JSI) between a recommended and ground truth area. However, to consider the tolerance area of an object, we modify the definition of the JSI from the previous chapters as follows:

$$J(GT, GT_{TOL}, R) = \frac{|GT_{TOL} \cap R|}{|GT \cup R|} \in [0,1] \tag{5.1}$$

The area of a recommendation is defined as R and the ground truth area as GT. The tolerance area of a restriction area, an area that can, but does not need to be covered by a recommendation, is denoted as $GT_{TOL} \supseteq GT$.

Regarding the recognition rate, the number of restriction areas in the environment is considered as T. The LSS can correctly recommend a restriction area, i.e. a true positive classification TP, or falsely recommend another area, which is not part of the restriction area, i.e. a false positive classification FP. To distinguish between these outcomes, we consider a true positive TP if the recommendation area intersects with the ground truth and its tolerance area by more than 50%, i.e. the JSI between the areas exceeds a threshold of 50%, otherwise the recommendation is considered

as false positive FP[5]. If objects of a restriction area with the same category are spatially close to each other, a single recommendation can cover multiple objects. For example, if a kids' corner is composed of multiple boxes, these can be covered by a single recommendation.

These definitions of the binary outcome of the LSS are used to define measures for the recognition rate. The first measure is the *recall*, which gives information about the ratio between the true positive classifications TP and the total number of restriction areas T. It is defined in Equation 5.2:

$$Recall = \frac{TP}{T} \tag{5.2}$$

Another measure is the *precision*, that indicates the number of true positive classifications TP with respect to the total number of recommendations consisting of true positives TP and false positives FP. It is defined in Equation 5.3:

$$Precision = \frac{TP}{TP + FP} \tag{5.3}$$

Both measures can be combined as harmonic mean resulting in the *F-score*, which is defined in Equation 5.4:

$$F - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.4}$$

This F-score is finally used as measure for the recognition rate.

Analysis & Results

Recognition Rate The results of the recognition rate are summarized in Table 5.2. In general, a high recall is more important than a high precision in our case because the recall indicates the percentage of restriction areas, that are recognized in the environment. In contrast to this, the precision indicates the percentage of correctly recognized restriction areas with respect to the total number of recommendations. Thus, a low precision implies much false recommendations, i.e. false positive classifications FP. However, since these are only recommendations of the system, these false recommendations do not need to be selected by a human in an interaction process and will thus not be integrated into the OGM of the environment M. Overall, the LSS achieves a recall of 93.5% and a precision of 89.7% resulting in an F-score of 91.5%. This shows that the LSS can correctly recognize restriction areas and that there are only a few false recommendations. Additionally, the results also reveal that some categories are better recognized than others, e.g. the F-score of carpets (97.2%) is higher than the F-score of dishes (86.9%).

[5]The overlap of 50% is a typical threshold to distinguish between true positives TP and false positives FP, e.g. in PASCAL VOC (EVERINGHAM *et al.*, 2010) or ILSVRC (RUSSAKOVSKY *et al.*, 2015).

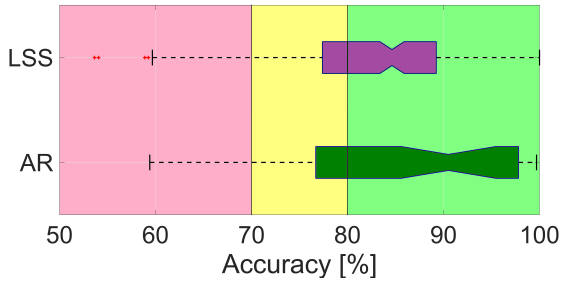


Figure 5.4: Boxplots showing the accuracy of the LSS compared to the standard AR approach. The background colors indicate the quality levels ranging from unacceptable (red) to acceptable (yellow) and good (green).

Some qualitative results of the recognition rate (and also accuracy) are depicted in the last row of Figure 5.3. The corresponding inference results also reveal that a category of a restriction area can encompass multiple different semantics. For example, a carpet is either recognized as *carpet* (red) or *grass* (green), which is indicated by two different colors.

Table 5.2: Results of the recognition rate.

Category	Recall	Precision	F-score
Carpet	0.958	0.986	0.972
Water dishes	0.875	0.864	0.869
Kids' corner	0.968	0.891	0.928
Weighted average	0.935	0.897	0.915

Accuracy The results of the accuracy evaluation considering the recognized restriction areas are visualized in Figure 5.4 as boxplots[6]. The LSS features a good accuracy on average ($M = 83.2\%$), which is slightly worse compared to the AR approach ($M = 86.2\%$). Moreover, the interquartile range of the AR approach ($IQR = 21.1\%$) is almost twice the interquartile range of the LSS ($IQR = 11.8\%$). In addition, the distribution of the accuracy values strongly overlaps between the interaction methods indicating no difference in accuracy between both approaches.

To statistically verify this observation, we first explore the data concerning outliers and normal distribution, which are assumptions of an unpaired t-test. The visual inspection of the boxplots reveals that there are only a few outliers in the data of the LSS and no outliers in the data of the AR approach. However, Shapiro-Wilk tests become significant for the data indicating a non-normal distribution. Due to this violation of normality, we prefer a non-parametric Mann-Whitney U test to an unpaired

[6]The accuracy results of the AR approach are taken from Experiment 2 of Chapter 3 comprising performances of 15 non-expert humans in three different scenarios.

t-test to check if the accuracy between both interaction methods differs significantly. The result of the Mann-Whitney U test suggests that there is no significant difference in the accuracy between the LSS and the AR approach without learning capabilities ($U = 3981.0$, $p = 0.062$).

Moreover, since the interaction with the LSS focuses on the selection of recommendations by a human, it is also important that a recommendation achieves an at least acceptable accuracy to be selected by a human. Table 5.3 shows the percentage of recommendations with a certain quality level with respect to the number of recognized restriction areas. The results reveal that 89.8% and 68.1% of the recommendations feature an acceptable and good accuracy. Compared to the results of the AR approach, these values are similar, which underlines that the accuracy of the LSS does not differ significantly from the AR approach.

Table 5.3: Percentage of recommendations (and user interactions) with a certain quality level.

Method	Acceptable	Good
LSS	0.898	0.681
AR	0.933	0.622

Discussion

The experimental results of the recognition rate demonstrate that the LSS achieves an average F-score of 91.5%. Since this value is higher than the threshold of 85% defining an acceptable recognition rate for our problem, we conclude that Hypothesis 1 of this experiment is supported by the results. Moreover, we observed that the recognition rates slightly differ between different categories of restriction areas, e.g. the recognition rate of carpets is higher than the recognition rate of dishes. This is due to the fact that small objects on the ground are often relatively small in image space compared to the size of an image. Thus, they are harder to recognize by the smart cameras integrated in the environment. Furthermore, a smaller precision is caused by false recommendations (FP) because there are objects of this category in the scene, but they are not part of a restriction area, e.g. dishes on a table. This case is extremely rare for carpets resulting in a higher precision.

Moreover, the inference results show that a restriction area's category can comprise multiple semantics, e.g. a carpet is recognized as *carpet* or *grass*. Nonetheless, our LSS can cope with this ambiguity since frequent semantics, which are the basis for recommendations of the system, are determined using frequent itemset mining. In case of an ambiguity, more user interactions would be necessary to consider a restriction area's category as frequent because each user interaction is encoded as a single semantic. Hence, a restriction area's category is divided among several semantics where each semantic has to exceed the threshold parameter $minSupport$ of the frequent itemset mining task. Thus, the learning phase would take longer, but this does not affect the recognition rate.

Concerning the accuracy, the statistical results reveal that there are no significant differences between the LSS and AR approach. Hence, the LSS performs at least as good as the AR approach, which supports Hypothesis 2 of this experiment. In addition, a large majority of the recommendations should be accepted by humans since 89.8% of the recognized restriction areas feature an at least acceptable accuracy. Hence, only 10.2% of the recommendations are unacceptable and should thus be rejected by a human. Moreover, the smaller interquartile range of the LSS is caused by the automatic recognition of the restriction areas by the smart camera network. Therefore, there are no inaccuracies introduced by different participants and their characteristics. Instead, the accuracy is mainly affected by results of the semantic segmentation in the *Scene Understanding* module and the calibration of the smart cameras to determine the 3D positions on the ground.

5.2.2 Experiment 2: Remaining User Requirements

After validating the LSS concerning an acceptable recognition rate and good accuracy without degradation compared to the standard AR approach, we perform another experiment including multiple participants, who evaluate the LSS in multiple scenarios in a lab experiment concerning the other user requirements, i.e. interaction time, completeness, learnability and user experience[7].

Independent Variables

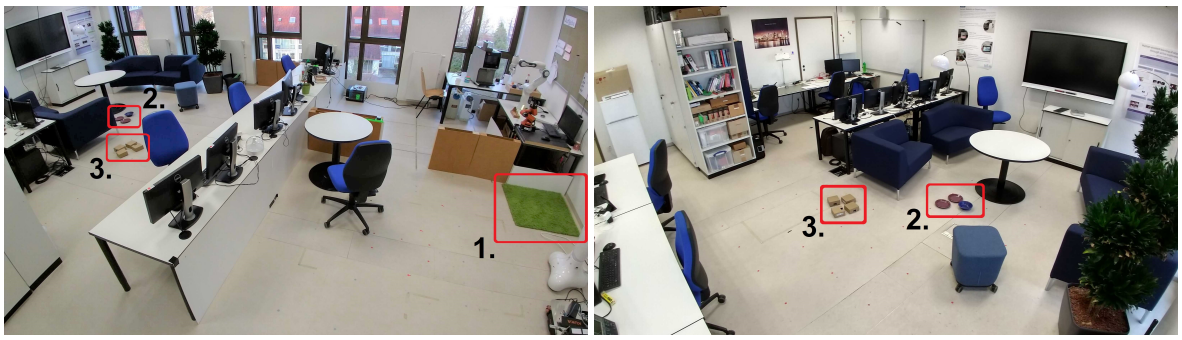
The LSS serves as interaction method in this experiment. However, to compare the results with the standard AR approach, we take the data from the experiments in Chapter 3. Thus, we manipulate the same independent variable as in the previous experiment, i.e. the interaction method.

Hypotheses

The objective of this experiment is the test of the following hypotheses:

- **Hypothesis 1:** The LSS achieves a better interaction time than the standard AR approach.
- **Hypothesis 2:** The completeness does not decrease when employing the LSS compared to the standard AR approach. Thus, the completeness should achieve a good quality level.
- **Hypothesis 3:** The learnability does not decrease when employing the LSS compared to the standard AR approach. Moreover, the learnability should achieve a good quality level.
- **Hypothesis 4:** The user experience does not decrease when employing the LSS compared to the standard AR approach. Thus, the user experience should achieve a good quality level.

[7] This experiment is an extension of the original experiment described in (SPRUTE *et al.*, 2019). We increased the number of participants with the goal to increase the fit between the characteristics of the experiment's participants and the intended user group of this work. Thus, the results slightly differ from the reported results of the original experiment.



(a) **(b)**

Figure 5.5: Images of the experimental environment comprising three restriction areas: (1) a carpet, (2) pets' water dishes and (3) a kids' corner.

Setup

These hypotheses are tested in an experimental evaluation, which takes place in the same 10 × 8 m indoor environment as the previous experiment. Thus, the environment is composed of different objects, such as tables, chairs, sofas, plants or displays as illustrated in Figure 5.5. This setup is inspired by typical home environments in the scope of this thesis. In addition, we set up a restriction area of each category in the environment, i.e. (1) a carpet, (2) pets' water dishes and (3) a kids' corner indicated by boxes. Due to the different categories, these restriction areas also have different border lengths and shapes. To guarantee reproducibility of the experiment, we choose three images of the dataset from the previous experiment as input for the *Learning and Support* module. Before conducting the experimental evaluation, the LSS accomplishes the same learning phase as described in the previous experiment. Hence, the LSS already identified frequent semantics $\mathcal{S}$ and can create recommendations $\mathcal{R}$ for the three virtual border categories to support a participant.

Procedure

After setting up the experimental environment, each participant is introduced to the LSS by an experimenter, i.e. description of the scenarios and how to use the AR interface for the restriction of the mobile robot's workspace (with and without recommendations of the LSS). Afterwards, each participant fills a form with general information, such as age, gender and experience with tablets. Subsequently, a participant is asked to specify the three virtual borders employing the LSS and its recommendations for virtual borders. The initial position of a participant in the environment is randomly chosen. The interaction device is the same 7-inch Google Tango tablet as used in the experiments of Chapter 3. A participant can choose the order of the restriction areas on his/her own,

e.g. first a kids' corner, then water dishes and finally a carpet area, to avoid order effects. This procedure is performed three times per participant to investigate improvements between consecutive runs of the experiment, which is necessary to demonstrate a good learnability. After the practical part, each participant fills a post-study questionnaire containing statements concerning learnability and user experience. The experiment takes approximately 15 minutes per participant.

Participants

This procedure is conducted by a total of 20 participants (14 male, 6 female), who are recruited from the local environment by word of mouth. Their ages range between 18 and 58 years with a mean age of $M = 30.90$ years and standard deviation of $SD = 11.58$ years. Participants rate their experience with tablets on a 5-point Likert item ranging from *no experience* (1) to *highly experienced* (5) with a mean of $M = 3.75$ and standard deviation of $SD = 1.25$. Thus, participants have a moderate to good confidence with tablets on average. Since we assume the intended user to be able to interact with common consumer devices, such as tablets or smartphones, the participants fulfill this criterion.

Measurement Instruments

In order to test the hypotheses of this experiment, we need to measure the interaction time, learnability, user experience and completeness. Since these criteria have already been evaluated in previous experiments of this work, we only summarize the measurement instruments in this section.

We define the interaction time as the time between the selection of a recommendation and the final integration of the virtual border into the OGM of the environment, which corresponds to the definition in Experiment 3 of Chapter 3. We do not include the whole interaction process, i.e. starting with a sign of an experimenter as in Experiment 2 of Chapter 3, because we want to avoid the linear interaction time with respect to the border length. Thus, the additional time needed to move in the environment would corrupt the measurements. In case of the learnability and user experience, we apply a questionnaire similar to the previous ones dealing with aspects, such as problems, intuitiveness, effort and feedback[8]. Additionally, participants are asked if they can realize a benefit/advantage of the proposed system. This question is not directly related to a certain requirement but gathers further insights into participants' opinions about the system. Each aspect of the questionnaire can be rated on a 5-point Likert item with numerical response format. Moreover, the questionnaire provides a possibility for free responses allowing additional comments of the participants. In addition to the questionnaire, we also check if a participant improves his/her performance with respect to the completeness and interaction time during the repeated runs of the experiment[9].

[8] The questionnaire also included other statements, which are irrelevant for this thesis.

[9] In this case, we include the whole interaction process in the time measurement starting with a sign of an experimenter.

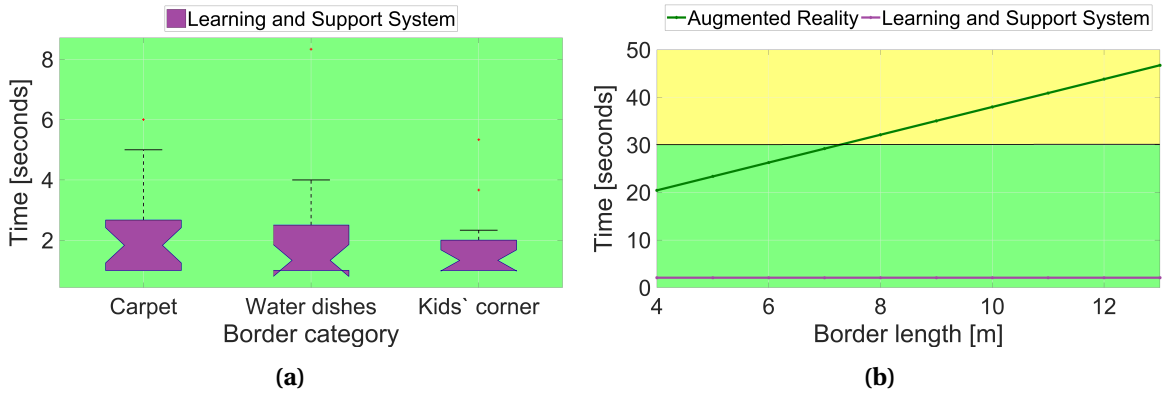


Figure 5.6: Results of the interaction time depending on (a) the category and (b) the length of a virtual border. The background colors indicate the quality levels ranging from acceptable (yellow) to good (green).

This is necessary to claim a good learnability as defined in Section 1.3. Finally, the completeness is assessed by the success rate of the performed runs of the experiment.

Analysis & Results

Interaction Time The time measurements per virtual border category are summarized as boxplots in Figure 5.6a. Each participant contributes a single data point to each category by calculating the mean value of his/her three performances. The results show that the interaction time is always on a good quality level independent of the virtual border category. Moreover, by visually inspecting the plots, there is no significant difference between the different categories of virtual borders.

To statistically verify this observation, we run a non-parametric Friedman test to identify differences between the time measurements of the three virtual border categories. Although the data contain only a few outliers, we prefer this test to a repeated measures analysis of variance (ANOVA) because Shapiro-Wilk tests become significant indicating a non-normal distribution of the data. Furthermore, the number of participants does not exceed a value of 25, which would justify neglecting this violation of assumption of a parametric statistical method. The result of the Friedman test does not become significant for the measurements of the interaction time ($\chi^2(2) = 3.313, p = 0.197$), which means that there is no difference between the different virtual border categories and that the interaction time is independent of the virtual border category. Thus, since the evaluation scenarios cover different virtual border lengths and sizes, the interaction time is also independent of the virtual border length and size. Therefore, the LSS features a constant interaction time with a mean of $M = 2$ seconds. This is the average time needed by a participant to select a recommendation for a virtual border and to integrate it into the environment's OGM M. In contrast to this, the AR

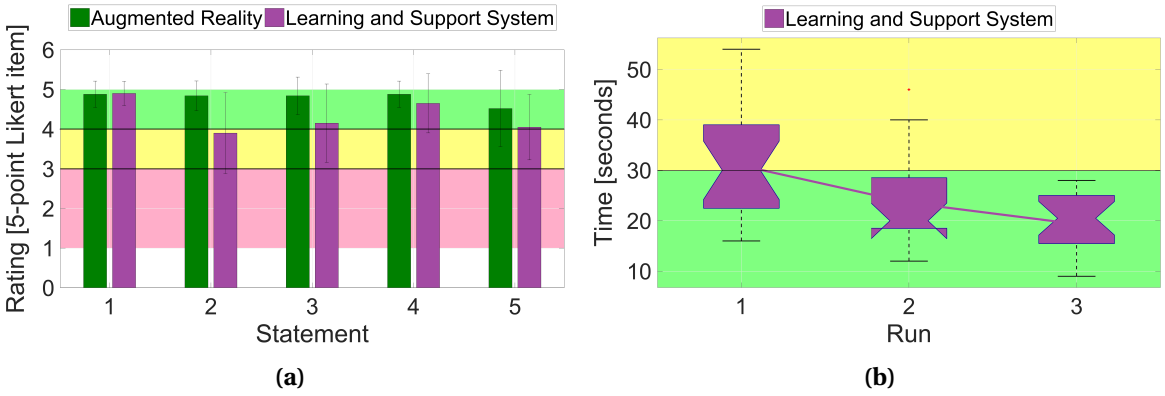


Figure 5.7: Results of (a) the answers to the questionnaire (mean and standard deviation) and (b) the interaction time depending on the runs of the experiment. The background colors indicate the quality levels ranging from unacceptable (red) to acceptable (yellow) and good (green).

approach features a linear interaction time with respect to the virtual border length as depicted in Figure 5.6b. The visualized data of the AR method are based on linear regression on the data of the experiment described in Subsection 3.4.6.

Completeness The completeness indicates the success rate with which a participant successfully performs an interaction process. In this experiment, all participants could successfully accomplish all interaction processes resulting in a completeness of 100%. Since the participants of the experiment described in Subsection 3.4.5 also achieved a good completeness on average with the AR method ($M = 97.8\%$), there is no significant difference between both interaction methods.

Learnability and User Experience Regarding the hypotheses concerning learnability and user experience, we consider the answers to the questionnaire, which are summarized in Figure 5.7a. The data of the AR approach are taken from the experiment described in Subsection 3.4.4, where participants compared the AR approach to the interaction methods based on a laser pointer and graphical user interface (GUI). The bar charts reveal that the LSS achieves a good quality level on all statements, except of the second statement where the value slightly falls below the threshold of a good quality level ($M = 3.90$, $SD = 1.02$). Besides, the mean values of the LSS do not reach the values of the AR approach in terms of the user experience aspects ($S2$-$S5$).

These observations are confirmed when statistically testing for differences between both interaction methods. Since Likert-item data violate the assumption of normality, we prefer non-parametric Mann-Whitney U tests to unpaired t-tests for comparison. The results of the statistical tests are reported in Table 5.4. There is no difference in case of the learnability statement ($S1$), where the

LSS ($M = 4.90$, $SD = 0.31$) performs similarly to the AR approach ($M = 4.88$, $SD = 0.33$). Moreover, the comfort/effort ($S4$) is similarly rated for both interaction methods (AR: $M = 4.88$, $SD = 0.33$ and LSS: $M = 4.65$, $SD = 0.75$). However, there are differences in case of problems, intuitiveness and feedback where the AR approach outperforms the LSS. Nonetheless, participants realized the benefit/advantage of the system well ($M = 4.45$, $SD = 0.69$), which is revealed by the answers to the additional question of the questionnaire.

Table 5.4: Statistical results of the answers to the questionnaire comparing both interaction methods. A * indicates a significant result.

Statement	Aspect	Statistic	p-value
S1	Learnability	$U = 245.0$	$p = 0.834$
S2	Problems	$U = 94.5$	$p < 0.001$*
S3	Intuitiveness	$U = 141.0$	$p = 0.002$*
S4	Comfort/Effort	$U = 216.0$	$p = 0.242$
S5	Feedback	$U = 153.0$	$p = 0.014$*

In addition to the answers to the questionnaire, we also investigate the development of the participants' performances in multiple runs of the experiment, which is necessary to demonstrate a good learnability. Since the completeness already achieved a value of 100% in all runs, we analyze the interaction time depending on the runs of the experiment. For this purpose, we consider the time that a participant needs to specify all three restriction areas in a run of the experiment. Thus, a time measurement comprises the specification of three virtual borders. The results are visualized in Figure 5.7b. The plot reveals a continuous improvement of the interaction time starting from a mean time of $M = 31$ seconds for the first run, $M = 24$ seconds for the second run and finally $M = 19$ seconds for the last run.

Discussion

The results of the interaction time reveal a significant reduction when employing the LSS compared to the AR approach. The LSS achieves a constant interaction time, which is independent of the length, shape or size of a virtual border. Thus, the experimental results support Hypothesis 1. This time reduction is caused by the novel workflow of the LSS that allows a human to simply select a recommendation for a virtual border instead of explicitly specifying all components of a virtual border (as it is the case for the AR method).

Regarding the completeness, the experimental results confirm a good completeness of the LSS, which is as good as the completeness of the AR interaction method. This is expectable since the AR approach already achieved a good quality level and the LSS even simplifies the interaction by

recommending virtual borders. Thus, there is even less potential for incorrect interactions. This leads to the acceptance of Hypothesis 2.

In case of the learnability, the participants' ratings are similar for both interaction methods. Moreover, we observed a continuous improvement of the interaction time when participants performed multiple runs of the experiment. Hence, the LSS reaches a good learnability and does not have a negative effect on the learnability of the AR approach. Therefore, we conclude that Hypothesis 3 is supported by the results of this experiment.

Finally, the LSS is rated worse than the AR approach when considering the user experience aspects. While the comfort/effort is on the same good level as the AR method, the LSS features worse ratings on the aspects of problems, intuitiveness and feedback. In case of the aspects of problems and intuitiveness, we identify reasons for this degradation in the free responses of the participants. There are four participants who wish to select the area instead of the boundary of a virtual border to select a recommendation. Thus, this could have negatively affected both aspects. However, in case of the feedback, the difference is hard to explain since both interaction methods feature the same feedback system. A reason could be that the data of the AR approach are taken from an experiment where participants also compared the method with other interaction methods. This could lead to higher ratings if the methods differ significantly. This comparison was not possible for participants evaluating the LSS. Nonetheless, the ratings for the LSS are still on a good quality level on average. In summary, these results partly support Hypothesis 4 because the LSS is worse rated than the AR approach but still on the same good quality level.

5.3 Summary

In this chapter, we dealt with Research Question 3, i.e. how can a network robot system learn from user interactions and apply the knowledge in future interaction processes. To this end, we proposed a LSS that learns from multiple user interactions and supports a human through appropriate recommendations for interactions. To encode an interaction process for machine learning algorithms, the LSS encodes an interaction process through a semantic of a restriction area. This is extracted using scene understanding, in particular semantic segmentation, performed on images acquired from the smart camera network introduced in the previous chapter. The identification of frequent semantics is accomplished using frequent itemset mining. Based on these frequent semantics, the LSS creates recommendations for virtual borders with identical semantics, which are conveyed to the human employing an AR device. This enables a human to simply select a recommendation instead of explicitly specifying all virtual border components. This should reduce the interaction time by avoiding the linear interaction time of the other interaction methods without learning capabilities. To test this, we first validated the proposed LSS concerning its recognition rate and accuracy. This was necessary to show that restriction areas can be robustly recognized by the system and

that the recommendations feature an equally high accuracy as the standard AR approach (without learning capabilities). Since the experimental results supported both hypotheses, we conducted a second experiment that dealt with the other user requirements. These results revealed that the LSS can reduce the interaction time to a constant level while preserving the performance of the other user requirements. Thus, we conclude that Objective 3 of this thesis could be achieved with the development of the LSS, which leads to a further improvement of the state of the art.

The results of this chapter in comparison to the other interaction methods are summarized in Table 5.5. Although the LSS features a constant interaction time, this comes with two assumptions: (1) The approach is limited to restriction areas whose semantic can be visually derived from the appearance, e.g. carpets or kids' corners. It is not possible to create recommendations for restriction areas where a semantic cannot be derived from the visual appearance, e.g. privacy zones without certain visual characteristics. (2) Moreover, the restriction areas need to be covered by the fields of view of the smart cameras in the environment. This is necessary to identify the semantic of a user-defined restriction area and to identify potential restriction areas for recommendations. Thus, our experiments were designed according to these assumptions to show the potential of the LSS. However, in real-life scenarios, it will be typically a mix of the standard and novel workflow resulting in a constant time in the best case and a linear time in the worst case. Nonetheless, the interaction time will be on a good quality level.

Table 5.5: Summary of the performance of the learning and support system regarding the user requirements. The symbols indicate an unacceptable ($-$), acceptable ($\circ$) and good ($+$) quality level. The $\oplus$ is used for an acceptable quality level if there is no good quality level defined for a certain requirement. The $++$ is used to indicate the constant interaction time of the learning and support system. Arrows indicate the change with respect to the AR method.

Method	Correctness	Flexibility	Completeness	Accuracy	Time	User exp.	Learnability[10]
GUI	$\oplus$	$\oplus$	$\circ$	$\circ$	$+$	$\circ$	$\circ$
Pointer	$\oplus$	$\oplus$	$\circ$	$+$	$-$	$\circ$	$\circ$
AR	$\oplus$	$\oplus$	$+$	$+$	$+$	$+$	$\circ$
NRS	$\oplus$	$\oplus$	$\circ$	$+$	$\circ$	$+$	$\circ$
LSS	$\oplus$ ($\rightarrow$)	$\oplus$ ($\rightarrow$)	$+$ ($\rightarrow$)	$+$ ($\rightarrow$)	$++$ ($\nearrow$)	$+$ ($\rightarrow$)	$+$ ($\nearrow$)

[10] The LSS features a good learnability and the other interaction methods an acceptable learnability. However, we have not evaluated the other interaction methods concerning a good learnability. Thus, it is unfair to compare the LSS with the other interaction methods concerning this requirement.

6

Concluding Remarks

6.1 Conclusions

We addressed the problem of the interactive restriction of a mobile robot's workspace in traditional and smart home environments. This problem is especially relevant for non-expert humans living in human-robot shared spaces, e.g. home environments with a mobile service robot. A solution to this problem requires an interaction process between human and mobile robot, in which spatial information about a restriction area is transferred from human to robot and feedback about the interaction process is provided from robot to human. This is challenging due to the complexity of spatial information and the limited mobile robot's interaction capabilities. Moreover, ambitious user requirements make the problem more challenging. Existing solutions to this problem do not optimally fulfill these user requirements, e.g. concerning flexibility, accuracy or user experience.

Objective 1 Therefore, the first objective of this work was to develop alternative interaction methods for a traditional home environment, that perform better than the current state-of-the-art solution, i.e. sketching restriction areas on an occupancy grid map (OGM) displayed on a graphical user interface (GUI), and that constitute an acceptable solution for our problem. To this end, we proposed two alternative interaction methods based on a laser pointer and an augmented reality (AR) user interface, which were identified as promising alternatives in a literature review. The first one is a robot-dependent interaction method, that requires the active participation of the mobile robot in the interaction process, while the latter one is a robot-independent interaction method where the user interface is responsible for the whole interaction (without the need for a mobile robot's participation in the interaction process). Both interaction methods employ virtual borders, that are incorporated into a human-aware navigation framework, to flexibly model restriction areas and change the mobile robot's navigational behavior. Experimental evaluations revealed that the AR approach based on a RGB-D tablet achieves good quality levels on almost all user requirements. Thus, this interaction method outperforms the state-of-the-art solution, which features mainly an acceptable quality level on most of the user requirements. Hence, Objective 1 of this thesis could be

achieved. The other alternative interaction method based on a laser pointer features a better accuracy than the baseline, but it does not outperform the state-of-the-art solution, in particular a linear interaction time leads to an unacceptable solution. Moreover, the restricted mobile robot's interaction capabilities do not allow an improvement of the user experience with respect to the baseline. Nonetheless, both proposed interaction methods were preferred to the baseline approach by the participants of our experiment.

Objective 2 To address these limitations of the laser pointer method, Objective 2 dealt with the investigation of the role of a smart home environment in the interaction process to improve the interaction time to an acceptable and user experience to a good quality level. Therefore, we proposed another interaction method which incorporates components of a smart home environment in the interaction process. Thus, a human does no more directly interact with the mobile robot but rather interacts with a network robot system (NRS) consisting of mobile robot and smart environment. This idea seemed especially promising in case of robot-dependent interaction methods due to additional sensors and actuators provided by a smart environment to extend the perceptual and interaction capabilities of the mobile robot. For this purpose, we incorporated a smart camera network, a smart speaker and a smart display into the interaction process. In addition to the selection of appropriate smart home components and the specification of the human-robot-environment interaction, the cooperative perception of multiple stationary and mobile cameras implied several challenges. These were adequately addressed by a multi-stage virtual border extraction algorithm, which extracts a virtual border from multiple camera observations. The proposed interaction method was evaluated in an experiment that compared the NRS with the laser pointer approach (without smart home support). The experimental results indicated that the NRS method features a significantly shorter interaction time and a better user experience while maintaining the quality levels of the other user requirements. Hence, the incorporation of a smart home into the interaction process turned the initial laser pointer approach from an unacceptable to an acceptable solution for our problem. Therefore, we conclude that Objective 2 of this thesis could be achieved. Furthermore, this solution performed better in the overall evaluation than the baseline method employing a GUI. However, the quality level of the AR interaction method was not achieved, especially in terms of completeness and interaction time.

Objective 3 Finally, the last objective of this thesis dealt with the investigation of learning capabilities with the goal to reduce the interaction time. The main idea was to learn from multiple user interactions and apply the knowledge in future interaction processes to support the human. To this end, we proposed a learning and support system (LSS), which encodes an interaction process employing semantic scene understanding and learns from multiple interaction processes through frequent itemset mining. The extracted knowledge is then conveyed to the human through appropriate recommendations for interactions using AR. Thus, a human does not have to explicitly

specify all components of a virtual border, but can rather select a recommendation of the LSS. After validating the LSS concerning its recognition rate and accuracy, experimental results showed that this approach reduces the interaction time to a constant good level while preserving the quality levels of the other user requirements. Thus, we conclude that the state of the art could be further improved and that Objective 3 of this thesis could be achieved. However, it is noted that the LSS is limited to restriction areas whose semantic can be visually derived from the appearance. Hence, the camera network, which is the basis for semantic scene understanding, has to cover these areas.

6.2 Limitations

Although the objectives of this thesis could be achieved, this work also has some limitations. The main limitation deals with the evaluation of the proposed solutions. The experimental evaluations were thoroughly conducted with multiple participants and multiple evaluation scenarios. The experimental environments were best possibly set up according to real home environments and the participants' characteristics approximately corresponded to the intended user group of this thesis. However, these experiments were performed in a lab environment under lab conditions, i.e. there was an artificial and controlled environment. Thus, there was no evaluation in the field in *real* home environments, which could support our conclusions.

Another point, that is not considered throughout this work, are the financial costs of a smart home. This entails the question of how much does the added value of a smart home in the interaction process cost and if it is worth to upgrade a traditional to a smart home environment. Such costs mainly consist of acquisition and installation costs. Since we presented acceptable solutions for both types of environments, a human is always able to interactively restrict the workspace of a mobile robot with an at least acceptable performance. In addition, the performance of the AR interaction method even achieves a good performance in most of the requirements, which makes it better than the current the state of the art. Therefore, an upgrade to a smart home environment only makes sense if a human wants to employ the laser pointer interface (as there is a significant improvement in interaction time and user experience) or if the interaction time should be further reduced to a constant level (enabled by learning capabilities). Hence, while an upgrade does not always make sense, the incorporation of smart home components in the interaction process is advantageous if the environment already includes smart components, e.g a new smart building. In this case, a human can take advantage of an improved performance without additional costs of an upgrade.

6.3 Future Work

Motivated by the limitations, a first work for the future could be the extension of the experimental evaluation. This could be a long-term study in the field comprising multiple real (smart) home

environments with residents. In this case, the environment would not be artificial and controlled but would rather constitute an everyday-life environment. Similar results of such a study could support our conclusions. Moreover, when considering a larger number of participants in a study, the evaluation of the interaction methods depending on age groups could bring further insights.

In addition to the extension of the experimental evaluation, the functionality of the interaction methods could be extended in the future. For example, to specify temporary restriction areas, such as dirty areas, a human currently has to add and later delete a virtual border, which results in two interaction processes. An extension could model this time constraint as an additional component of a virtual border, which could be then specified in an additional state of an interaction method, e.g. the AR application could provide an additional input to specify a time constraint. Thus, a virtual border could be automatically deleted when a user-defined time expires. This would facilitate the definition of temporary restriction areas.

Another work for the future arises from the evaluation of the participants' free responses in the experiments. For example, a participant wished a stronger feedback channel of the smart home environment. In this case, projectors integrated in the environment could be a promising solution. For example, LEUTERT *et al.* (2013) use a stationary projector mounted on the ceiling and a mobile projector on a robotic arm to visualize robotic data, and GANESAN *et al.* (2018) use a projector to visualize cues in a human-robot collaboration scenario. However, these hardware devices are currently not widespread in smart homes, but progress in the deployment of smart environments could mitigate this limitation in the future. An alternative would be the installation of a projector on the mobile robot to give feedback to a human (CHADALAVADA *et al.*, 2015)(SHRESTHA *et al.*, 2018). This would not only be applicable in smart homes but also in traditional home environments.

Furthermore, we evaluated our NRS interaction method with a single mobile robot, which is valid for most households and the scope of this thesis. However, it would be interesting to incorporate multiple robots into the interaction process to increase the number of mobile cameras. Our proposed architecture already rudimentarily considers this aspect as depicted in Figure 4.3, but more work on the cooperation between multiple mobile robots in such a scenario is needed to address additional challenges arising from this setting.

Regarding the LSS, the user interaction focuses on the selection of recommendations by a human. Although the recognition rate is high (F-score of 91.5%) and most of the recommendations (89.8% of the recognized restriction areas) feature an at least acceptable accuracy and should thus be accepted by a human, future work could deal with the improvement of these values. For example, false positive classifications FP could be reduced by considering additional depth data of a camera. This could avoid recommendations of objects in a scene, that are not part of a restriction area, e.g. dishes on a table or boxes on a shelf. Besides, progress in the field of semantic segmentation could further increase the accuracy and thus the percentage of recommendations with an acceptable and good quality level.

Moreover, the functionality of the LSS could be extended. Currently, the semantic scene understanding is performed on images acquired from cameras integrated in the environment. Due to the partial camera observation of the environment, the LSS is restricted to the observed areas. To address this issue, future works could consider the AR device or the mobile robot as additional sensor for scene understanding. This requires the re-training of a new deep neural network for semantic segmentation from different viewpoints and the optimization of the algorithms for a hardware-limited device, such as a tablet without graphics processing unit (GPU). Additionally, the LSS could be extended by a "track-and-adapt"-behavior. This could allow the system to track virtual borders and automatically adapt their locations in the OGM of the environment if they are moved.

Apart from these suggestions for future work, the commercialization of the contributions of this work would be a next step. Currently, the interaction methods are implemented as prototypical systems, but the realization of a commercial product would take some additional steps. For example, it would be necessary to develop an automatic registration method in case of the AR approach to circumvent the requirement concerning a manual registration between the *Map* and *ADF* coordinate frames as described in Subsection 3.3.2. Besides, the AR method requires special hardware, i.e. a RGB-D tablet or smartphone, limiting the potential number of users. Nonetheless, major companies started to release AR toolkits (ARCore by Google[1] and ARKit by Apple[2]) in the last years, that work without specialized hardware. Thus, the AR method could be widely deployed on common smartphones and tablets without additional costs for the user.

Finally, the scope of this thesis could be extended in future works, especially in case of the environment. While this thesis dealt with traditional and smart home environments, the transfer of the interaction methods to outdoor environments could be of interest. In particular, the example of a robotic lawnmower operating in a garden is an insufficiently solved problem because wires need to be buried in the garden to restrict the robot's workspace. This is inflexible and involves additional costs and installation effort. Such an outdoor environment entails new challenges dealing with uneven terrain, localization issues and uncontrolled light conditions, which could affect the performance of the interaction methods.

[1] `https://developers.google.com/ar` [Accessed: 26.03.2020]
[2] `https://developer.apple.com/augmented-reality` [Accessed: 26.03.2020]

A

List of Figures

B

List of Tables

C

List of Algorithms

D

List of Abbreviations

ANOVA	ANalysis Of VAriance
AR	Augmented Reality
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HRI	Human-Robot Interaction
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IMU	Inertial Measurement Unit
JSI	Jaccard Similarity Index
LSS	Learning and Support System
NRS	Network Robot System
OGM	Occupancy Grid Map
PASCAL VOC	PASCAL Visual Object Classes
PEIS	Physically Embedded Intelligent Systems
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle
VO	Visual Odometry
VSLAM	Visual Simultaneous Localization and Mapping

E

Glossary

Interaction method An interaction method describes the way of how to employ a user interface in an interaction process to achieve a goal, e.g. the restriction of a mobile robot's workspace. . . .

Interaction process In the context of this thesis, an interaction process describes the interaction between human and mobile robot or network robot system. . . . . .

Mobile robot A mobile robot is a robot with a locomotion system, i.e. it is able to move in the environment using its actuators. . . . . .

Navigational behavior The navigational behavior describes the way how a mobile robot moves from a starting to a goal pose. Thus, it is a synonym for the path between both poses. . . . . .

Restriction area A restriction area is an area that is excluded from a mobile robot's workspace. Thus, a mobile robot does not enter this area. . . . . .

User interaction A user interaction refers to the execution of an interaction process. . . . . .

User interface A user interface gives an opportunity for interaction between a human and robot, e.g. a tablet or laser pointer. . . . . .

Virtual border A virtual border is used to model a restriction area in terms of virtual border points, a seed point and an occupancy probability. . . . . .

Workspace The workspace of a mobile robot is the space that can be reached by the robot using its locomotion and path planning system. . . . . .

F

References

ACKERMAN, E. (2013). TurtleBot inventors tell us everything about the robot. *IEEE Spectrum* (26.03.2013).

ACKERMAN, E. (2017). Neato adds persistent, actionable maps to new D7 robot vacuum. *IEEE Spectrum* (31.08.2017).

AGRAWAL, R. and R. SRIKANT (1994). Fast algorithms for mining association rules in large databases. In *International Conference on Very Large Data Bases (VLDB)*, pp. 487–499.

AHN, H., Y. OH, S. CHOI, C. J. TOMLIN, and S. OH (2018). Online learning to approach a person with no regret. *IEEE Robotics and Automation Letters 3*(1), 52–59.

AKGUN, B., M. CAKMAK, J. W. YOO, and A. L. THOMAZ (2012). Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 391–398.

ALBEN, L. (1996). Quality of experience: Defining the criteria for effective interaction design. *Interactions 3*(3), 11–15.

ALDRICH, F. K. (2003). Smart homes: Past, present and future. In R. Harper (Ed.), *Inside the Smart Home*, pp. 17–39. Springer London.

ALEMPIJEVIC, A., R. FITCH, and N. KIRCHNER (2013). Bootstrapping navigation and path planning using human positional traces. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1242–1247.

ALTHAUS, P., H. ISHIGURO, T. KANDA, T. MIYASHITA, and H. I. CHRISTENSEN (2004). Navigation for human-robot interaction tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1894–1900.

APTHORPE, N., Y. SHVARTZSHNAIDER, A. MATHUR, D. REISMAN, and N. FEAMSTER (2018). Discovering smart home internet of things privacy norms using contextual integrity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2*(2), 59:1–59:23.

ASSAD, C., M. WOLF, T. THEODORIDIS, K. GLETTE, and A. STOICA (2013). BioSleeve: A natural EMG-based interface for HRI. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 69–70.

AUGUSTO, J. C. (2007). Ambient intelligence: The confluence of ubiquitous/pervasive computing and artificial intelligence. In A. J. Schuster (Ed.), *Intelligent Computing Everywhere*, pp. 213–234. Springer London.

AUGUSTO, J. C., V. CALLAGHAN, D. COOK, A. KAMEAS, and I. SATOH (2013). Intelligent environments: a manifesto. *Human-centric Computing and Information Sciences 3*(12), 1–18.

AUGUSTO, J. C., H. NAKASHIMA, and H. AGHAJAN (2010). Ambient intelligence and smart environments: A state of the art. In H. Nakashima, H. Aghajan, and J. C. Augusto (Eds.), *Handbook of Ambient Intelligence and Smart Environments*, pp. 3–31. Springer US.

AUGUSTO, J. C. and C. D. NUGENT (2006). Smart homes can be smarter. In J. C. Augusto and C. D. Nugent (Eds.), *Designing Smart Homes: The Role of Artificial Intelligence*, pp. 1–15. Springer Berlin Heidelberg.

BAILEY, T. and H. DURRANT-WHYTE (2006). Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine 13*(3), 108–117.

BARAKA, K., S. ROSENTHAL, and M. VELOSO (2016). Enhancing human understanding of a mobile robot's state and actions using expressive lights. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 652–657.

BARAKA, K. and M. M. VELOSO (2018). Mobile service robot state revealing through expressive lights: Formalism, design, and evaluation. *International Journal of Social Robotics 10*(1), 65–92.

BEKEY, G. (2005). Localization, navigation, and mapping. In *Autonomous Robots: From Biological Inspiration to Implementation and Control*, pp. 473–508. The MIT Press.

BISWAS, J. and M. VELOSO (2016). The 1,000-km challenge: Insights and quantitative and qualitative results. *IEEE Intelligent Systems 31*(3), 86–96.

BLOCHLIGER, F., M. FEHR, M. DYMCZYK, T. SCHNEIDER, and R. SIEGWART (2018). Topomap: Topological mapping and navigation based on visual SLAM maps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9.

BOHREN, J., R. B. RUSU, E. GIL JONES, E. MARDER-EPPSTEIN, C. PANTOFARU, M. WISE, L. MÖSENLECHNER, W. MEEUSSEN, and S. HOLZER (2011). Towards autonomous robotic butlers: Lessons learned with the PR2. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5568–5575.

BORENSTEIN, J. and Y. KOREN (1991). The vector field histogram – Fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation 7*(3), 278–288.

BOUCHARD, K., D. FORTIN-SIMARD, J. LAPALU, S. GABOURY, A. BOUZOUANE, and B. BOUCHARD (2015). Unsupervised spatial data mining for smart homes. In *IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1433–1440.

BOWYER, S. A., B. L. DAVIES, and F. RODRIGUEZ Y BAENA (2014). Active constraints/virtual fixtures: A survey. *IEEE Transactions on Robotics 30*(1), 138–157.

BRDICZKA, O., J. L. CROWLEY, and P. REIGNIER (2009). Learning situation models in a smart home. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39*(1), 56–63.

BRDICZKA, O., M. LANGET, J. MAISONNASSE, and J. L. CROWLEY (2009). Detecting human behavior models from multimodal observation in a smart home. *IEEE Transactions on Automation Science and Engineering 6*(4), 588–597.

BUNIYAMIN, N., N. SARIFF, W. A. J. WAN NGAH, and Z. MOHAMAD (2011). Robot global path planning overview and a variation of ant colony system algorithm. *International Journal of Mathematics and Computers in Simulation 5*(1), 9–16.

BUSCHKA, P. and A. SAFFIOTTI (2004). Some notes on the use of hybrid maps for mobile robots. In *International Conference on Intelligent Autonomous Systems (IAS)*, pp. 547–556.

BUTZ, A. (2010). User interfaces and HCI for ambient intelligence and smart environments. In H. Nakashima, H. Aghajan, and J. C. Augusto (Eds.), *Handbook of Ambient Intelligence and Smart Environments*, pp. 535–558. Springer US.

CADENA, C., L. CARLONE, H. CARRILLO, Y. LATIF, D. SCARAMUZZA, J. NEIRA, I. REID, and J. J. LEONARD (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics 32*(6), 1309–1332.

CASQUEIRO, A., D. RUIVO, A. MOUTINHO, and J. MARTINS (2016). Improving teleoperation with vibration force feedback and anti-collision methods. In *Robot 2015: Second Iberian Robotics Conference*, pp. 269–281.

CHA, E. and M. MATARIĆ (2016). Using nonverbal signals to request help during human-robot collaboration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5070–5076.

CHA, E., T. TREHON, L. WATHIEU, C. WAGNER, A. SHUKLA, and M. J. MATARIĆ (2017). ModLight: Designing a modular light signaling tool for human-robot interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1654–1661.

CHADALAVADA, R. T., H. ANDREASSON, R. KRUG, and A. J. LILIENTHAL (2015). That's on my mind! Robot to human intention communication through on-board projection on shared floor space. In *European Conference on Mobile Robots (ECMR)*, pp. 1–6.

CHAN, A. H. and A. W. NG (2009). Perceptions of implied hazard for visual and auditory alerting signals. *Safety Science 47*(3), 346 – 352.

CHARALAMPOUS, K., I. KOSTAVELIS, and A. GASTERATOS (2017). Recent trends in social aware robot navigation. *Robotics and Autonomous Systems 93*(1), 85–104.

CHAUMETTE, F., S. HUTCHINSON, and P. CORKE (2016). Visual servoing. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 841–866. Springer International Publishing.

CHAVEZ, F., F. FERNANDEZ, M. J. GACTO, and R. ALCALA (2012). Automatic laser pointer detection algorithm for environment control device systems based on template matching and genetic tuning of fuzzy rule-based systems. *International Journal of Computational Intelligence Systems 5*(2), 368–386.

CHEN, D. Z., R. J. SZCZERBA, and J. J. UHRAN (1997). A framed-quadtree approach for determining euclidean shortest paths in a 2-d environment. *IEEE Transactions on Robotics and Automation 13*(5), 668–681.

CHEN, T. L. and C. C. KEMP (2010). Lead me by the hand: Evaluation of a direct physical interface for nursing assistant robots. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 367–374.

CHIK, S. F., C. F. YEONG, E. L. M. SU, T. LIM, Y. SUBRAMANIAM, and P. J. H. CHIN (2016). A review of social-aware navigation frameworks for service robot in dynamic human environments. *Journal of Telecommunication, Electronic and Computer Engineering 8*(11), 41–50.

CHIU, T.-Y. (2011). Virtual wall system for a mobile robotic device. European Patent Application EP2388673A1.

CHOI, Y. S., C. D. ANDERSON, J. D. GLASS, and C. C. KEMP (2008). Laser pointers and a touch screen: Intuitive interfaces for autonomous mobile manipulation for the motor impaired. In *International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, pp. 225–232.

CHONG, K. S. and L. KLEEMAN (1999). Feature-based mapping in real, large scale environments using an ultrasonic array. *The International Journal of Robotics Research 18*(1), 3–19.

CHUNG, W. and K. IAGNEMMA (2016). Wheeled robots. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 575–594. Springer International Publishing.

CHUNG, W., S. KIM, M. CHOI, J. CHOI, H. KIM, C. MOON, and J. SONG (2009). Safe navigation of a mobile robot considering visibility of environment. *IEEE Transactions on Industrial Electronics 56*(10), 3941–3950.

CHUY, O., Y. HIRATA, and K. KOSUGE (2006). A new control approach for a robotic walking support system in adapting user characteristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 36*(6), 725–733.

COLLINS, E. C., T. J. PRESCOTT, and B. MITCHINSON (2015). Saying it with light: A pilot study of affective communication using the MIRO robot. In *Living Machines: International Conference on Biomimetic and Biohybrid Systems*, pp. 243–255.

COOK, D. J. (2012). Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems 27*(1), 32–38.

COOK, D. J. and S. K. DAS (2005). *Smart Environments: Technology, Protocols and Applications*. John Wiley & Sons.

CORKE, P. (2017). Mobile robot vehicles. In *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, pp. 99–124. Springer International Publishing.

COSGUN, A. and H. I. CHRISTENSEN (2018). Context-aware robot navigation using interactively built semantic maps. *Paladyn, Journal of Behavioral Robotics 9*(1), 254–276.

DAS, S. K., D. J. COOK, A. BATTACHARYA, E. O. HEIERMAN, and TZE-YUN LIN (2002). The role of prediction algorithms in the MavHome smart home architecture. *IEEE Wireless Communications 9*(6), 77–84.

DAVISON, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1403–1410.

DEN BERGH, M. V., D. CARTON, R. D. NIJS, N. MITSOU, C. LANDSIEDEL, K. KUEHNLENZ, D. WOLLHERR, L. V. GOOL, and M. BUSS (2011). Real-time 3d hand gesture interaction with a robot for understanding directions from humans. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 357–362.

DESAI, K., Y. LIU, and G. LIU (2012). A graphical user interface for tele-operated robotic sample acquisition. In *IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1202–1207.

DOCTOR, F., H. HAGRAS, and V. CALLAGHAN (2005). A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A (Systems and Humans) 35*(1), 55–65.

DROESCHEL, D., J. STÜCKLER, D. HOLZ, and S. BEHNKE (2011). Towards joint attention for a domestic service robot - person awareness and gesture recognition using time-of-flight cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1205–1210.

DUDA, R. O., P. E. HART, and D. G. STORK (2000). *Pattern Classification*. John Wiley & Sons.

DURRANT-WHYTE, H. and T. BAILEY (2006). Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine 13*(2), 99–110.

ELSDON, J. and Y. DEMIRIS (2018). Augmented reality for feedback in a shared control spraying task. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1939–1946.

ESTER, M., H.-P. KRIEGEL, J. SANDER, and X. XU (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231.

EVERINGHAM, M., L. VAN GOOL, C. K. I. WILLIAMS, J. WINN, and A. ZISSERMAN (2010). The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision 88*(2), 303–338.

FERRER, G., A. G. ZULUETA, F. H. COTARELO, and A. SANFELIU (2017). Robot social-aware navigation framework to accompany people walking side-by-side. *Autonomous Robots 41*(4), 775–793.

FLECK, S. and W. STRASSER (2008). Smart camera based monitoring system and its application to assisted living. *Proceedings of the IEEE 96*(10), 1698–1714.

FOURNIER-VIGER, P., J. C.-W. LIN, B. VO, T. T. CHI, J. ZHANG, and H. B. LE (2017). A survey of itemset mining. *WIREs Data Mining and Knowledge Discovery 7*(4), 1–18.

FOX, D. (2003). Adapting the sample size in particle filters through KLD-sampling. *The International Journal of Robotics Research 22*(12), 985–1003.

FOX, D., W. BURGARD, and S. THRUN (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine 4*(1), 23–33.

FRANK, J. A., M. MOORHEAD, and V. KAPILA (2016). Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 302–307.

FRAUNDORFER, F. and D. SCARAMUZZA (2012). Visual odometry: Part II - matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine 19*(2), 78–90.

FUENTES-PACHECO, J., J. RUIZ-ASCENCIO, and J. M. RENDÓN-MANCHA (2015). Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review 43*(1), 55–81.

GANESAN, R. K., Y. K. RATHORE, H. M. ROSS, and H. BEN AMOR (2018). Better teaming through visual cues: How projecting imagery in a workspace can improve human-robot collaboration. *IEEE Robotics & Automation Magazine 25*(2), 59–71.

GARCIA-FIDALGO, E. and A. ORTIZ (2015). Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems 64*(1), 1–20.

GARCIA-GARCIA, A., S. ORTS-ESCOLANO, S. OPREA, V. VILLENA-MARTINEZ, P. MARTINEZ-GONZALEZ, and J. GARCIA-RODRIGUEZ (2018). A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing 70*(Sept), 41–65.

GOCKLEY, R., J. FORLIZZI, and R. SIMMONS (2007). Natural person-following behavior for social robots. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 17–24.

GOMEZ-DONOSO, F., F. ESCALONA, F. M. RIVAS, J. M. CAÑAS, and M. CAZORLA (2019). Enhancing the ambient assisted living capabilities with a mobile robot. *Computational Intelligence and Neuroscience 2019*(9412384), 1–15.

GOODRICH, M. A. and A. C. SCHULTZ (2007). Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction 1*(3), 203–275.

GRANATA, C., M. CHETOUANI, A. TAPUS, P. BIDAUD, and V. DUPOURQUÉ (2010). Voice and graphical -based interfaces for interaction with a robot dedicated to elderly and people with cognitive disorders. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 785–790.

GRISETTI, G., C. STACHNISS, and W. BURGARD (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics 23*(1), 34–46.

GROMOV, B., G. ABBATE, L. GAMBARDELLA, and A. GIUSTI (2019). Proximity human-robot interaction using pointing gestures and a wrist-mounted IMU. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8084–8091.

GROMOV, B., L. M. GAMBARDELLA, and A. GIUSTI (2018). Robot identification and localization with pointing gestures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3921–3928.

GROSS, H. M., S. MUELLER, C. SCHROETER, M. VOLKHARDT, A. SCHEIDIG, K. DEBES, K. RICHTER, and N. DOERING (2015). Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5992–5999.

GROSS, H. M., C. SCHROETER, S. MUELLER, M. VOLKHARDT, E. EINHORN, A. BLEY, C. MARTIN, T. LANGNER, and M. MERTEN (2011). Progress in developing a socially assistive mobile home robot companion for the elderly with mild cognitive impairment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2430–2437.

HÄGELE, M. (2016). Robots conquer the world [turning point]. *IEEE Robotics & Automation Magazine 23*(1), 120–118.

HÄGELE, M., K. NILSSON, J. N. PIRES, and R. BISCHOFF (2016). Industrial robotics. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 1385–1422. Springer International Publishing.

HALL, E. (1966). *The Hidden Dimension*. Doubleday.

HALMETSCHLAGER-FUNEK, G., M. SUCHI, M. KAMPEL, and M. VINCZE (2019). An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments. *IEEE Robotics & Automation Magazine 26*(1), 67–77.

HAN, J., H. CHENG, D. XIN, and X. YAN (2007). Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery 15*(1), 55–86.

HAN, J., J. PEI, and Y. YIN (2000). Mining frequent patterns without candidate generation. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 1–12.

HARPE, S. E. (2015). How to analyze Likert and other rating scale data. *Currents in Pharmacy Teaching and Learning 7*(6), 836–850.

HARRISON, C., J. HORSTMAN, G. HSIEH, and S. HUDSON (2012). Unlocking the expressivity of point lights. In *SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1683–1692.

HARTLEY, R. and A. ZISSERMAN (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

HAWES, N., C. BURBRIDGE, F. JOVAN, L. KUNZE, B. LACERDA, L. MUDROVA, J. YOUNG, J. WYATT, D. HEBESBERGER, T. KORTNER, R. AMBRUS, N. BORE, J. FOLKESSON, P. JENSFELT, L. BEYER, A. HERMANS, B. LEIBE, A. ALDOMA, T. FAULHAMMER, M. ZILLICH, M. VINCZE, E. CHINELLATO, M. AL-OMARI, P. DUCKWORTH, Y. GATSOULIS, D. C. HOGG, A. G. COHN, C. DONDRUP, J. P. FENTANES, T. KRAJNIK, J. M. SANTOS, T. DUCKETT, and M. HANHEIDE (2017). The STRANDS project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine 24*(3), 146–156.

HE, K., X. ZHANG, S. REN, and J. SUN (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

HEBERT, P., J. MA, J. BORDERS, A. AYDEMIR, M. BAJRACHARYA, N. HUDSON, K. SHANKAR, S. KARUMANCHI, B. DOUILLARD, and J. BURDICK (2015). Supervised remote robot with guided autonomy and teleoperation (SURROGATE): A framework for whole-body manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5509–5516.

HEDAYATI, H., M. WALKER, and D. SZAFIR (2018). Improving collocated robot teleoperation with augmented reality. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 78–86.

HELAL, S., W. MANN, H. EL-ZABADANI, J. KING, Y. KADDOURA, and E. JANSEN (2005). The Gator Tech Smart House: a programmable pervasive space. *Computer 38*(3), 50–60.

HO, C., N. REED, and C. SPENCE (2007). Multisensory in-car warning signals for collision avoidance. *Human Factors 49*(6), 1107–1114.

HORIKAWA, Y., A. EGASHIRA, K. NAKASHIMA, A. KAWAMURA, and R. KURAZUME (2017). Previewed reality: Near-future perception system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 370–375.

ISHII, K., S. ZHAO, M. INAMI, T. IGARASHI, and M. IMAI (2009). Designing laser gesture interface for robot control. In *IFIP Conference on Human-Computer Interaction (INTERACT)*, pp. 479–492.

ISO (2010). ISO 13855:2010. Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body.

ISO (2018). ISO 9241-11:2018. Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts.

JAKKULA, V. R., A. S. CRANDALL, and D. J. COOK (2009). Enhancing anomaly detection using temporal pattern discovery. In A. D. Kameas, V. Callagan, H. Hagras, M. Weber, and W. Minker (Eds.), *Advanced Intelligent Environments*, pp. 175–194. Springer US.

JEVTIC, A., G. DOISY, Y. PARMET, and Y. EDAN (2015). Comparison of interaction modalities for mobile indoor robot guidance: Direct physical interaction, person following, and pointing control. *IEEE Transactions on Human-Machine Systems 45*(6), 653–663.

JONES, J. L. (2006). Robots at the tipping point: The road to iRobot Roomba. *IEEE Robotics & Automation Magazine 13*(1), 76–78.

KANG, K., X. LIN, C. LI, J. HU, B. HENGEVELD, C. HUMMELS, and G. RAUTERBERG (2018). Designing interactive public displays in caring environments: A case study of OutLook. *Journal of Ambient Intelligence and Smart Environments 10*(6), 427–443.

KAPADIA, R., S. STASZAK, L. JIAN, and K. GOLDBERG (2017). EchoBot: Facilitating data collection for robot learning with the Amazon echo. In *IEEE Conference on Automation Science and Engineering (CASE)*, pp. 159–165.

KEMP, C. C., C. D. ANDERSON, H. NGUYEN, A. J. TREVOR, and Z. XU (2008). A point-and-click interface for the real world: Laser designation of objects for mobile manipulation. In *ACM/IEEE International Conference on Human Robot Interaction (HRI)*, pp. 241–248.

KIM, B. M., S. S. KWAK, and M. S. KIM (2009). Design guideline of anthropomorphic sound feedback for service robot malfunction - with emphasis on the vacuum cleaning robot. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 352–357.

KIM, H.-J., K.-H. JEONG, S.-K. KIM, and T.-D. HAN (2011). Ambient wall: Smart wall display interface which can be controlled by simple gesture for smart home. In *SIGGRAPH Asia 2011 Sketches*, pp. 1:1–1:2.

KIM, J. H., K. H. LEE, Y. D. KIM, N. S. KUPPUSWAMY, and J. JO (2007). Ubiquitous robot: A new paradigm for integrated services. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2853–2858.

KIRBY, R., R. SIMMONS, and J. FORLIZZI (2009). COMPANION: A constraint-optimizing method for person-acceptable navigation. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 607–612.

KLANCAR, G., A. ZDESAR, S. BLAZIC, and I. SKRJANC (2017). Path planning. In *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*, pp. 161–206. Butterworth-Heinemann.

KORTENKAMP, D. and T. WEYMOUTH (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. In *National Conference on Artificial Intelligence (AAAI)*, pp. 979–984.

KRIZHEVSKY, A., I. SUTSKEVER, and G. E. HINTON (2012). ImageNet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems - Volume 1 (NIPS)*, pp. 1097–1105.

KRUSE, T., A. K. PANDEY, R. ALAMI, and A. KIRSCH (2013). Human-aware robot navigation: A survey. *Robotics and Autonomous Systems 61*(12), 1726–1743.

KUNZE, L., M. BEETZ, M. SAITO, H. AZUMA, K. OKADA, and M. INABA (2012). Searching objects in large-scale indoor environments: A decision-theoretic approach. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4385–4390.

KÖNIG, M. and D. SPRUTE (2019). Verfahren und Robotersystem zur Eingabe eines Arbeitsbereichs. DPMA Patent DE102018125266B3.

KÖNIG, M., D. SPRUTE, and P. VIERTEL (2020). Verfahren und Robotersystem zur Eingabe eines Arbeitsbereichs. DPMA Patent DE102019126903B3.

LAPALU, J., K. BOUCHARD, A. BOUZOUANE, B. BOUCHARD, and S. GIROUX (2013). Unsupervised mining of activities for smart home prediction. *Procedia Computer Science 19*(1), 503 – 510.

LEUTERT, F., C. HERRMANN, and K. SCHILLING (2013). A spatial augmented reality system for intuitive display of robotic data. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 179–180.

LI, R., M. A. OSKOEI, and H. HU (2013). Towards ROS based multi-robot architecture for ambient assisted living. In *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3458–3463.

LIKERT, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology 22*(140), 5–55.

LINDNER, F. (2015). A conceptual model of personal space for human-aware robot activity placement. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5770–5775.

LIPTON, J. I., A. J. FAY, and D. RUS (2018). Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robotics and Automation Letters 3*(1), 179–186.

LIU, H., Y. ZHANG, W. SI, X. XIE, Y. ZHU, and S. ZHU (2018). Interactive robot knowledge patching using augmented reality. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1947–1954.

LU, F. and E. MILIOS (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots 4*(4), 333–349.

LUMELSKY, V. J. and T. SKEWIS (1990). Incorporating range sensing in the robot navigation function. *IEEE Transactions on Systems, Man, and Cybernetics 20*(5), 1058–1069.

LV, X., M. ZHANG, and H. LI (2008). Robot control based on voice command. In *IEEE International Conference on Automation and Logistics (ICAL)*, pp. 2490–2494.

MASTROGIOVANNI, F., A. SGORBISSA, and R. ZACCARIA (2010). From autonomous robots to artificial ecosystems. In H. Nakashima, H. Aghajan, and J. C. Augusto (Eds.), *Handbook of Ambient Intelligence and Smart Environments*, pp. 635–668. Springer US.

MEAD, R. and M. J. MATARIĆ (2016). Perceptual models of human-robot proxemics. In M. A. Hsieh, O. Khatib, and V. Kumar (Eds.), *Experimental Robotics: The 14th International Symposium on Experimental Robotics*, pp. 261–276. Springer International Publishing.

MEAD, R. and M. J. MATARIĆ (2017). Autonomous human–robot proxemics: socially aware navigation based on interaction potential. *Autonomous Robots 41*(5), 1189–1201.

MIKAWA, M., Y. MORIMOTO, and K. TANAKA (2010). Guidance method using laser pointer and gestures for librarian robot. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 373–378.

MILGRAM, P. and F. KISHINO (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems E77-D*(12), 1321–1329.

MILLER, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review 63*(2), 81–97.

MINGUEZ, J., F. LAMIRAUX, and J.-P. LAUMOND (2016). Motion planning and obstacle avoidance. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 1177–1202. Springer International Publishing.

MITRA, S. and T. ACHARYA (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37*(3), 311–324.

MONAJJEMI, M., S. MOHAIMENIANPOUR, and R. VAUGHAN (2016). UAV, come to me: End-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4410–4417.

MONTEMERLO, M., S. THRUN, D. KOLLER, and B. WEGBREIT (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *AAAI National Conference on Artificial Intelligence*, pp. 593–598.

MONTEMERLO, M., S. THRUN, D. KOLLER, and B. WEGBREIT (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1151–1156.

MORAVEC, H. and A. ELFES (1985). High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 116–121.

MOZER, M. C. (1998). The neural network house: An environment that adapts to its inhabitants. In *AAAI Spring Symposium Intelligent Environments*, pp. 110–114.

NAGI, J., A. GIUSTI, L. M. GAMBARDELLA, and G. A. DI CARO (2014). Human-swarm interaction using spatial gestures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3834–3841.

NAKAUCHI, Y. and R. SIMMONS (2002). A social robot that stands in line. *Autonomous Robots 12*(3), 313–324.

NEATO (2017). What are boundary markers and how do i use them? `https://support.neatorobotics.com/hc/en-us/articles/225370067-What-are-boundary-markers-and-how-do-I-use-them-`. [Accessed: 01.12.2017].

NGUYEN, H., A. JAIN, C. ANDERSON, and C. C. KEMP (2008). A clickable world: Behavior selection through pointing and context for mobile manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 787–793.

NICKEL, K. and R. STIEFELHAGEN (2007). Visual recognition of pointing gestures for human-robot interaction. *Image and Vision Computing 25*(12), 1875–1884.

NIELSEN, C. W. and M. A. GOODRICH (2006). Comparing the usefulness of video and map information in navigation tasks. In *ACM SIGCHI/SIGART Conference on Human-Robot Interaction (HRI)*, pp. 95–101.

NISTER, D., O. NARODITSKY, and J. BERGEN (2004). Visual odometry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–659.

NISTER, D., O. NARODITSKY, and J. BERGEN (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics 23*(1), 3–20.

NOR, N. S. M. and M. MIZUKAWA (2014). Robotic services at home: An initialization system based on robots' information and user preferences in unknown environments. *International Journal of Advanced Robotic Systems 11*(7), 112.

O'CALLAGHAN, S. T., S. P. N. SINGH, A. ALEMPIJEVIC, and F. T. RAMOS (2011). Learning navigational maps by observing human motion patterns. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4333–4340.

PACCHIEROTTI, E., H. I. CHRISTENSEN, and P. JENSFELT (2005). Human-robot embodied interaction in hallway settings: a pilot user study. In *IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, pp. 164–171.

PAGES, J., L. MARCHIONNI, and F. FERRO (2016). TIAGo: the modular robot that adapts to different research needs. In *International Workshop on Robot Modularity*, pp. 1–4.

PANDEY, A. K. and R. ALAMI (2010). A framework towards a socially aware mobile robot motion in human-centered dynamic environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5855–5860.

PANDEY, A. K. and R. GELIN (2018). A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine 25*(3), 40–48.

PARK, K.-H., Z. BIEN, J.-J. LEE, B. K. KIM, J.-T. LIM, J.-O. KIM, H. LEE, D. H. STEFANOV, D.-J. KIM, J.-W. JUNG, J.-H. DO, K.-H. SEO, C. H. KIM, W.-G. SONG, and W.-J. LEE (2007). Robotic smart house to assist people with movement disabilities. *Autonomous Robots 22*(2), 183–198.

PAROMTCHIK, I. E. and H. ASAMA (2001). Optical guidance system for multiple mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2935–2940.

POURMEHR, S., V. MONAJJEMI, J. WAWERLA, R. VAUGHAN, and G. MORI (2013). A robust integrated system for selecting and commanding multiple mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2874–2879.

PRASSLER, E., M. E. MUNICH, P. PIRJANIAN, and K. KOSUGE (2016). Domestic robotics. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 1729–1758. Springer International Publishing.

PRATI, A., C. SHAN, and K. I.-K. WANG (2019). Sensors, vision and networks: From video surveillance to activity recognition and health monitoring. *Journal of Ambient Intelligence and Smart Environments 11*(1), 5–22.

PYO, Y., K. NAKASHIMA, S. KUWAHATA, R. KURAZUME, T. TSUJI, K. MOROOKA, and T. HASEGAWA (2015). Service robot system with an informationally structured environment. *Robotics and Autonomous Systems 74*(Part A), 148 – 165.

PÖRTNER, A., L. SCHRÖDER, R. RASCH, D. SPRUTE, M. HOFFMANN, and M. KÖNIG (2018). The power of color: A study on the effective use of colored light in human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3395–3402.

QUIGLEY, M., K. CONLEY, B. GERKEY, J. FAUST, T. B. FOOTE, J. LEIBS, R. WHEELER, and A. Y. NG (2009). ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

QUINTERO, C. P., R. T. FOMENA, A. SHADEMAN, N. WOLLEB, T. DICK, and M. JAGERSAND (2013). SEPO: Selecting by pointing as an intuitive human-robot command interface. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1166–1171.

QUINTERO, C. P., S. LI, M. K. PAN, W. P. CHAN, H. M. V. DER LOOS, and E. CROFT (2018). Robot programming through augmented trajectories in augmented reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1838–1844.

RAMÍREZ, O. A. I., H. KHAMBHAITA, R. CHATILA, M. CHETOUANI, and R. ALAMI (2016). Robots learning how and where to approach people. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 347–353.

RASCH, R., D. SPRUTE, A. PÖRTNER, S. BATTERMANN, and M. KÖNIG (2019). Tidy up my room: Multi-agent cooperation for service tasks in smart environments. *Journal of Ambient Intelligence and Smart Environments 11*(3), 261–275.

RASHIDI, P. and D. J. COOK (2009). Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Transactions on Systems, Man, and Cybernetics, Part A (Systems and Humans) 39*(5), 949–959.

RASHIDI, P., D. J. COOK, L. B. HOLDER, and M. SCHMITTER-EDGECOMBE (2011). Discovering activities to recognize and track in a smart environment. *IEEE Transactions on Knowledge and Data Engineering 23*(4), 527–539.

RATNER, B. (2009). The correlation coefficient: Its values range between +1/-1, or do they? *Journal of Targeting, Measurement and Analysis for Marketing 17*(2), 139–142.

RIOS-MARTINEZ, J., A. SPALANZANI, and C. LAUGIER (2015). From proxemics theory to socially-aware navigation: A survey. *International Journal of Social Robotics 7*(2), 137–153.

ROSEN, E., D. WHITNEY, E. PHILLIPS, G. CHIEN, J. TOMPKIN, G. KONIDARIS, and S. TELLEX (2019). Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays. *The International Journal of Robotics Research 38*(12-13), 1513–1526.

ROUANET, P., P. Y. OUDEYER, F. DANIEAU, and D. FILLIAT (2013). The impact of human-robot interfaces on the learning of visual objects. *IEEE Transactions on Robotics 29*(2), 525–541.

RUSSAKOVSKY, O., J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATHY, A. KHOSLA, M. BERNSTEIN, A. C. BERG, and L. FEI-FEI (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision 115*(3), 211–252.

RUSSELL, S. and P. NORVIG (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press.

RUSU, R. B., B. GERKEY, and M. BEETZ (2008). Robots in the kitchen: Exploiting ubiquitous sensing and actuation. *Robotics and Autonomous Systems 56*(10), 844–856.

SAFFIOTTI, A., M. BROXVALL, M. GRITTI, K. LEBLANC, R. LUNDH, J. RASHID, B. S. SEO, and Y. J. CHO (2008). The PEIS-ecology project: Vision and results. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2329–2335.

SAKAMOTO, D., K. HONDA, M. INAMI, and T. IGARASHI (2009). Sketch and run: A stroke-based interface for home robots. In *SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 197–200.

SAKAMOTO, D., Y. SUGIURA, M. INAMI, and T. IGARASHI (2016). Graphical instruction for home robots. *Computer 49*(7), 20–25.

SAKAMOTO, J., K. KIYOYAMA, K. MATSUMOTO, Y. PYO, A. KAWAMURA, and R. KURAZUME (2018). Development of ROS-TMS 5.0 for informationally structured environment. *Robomech Journal 5*(24), 1–11.

SANFELIU, A., N. HAGITA, and A. SAFFIOTTI (2008). Network robot systems. *Robotics and Autonomous Systems 56*(10), 793–797.

SCARAMUZZA, D. and F. FRAUNDORFER (2011). Visual odometry: Part I - the first 30 years and fundamentals. *IEEE Robotics & Automation Magazine 18*(4), 80–92.

SCHEPELMANN, A., R. E. HUDSON, F. L. MERAT, and R. D. QUINN (2010). Visual segmentation of lawn grass for a mobile robotic lawnmower. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 734–739.

SCHROETER, C., S. MUELLER, M. VOLKHARDT, E. EINHORN, C. HUIJNEN, H. VAN DEN HEUVEL, A. VAN BERLO, A. BLEY, and H. M. GROSS (2013). Realization and user evaluation of a companion robot for people with mild cognitive impairments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1153–1159.

SCHULZ, D., W. BURGARD, D. FOX, S. THRUN, and A. B. CREMERS (2000). Web interfaces for mobile robots in public places. *IEEE Robotics & Automation Magazine 7*(1), 48–56.

SEIFRIED, T., M. HALLER, S. D. SCOTT, F. PERTENEDER, C. RENDL, D. SAKAMOTO, and M. INAMI (2009). CRISTAL: A collaborative home media and device controller based on a multi-touch display. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS)*, pp. 33–40.

SHIBATA, S., T. YAMAMOTO, and M. JINDAI (2011). Human-robot interface with instruction of neck movement using laser pointer. In *IEEE/SICE International Symposium on System Integration (SII)*, pp. 1226–1231.

SHRESTHA, M. C., T. ONISHI, A. KOBAYASHI, M. KAMEZAKI, and S. SUGANO (2018). Communicating directional intent in robot navigation using projection indicators. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 746–751.

SICILIANO, B. and O. KHATIB (2016). Robotics and the handbook. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 1–6. Springer International Publishing.

SICILIANO, B., L. SCIAVICCO, L. VILLANI, and G. ORIOLO (2009). Introduction. In *Robotics: Modelling, Planning and Control*. Springer London.

SIEGWART, R., I. R. NOURBAKHSH, and D. SCARAMUZZA (2011). Planning and navigation. In *Introduction to Autonomous Mobile Robots*, pp. 369–424. The MIT Press.

SIMOENS, P., M. DRAGONE, and A. SAFFIOTTI (2018). The internet of robotic things: A review of concept, added value and applications. *International Journal of Advanced Robotic Systems 15*(1), 1–11.

SIMONYAN, K. and A. ZISSERMAN (2014). Very deep convolutional networks for large-scale image recognition. http://arxiv.org/abs/1409.1556.

SISBOT, E. A., L. F. MARIN-URIAS, R. ALAMI, and T. SIMEON (2007). A human aware mobile robot motion planner. *IEEE Transactions on Robotics 23*(5), 874–883.

SISBOT, E. A., L. F. MARIN-URIAS, X. BROQUÈRE, D. SIDOBRE, and R. ALAMI (2010). Synthesizing robot motions adapted to human presence. *International Journal of Social Robotics 2*(3), 329–343.

SMITH, R., M. SELF, and P. CHEESEMAN (1990). Estimating uncertain spatial relationships in robotics. In I. J. Cox and G. T. Wilfong (Eds.), *Autonomous Robot Vehicles*, pp. 167–193. Springer New York.

SONG, S. and S. YAMADA (2017). Expressing emotions through color, sound, and vibration with an appearance-constrained social robot. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 2–11.

SPRUTE, D. and M. KÖNIG (2016). On-chip activity recognition in a smart home. In *International Conference on Intelligent Environments (IE)*, pp. 95–102.

SPRUTE, D., A. PÖRTNER, R. RASCH, S. BATTERMANN, and M. KÖNIG (2017). Ambient assisted robot object search. In *International Conference on Smart Homes and Health Telematics (ICOST)*, pp. 112–123.

SPRUTE, D., R. RASCH, A. PÖRTNER, S. BATTERMANN, and M. KÖNIG (2018). Gesture-based object localization for robot applications in intelligent environments. In *International Conference on Intelligent Environments (IE)*, pp. 48–55.

SPRUTE, D., R. RASCH, K. TÖNNIES, and M. KÖNIG (2017). A framework for interactive teaching of virtual borders to mobile robots. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1175–1181.

SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2018). Virtual borders: Accurate definition of a mobile robot's workspace using augmented reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8574–8581.

SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019a). Interactive restriction of a mobile robot's workspace in a smart home environment. *Journal of Ambient Intelligence and Smart Environments 11*(6), 475–494.

SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019b). A study on different user interfaces for teaching virtual borders to mobile robots. *International Journal of Social Robotics 11*(3), 373–388.

SPRUTE, D., K. TÖNNIES, and M. KÖNIG (2019c). This far, no further: Introducing virtual borders to mobile robots using a laser pointer. In *IEEE International Conference on Robotic Computing (IRC)*, pp. 403–408.

SPRUTE, D., P. VIERTEL, K. TÖNNIES, and M. KÖNIG (2019). Learning virtual borders through semantic scene understanding and augmented reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4607–4614.

STACHNISS, C. (2009). Introduction. In *Robotic Mapping and Exploration*, pp. 3–6. Springer Berlin Heidelberg.

STACHNISS, C. and W. BURGARD (2003). Mapping and exploration with mobile robots using coverage maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 467–472.

STACHNISS, C., J. J. LEONARD, and S. THRUN (2016). Simultaneous localization and mapping. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 1153–1176. Springer International Publishing.

STATISTA (2018). Smart home report 2019.

STEIN, P., V. SANTOS, A. SPALANZANI, and C. LAUGIER (2013). Navigating in populated environments by following a leader. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 527–532.

SUZUKI, T., A. OHYA, and S. YUTA (2005). Operation direction to a mobile robot by projection lights. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp. 160–165.

SVENSTRUP, M., T. BAK, and H. J. ANDERSEN (2010). Trajectory planning for robots in dynamic human environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4293–4298.

SZAFIR, D., B. MUTLU, and T. FONG (2015). Communicating directionality in flying robots. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 19–26.

SZEGEDY, C., W. LIU, Y. JIA, P. SERMANET, S. REED, D. ANGUELOV, D. ERHAN, V. VANHOUCKE, and A. RABINOVICH (2015). Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.

TAKETOMI, T., H. UCHIYAMA, and S. IKEDA (2017). Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications 9*(16), 1–11.

TAPIA, E. M., S. S. INTILLE, and K. LARSON (2004). Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing (PerCom)*, pp. 158–175.

TAX, N., N. SIDOROVA, R. HAAKMA, and W. M. P. VAN DER AALST (2018). Mining local process models with constraints efficiently: Applications to the analysis of smart home data. In *International Conference on Intelligent Environments (IE)*, pp. 56–63.

THRUN, S., M. BEETZ, M. BENNEWITZ, W. BURGARD, A. B. CREMERS, F. DELLAERT, D. FOX, D. HÄHNEL, C. ROSENBERG, N. ROY, J. SCHULTE, and D. SCHULZ (2000). Probabilistic algorithms and the interactive museum tour-guide robot minerva. *The International Journal of Robotics Research 19*(11), 972–999.

THRUN, S., W. BURGARD, and D. FOX (2005). *Probabilistic Robotics*. The MIT Press.

THRUN, S., J.-S. GUTMANN, D. FOX, W. BURGARD, and B. J. KUIPERS (1998). Integrating topological and metroc maps for mobile robot navigation: A statistical approach. In *Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pp. 989–995.

TIPALDI, G. D. and K. O. ARRAS (2011). Please do not disturb! Minimum interference coverage for social robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1968–1973.

TÖLGYESSY, M., M. DEKAN, F. DUCHOŇ, J. RODINA, P. HUBINSKÝ, and L. CHOVANEC (2017). Foundations of visual linear human–robot interaction via pointing gesture navigation. *International Journal of Social Robotics 9*(4), 509–523.

TRINH, T. Q., C. SCHROETER, J. KESSLER, and H.-M. GROSS (2015). "Go ahead, please": Recognition and resolution of conflict situations in narrow passages for polite mobile robot navigation. In *International Conference on Social Robotics (ICSR)*, pp. 643–653.

TROUVAIN, B. A., F. E. SCHNEIDER, and D. WILDERMUTH (2001). Integrating a multimodal human-robot interaction method into a multi-robot control station. In *IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, pp. 468–472.

TRUONG, X.-T. and T.-D. NGO (2016). Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. *International Journal of Social Robotics 8*(5), 663–684.

VAUGHAN, J., S. KRATZ, and D. KIMBER (2016). Look where you're going: Visual interfaces for robot teleoperation. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 273–280.

VEGA, A., L. J. MANSO, D. G. MACHARET, P. BUSTOS, and P. NÚÑEZ (2019). Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances. *Pattern Recognition Letters 118*(1), 72–84.

VELOSO, M., J. BISWAS, B. COLTIN, and S. ROSENTHAL (2015). CoBots: Robust symbiotic autonomous mobile service robots. In *International Conference on Artificial Intelligence (IJCAI)*, pp. 4423–4429.

VELOSO, M., J. BISWAS, B. COLTIN, S. ROSENTHAL, T. KOLLAR, C. MERICLI, M. SAMADI, S. BRANDÃO, and R. VENTURA (2012). CoBots: Collaborative robots servicing multi-floor buildings. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5446–5447.

WACHS, J. P., M. KÖLSCH, H. STERN, and Y. EDAN (2011). Vision-based hand-gesture applications. *Communications of the ACM 54*(2), 60–71.

WALLGRÜN, J. O. (2010). Robot mapping. In *Hierarchical Voronoi Graphs: Spatial Representation and Reasoning for Mobile Robots*, pp. 11–43. Springer Berlin Heidelberg.

WEISS, A., R. BERNHAUPT, M. LANKES, and M. TSCHELIGI (2009). The USUS evaluation framework for human-robot interaction. In *Symposium on New Frontiers in Human-Robot Interaction*, pp. 158–165.

WILDE, N., D. KULIĆ, and S. L. SMITH (2018). Learning user preferences in robot motion planning through interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 619–626.

WITTEN, I. H., E. FRANK, and M. A. HALL (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc.

WOLF, M. T., C. ASSAD, M. T. VERNACCHIA, J. FROMM, and H. L. JETHANI (2013). Gesture-based robot control with variable autonomy from the JPL BioSleeve. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1160–1165.

YAMAOKA, F., T. KANDA, H. ISHIGURO, and N. HAGITA (2010). A model of proximity control for information-presenting robots. *IEEE Transactions on Robotics 26*(1), 187–195.

YOUSIF, K., A. BAB-HADIASHAR, and R. HOSEINNEZHAD (2015). An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intelligent Industrial Systems 1*(4), 289–311.

ZENDER, H., P. JENSFELT, and G. J. M. KRUIJFF (2007). Human- and situation-aware people following. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1131–1136.

ZHANG, Z., C. CONLY, and V. ATHITSOS (2015). A survey on vision-based fall detection. In *ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA)*, pp. 46:1–46:7.

ZHAO, H., J. SHI, X. QI, X. WANG, and J. JIA (2017). Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239.

ZHOU, B., H. ZHAO, X. PUIG, T. XIAO, S. FIDLER, A. BARRIUSO, and A. TORRALBA (2019). Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision 127*(3), 302–321.

ZIEFLE, M., S. HIMMEL, and W. WILKOWSKA (2011). When your living space knows what you do: Acceptance of medical home monitoring by different technologies. In *Information Quality in e-Health: Conference of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB)*, pp. 607–624.

ZOLOTAS, M., J. ELSDON, and Y. DEMIRIS (2018). Head-mounted augmented reality for explainable robotic wheelchair assistance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1823–1829.

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, den 17.04.2020

Dennis Sprute