International Journal of Social Robotics (2019) 11:373–388
https://doi.org/10.1007/s12369-018-0506-3

A Study on Different User Interfaces for Teaching Virtual Borders to Mobile Robots

Dennis Sprute[1,2] · Klaus Tönnies[2] · Matthias König[1]

Accepted: 9 November 2018 / Published online: 16 November 2018

Abstract
Human-aware robot navigation is an essential aspect to increase the acceptance of mobile service robots in human-centered environments, e.g. home environments. Robots need to navigate in a human-acceptable way according to the users' conventions, presence and needs. In order to address the users' needs, we employ virtual borders, which are non-physical borders and respected by the robots while working, to effectively restrict the workspace of a mobile robot and change its navigational behavior. To this end, we consider different user interfaces, i.e. visual markers, a laser pointer, a graphical user interface and a RGB-D Google Tango tablet with augmented reality application, to allow non-expert users the flexible and interactive definition of virtual borders. These user interfaces were evaluated with respect to their correctness, flexibility, accuracy, teaching effort and user experience. Experimental results show that the RGB-D Google Tango tablet as user interface yields the best overall results compared to the other user interfaces. Apart from a low teaching effort and high flexibility and accuracy, it features the highest user ratings acquired from a comprehensive user study with 25 participants for intuitiveness, comfort, learnability and its feedback system.

Keywords Human-aware navigation · Socially-aware navigation · Human-centered environments · Social human–robot interaction · Virtual borders · User interfaces

1 Introduction

We consider a human–robot shared space, such as an office or home environment, where service robots support the residents in their everyday life. These are vacuum cleaning robots cleaning the floor [1], companion robots assisting elderly people [2], collaborative robots interacting with humans [3] or service robots searching for and bringing objects [4]. They work in the entire building alongside humans living there, and the humans appreciate the coexistence and support of the robots.

This work is financially supported by the German Federal Ministry of Education and Research (BMBF, Funding Number: 03FH006PX5).

✉ Dennis Sprute
dennis.sprute@fh-bielefeld.de

Klaus Tönnies
klaus@isg.cs.uni-magdeburg.de

Matthias König
matthias.koenig@fh-bielefeld.de

1 Bielefeld University of Applied Sciences, Campus Minden, 32427 Minden, Germany

2 Faculty of Computer Science, Otto-von-Guericke University Magdeburg, 39106 Magdeburg, Germany

1.1 Motivation

However, if a cleaning robot enters a kid's corner and vacuums up toy blocks or a mobile companion robot tackles a pet's water dish and spills water yet again, the residents become annoyed. Moreover, if a service robot damages an expensive carpet while crossing it, humans stop deploying robots in their home environment. Besides, humans do not like robots, equipped with cameras and network connectivity, to enter certain places in the building, such as bed- or bathrooms. There are still privacy concerns related to such places and these must be respected by the robots while working. If mobile robots do not consider such situations, it will lead to a stagnating acceptance of robots, and robots will then not pervasively find their ways into human-centered environments. Therefore, mobile service robots should navigate in a socially-aware manner considering the presence and demands of the people. This is an essential aspect to deploy robots in human-centered environments.

In this work, we consider *virtual borders* as a possibility to effectively restrict the workspace of 3-DoF mobile service robots and ensure human-aware navigation. These are non-physical borders in the environment, that are not directly visible by humans, but are respected by the mobile robots while navigating in the environment. These virtual borders need to be defined by humans who are non-expert users (1) with no programming skills. Besides, they (2) do not have much experience with robots or their insights, but they (3) are used to interact with common consumer devices, such as tablets or smartphones. Due to their nature, they (4) prefer a feature-complete system correctly fulfilling its task to a highly complex and experimental system. Finally, we characterize a non-expert to (5) have no cognitive impairments or upper limb disorders.

A teaching method, that allows non-experts to interactively restrict the workspace of their mobile robots using virtual borders, needs to fulfill certain requirements to be accepted by the users:

1. **Correctness** Users expect a teaching method to be correct, i.e. the user-defined virtual borders in the teaching process should be correctly incorporated into the navigational map of the mobile robot. Thus, the mobile robot changes its navigational behavior according to the user's needs.
2. **Flexibility** A high flexibility is expected, i.e. the user can define arbitrary virtual borders with different shapes and complexities. Thus, the user can define the workspace as desired.
3. **Accuracy** An important aspect for a high user acceptance is the accuracy of the defined virtual borders. A high accuracy entails precisely defined virtual borders, i.e. the virtual borders are integrated into the mobile robot's navigational map where the user wants them to be.
4. **Teaching effort** A low teaching effort, i.e. the time to teach a virtual border, is an indicator for a good usability. Hence, defining virtual borders should take as less effort as possible.
5. **User experience** A good user experience is characterized by an intuitive and comfortable teaching method and user interface (UI). Moreover, a powerful feedback system giving information about the current state of the teaching process and the defined virtual borders to the user increases the user experience. Another important point is the learnability of the teaching method that should be easy to learn.

Finally, the teaching method should not require additional hardware installed in the environment, such as cameras, to allow an easy portability of the system.

1.2 Problem Statement

This leads to the following problem definition: we consider an indoor environment, such as an office or home environment. The environment consists of free space, e.g. rooms and corridors, and physical borders, e.g. walls or furniture. A 3-DoF mobile robot equipped with sensors, e.g. laser scanner and camera, operates in this environment, that does not have any sensory capabilities. In order to autonomously navigate in the environment, the mobile robot has access to a 2D occupancy grid map (OGM) [5] of the environment, that models the physical environment in terms of discrete cells. Each cell contains a probability for the occupancy of the corresponding area in the environment. Based on this OGM, the robot can calculate a costmap for navigation purposes, i.e. the mobile robot can move to all locations marked as free space in the OGM (except of inflation areas around obstacles and borders). This space is referred to as *workspace* of the mobile robot.

Although a user does not want his or her mobile robot to cross certain areas as described in the previous section, the robot will cross these areas if it considers the OGM of the environment as basis for navigation. The problem is that these areas are not modeled as occupied areas in the OGM and that they cannot be identified as an obstacle by the robot's on-board sensors. Therefore, these areas need to be explicitly defined by a user and integrated as occupied areas into the OGM to restrict the robot's workspace. Thus, we obtain a OGM containing physical as well as *virtual borders*. This OGM can then be used in navigation tasks as basis for costmaps to change the mobile robot's navigational behavior according to the user's needs. In this work, we focus on the question of *which UI is best suited for the definition of arbitrary virtual borders with respect to the identified requirements*.

For reasons of clarity, we consider a consistent use case scenario throughout this article: the scenario takes place in a home environment consisting of several rooms with a non-expert resident as specified in the introductory part. Additionally, a mobile service robot equipped with an on-board RGB-D camera and a laser scanner for localization purposes is working in the home environment. Since the non-expert is afraid that the mobile service robot damages his or her expensive carpet with its wheels, he or she wants the robot to circumvent the carpet area while moving in the environment. Therefore, the resident wants to define virtual borders in the workspace of his or her mobile robot.

1.3 Contribution

Parts of this article are based on our previous publications [6–8]. In this work, we contribute the following points, that have not been addressed in the other works:

1. We performed a **comprehensive user study** with 25 participants evaluating four different UIs for teaching virtual borders and report the results for the criteria accuracy, teaching effort and user experience.
2. We consider **drawing on a graphical user interface (GUI)** as a new interface in the evaluation due to feedback from users and the emergence of robot mapping capabilities in recent consumer robots, e.g. vacuum cleaning robots.
3. We consider **additional evaluation criteria** for the teaching methods and UIs, such as flexibility of the teaching method and the users' success rate during the experiments. Besides, we performed a survey covering topics related to the user experience with the teaching method, e.g. intuitiveness, comfort, learnability or feedback.
4. We report **certain characteristics of the UIs** and teaching methods that have been identified during the experiments and a more detailed evaluation, e.g. a correspondence problem for the GUI interface.
5. We give a **comparative overview** of UIs and teaching methods concerning eight features and give a recommendation for a UI based on the comparison.

1.4 Structure of this Article

In the next section of this article, we give a literature overview of relevant works concerning related topics, such as human-aware robot navigation and human–robot interaction. Afterwards, we formally describe a virtual border to be used by our generic algorithm for the integration into an OGM. Subsequently, we introduce four teaching methods and UIs that allow the definition of a virtual border's components. The first teaching method is based on visual markers with different IDs used to guide a mobile robot. While following the markers, the mobile robot stores its trajectory that is used to define virtual border points. Similarly, a user can employ a laser pointer to guide the mobile robot and define virtual borders. The third method employs a GUI showing a 2D OGM of the environment. A user can simply integrate virtual borders by drawing on a tablet's display. As a last UI, a RGB-D Google Tango tablet allows a user the definition of virtual borders with an augmented reality (AR) application. These different UIs are evaluated in different experiments with respect to the identified requirements, and the results are presented in Sect. 5. Finally, we conclude our work and recommend work for the future based on a comparative summary of the UIs.

2 Literature Review

There are basically two possibilities to restrict the workspace of a mobile robot and change the robot's navigational behavior: (1) the adaption of the mobile robot's navigation path based on implicit observations or prior knowledge known from the field of human-aware robot navigation and (2) explicit methods from the field of human–robot interaction (HRI) allowing the definition of virtual borders. In this section, we give an insight into both possibilities and explain why we prefer virtual borders for our scenario. Furthermore, we point out the need for our work by giving an overview of different UIs that can be used for virtual border definition.

2.1 Human-Aware Robot Navigation

Nowadays, mobile robots do not only need to navigate in a safe but also in a socially-aware way [9]. This challenge arises due to the presence of humans in the mobile robot's workspace. Rios-Martinez et al. define a socially-aware robot navigation as a strategy exhibited by a social robot that acts according to identified social conventions to enable a comfortable interaction between humans and robots [10]. This implies respecting personal space [11,12], minimizing the probability to encounter humans [13], passing people [14], approaching people [15,16], following people [17] and walking side-by-side with people [18]. Kruse et al. similarly identified three categories of properties to enable a human-aware robot navigation, i.e. comfort, naturalness and sociability [19]. Other works in this field comprise an effective human comfortable safety framework [20], works analyzing human motion patterns [21,22] and a framework for socially adaptive path planning in dynamic environments [23]. A recent overview of trends in socially-aware robot navigation is given by Charalampous et al. [24]. All these approaches have in common that mobile robots adapt their navigational behavior according to the user who is present in the human–robot shared space. Hence, it is one possibility to restrict the workspace of a mobile robot.

2.2 Human–Robot Interaction

However, for our challenge described in the introduction, i.e. a definition of arbitrary areas according to the users' needs, these approaches are not suitable. They modify the robot's navigational behavior based on implicit observations, e.g. human presence or trajectories [25], and prior knowledge of the virtual personal space, known as proxemics [26], is leveraged to enhance robot's navigation capabilities. But we think that explicit HRI is the only way to allow users the flexible definition of *arbitrary* areas. For this purpose, adequate HRI interfaces must be employed to satisfy the requirements of non-expert users. The required HRI for this task can be generally broken down to providing 2D positions on the ground plane that can be used for the definition of virtual borders.

There are basically two categories on how to provide such spatial information: direct and indirect methods. An example for a direct method is the approach by Sakamoto et al. who use cameras mounted in the home environment to show a live stream of the environment on a tablet [27]. The user can then define the workspace of a vacuum cleaning robot by drawing on the screen. However, this approach needs additional cameras mounted in the environment. Other direct approaches for mobile service robots comprise beacon devices [28] or magnetic strips [29], but these methods are intrusive, power-consuming and inflexible. An alternative to these intrusive methods are pointing gestures to define positions in space, e.g. through human gestures [30] or pointing devices [31,32]. A comparison between a laser pointer and touch screen interface for providing 3D locations is given by Choi et al. reporting a high satisfaction by the users for both devices [33]. The advantage of touch screen interfaces combined with AR techniques is another approach allowing users to directly interact with the environment [8].

An alternative to these methods are indirect methods to provide 2D positions on the ground plane. These can be obtained as a trajectory of a mobile robot that is guided by a user. For example, visual markers [6] and laser pointers [7] were used to guide a mobile robot using visual servoing technique. Both approaches leverage the high accuracy of the robot localization yielding to accurate virtual borders. However, the UIs were only evaluated with respect to a limited number of criteria and users. Along these UIs, a mobile robot can also be guided employing direct physical control [34,35], game pads [36] or tablets [37].

To sum up, there are two main possibilities for changing the mobile robot's navigational behavior according to the users' needs: (1) implicit methods based on observations or prior knowledge and (2) explicit methods based on the interaction between human and robot. In order to flexibly define arbitrary virtual borders, we need to focus on the second category, explicit HRI. For this purpose, there are a lot of different UIs and teaching methods imaginable but there is no comprehensive user study on different UIs with respect to teaching virtual borders.

3 The Role of Virtual Borders

In this work, we employ virtual borders, that are non-physical borders, to interactively modify the workspace of a 3-DoF mobile robot and change the robot's navigational behavior. The mobile robot will respect the user-defined virtual borders allowing a socially-aware robot navigation according to the user's needs. To this end, a user interactively defines virtual borders, and these are incorporated into a prior map of the environment. The resulting posterior map can be used by a common global path planner as basis for a costmap. Thus, we integrate social costs, explicitly defined by the user, into the navigation framework of a robot.

The following teaching methods have in common that they are based on the same algorithm for creating a posterior map from a given prior map and a virtual border. This algorithm was formalized in our previous work [8]. Since this algorithm is employed by the different teaching methods but is out of this article's scope, we only describe the interface to this *generic algorithm*.

The algorithm requires a 2D OGM of the physical environment as input (*prior map*) and outputs a 2D OGM containing physical as well as virtual borders (*posterior map*). Virtual borders are defined by a user in a teaching process as a triple $V = (\mathcal{P}, \boldsymbol{s}, \delta)$.

- The **virtual border points** $\mathcal{P}$ specify boundaries of the virtual border and are structured as a polygonal chain. The polygonal chain consists of n points $\boldsymbol{p}_i \in \mathbb{R}^2$ with $i \leq n$ corresponding to coordinates on the ground plane of the environment. We distinguish between (1) closed and (2) simple polygonal chains allowing the user a flexible definition of arbitrary virtual borders. In case of a simple polygonal chain, that does not partition the map, we apply a linear extension to the first and the last line segments. Thus, the beginning and the ending of the polygonal chain are automatically extended to the borders of the prior map allowing the user to easily exclude large areas from the mobile robot's workspace.
- A **seed point** $\boldsymbol{s} \in \mathbb{R}^2$ is the user-defined component of a virtual border V that indicates the area to be modified during the teaching process.
- $\delta \in [0, 1]$ is the user-defined component which indicates the **occupancy probability** of the area to be modified (as indicated by $\boldsymbol{s}$).

In order to allow a user the definition of arbitrary virtual borders, this teaching process can be performed N times resulting in a sequence of virtual borders $V^* = \{V_1, V_2, \ldots, V_N\}$. Hence, the posterior map of the i-th teaching process becomes the prior map of the $i+1$-th teaching process. A user needs to define the components of a virtual border $V = (\mathcal{P}, \boldsymbol{s}, \delta)$ to apply this generic algorithm and integrate the virtual borders into the prior map. The question of *how to define these virtual border components with different UIs* is investigated in the following section by considering four different UIs and teaching methods. Details on how the virtual borders are integrated into the prior map can be found in [8].

4 User Interfaces and Teaching Methods

A UI offers the possibility for interaction between a human and a robot, while a teaching method describes how to interact with the robot using the UI to achieve the goal, i.e.

(a) **(b)** **(c)** **(d)**

Fig. 1 Example screenshots of the teaching process using visual markers. A user guides a mobile robot around a carpet using a marker cube with different marker IDs. The robot follows the marker using its on-board camera and stores its trajectory for virtual border definition

the definition of virtual borders. For this purpose, different UIs are applicable as described in Sect. 2.2 and one cannot evaluate all of them. We selected four different UIs that we think would well address the requirements identified in Sect. 1.1. In this study, we consider visual markers, a laser pointer, a GUI on a common tablet and a RGB-D Google Tango tablet with AR application as UIs. We give details on the UIs, the teaching methods and reasons for their selection in the following subsections. Furthermore, we apply the UIs in two types of teaching methods. Direct teaching methods directly interact with the environment or map, while indirect methods need an additional step for interaction, e.g. interaction based on robot guidance. The robot's position data acquired during guidance can be further used as a definition of a virtual border. Hence, the user indirectly interacts with the map using the 2D positions of the mobile robot.

4.1 Visual Markers

Visual markers are used to guide the mobile robot along the desired virtual borders while the mobile robot keeps track of its path. The approach is based on a teaching framework using robot guidance that we introduce in [6]. Since the user does not directly interact with the map or environment but rather leverages a mobile robot for the teaching process, this approach is an example for an indirect teaching method. The teaching framework mainly consists of three states: (1) a state to guide the robot to a start position for virtual border definition, (2) a state to guide the mobile robot along the desired virtual border points $\mathcal{P}$ and (3) a state to define the seed point $\boldsymbol{s}$ indicating the keep off area ($\delta = 1$). We choose visual markers [38] as UI since visual markers are easy to use and often used for preliminary studies.

In the concrete implementation, different marker IDs correspond to different states of the teaching framework. Thus, we use three different marker IDs to define a virtual border. A change of the marker ID triggers a state transition. These markers are employed by the user to guide the robot in the environment using visual servoing technique. Feedback to the user is provided in two ways: (1) on-board LEDs indicate different states of the teaching framework by changing the color and (2) simple sounds of the robotic platform signalize an internal state change and the recording of the virtual border points $\mathcal{P}$. In order to realize the described behavior, a localized mobile robot needs to be available in the environment, and it needs a calibrated monochrome camera pointing towards the front. Figure 1 shows some exemplary screenshots of a teaching process using visual markers. A user holds a marker cube with different IDs on its sides to guide the robot. First, the user guides the mobile robot in the direction of the carpet using the first ID. After reaching the desired start position, the user changes the marker to the second ID and guides the mobile robot around the carpet. Finally, the marker with the third ID is used to rotate the robot towards the keep off area. A more detailed description of the teaching method can be found in [6], and a full video can be found online at: https://youtu.be/g-Yjtqgibog.

4.2 Laser Pointer

Based on the previously described teaching framework, we propose a teaching method using a laser pointer as HRI interface [7]. We choose a laser pointer for several reasons: (1) a laser pointer is more accurate in providing 2D positions compared to human gestures due to the inherent uncertainties in gesture recognition [39], and (2) studies have shown that mediator devices are judged more efficient for visual object learning and providing positions in space while being equally intuitive compared to human gestures [40]. Furthermore, (3) laser pointers are common everyday devices that make their application by non-expert users intuitive and (4) they provide inherent visual feedback. Finally, (5) guiding a mobile robot using a pointing device is easier compared to remote controlling using tablets, smartphones or control pads. The user only has to provide 2D positions instead putting themselves into the perspective of the mobile robot. Besides, driving a mobile robot by viewing a live video captured from a robot's on-board camera leads to a "tunnel vision" and a lack of awareness as identified by Vaughan et al. [41].


(a) (b) (c) (d)

Fig. 2 Example images of the teaching process using a laser pointer. The mobile robot follows the laser spot using its on-board camera and stores its trajectory depending on its internal state. The trajectory is used for the definition of the virtual border


(a) (b) (c) (d)

Fig. 3 Example screenshots of the teaching process by drawing on a GUI. The tablet shows an OGM of the environment on the screen, and a user defines the virtual border around the carpet by drawing on the tablet's display using his or her fingers. The resulting OGM is directly visualized on the tablet's display

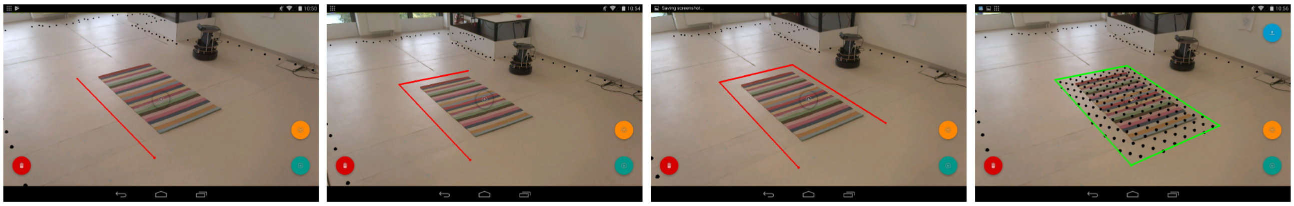
By employing this teaching method, the user guides the mobile robot along the desired virtual border using a laser pointer. In order to switch between different states of the teaching framework, e.g. recording the robot's path $\mathcal{P}$ or defining a seed point $\boldsymbol{s}$, the user can employ the laser pointer to generate visual codes recognized by the robot's on-board camera. Alternatively, the user can press buttons on the robot platform to change the internal state of the system. For this purpose, the mobile robot needs to be localized inside the environment and requires a front-mounted RGB-D camera. The depth information is necessary to estimate the distance to the laser spot and adjust the mobile robot's velocity according to the distance. Feedback to the user is, analogously to the marker approach, provided through on-board LEDs on the robot to indicate the current state of the teaching framework and through a sound signalizing a change of internal state. Furthermore, a laser pointer provides inherent visual feedback to the user. Figure 2 shows four example images of a teaching process with a laser pointer. Similar to the visual marker approach, the user first guides the mobile robot in proximity to the carpet. Subsequently, the user generates a visual code with the laser pointer to change the internal state of the robot and start recording of the mobile robot's path. After guiding the robot along the desired virtual border, a visual code is again used to change the internal state of the robot. Finally, the user employs the laser pointer to rotate the robot in the direction of the desired keep off area ($\boldsymbol{s}$ and $\delta = 1$), i.e. the carpet area. More details about the approach are described in [7], and a full video of a teaching process can be found online at: https://youtu.be/lKsGp8xtyIc.

4.3 GUI Drawing

The first direct teaching method is based on a common tablet with a 2D OGM of the environment shown on the display. We choose this UI because (1) tablets are common consumer products that are known to non-experts and (2) handling a tablet should be intuitive for non-experts. A user holds the common tablet and can move freely in the environment. In order to define virtual borders, the user directly draws the desired virtual border points $\mathcal{P}$ in the OGM of the environment shown on the tablet. For this purpose, the person can use his or her fingers to draw a polygonal chain in the map. Additionally, a flood filling tool is available to indicate the seed point $\boldsymbol{s}$ and fill the corresponding area. The occupancy probability δ of the indicated area is defined by the selected filling color. As a feedback of the system, the user can directly see the posterior OGM including the virtual borders. Figure 3 depicts exemplary screenshots of a teaching process using drawing on a GUI. The non-expert draws four edges around the carpet area and fills out the area using flood filling. This teaching method does not have any additional requirements.

4.4 Google Tango Tablet

In contrast to common tablets, a Google Tango tablet is a RGB-D tablet that can perceive depth information of a scene additional to color data. With its high-accuracy on-board visual-inertial odometry it can keep track of



(a) (b) (c) (d)

Fig. 4 Example screenshots of the teaching process using a RGB-D Google Tango tablet with AR application. A user freely moves in the environment with the Tango tablet while an augmented live video stream of the tablet's camera is shown on the display. Virtual border points and the seed point can be added or removed using the software buttons (green, yellow and red) while simultaneously pointing the center of the display towards the desired physical point in the environment. Red lines show virtual borders during the teaching process, and green lines show the virtual borders integrated into the global OGM. Black dots indicate occupied areas in the OGM. (Color figure online)

its 6-DoF pose in space. This tablet is used to directly interact with the environment through AR techniques. We choose this UI for interaction because of several reasons: (1) non-expert users are familiar with tablets making it especially attractive for this user group, (2) the display provides an opportunity for direct visual feedback to the user, (3) the accurate on-board visual-inertial odometry and depth sensor allow the creation of user-friendly AR applications.

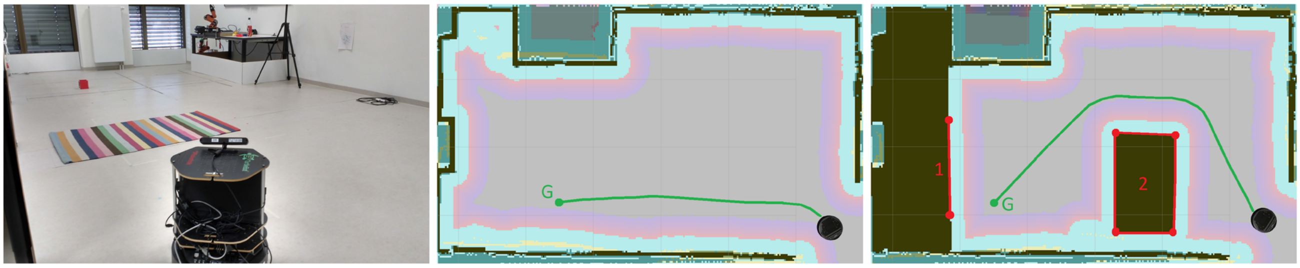
In order to define virtual borders, the user moves around in the environment with the Tango tablet and selects virtual border points $\mathcal{P}$ and the seed point $\boldsymbol{s}$ by pointing the device to the desired position in 3D space. The tablet's display shows the live video stream of the tablet's camera augmented with additional information, such as user-defined virtual borders and the global OGM showing the workspace of the mobile robot. This inherent visual feedback system is tailored to non-expert users to easily interact with the system and understand the robot's workspace. Such an immediate feedback also gives opportunities to correct eventual mistakes during the teaching process. If the user wants to close a virtual border polygon and points at a previously defined point, the tablet also provides vibration feedback. Additionally, the occupancy probability δ can be chosen on a simple menu on the display. This teaching method requires the Tango tablet's base coordinate frame to be registered with the map's coordinate frame beforehand to allow transformations between them. Moreover, the Tango tablet needs to be initially localized inside the environment to allow an AR application. Figure 4 shows screenshots of a teaching process where a user defines a virtual border around a carpet. To this end, he or she defines a polygon $\mathcal{P}$ by selecting four corner points around the carpet and a seed point $\boldsymbol{s}$ on the carpet area. The live view on the display is augmented with physical and virtual borders shown as black dots representing the global OGM. We refer the reader to [8] for a detailed description of the approach. A full video of a teaching process using the RGB-D Google Tango tablet can be found online at: https://youtu.be/oQO8sQ0JBRY.

5 Experimental Evaluation

In this section, we describe our experimental evaluation that aims to assess the different UIs and teaching methods with respect to the five criteria stated in Sect. 1.1, i.e. correctness, flexibility, accuracy, teaching effort and user experience. The correctness indicates if the user-defined virtual borders are correctly integrated into the robot's navigational map and if the robot changes its navigational behavior accordingly. The second criterion is used to assess the flexibility of the teaching method, i.e. a user can define arbitrary workspaces with different shapes and complexities. The evaluation of the accuracy gives an insight on how accurate a given UI can be used for the definition of virtual borders. This is especially important for precision tasks, such as robot vacuum cleaning or mopping around carpets or close to stairs. In order to assess the effort of teaching virtual borders with different UIs, we measure the time to teach different virtual borders. Finally, we comprise aspects, such as intuitiveness, comfort, learnability and feedback, under the criterion of user experience. This criterion reflects the subjective perception of the users with respect to the teaching methods and UIs and is therefore a strong indicator for the applicability by non-expert users. We evaluated the correctness independent of the UIs since it only depends on the generic algorithm and its resulting posterior maps. Although it is independent of the UI, we present the results for reasons of completeness. The other four evaluation criteria were evaluated depending on the UI and teaching method.

5.1 Experimental Setup

All experiments were performed in our 6.1 m × 3.5 m lab environment. A 2D OGM of the environment was created in advance using a common Simultaneous Localization and Mapping (SLAM) algorithm [42] performed on a Turtlebot v2 equipped with an on-board laser scanner and front-mounted RGB-D camera. This map had a resolution of 2.5 cm



(a) (b) (c)

Fig. 5 **a** Lab environment and costmaps for a navigation scenario before and after defining virtual borders. **b** Before defining virtual borders, the mobile robot takes the shortest path (green) to the navigation goal G. **c** After defining two virtual borders (red), the mobile robot circumvents the carpet area while navigating to the same goal G. (Color figure online)

per pixel and served as prior map of the environment. The Turtlebot v2 was used for both indirect teaching methods using visual markers and a laser pointer, while an Asus Nexus 7 tablet with a display size of 7 in. was used for drawing on a GUI. The teaching method based on AR technology was realized on a 7 in. Google Tango tablet.

5.2 Tools

The generic algorithm as well as the teaching methods were implemented as ROS packages [43], the de facto standard for robot applications. ROS is a middleware architecture that enables the communication between several *nodes* in a system, e.g. a node for the detection of markers or laser spots. Related nodes can be summarized in packages that allow the easy distribution of software components. We implemented the generic algorithm as a node providing a service to integrate virtual borders. The interface was the same as described in Sect. 3. This service was called during the experiments by the nodes for the different teaching methods, e.g. a node for the Tango approach. If the service was called, the internal algorithm calculated the posterior map containing the virtual borders and simultaneously saved it to a hard disk. Thus, we obtained all posterior maps defined in the experiments for further evaluation. In case of the accuracy evaluation, we additionally implemented a program that calculated the similarity between two OGMs as described later in Sect. 5.5. For the time measurement of a teaching process, we used internal ROS functionality.

5.3 Correctness

We evaluated this criterion to show the correctness of the generic algorithm, i.e. the user-defined virtual borders are correctly integrated into the prior map of the environment. The resulting posterior map can be used in future navigation tasks as basis for path planning to change the mobile robot's behavior according to the user's needs. Since the correctness is independent of a concrete UI, we prove it in a simple navigation scenario as depicted in Fig. 5. For this purpose, we investigated the robot's navigational behavior before and after defining virtual borders. Figure 5a shows an image of the lab environment with the mobile robot at the bottom of the image and a navigation goal represented by a red cube in the top left of the image. Additionally, we placed a carpet between both locations. If we use the prior map of the environment as basis for path planning, the resulting costmap and the robot's path to the goal are shown in Fig. 5b. The mobile robot takes the path with the fewest costs to the navigation goal, i.e. the shortest path, and the robot crosses the carpet. But if the user defines a virtual border around the carpet area in advance (border 2), the costmap and the path to the same goal changes as visualized in Fig. 5c. Instead of crossing the carpet area, the robot now circumvents the carpet as desired by the user. This shows that the generic algorithm successfully integrates the virtual borders into the prior map. The output of this algorithm is a modified 2D OGM of the environment that can be used by a path planning algorithm to change the robot's navigational behavior. We used a navigation function computed with Dijkstra's algorithm as a global planner in this example.

5.4 Flexibility

The flexibility of a UI and teaching method depends on two aspects: (1) the generic algorithm and (2) the type of teaching method. The generic algorithm models virtual border points as a polygonal chain. Thus, it reaches maximal flexibility in the definition of virtual borders. Furthermore, we consider simple as well as closed polygonal chains to make the teaching process even more flexible. The first and second virtual border visualized as red lines in Fig. 5c are examples for a simple and a closed polygonal chain, respectively. If the user defines a simple polygonal chain, the generic algorithm automatically extends the beginning and the ending of the chain to the borders of the prior map. Therefore, the user can easily define large areas, e.g. excluding a whole room from the mobile robot's workspace. Finally, the generic algorithm features an iterative nature such that arbitrary virtual borders can be defined iteratively. The resulting map in Fig. 5c is an example for this feature where two virtual borders are integrated, i.e. a simple polygonal chain excluding the window area (border 1) and a closed polygonal chain excluding the carpet area from the workspace (border 2). Since all teaching methods employ this generic algorithm, they all feature the same high flexibility in this aspect.

The second aspect deals with the type of teaching method, i.e. indirect or direct method. If we consider an indirect teaching method, the user guides the mobile robot using either visual markers or a laser pointer. This works as long as there are no physical obstacles, e.g. a table, on the desired virtual border that the mobile robot has to cross. If we instead consider a direct teaching method, the user can directly interact with the physical environment or map. Hence, the user can define arbitrary virtual borders without being restricted by obstacles. Furthermore, the user can define virtual borders very close to physical obstacles or other elements, such as down stairs, without requiring the robot to access that area. Therefore, we assess the flexibility of the indirect teaching methods worse than the flexibility of the direct teaching methods.

5.5 Accuracy

The accuracy states how precisely the user defined the virtual borders, and a high accuracy is desired, especially in cases such as robotic vacuum cleaning or mopping where accurate borders are necessary.

5.5.1 Procedure

We evaluated the accuracy on a self-recorded dataset containing ten different maps of our lab environment with manually integrated virtual borders. The virtual borders are polygonal shaped, convex and non-convex and their lengths range from 4 to 13 m. Exemplary OGMs of the dataset are visualized in the first row of Fig. 6. Black, white and gray pixels in the OGMs represent the occupancy probability of the physical environment, while yellow pixels show the ground truth virtual borders. In order to evaluate the accuracy, we asked a single non-expert to define the virtual borders as defined in the dataset employing the four different UIs. The non-expert had time to get familiar with the different UIs and teaching methods. We asked a single non-expert because it is not practical to ask a group of non-experts to evaluate all maps of the dataset with all UIs and some variations. This is a tedious work and would take several hours for a single person. Thus, a single non-expert evaluated the UIs in terms of their accuracy on an extensive dataset in this experiment. However, we also evaluated the accuracy of a group of non-experts on a single map instead of an extensive dataset. These results are presented in Sect. 5.7.3 along with our user study. Before conducting the experiments, we marked the corresponding ground truth areas in the physical environment to allow the non-expert the definition of the virtual borders. Afterwards, the non-expert used one of the UIs and defined the virtual borders according to the ten ground truth maps. In order to get meaningful results, each virtual border was defined five times per UI resulting in 50 runs on the dataset for each UI. These five runs per map and UI also introduced some variation in the teaching process, e.g. through different start positions of the virtual border polygon $\mathcal{P}$.

After performing the experiments, we determined the accuracy by calculating the Jaccard similarity index (JSI) between two virtual borders GT and UD:

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

We define GT and UD in the following:

1. GT (ground truth): This is the set of cells that belongs to the ground truth virtual borders. They are defined manually and are part of the self-recorded dataset. These areas are visualized as yellow cells in the first row and as yellow and green cells in the following rows of Fig. 6.
2. UD (user defined): This is the set of cells that belongs to the user-defined virtual borders. They are defined in the teaching process by the non-expert user employing one of the four UIs. These areas are visualized as green and red cells in Fig. 6.

Moreover, $GT \cap UD$ is the intersection set that is visualized as green pixels in Fig. 6, and $GT \cup UD$ is the union set shown as an area encompassed by a blue contour. Hence, the Jaccard index can be visually interpreted as the green area with respect to the area enclosed by the blue contour. For reasons of completeness, red cells indicate areas defined by the user that are not part of the ground truth cells $UD \setminus GT$.

5.5.2 Results

We provide qualitative and quantitative results for the accuracy experiments. First, we take a look at the qualitative results that are visualized in Fig. 6. The first row shows four exemplary ground truth maps of the self-recorded dataset, while the following three rows depict overlapping maps of the ground truth with the user-defined virtual borders. It is apparent that the approaches based on visual markers, a laser pointer and a Tango tablet are highly accurate. This means that the green intersection area fills out the blue enclosed

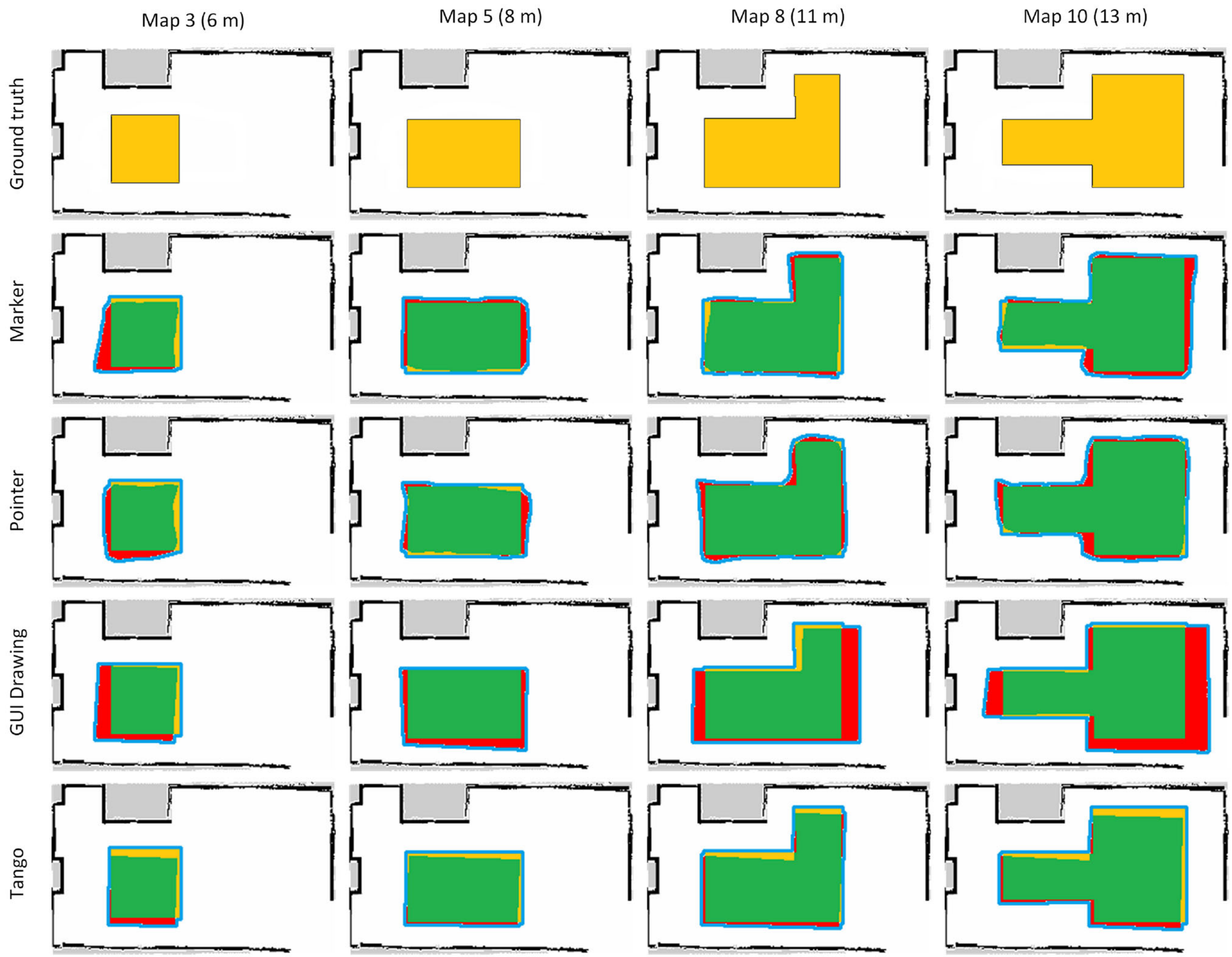


Fig. 6 Qualitative accuracy results for the different UIs dependent on exemplary maps in the dataset. The first row shows four ground truth maps consisting of the physical environment and a virtual border (yellow). The following rows show the overlap between user-defined and ground truth maps with respect to the UIs. Green areas are correctly overlapping areas, while red areas are wrongly defined areas by the user. The blue contour surrounds the union set of the user-defined and ground truth virtual borders. If the green area fills out the blue contour well, it results in a high accuracy. (Color figure online)

union area well, and that there are few yellow and red cells. There are mainly two reasons for the high accuracy: (1) The indirect approaches (marker and laser pointer) are based on robot guidance and leverage the small localization error of the robot. Thus, one source of error is eliminated, and the small error is only caused by the interaction between human and robot. (2) The small error of the Tango device is the result of the direct interaction with the physical environment using AR techniques. Furthermore, the on-board visual-inertial odometry is highly accurate allowing robust pose tracking of the Tango device. A more detailed look at the results of the Tango approach reveals that the ground truth and the user-defined areas only differ by a rotational transformation. This clearly indicates a constant registration error between the tablet's and the map's coordinate frame. If this error could be fixed in future, the accuracy results would be even better. In contrast to these approaches, the accuracy of the teaching method based on GUI drawing is less accurate. This is due to the lack of correspondence between the physical environment and the OGM shown on the tablet. Hence, it is hard for the user to relate physical positions in the environment to coordinates on the tablet's GUI. We refer this lack to as *correspondence problem*, and we assume that this problem will be even more meaningful if the mapped physical environment becomes larger and featureless. Then it would be even harder for the user to establish a robust correspondence between the OGM and physical environment resulting in a drop of accuracy.

These qualitative results are underlined by the quantitative accuracy results presented in Fig. 7. The figure

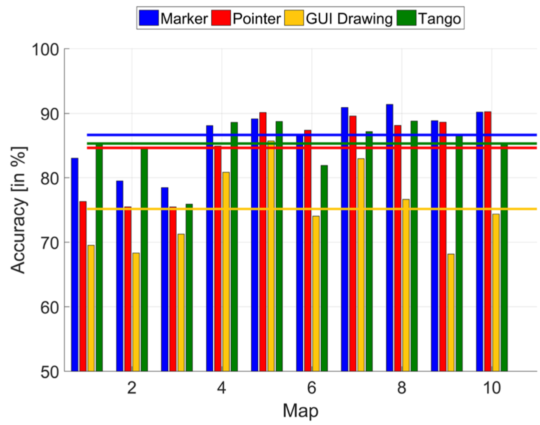


Fig. 7 Quantitative accuracy results for the different UIs dependent on the maps in the dataset. A horizontal line represents the average over all maps

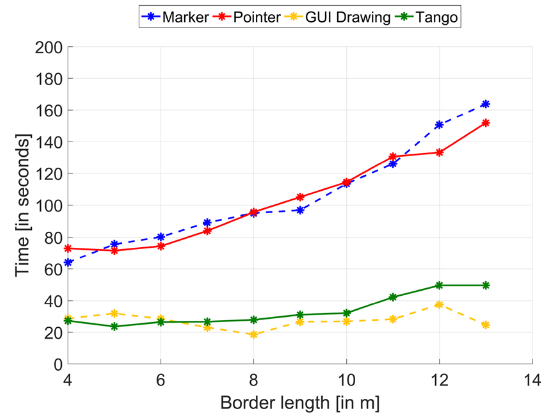


Fig. 8 Teaching time for the different UIs dependent on the border length

Table 1 Speedup of the GUI approach with respect to the other UIs

Baseline	Border length (in m)										
	4	5	6	7	8	9	10	11	12	13	Avg.
Marker	2.2	2.4	2.8	3.9	5.1	3.6	4.2	4.5	4.0	6.7	3.9
Pointer	2.6	2.3	2.6	3.7	5.1	3.9	4.3	4.6	3.6	6.2	3.9
Tango	1.0	0.7	0.9	1.2	1.5	1.2	1.2	1.5	1.3	2.0	1.2

shows the JSI averaged on five runs per map and UI as bars. The overall average per approach is indicated by a horizontal line. As identified before, the GUI drawing approach has a small accuracy (75.2%), while the other three approaches feature a high accuracy (marker: 86.6%, pointer: 84.6%, Tango: 85.3%). The last three results were already reported in [8].

5.6 Teaching Effort

The teaching effort is an indicator for the usability of a system. The smaller the teaching effort is, the higher is the acceptance of the user. We consider the time to teach a virtual border as a measure to assess the effort.

5.6.1 Procedure

While conducting the previously described accuracy experiments, we measured the time for each of the 50 runs per UI. The teaching time is the time between the definition of the first virtual border point and the successful creation of the posterior map.

5.6.2 Results

Figure 8 shows the teaching time dependent on the border length for the different UIs. The results underline that there are two groups of teaching methods: direct and indirect ones. Indirect teaching methods, in particular the approaches based on visual markers and a laser pointer, entail a significantly higher teaching effort than the direct teaching methods, especially the interaction with the GUI and Tango tablet. Furthermore, the visualization shows that the teaching time of the indirect approaches linearly depends on the border length. This is due to the nature of the teaching framework based on robot guidance. The longer a virtual border is, the longer it takes the robot to move along the desired virtual border. In contrast to the indirect approaches, the direct approaches similarly feature a linear relationship but with a smaller gradient. Thus, the teaching time increases slower compared to the indirect approaches making the direct approaches more attractive for non-expert users. This significant difference between both groups is caused by the direct interaction with the map or the environment, while an indirect interaction based on robot guidance leads to an overhead of the teaching time. The GUI-based approach has the smallest teaching effort on average. Table 1 shows the speedup of this approach with respect to the other three UIs. While the GUI approach is 1.2 times faster than the Tango approach, there is even an average speedup of 3.9 with respect to both indirect approaches.

5.7 User Experience

The user experience is a crucial factor for the acceptance of a UI and teaching method. Therefore, we evaluated different aspects, such as intuitiveness, comfort, learnability and feedback, for each UI.

5.7.1 Procedure

In order to evaluate the user experience, we conducted a user study with a total of 25 participants (18 males, 7 females; ages 16–56). The mean age was $M = 31.92$ years with a standard deviation of $SD = 11.54$ years. Participants rated their robotic skills on an 11-point Likert scale ($M = 3.44$, $SD = 3.20$). Each participant entered the lab environment and was briefly introduced to the study by the experimenter. Afterwards, one of the four UIs was randomly selected to avoid order effects, and the experimenter explained and demonstrated the use of the teaching method and UI. Subsequently, the participant got some time to get familiar with the UI before he or she was asked to define a virtual border around a fixed-placed carpet. Simultaneously, the experimenter measured the time needed for the definition of the virtual borders, saved the resulting posterior map and protocolled the success. The success indicates if the participant could correctly handle the UI and was able to specify all components of a virtual border V. This procedure was repeated for every UI. Hence, every participant could compare the different teaching methods and UIs. After practically evaluating the four UIs, the participant was asked to rate the user experience on 5-point Likert scales in a post-study questionnaire. The questionnaire consisted of the following statements:

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

Additionally, the participant was asked which UI he or she prefers for the given task. The participant could give multiple responses allowing the selection of none, one or more UIs. Finally, the participant had the possibility to give comments on the statements or reasons for a rating. The whole experiment including the practical application of the UIs and the answering of the questionnaire took between 20 and 30 min per participant.

5.7.2 Results: Questionnaire

The answers to the questionnaire are presented in Fig. 9 showing the mean ratings and standard deviations per statement. Since we used a within-subjects design allowing the comparison of different UIs by each participant, we performed a repeated measures analysis of variance (ANOVA) with a significance level of $\alpha = 0.05$ on the statements S1–S6. The UI was the independent and the corresponding user rating the dependent variable. $F(df_1, df_2)$ denotes the F-distribution

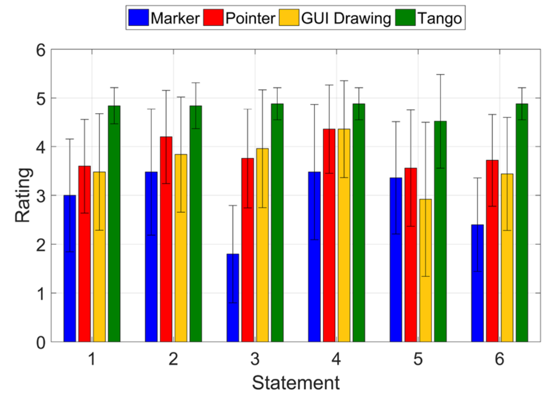


Fig. 9 Results (mean and standard deviation) of the questionnaire on a 5-point Likert scale per statement

with its two parameters df_1 and df_2 that depend on the number of groups and measurements. We found statistically significant differences for all statements:

- S1: $F(3, 72) = 17.12$, $p < 0.001$
- S2: $F(3, 72) = 9.10$, $p < 0.001$
- S3: $F(2.02, 48.49) = 52.84$, $p < 0.001$
- S4: $F(2.05, 49.29) = 10.48$, $p < 0.001$
- S5: $F(1.70, 40.83) = 8.16$, $p = 0.002$
- S6: $F(2.15, 51.61) = 36.12$, $p < 0.001$

A Greenhouse–Geisser adjustment was used for statements S3–S6 to correct violations of sphericity. A Bonferroni-adjusted post-hoc analysis showed that users have significantly ($p < 0.001$) less problems to define virtual borders using a Tango tablet compared to the other UIs. Furthermore, users find the Tango approach more intuitive compared to the marker ($p = 0.001$) and GUI ($p = 0.007$) approach, but there is no significant difference compared to the laser pointer approach ($p = 0.050$). In case of the comfort, visual markers are significantly ($p < 0.001$) less comfortable compared to the other UIs. The Tango tablet is rated to be the most comfortable UI for the given task significantly differing from the marker, laser pointer ($p < 0.001$) and GUI ($p = 0.04$) interface. Additionally, the teaching method with the Tango tablet features the highest learnability ($M = 4.88$, $SD = 0.33$) with a significant difference compared to the marker ($p < 0.001$) and laser pointer ($p = 0.039$), but no significant difference ($p = 0.122$) compared to the GUI approach. The AR application on the Tango tablet also causes the highest rating for the feedback system ($M = 4.52$, $SD = 0.96$) which is significantly different from the marker ($p = 0.012$), laser pointer ($p = 0.045$) and GUI ($p < 0.001$) approach. The overall satisfaction is also leaded by the Tango tablet ($M = 4.88$, $SD = 0.33$) followed by the laser pointer ($M = 3.72$, $SD = 0.94$), GUI ($M = 3.44$, $SD = 1.16$) and marker approach ($M = 2.40$, $SD = 0.96$). These results

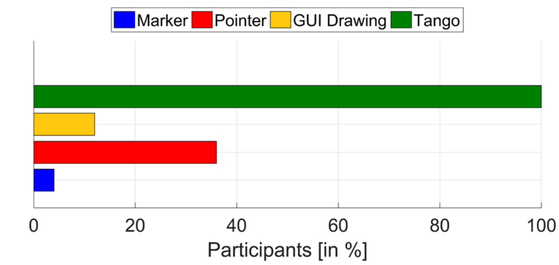


Fig. 10 Overall preferences of the participants for a UI

are confirmed by the preferences of the participants shown in Fig. 10. All participants selected the Tango tablet for the given task followed by the laser pointer approach (9 out of 25). Participants often suggested to use a stick attached with a marker to increase the user experience of the marker approach. Note that, for reasons of readability, we mean statistically significant differences when we report a significant difference with a p value.

5.7.3 Results: Experimenter's Protocol

Additional to the answers of the questionnaire, the experimenter documented the success rate, accuracy and teaching time for each participant and UI. The averaged results are shown in Table 2. All participants could successfully integrate the virtual borders using the direct teaching methods, while only 80% (laser pointer) and 84% (marker) of the participants were successful using one of the indirect teaching methods. It was observed that these participants had problems to specify the seed point $\boldsymbol{s}$ by rotating the mobile robot around its vertical axis.

The quantitative results for the accuracy and teaching effort shown in Table 2 were obtained by a group of non-experts (25 participants of the user study) defining a single virtual border for each UI for the first time. In contrast to that, the accuracy and teaching effort evaluation presented in Sects. 5.5 and 5.6 was the result of a single non-expert defining multiple virtual borders of an extensive dataset. Thus, this non-expert also had some experience defining virtual borders and employing the UIs. The quantitative accuracy results discussed here show that the indirect and Tango tablet approaches feature a similar accuracy (63.2–68.3%), whereas the accuracy of the GUI approach dramatically drops to 38.2%. This can be explained by the previously identified correspondence problem between coordinates on the tablet's display and in the physical environment. Compared to the accuracy results based on the dataset evaluation presented in Sect. 5.5, these results are lower (pointer: − 16.3%–GUI: − 37.0%). This is due to the learning effect. The former results were performed by a non-expert who got familiar with the UI and teaching method, while the results in this section are results from participants teaching virtual bor-

Table 2 Additional measures evaluated in the user study and their differences to the dataset evaluation performed by a single non-expert

	Success rate (%)	Accuracy	Effort
Marker	84	63.2% (− 23.4%)	132 s (+ 47 s)
Pointer	80	68.3% (− 16.3%)	112 s (+ 33 s)
GUI	100	38.2% (− 37.0%)	29 s (+ 1 s)
Tango	100	65.6% (− 19.7%)	39 s (+ 12 s)

ders for the first time. Thus, experience in using the UI and teaching method increases the accuracy. Similarly, teaching takes longer for the indirect teaching methods compared to the results based on a non-expert who got familiar with the UI (marker: + 47 s and pointer: + 33 s). But the teaching time for the direct approaches only differs minimally (GUI: + 1 s and Tango: + 12 s). Therefore, it is easier to learn the handling of the direct teaching methods.

5.8 Discussion

All teaching methods and UIs have in common that they are based on a generic algorithm, and we showed that this algorithm is correct and flexible. Hence, a user can define arbitrary virtual borders, and the mobile robot changes its navigational behavior respecting the virtual borders. Nonetheless, we assess the flexibility of the indirect teaching methods worse than the flexibility of the direct teaching methods because the mobile robot can be restricted by physical obstacles during guidance. In terms of accuracy, the approaches based on visual markers, a laser pointer and a Tango tablet have an equally high accuracy (84.6–86.6%). Only the approach of drawing on a GUI has a lower accuracy (75.2%) due to the identified correspondence problem between coordinates on the tablet's display and in the physical environment. Direct teaching methods, i.e. the GUI and Tango approach, feature a lower teaching effort (0.1–2.9 s/m) compared to the indirect teaching methods, i.e. the marker and laser pointer approach (9.2–10.5 s/m). The values are gradients obtained from linear regression on the data points. This difference is caused by the direct interaction with the map or environment and the nature of the teaching framework based on robot guidance. Both criteria, accuracy and teaching effort, were evaluated based on data obtained by a single non-expert performing experiments on an extensive dataset and by a group of non-experts defining a single virtual border for each UI. We made this distinction due to practical reasons: it is not reasonable to perform extensive experiments on a comprehensive dataset with multiple participants since it would take several hours per participant. With this method, we could evaluate the accuracy and teaching effort extensively on a large dataset but also got results of multiple non-experts that we compared with each other. Although the results of the single non-expert were better due

Table 3 Comparative summary of the experimental evaluation concerning the evaluation criteria

	Correctness	Flexibility	Accuracy (%)	Effort (s/m)	Intuitiveness	Comfort	Learnability	Feedback
Marker	Yes	Medium	86.6	10.5	3.48	1.80	3.48	3.36
Pointer	Yes	Medium	84.6	9.2	4.20	3.76	4.36	3.56
GUI drawing	Yes	High	75.2	0.1	3.84	3.96	4.36	2.92
Tango	Yes	High	85.3	2.9	4.84	4.88	4.88	4.52

to his experience in handling the UIs in the teaching process (see Table 2), both evaluations reveal the same characteristics, i.e. (1) drawing on a GUI is less accurate than the other approaches and (2) direct teaching methods are less time consuming than indirect ones. Furthermore, experience in using the UIs in the teaching process increases accuracy and decreases the teaching effort. In case of intuitiveness, the Tango tablet is the most intuitive ($M = 4.84$) UI which can be explained by the combination of a common tablet and an AR application. Moreover, direct teaching methods are more comfortable compared to the indirect approaches since users can directly interact with the map or environment. The marker approach is the least comfortable one ($M = 1.80$) which is due to the realization: the user has to bend down while guiding the mobile robot. This can be improved in the future by employing a stick attached with the markers as suggested by participants of the user study. The teaching method based on the Tango tablet is also the UI that is rated with the highest learnability ($M = 4.88$). The laser pointer and GUI approach feature the same learnability ($M = 4.36$) because both UIs are known from everyday life which makes learning easier compared to the marker approach ($M = 3.48$). Finally, the Tango tablet is the best-rated UI in terms of the provided feedback ($M = 4.52$). The AR application directly provides visual feedback to the user about the virtual borders. Both indirect teaching methods have a similar feedback rating (marker: $M = 3.36$, pointer: $M = 3.56$) because of the same feedback system. The feedback system of the GUI approach got the worst rating ($M = 2.92$) since there is no real feedback for the user underlining the correspondence problem. Overall, the Tango tablet is the best UI in terms of user experience. This can be explained by the combination of a popular consumer tablet, that is known to non-expert users, and the user-friendliness of AR applications. We summarized the results in Table 3 for ease of comparison.

6 Conclusion and Future Work

Virtual borders are a possibility to interactively and flexibly modify the workspace of a 3-DoF mobile robot, such as a vacuum cleaning or service robot. This allows non-expert users to easily change the navigational behavior of their mobile robots according to their needs and ensure a human-aware robot navigation. The opportunity is especially interesting for non-experts living in a human–robot shared space, e.g. home environment. We evaluated four different UIs and teaching methods to optimally address the needs of non-expert users for this task. All teaching methods are based on the same generic algorithm for incorporating virtual borders into prior maps of the environment. We proved the algorithm to be correct in terms of integrating the virtual borders into a given prior map and changing the robot's navigational behavior. Moreover, the algorithm is flexible allowing users to define arbitrary virtual borders. However, we assessed the flexibility of the direct teaching methods better than the flexibility of the indirect ones since they can be limited by physical obstacles during robot guidance. The experimental results showed that the indirect teaching methods based on visual markers and a laser pointer have a high accuracy but also a higher teaching effort compared to the direct teaching methods based on drawing on a GUI and interacting with a Tango tablet. We also identified a correspondence problem when using the GUI approach. Almost all users had a problem to relate 2D positions on the tablet to coordinates in the physical environment. Thus, the accuracy decreases significantly compared to the other UIs. The best compromise of flexibility, accuracy, teaching effort and user experience is given by the RGB-D Google Tango tablet as UI. It features a high flexibility and accuracy, low teaching effort and is preferred by all participants of the user study leading to the highest ratings for intuitiveness, comfort, learnability and feedback. Especially, the direct interaction with the environment using AR techniques makes it favorable for the task of teaching virtual borders.

Therefore, in the future we plan to follow up on the RGB-D Google Tango approach with AR application making the teaching process more intelligent. Currently, the definition of the virtual borders only relies on sole HRI. We plan to learn from the users' interactions and to use this knowledge in a recommendation system suggesting possible virtual borders to the user, e.g. through automatic edge detection. This aims to further increase the accuracy and to decrease the teaching effort by supporting the user in the teaching process.

Acknowledgements We would like to thank all participants of the user study for their time and valuable feedback.

Funding This study was funded by the German Federal Ministry of Education and Research (Grant Number 03FH006PX5).

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Hess J, Beinhofer M, Burgard W (2014) A probabilistic approach to high-confidence cleaning guarantees for low-cost cleaning robots. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 5600–5605
2. Gross HM, Mueller S, Schroeter C, Volkhardt M, Scheidig A, Debes K, Richter K, Doering N (2015) Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5992–5999
3. Biswas J, Veloso M (2016) The 1,000-km challenge: insights and quantitative and qualitative results. IEEE Intell Syst 31(3):86–96
4. Sprute D, Pörtner A, Rasch R, Battermann S, König M (2017) Ambient assisted robot object search. In: Enhanced quality of life and smart living: 15th international conference on smart homes and health telematics, pp 112–123
5. Moravec H, Elfes A (1985) High resolution maps from wide angle sonar. In: 1985 IEEE international conference on robotics and automation. Proceedings, pp 116–121
6. Sprute D, Rasch R, Tönnies K, König M (2017a) A framework for interactive teaching of virtual borders to mobile robots. In: 2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN), pp 1175–1181
7. Sprute D, Tönnies K, König M (2017b) This far, no further: introducing virtual borders to mobile robots using a laser pointer. ArXiv:1708.06274 (currently under review)
8. Sprute D, Tönnies K, König M (2018) Virtual borders: Accurate definition of a mobile robot's workspace using augmented reality. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 8574–8581
9. Sisbot EA, Marin-Urias LF, Broquère X, Sidobre D, Alami R (2010) Synthesizing robot motions adapted to human presence. Int J Soc Robot 2(3):329–343
10. Rios-Martinez J, Spalanzani A, Laugier C (2015) From proxemics theory to socially-aware navigation: a survey. Int J Soc Robot 7(2):137–153
11. Lindner F (2015) A conceptual model of personal space for human-aware robot activity placement. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5770–5775
12. Mead R, Matarić MJ (2017) Autonomous human–robot proxemics: socially aware navigation based on interaction potential. Auton Robots 41(5):1189–1201
13. Tipaldi GD, Arras KO (2011) Please do not disturb! minimum interference coverage for social robots. In: 2011 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1968–1973
14. Pandey AK, Alami R (2010) A framework towards a socially aware mobile robot motion in human-centered dynamic environment. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5855–5860
15. Yamaoka F, Kanda T, Ishiguro H, Hagita N (2010) A model of proximity control for information-presenting robots. IEEE Trans Robot 26(1):187–195
16. Ramrez OAI, Khambhaita H, Chatila R, Chetouani M, Alami R (2016) Robots learning how and where to approach people. In: 2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN), pp 347–353
17. Zender H, Jensfelt P, Kruijff GJM (2007) Human- and situation-aware people following. In: RO-MAN 2007—the 16th IEEE international symposium on robot and human interactive communication, pp 1131–1136
18. Ferrer G, Zulueta AG, Cotarelo FH, Sanfeliu A (2017) Robot social-aware navigation framework to accompany people walking side-by-side. Auton Robots 41(4):775–793
19. Kruse T, Pandey AK, Alami R, Kirsch A (2013) Human-aware robot navigation: a survey. Robo Auton Syst 61(12):1726–1743
20. Truong XT, Ngo TD (2016) Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. Int J Soc Robot 8(5):663–684
21. Hansen ST, Svenstrup M, Andersen HJ, Bak T (2009) Adaptive human aware navigation based on motion pattern analysis. In: RO-MAN 2009—the 18th IEEE international symposium on robot and human interactive communication, pp 927–932
22. Alempijevic A, Fitch R, Kirchner N (2013) Bootstrapping navigation and path planning using human positional traces. In: 2013 IEEE international conference on robotics and automation (ICRA), pp 1242–1247
23. Kim B, Pineau J (2016) Socially adaptive path planning in human environments using inverse reinforcement learning. Int J Soc Robot 8(1):51–66
24. Charalampous K, Kostavelis I, Gasteratos A (2017) Recent trends in social aware robot navigation. Robot Auton Syst 93:85–104
25. O'Callaghan ST, Singh SPN, Alempijevic A, Ramos FT (2011) Learning navigational maps by observing human motion patterns. In: 2011 IEEE international conference on robotics and automation (ICRA), pp 4333–4340
26. Hall E (1969) The hidden dimension: man's use of space in public and private. Doubleday Anchor Books, Bodley Head, London
27. Sakamoto D, Honda K, Inami M, Igarashi T (2009) Sketch and run: a stroke-based interface for home robots. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, CHI '09, pp 197–200
28. Chiu TY (2011) Virtual wall system for a mobile robotic device. European patent application EP000002388673A1
29. Neato (2017) Boundary markers for Neato Botvac series. https://www.neatorobotics.com/accessories/botvac/boundary-markers-for-neato-robot-vacuums/. Access: 12/17
30. Tölgyessy M, Dekan M, Duchoň F, Rodina J, Hubinský P, Chovanec L (2017) Foundations of visual linear human–robot interaction via pointing gesture navigation. Int J Soc Robot 9(4):509–523
31. Kemp CC, Anderson CD, Nguyen H, Trevor AJ, Xu Z (2008) A point-and-click interface for the real world: laser designation of objects for mobile manipulation. In: Proceedings of the 3rd ACM/IEEE international conference on human robot interaction, HRI '08, pp 241–248
32. Ishii K, Zhao S, Inami M, Igarashi T, Imai M (2009) Designing laser gesture interface for robot control. Human–computer interaction—INTERACT 2009, pp 479–492
33. Choi YS, Anderson CD, Glass JD, Kemp CC (2008) Laser pointers and a touch screen: intuitive interfaces for autonomous mobile manipulation for the motor impaired. In: Proceedings of the 10th international ACM SIGACCESS conference on computers and accessibility, ASSETS '08, pp 225–232
34. Chen TL, Kemp CC (2010) Lead me by the hand: evaluation of a direct physical interface for nursing assistant robots. In: 2010 5th

ACM/IEEE international conference on human–robot interaction (HRI), pp 367–374
35. Jevtic A, Doisy G, Parmet Y, Edan Y (2015) Comparison of interaction modalities for mobile indoor robot guidance: direct physical interaction, person following, and pointing control. IEEE Trans Hum Mach Syst 45(6):653–663
36. Marques C, Cristoao J, Lima P, Frazao J, Ribeiro I, Ventura R (2006) RAPOSA: semi-autonomous robot for rescue operations. In: 2006 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3988–3993
37. Hebert P, Ma J, Borders J, Aydemir A, Bajracharya M, Hudson N, Shankar K, Karumanchi S, Douillard B, Burdick J (2015) Supervised remote robot with guided autonomy and teleoperation (surrogate): a framework for whole-body manipulation. In: 2015 IEEE international conference on robotics and automation (ICRA), pp 5509–5516
38. Garrido-Jurado S, Munoz-Salinas R, Madrid-Cuevas F, Marín-Jiménez M (2014) Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognit 47(6):2280–2292
39. Sprute D, Rasch R, Pörtner A, Battermann S, König M (2018) Gesture-based object localization for robot applications in intelligent environments. In: 2018 international conference on intelligent environments (IE), pp 48–55
40. Rouanet P, Oudeyer PY, Danieau F, Filliat D (2013) The impact of human–robot interfaces on the learning of visual objects. IEEE Trans Robot 29(2):525–541
41. Vaughan J, Kratz S, Kimber D (2016) Look where you're going: visual interfaces for robot teleoperation. In: 2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN), pp 273–280
42. Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans Robot 23(1):34–46
43. Quigley M, Conley K, Gerkey B, Faust J, Foote TB, Leibs J, Wheeler R, Ng AY (2009) ROS: an open-source robot operating system. In: ICRA workshop on open source software

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dennis Sprute is a research assistant at the Bielefeld University of Applied Sciences, Germany and Ph.D. student at the Otto-von-Guericke University Magdeburg, Germany. He received a B.Sc. and M.Sc. degree in Computer Science from Bielefeld University of Applied Sciences in 2014 and 2015. His research interests are in the fields of human–robot interaction and human-aware robot navigation in human-centered environments. Currently, he is involved in a research project dealing with mobile service robots in smart home environments.

Klaus Tönnies is Professor for Computer Vision at the Department of Computer Science of the Otto-von-Guericke-University Magdeburg, Germany. He received his Diploma degree in 1983 and his Ph.D. degree in 1987 from the Department of Computer Science of the Technical University of Berlin. His research interests are in the field of model-driven computer vision with applications in engineering and medical image analysis.

Matthias König is Professor for Embedded Software Engineering at the Bielefeld University of Applied Sciences, Germany. He holds a Diploma degree in Computer Science from the University of Paderborn, a Master degree in Business Administration from the Open University, a PhD degree in Computer Science from the Otto-von-Guericke University Magdeburg and a Ph.D. degree in Management Science from the University of Lancaster. His research interests include image processing, computer vision, embedded systems and robotics.