Virtual Borders: Accurate Definition of a Mobile Robot’s Workspace
Using a RGB-D Google Tango Tablet

Dennis Sprute¹,², Klaus Tönies² and Matthias König¹

Abstract—We address the problem of interactively controlling the workspace of a mobile robot to ensure a human-aware navigation. This is especially of relevance for non-expert users living in human-robot shared spaces, e.g. home environments, since they want to keep the control of their mobile robots, such as vacuum cleaning or companion robots. Therefore, we introduce virtual borders that are respected by a robot while performing its tasks. For this purpose, we employ a RGB-D Google Tango tablet as human-robot interface to flexibly specify virtual borders in the environment. We evaluated our system concerning correctness, accuracy and teaching effort, and compared the results with other baseline methods. Our method features an equally-high accuracy while reducing the teaching effort by a factor of 3.1 compared to the baseline.

I. INTRODUCTION

Humans and robots increasingly live together in shared spaces, such as a home environments. Robots support the residents in their everyday life, e.g. as household or companion robots, and people appreciate the help of robots. But from our experience, we know that there are sometimes areas that should not be entered by a robot. These can be social places, e.g. bathrooms or bedrooms, that should be avoided by the robot due to privacy concerns. Another use case is the accurate definition of the workspace of a mobile vacuum or mopping robot to operate in certain areas. Therefore, non-expert users need the ability to interactively and easily control a mobile robot’s workspace to address this challenge.

For this purpose, we propose virtual borders that are not directly visible to the user but indicate occupied areas to the robot. These are respected by the mobile robot while performing its task. We want to address the question of how to allow non-expert users to flexibly teach virtual borders to their robots and change their navigational behavior accordingly. This teaching method needs to allow accurate border teaching while featuring little effort. Additionally, a feedback system giving information about learned virtual borders is desirable. In this context, we refer a non-expert to as a person that (1) has no programming skills, (2) no experience with robotics and its insights, (3) has no cognitive impairments or upper limb disorders, but (4) has experiences with common consumer products, such as tablets or smartphones. Moreover, a non-expert (5) prefers a robust and feature-complete system to a highly sophisticated and non-intuitive one.

¹The authors are with Campus Minden, Bielefeld University of Applied Sciences, 32427 Minden, Germany
²The authors are with the Faculty of Computer Science, Otto-von-Guericke University Magdeburg, 39106 Magdeburg, Germany

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

Fig. 1: A user restricts the mobile robot’s workspace using a RGB-D Google Tango tablet by specifying an area around a carpet. The robot avoids this area while working, and the tablet’s display provides direct visual feedback to the user.

Several human-robot interfaces are imaginable for the teaching process, e.g. approaches using simple GUIs, remote controllers, smartphones or tablets, direct physical interaction with the robot or pointing gestures (with or without auxiliary device). In order to address the above-mentioned requirements optimally, we propose a teaching method employing a RGB-D tablet, such as a Google Tango device, to interact with the robot. We chose a Tango tablet for several reasons: (1) a high-accurate onboard visual-inertial odometry allows robust 6-DoF pose tracking of the device, (2) tablet and smartphone interfaces are well established which makes them attractive for non-experts, (3) first commercial smartphones are shipped with Google Tango technology making them accessible for a wider user group, (4) no additional devices (robot or cameras) are necessary for teaching and (5) the integrated display allows direct visual feedback to the user.

Fig. 1 shows a user with a Google Tango tablet excluding the carpet area from the mobile robot’s workspace. After completing the teaching process, the robot will not cross the carpet area while navigating in the environment. Our main contribution is a teaching method leveraging a RGB-D device to allow non-experts the flexible definition of a 3-DoF robot’s workspace. Such a method is especially interesting for robot navigation in human-centered environments. To the best of our knowledge, this is the first time a RGB-D tablet or smartphone is used for human-robot interaction.

The remainder of this paper is structured as follows: in the next section, we give an overview of related work concerning the topic before we formally define the problem. Subsequently, we give details about our proposed teaching method.
based on a RGB-D device. We also evaluated the proposed method concerning correctness, accuracy and teaching effort and compared the results with selected baseline methods. These experimental results are presented in the following section. Finally, we summarize our method and point out work for the future.

II. RELATED WORK

There are different types of maps that differ in their way they model the environment, e.g. metric maps represent geometric properties of the environment. A typical representative of this category are occupancy grid maps (OGM) [1] that are widely used in robot navigation and path planning. They model the environment by means of cells containing a probability for the occupancy of the corresponding area. In order to create an OGM of an environment and localize the robot with respect to it, Simultaneous Localization and Mapping (SLAM) algorithms [2] are widely established. Cadena et al. [3] give a comprehensive overview of the evolution of SLAM from the past to the future.

Along the occupancy information modeled in an OGM, maps can contribute additional information, such as semantics [4] or social information [5]. Especially social information can be used with the purpose of changing the robot’s navigational behavior in human-centered environments, e.g. O’Callaghan et al. [6] incorporate motion patterns of people into the robot’s trajectories and Alempijevic et al. [7] jointly learn a map from robots’ sensor measurements and human trajectories as basis for path planning. Other works use social costmaps built from sensor measurements to realize a human-aware navigation [8], integrate social norms into the costmap to change the way a robot approaches a human [9] and propose human motion maps to represent the distribution of human motion in a map [10]. A survey on recent trends in social aware robot navigation and a historical overview is given by Charalampos et al. [11].

These implicit approaches to change the robot’s navigational behavior are based on observations. They are user-friendly because no explicit interaction is necessary, but they are not flexible enough for the problem addressed in this work. We argue that teaching of arbitrary virtual borders can only be accomplished through explicit user interaction. Examples for this category are the GUI-based user interface to sketch the area for a vacuum cleaning robot [12] and a virtual wall systems based on beacon devices [13]. The first approach needs several top-view cameras installed in the environment to stream images to the user’s display, while the latter is restricted by the conic beam of the beacon devices. These only allow to block certain areas using a straight line while consuming power and being intrusive. Magnetic stripes placed on the ground known from commercial vacuum cleaning robots are intrusive as well. To address these aspects of intrusiveness, power-consumption and small flexibility, Sprute et al. [14] proposed a framework for interactive teaching of virtual borders and an implementation based on visual markers. Furthermore, they use a laser pointer as human-robot interface to guide the robot along the virtual border [15]. Although both approaches are flexible and allow teaching of arbitrary virtual borders, they do not provide an inherent feedback system. Thus, the user cannot directly notice the learned virtual borders.

To purposely address this lack and for the reasons mentioned in the introductory part, we chose a RGB-D Google Tango device as interaction device. It has been used in several robotics-related applications, e.g. indoor-localization given a 2D floor plan [16] and real-time 3D reconstruction [17]. Other use cases include optimization of SLAM by text spotting [18] or controlling of a quadrotor equipped with a Tango smartphone [19]. These applications show the potential of mobile RGB-D devices in the context of robotics.

III. PROBLEM STATEMENT

Before we give details on the proposed teaching method, we introduce the notation we use throughout the paper and formally define the problem of interactively manipulating an OGM using a RGB-D device. It is the goal to change the robot’s navigational behavior in future tasks according to the users’ needs. An OGM models the physical environment in terms of cells containing probabilities for the occupancy of the corresponding area. \( M(x, y) \in [0, 1] \) denotes the occupancy probability for the cell \((x, y)\) in the map \( M \). Furthermore, we define all possible coordinates \((x, y)\) as the domain of the map \(\Omega(M) \subseteq \mathbb{R}^2\). At the beginning, an OGM of the physical environment \( M_{\text{prior}} \) containing walls and furniture is given. Due to the iterative nature of the teaching method, \( M_{\text{prior}} \) can also contain virtual borders from previous teaching processes. Since we want to integrate virtual borders into the map, the user defines a manipulation so that \( M_{\text{prior}} \rightarrow M_{\text{posterior}} \). This posterior map \( M_{\text{posterior}} \) contains the physical environment as well as the user-defined virtual borders and can be used for navigation and path planning.

IV. TEACHING USING A RGB-D DEVICE

We propose a teaching method based on a RGB-D tablet to address the problem of interactive teaching of virtual borders and changing the mobile robot’s navigational behavior accordingly. Therefore, a person uses a Google Tango tablet to move around in the environment and select points on the ground plane by interacting with the mobile device. The tablet simultaneously acts as feedback device showing an augmented live video of its onboard camera. The user-defined virtual borders are integrated into the prior map of the environment to ensure a human-aware navigation.

A. Requirements

In order to realize this behavior, the robot as well as the Tango tablet need to have access to the same global OGM \( M_{\text{prior}} \) that was created in advance. For this purpose, we relate the relevant coordinate frames to each other as shown in Fig. 2. We assume the Tango device to be localized within the environment. The origin of this previously constructed environment is the \( ADF \) (Area Description File) coordinate frame. When starting the teaching application, the
Fig. 2: Relevant coordinate frames and their relations shown as black dotted lines. The red line depicts an illustrative virtual border polygon \( \mathcal{P} \).

*SoS* (Start of Service) coordinate frame marks the current pose of the Tango device. While localizing in the environment employing visual features, the transformation between SoS and ADF is established. The Tango device uses its accurate onboard visual-inertial odometry to keep track of its current pose *Device* with respect to *SoS*. The ADF coordinate frame is manually related to the *Map* coordinate frame once a visual model of the environment is learned. Finally, the dynamic pose of the mobile robot *Robot* is related to the *Map* frame using adaptive Monte Carlo localization (AMCL) \([20]\) using the robot’s laser scanner. This ensures transformations between all relevant coordinate frames. All of these transformations belong to \( SE(3) \).

### B. Area Definition

We define a virtual border as a triple \( V = (\mathcal{P}, s, \delta) \) where each component is specified in the interactive teaching process using the RGB-D device. It is capable of perceiving a 3D point cloud of its environment that is used to specify \( n \) points \( \mathbf{p}_i \in \mathbb{R}^3, 1 < i < n \) on the ground plane. Only points on the ground plane are interesting because the mobile robot operates in the plane. By transforming these points into the *Map* coordinate frame, we obtain \( n \) points \( \mathbf{p}_i \in \mathbb{R}^2 \) building a polygonal chain \( \mathcal{P} \):

\[
\mathcal{P} = \bigcup_{i=1}^{n-1} [\mathbf{p}_i \mathbf{p}_{i+1}],
\]

(1)

with

\[
[\mathbf{p}_i \mathbf{p}_{i+1}] = \{(1 - \lambda)\mathbf{p}_i + \lambda\mathbf{p}_{i+1} \mid \lambda \in [0, 1]\}
\]

(2)

being a line segment between two points. We distinguish between simple and closed polygonal chains to define arbitrary areas in the environment as polygons or separating curves.

Additionally, the user employs the RGB-D device to select a seed point \( s \in \mathbb{R}^3 \) that indicates the area to be manipulated. The corresponding cell in the global OGM is denoted as \( s^* \in \Omega(M_{\text{prior}}) \). Finally, the user has the possibility to specify the occupancy probability \( \delta \in [0, 1] \) for the area indicated by \( s \).

### C. Map Creation

After defining a virtual border \( V \), we use the polygonal chain \( \mathcal{P} \) to partition the map into two areas:

\[
A_c = \{ c \in \Omega(M_{\text{prior}}) \mid c \text{ connected to } s^* \},
\]

(3)

which is the area that is directly connected to the cell corresponding to the seed point \( s^* \) and

\[
A_{nc} = \Omega(M_{\text{prior}}) - A_c,
\]

(4)

which is the complementary area containing coordinates disconnected from the seed point \( s^* \). Two cells \( a \in \Omega(M) \) and \( b \in \Omega(M) \) in a map \( M \) are connected if:

\[
\exists f : [0, 1] \to \Omega(M) : f(0) = a, f(1) = b,
\]

\[
\forall i, j \in [0, 1] : M(f(i)) = M(f(j))
\]

(5)

where \( f \) is a continuous mapping.

If the border polygon \( \mathcal{P} \) is a simple polygonal chain, we linearize the first \([\mathbf{p}_1 \mathbf{p}_2]\) and the last \([\mathbf{p}_{n-1} \mathbf{p}_n]\) line segments to partition the map. An example for such a simple polygonal chain and its linearization is shown in Fig. 3b where the green line indicates the actual polygon \( \mathcal{P} \) and the black dots the resulting occupied space. The system automatically extends the virtual border to the borders of the prior map \( M_{\text{prior}} \). This allows the user to easily exclude big areas from the robot’s workspace with a single curve or line. Finally, we construct the posterior map \( M_{\text{posterior}} \), dependent on the given prior map \( M_{\text{prior}} \) and the components of the virtual border \( V \) as follows:

\[
M_{\text{posterior}}(x, y) = \begin{cases} 
\delta & \text{if } (x, y) \in A_c \\
M_{\text{prior}}(x, y) & \text{if } (x, y) \in A_{nc}
\end{cases}
\]

(6)

By iterating this teaching process \( N \) times and defining a sequence of virtual borders \( V^* = \{V_1, V_2, ..., V_N\} \), the user can define arbitrary virtual borders in the environment. This allows the flexible definition of a mobile robot’s workspace.

### D. Interaction & Feedback

The user only needs the Google Tango tablet to specify a virtual border \( V \) with all its components. The person moves around in the environment with the tablet, and selects virtual border points \( \mathcal{P} \) by pointing the device towards the desired points on the ground plane. The seed point \( s \) is selected analogously, and a simple menu allows the definition of the occupancy probability \( \delta \). Simultaneously, the Tango’s camera image augmented with the virtual border points \( \mathcal{P} \) is displayed on its screen. Additionally, the global OGM, that is the basis for navigational costmaps, is integrated into the view. This makes it easy for the user to understand the workspace of the mobile robot. Besides, the user immediately gets visual feedback by system and can correct eventual mistakes. This is a crucial feature that has not been addressed by previous works yet. Fig. 3 shows some screenshots of the
teaching process where the mobile application on the Tango device is used to specify two virtual borders: a separating curve and a polygon. A full video of a teaching process can be found in the supplementary material or online at: https://youtu.be/oQ08sQ0JBRY.

V. EVALUATION

We provide quantitative and qualitative results for our proposed method concerning three criteria: correctness, accuracy and teaching effort. The results are compared with two other teaching methods that are described in the following subsection. All experiments were performed by a typical non-expert user, that got familiar with the interaction device and teaching method, as specified in the introductory part. We assume that the three evaluation criteria are not significantly affected when teaching is performed by different users who got familiar with the interaction device and teaching method. The evaluation criteria rather depend on the interaction device than on the user itself. Therefore, it is sufficient to evaluate the criteria with a single non-expert user since we do not focus on subjective perception in this work, such as the user’s perspective with respect to different interaction devices or the learning ability of the system. For the proposed method, we used a Google Tango tablet as mobile RGB-D device to acquire depth measurements and colored images from the environment. All methods are implemented as a ROS package, and we performed the following experiments on OGM with a resolution of 2.5 cm per pixel in our 6.1 m × 3.5 m lab environment. A prior map $M_{\text{prior}}$ of the environment was created with a common SLAM algorithm [21] using a particle filter and the robot’s onboard laser scanner to acquire measurements.

A. Baseline Methods

We compared our proposed method with two other teaching methods. Both methods allow the incorporation of virtual borders into a given prior map, but do not consider other occupancy probabilities except of free and occupied. Despite this limitation, both methods can be used to define certain areas in an environment and are suitable for the evaluation of the accuracy and the teaching effort. Since both comparative methods require a robotic platform for teaching, their evaluation is based on a TurtleBot v2 equipped with a laser scanner and a front-mounted RGB-D camera:

1) Marker [14]: The first method employs visual markers to teach virtual borders to a mobile robot. The user guides the mobile robot by showing visual markers, and the robot records its trajectory while following the marker. Different marker IDs indicate different states of the teaching process, e.g. recording borders or defining a seed point. The trajectory is used to define the area in the environment.

2) Pointer [15]: This method uses a laser pointer as human-robot interface and allows the user to define arbitrary areas in an environment. The user guides the robot by showing visual markers, and the robot takes its trajectory to define the area. If the laser point leaves the mobile robot’s field of view, the robot follows the direction of the laser point. Visual Morse code is used to switch between different states of the teaching process like marker IDs in the previous described method.

B. Correctness

The correctness is evaluated to answer the question whether the teaching process successfully changes the navigational behavior of the mobile robot. Therefore, we setup a simple navigation scenario as shown in Fig. 4 where the
mobile robot is instructed to navigate to the red cube in the left image of the lab environment. The centered image shows the global costmap and the path to the navigation goal based on the physical OGM $M_{prior}$ of the environment. As expected, the mobile robot crosses the carpet area while driving to its goal because it is the shortest path (the path with the fewest costs). In order to avoid the robot from crossing the carpet, we use the posterior OGM $M_{posterior}$ from the teaching process visualized in Fig. 3 as basis for a global costmap. This costmap and the calculated path to the same goal is shown in the right image of Fig. 4. It is apparent that now the mobile robot circumvents the carpet as desired. The results show that the teaching method successfully integrates the virtual borders into the global OGM and effectively changes the navigational behavior of the mobile robot. Thus, a user can easily control the workspace of a mobile robot. Note that the actual teaching process is independent of a concrete path planner for navigation.

C. Accuracy

The evaluation of this criterion answers the questions of how accurate are the virtual borders transferred from a user to the system. Accurately user-defined borders are especially important for task, such as vacuum cleaning around a carpet. We evaluated our method on a self-recorded dataset containing ten different maps with polygonal-shaped virtual borders that were manually integrated into the OGM of the lab environment beforehand. The lengths of the virtual borders range from 4 m to 13 m. Three example ground truth maps of the dataset are visualized in the first column of Fig. 6. Subsequently, the non-expert were ask to specify virtual borders according to the ground truth maps using the interaction device. We performed five runs for each map resulting in 50 runs in total. The same evaluation was performed for the baseline approaches. In order to assess the accuracy of a virtual border specified by the user, we considered the Jaccard index between two virtual areas $GT$ and $UD$ as similarity score:

$$J(GT, UD) = \frac{|GT \cap UD|}{|GT \cup UD|} \in [0, 1]$$ (7)

These two variables are defined as follows:

1) GT (ground truth): This set contains all cells of the OGM that belong to the ground truth virtual area that was manually created before evaluation. It is visualized as yellow pixels in the first column and yellow and green pixels in the remaining columns of Fig 6.

2) UD (user defined): This set contains all cells of the OGM that belong to a user-defined virtual area that was defined by the user in the teaching process. It is visualized as red and green pixels in the last three columns of Fig 6.

$|GT \cap UD|$ is the number of overlapping pixels between the ground truth and user-defined areas, whereas $|GT \cup UD|$ is the size of the union set. The Jaccard index can be visually interpreted as the size of the green area with respect to the area enclosed by the blue contour in Fig 6. Since this measure is independent of the size of the map, it can be used to compare different teaching methods easily.
The quantitative results for the accuracy evaluation are visualized in Fig. 5. In general, the accuracies of all teaching methods are high for each map (>75%), and there is only a non-significant difference between the overall averages per approach (marker: 86.6%, laser pointer: 84.6%, Tango: 85.3%). Inaccuracies occur due to localization inaccuracies and the interaction of the user. These results demonstrate the high accuracy of all methods resulting in accurately virtual maps. This is also underlined in Fig 6 that depicts the qualitative results. Note that, since the reported accuracy values of the marker approach [14] use another similarity calculation, we adapted these values to our similarity index to ensure comparability.

D. Teaching Effort

The third criterion is considered to answer the question of how much effort does it take a user to teach virtual borders. Therefore, we measured the time while performing the above-mentioned accuracy experiments. The duration starts with the selection of the first border point and ends on completion of the posterior map. The same procedure was performed for the baseline methods.

Fig. 7 shows the teaching time dependent on the border length. While there is a linear relationship between the teaching time and the border length, our approach features a smaller gradient. Thus, teaching with a RGB-D device increases the teaching time slower compared to the baseline methods. This is due to the nature of these methods: they directly use the mobile robot to define the border points and are limited by the velocity of the robot. In contrast to the baseline methods, our method only depends on the
E. Discussion

The experimental results show that the proposed method correctly incorporates the virtual borders into the given map. This is also the case for both comparative methods. The teaching method based on a RGB-D device is approximately as accurate as the baseline methods while it significantly takes less teaching effort which makes it more attractive for the users. Additionally, the RGB-D tablet provides an inherent feedback system that allows users to effectively keep track of the learned virtual borders. The absence of such a feedback system is a major drawback of the comparative methods that is purposely addressed by the proposed method.

VI. CONCLUSIONS & FUTURE WORK

We developed a teaching method for incorporating virtual borders into given OGM using a RGB-D device. This allows non-expert users to flexibly and interactively define arbitrary virtual borders in their mobile robots’ workspaces. Thus, users can prevent their robots to enter certain places, e.g. bath rooms or carpet areas, which gives them the ability to effectively control their robots in a simple way and allow human-aware navigation. We compared our method with other approaches, and the results revealed an accuracy on the same high level as the baseline methods while featuring significantly lower teaching effort. Additionally, our method integrates a feedback system that visualizes the modified workspace for the user and does not rely on additional sensors for teaching, e.g. cameras in the environment or on a robot. The modified OGM is used as basis for costmaps, and experiments demonstrated the change of the robot’s navigational behavior according to the new virtual borders.

Future work focuses on the evaluation of the user’s perspective with respect to different teaching methods because this aspect has not been addressed yet. Finally, we work on supporting the teaching process, that is currently based on sole human interaction, with a recommendation system. This aims to further alleviate the teaching effort.