2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Learning Virtual Borders through Semantic Scene Understanding and Augmented Reality

Dennis Sprute[1,2], Philipp Viertel[1], Klaus Tönnies[2] and Matthias König[1]

***Abstract*— Virtual borders are an opportunity to allow users the interactive restriction of their mobile robots' workspaces, e.g. to avoid navigation errors or to exclude certain areas from working. Currently, works in this field have focused on human-robot interaction (HRI) methods to restrict the workspace. However, recent trends towards smart environments and the tremendous progress in semantic scene understanding give new opportunities to enhance the HRI-based methods. Therefore, we propose a novel learning and support system (LSS) to support users during teaching of virtual borders. Our LSS learns from user interactions employing methods from visual scene understanding and supports users through recommendations for interactions. The bidirectional interaction between the user and system is realized using augmented reality. A validation of the approach shows that the LSS robustly recognizes a limited set of typical areas for virtual borders based on previous user interactions ($F1 - Score = 91.5\%$) while preserving the high accuracy of standard HRI-based methods with a median of $Mdn = 84.6\%$. Moreover, this approach allows the reduction of the interaction time to a constant mean value of $M = 2$ seconds making it independent of the border length. This avoids a linear interaction time of standard HRI-based methods.**

I. Introduction

Mobile service robots start to pervasively find their ways into human-centered environments, such as home environments or offices [1]. A major capability is the robust and safe navigation in the environment to support residents, e.g. carrying and placing objects [2], tidying up [3] or acting as social robots and answering users' questions [4]. However, since humans and robots share the same space, robots are expected to not only navigate in a robust and safe way, but also in a human-aware way [5]. In order to consider human needs during navigation, e.g. a robot should not cross a certain area or enter a specific room, virtual borders are an opportunity to flexibly and interactively define arbitrary robots' workspaces [6]. These virtual borders are non-physical borders defined by humans using methods from human-robot interaction (HRI) and are respected by robots during navigation enabling a human-aware navigation. From our experience we know that users often want to define virtual borders around areas with similar characteristics. For example, carpet areas should be excluded from a robot's workspace to prevent it from getting stuck, robots should circumvent pets' water dishes to avoid tackling it and spilling water on the floor, or a vacuum cleaning robot should not intrude a kid's corner and vacuum toy blocks. This knowledge gives us the opportunity to support the user during the definition of virtual borders in an interaction process.

Therefore, we propose a learning and support system (LSS) that learns from user interactions, i.e. how do previously user-defined workspaces look like, and that supports the user in future interaction processes, i.e. giving the user recommendations for interactions based on previously defined virtual borders. The objective of this LSS is a reduced interaction time compared to approaches based on sole HRI. For this purpose, we employ a smart environment including multiple cameras to perceive and semantically understand the scene. Based on this scene understanding, we learn from user interactions and give recommendations for future interactions. To this end, we employ an augmented reality (AR) interface to convey the system's recommendations to the user. Moreover, the AR interface can be used to explicitly specify virtual borders and to give visual feedback about the interaction process. In summary, we contribute a novel LSS for the definition of virtual borders to mobile robots based on semantic scene understanding and AR. This is especially relevant to support users living in human-robot shared spaces.

II. Related Work

Human-aware robot navigation is as an active research field on the intersection of robot path planning and human-robot interaction [7]. In addition to a safe navigation, works in this research field deal with the integration of social conventions into the path planning to enable a comfortable interaction between human and robot [8]. Kruse et al. [5] point out three main properties of human-aware navigation, i.e. comfort, naturalness and sociability. Recent works in this field comprise the approaching of people [9], learning of navigation paths from demonstration [10] and predicting human walking paths [11].

In contrast to these implicit methods based on observations or prior knowledge, our focus is on the explicit integration of users' needs into the robot's navigation framework, i.e. the interactive restriction of mobile robots' workspaces as the basis for a human-aware navigation. An example is sketching avoidance areas into a map of the environment to restrict a robot's workspace [12]. Since it is hard for a human to establish correspondences between coordinates in the map and the physical environment, Sakamoto et al. [13] propose a GUI-based interaction method where a user directly sketches borders onto a top-view camera stream of the environment. Moreover, AR is employed to directly interact with the environment and also visualize user-defined workspaces [14].

[1]The authors are with Campus Minden, Bielefeld University of Applied Sciences, 32427 Minden, Germany

[2]The 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: 13FH006PX5).

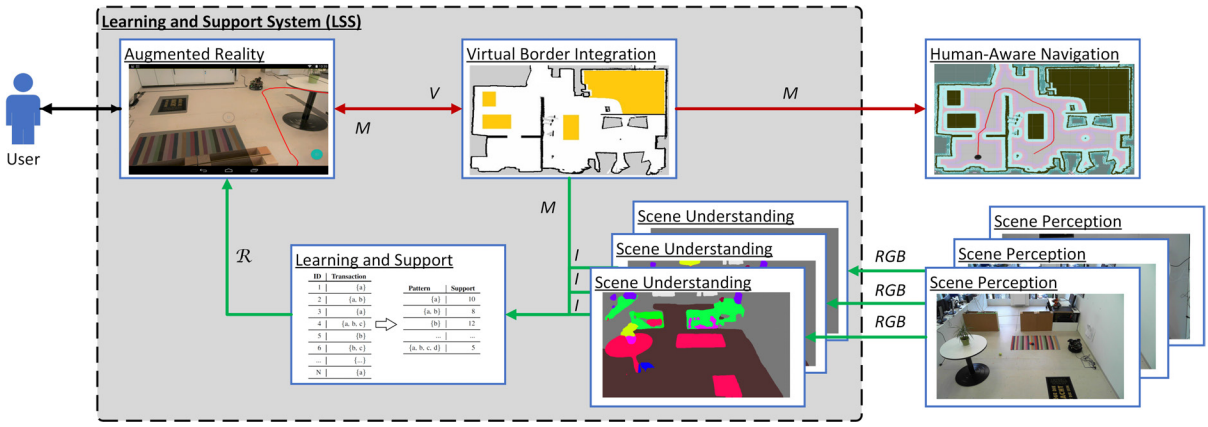

Fig. 1. System architecture consisting of several modules and a standard (red arrows) and novel (green arrows) workflow.

Current solutions for consumer service robots include beacon devices, magnetic strips and GUI-based interfaces. A study on various user interfaces for teaching virtual borders revealed that AR-based solutions yield the best results in terms of high flexibility, accuracy and user experience, while featuring a short teaching time [15]. This also explains the recent trend towards mobile AR in robotics applications, e.g. for teaching and patching robot knowledge [16], as feedback device in shared control tasks [17], for object manipulation [18], robot trajectory programming [19] or robot wheelchair navigation [20]. Therefore, our user interface is also based on AR to benefit from its usability.

However, all mentioned works are solely based on HRI without learning from user interactions. Specifically, the trend towards the integration of robots into smart environments consisting of several heterogeneous devices provides an opportunity to support the interaction process. These systems are known as ubiquitous robots [21], network robot systems [22] or informationally structured environments [23]. A recent overview of concepts and applications covering the topic of Internet of Robotic Things (IoRT), which is another synonym for the previously mentioned, is given by Simoens et al. [24]. The authors point out how the combination of robotics and IoT can enhance the robot's perception and interaction abilities.

Finally, the progress in semantic scene understanding, especially semantic segmentation using deep learning techniques, is an enabler for this work. Garcia-Garcia et al. [25] give a recent overview of deep learning techniques applied to semantic segmentation. The article highlights important deep network architectures, such as AlexNet [26], VGG [27], GoogLeNet [28] or ResNet [29], and their performances in the important ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in the last years, underlining the strong progress in this field. However, semantic segmentation has not yet been applied to encode user interactions and in the context of restricting a mobile robot's workspace. Therefore, we take the opportunity to propose a system that learns from user interactions based on semantic segmentation performed on images acquired from a smart environment.

III. Virtual Borders

Since a virtual border is the fundamental data structure in this work, we shortly characterize it in this section. Virtual borders are used to interactively and flexibly restrict the workspaces of 3-DoF mobile robots. They are non-physical borders, that are not directly visible to the user, but that are respected by mobile robots during navigation. Thus, mobile robots adapt their navigational behavior according to the user-defined workspaces enabling a human-aware navigation. Such a virtual border $V = (\mathcal{P}, \boldsymbol{s}, \delta)$ is composed of three components [14]. **Virtual border points** $\mathcal{P}$ indicate the boundary of a virtual border on the ground plane. They are structured as polygonal chain $\mathcal{P} = \bigcup_{i=1}^{n-1} [\boldsymbol{p}_i \boldsymbol{p}_{i+1}]$ consisting of n points $\boldsymbol{p_i} \in \mathbb{R}^2$. This polygonal chain separates the environment into two disjunct areas. In order to specify the area to be modified by a user, a **seed point** $\boldsymbol{s} \in \mathbb{R}^2$ on the ground plane indicates this area. The third component is the **occupancy probability** δ that specifies the occupancy probability of the area to be modified as indicated by the seed point $\boldsymbol{s}$. A map integration algorithm incorporates a virtual border into a given occupancy grid map (OGM) of the environment resulting in an OGM containing physical as well as virtual borders. Hence, a user must specify these three components using an arbitrary user interface to modify the robot's workspace and change the navigational behavior. Furthermore, this is an iterative process allowing the definition of arbitrary virtual borders.

IV. Learning and Support System (LSS)

In this work, we employ virtual borders and propose a novel LSS to learn from users' interactions and support users in future interaction processes through adequate recommendations. The goal is to reduce the interaction time resulting in a faster restriction of mobile robots' workspaces. An overview of the system architecture and its modules is visualized in Fig. 1.

A user can interact with our system through an *Augmented Reality* module enabling a human to interactively define virtual borders and to get visual feedback concerning the interaction process and its result. A user-defined virtual

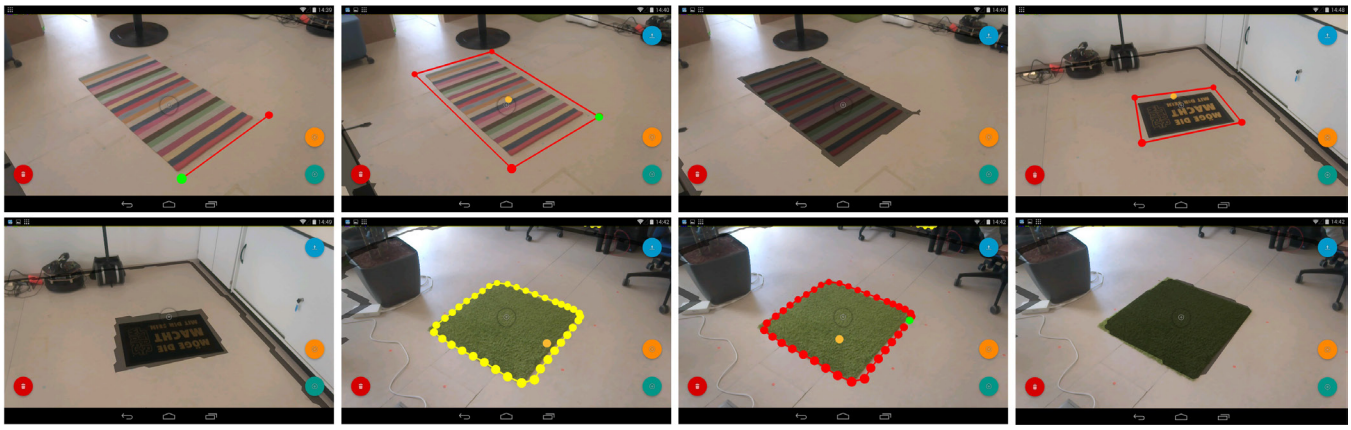

Fig. 2. Row-wise from top left to bottom right (1-8): a user defines virtual borders around two different carpets by defining the corner points of the carpets shown as red dots (1-3 and 4-5). The green dot highlights the last point of the polygonal chain. After defining these virtual borders, the corresponding areas are visualized as occupied (black) areas (3 and 5). The system suggests another virtual border shown as yellow boundary based on the previous user interactions (6). The user can simply select the recommendation without explicit definition of corner points (7) and integrate it into the map (8).

border V is passed to the *Virtual Border Integration* module which incorporates the virtual border into an OGM of the environment. This map M is used in the *Human-Aware Navigation* module as basis for a global costmap. Since the resulting map contains physical as well as virtual borders, a mobile robot respects the user-defined workspace and changes its navigational behavior. This *standard workflow* (red arrows) was basically developed in a previous work and is solely based on HRI [14], i.e. a user explicitly defines all virtual border components without any support.

To address this lack of support, we extend the system by incorporating a *novel workflow* (green arrows) into our system that is based on additional modules. A main contribution is the integration of a *Scene Understanding* module that extracts semantic knowledge about the scene. It depends on a *Scene Perception* module that captures images of the environment from different viewpoints. For this purpose, we consider a set of RGB cameras $\mathcal{C} = \{C_1, C_2, ..., C_m\}$ integrated into the environment as basis for scene perception. Another major contribution is the *Learning and Support* module that combines a previously user-defined virtual border map M and semantically segmented images I. This allows the system to semantically describe the areas defined by the user and learn from these interactions. Based on this knowledge, the module can recommend certain virtual borders $\mathcal{R} = \{V_1, V_2, ..., V_n\}$ and support the user in future interaction processes. To this end, support is finally conveyed to the user through the AR interface. In summary, our LSS extends the standard workflow by a novel workflow incorporating knowledge of the scene and learning capabilities with the objective to reduce a user's interaction time. We give details on the system modules in the following subsections, omitting the *Virtual Border Integration* module that was previously published [14].

A. Augmented Reality Module

The module acts as a bidirectional interface between the user and system. It is an extension of our previous work which allowed users to explicitly define virtual borders using a RGB-D tablet [14]. Our extension includes the support for recommended virtual borders $\mathcal{R}$, i.e. areas that are suggested by the *Learning and Support* module. Additional to the advantages of AR in the interaction process as mentioned in Sect. II, we also chose this kind of user interface due to the wide distribution of tablets.

In order to interact with the system, a user freely moves with the tablet in the environment, and the tablet's screen simultaneously shows an augmented video stream of the camera containing information, such as occupied areas, virtual borders and a robot's navigation path (see Fig. 2). The explicit definition of a virtual border V with its boundary points $\mathcal{P}$ and seed point $\boldsymbol{s}$ is realized by pointing the tablet's center towards the desired physical locations in the environment and selecting these points using software buttons. The occupancy probability δ can be changed in a pop-up menu. Additionally, recommendations for virtual borders are visualized if the *Learning and Support* module identifies appropriate areas (see image 6 in Fig. 2). In this case, the user does not need to explicitly define all virtual border components $(\mathcal{P}, \boldsymbol{s}, \delta)$, but can rather select the recommendation by pointing towards the corresponding area (see image 7 in Fig. 2). Since a recommendation already comprises all components of a virtual border, a recommendation can be directly selected by the user and send to the *Virtual Border Integration* module. This process avoids the linear interaction time with respect to the border length that goes along with the explicit definition of the virtual border components (standard workflow). Thus, we hypothesize that our system can reduce the interaction time compared to the standard workflow.

B. Scene Understanding Module

This module extracts semantic information about the scene from different camera views. Therefore, there is one instance of this module for each camera $C_i \in \mathcal{C}$ integrated into the environment. The input of this module is a color image RGB

provided by the *Scene Perception* module, and the output is a semantically segmented image I assigning a semantic class to each pixel, e.g. carpet, ground or wall. To this end, we employ an encoder-decoder architecture to perform pixel-level semantic segmentation. We use ResNet101 [29] with dilated convolutions as encoder and a pyramid pooling module [30] as decoder. The model is pre-trained on the MIT Scene Parsing Benchmark (SceneParse150) which is a standard training and evaluation platform (20K/2K/3K images for training, validation and testing) based on the ADE20K dataset [31]. This dataset contains more than 20K scene-centric images annotated with semantic categories on pixel-level. The benchmark comprises 150 semantic categories from outdoor as well as indoor scenes. We chose this dataset because it contains indoor semantic categories, that are relevant for our scenario. Furthermore, since the images are scene-centric, they can be successfully applied to similar scenes. The deep network architecture was chosen due to its state-of-the-art performance on the benchmark. The second row of Fig. 3 shows some inference results of the model on images in our indoor environment.

C. Learning and Support Module

A major contribution of this work is the *Learning and Support* module that consists of (1) learning from user interactions, i.e. the semantics of previously user-defined virtual borders, and the (2) support of the interaction process through the creation of appropriate recommendations for virtual borders $\mathcal{R}$. To this end, we formulate the problem of learning from user interactions as a *frequent itemset mining* task. We denote the set of all items as $\mathcal{I} = \{i_1, i_2, ..., i_m\}$, and a transaction T as a subset of the itemset $\mathcal{I}$, thus $T \subseteq \mathcal{I}$. The input for this task is a transactions database $D = \{T_1, T_2, ..., T_n\}$ consisting of n transactions. The support of an itemset $support(\mathcal{X})$ is the number of transactions in the database D containing the itemset $\mathcal{X}$, i.e. $support(\mathcal{X}) = |\{T | T \in D \wedge \mathcal{X} \subseteq T\}|$. It is the objective of this task to determine frequent itemsets $\mathcal{F}$ with $support(\mathcal{F}) \geq minsupport$ where $minsupport$ is a parameter specifying a minimum support value.

To adapt this problem definition, we consider a semantic of a user-defined virtual border as an item $\alpha \in \mathcal{I}$, and a transaction T is a session in which a user specifies multiple virtual borders. To this end, we perform a semantic extraction step whenever a new OGM M_t is generated by the *Virtual Border Integration* module at timestamp t. Algorithm 1 gives details on the realization that depends on the results of the *Scene Understanding* module. At the beginning, a $mask$ is created that indicates map coordinates that belong to the last user-defined virtual border (l. 3). Subsequently, for each point $p \in mask$, we determine the cameras whose field of view cover this position (l. 4f.). If a field of view of a camera c covers the point p, this point is projected into the image space of c and the corresponding semantic value is extracted and incremented in the $histogram$ (l. 7ff.). After iterating over all points $p \in mask$, the majority semantic α is determined and assigned to the last user-defined virtual border (l. 10).

This semantic is subsequently added as an item α to a new transaction T_{new} and stored in the transactions database D. Additionally, we store some morphological characteristics of the virtual border, such as area or fill degree, to validate recommendations according to their semantic-specific characteristics at a later time. If the database already contains a transaction in the same user session, i.e. within a certain time interval, the item α is added to an existing transaction T_{old}. Hence, transactions with a cardinality $|T| > 1$ emerge. If a user defines multiple virtual borders with the same semantic α, we also insert multiple transactions into the database D.

Algorithm 1: Semantic extraction step.

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

In order to learn from the data and extract frequent itemsets, we apply the FP-Growth algorithm proposed by Han et al. [32] on the transactions database D. It is an efficient method for mining the complete set of frequent patterns by pattern fragment growth. As a result, we obtain frequent itemsets, that we use to identify interesting semantics $\mathcal{S}$ and support future interaction processes.

To this end, we create appropriate recommendations $\mathcal{R}$ for virtual borders based on the interesting semantics $\mathcal{S}$. This procedure is described in Algorithm 2 that is triggered whenever a new semantic image I_C of a camera $C \in \mathcal{C}$ is available. The output of the algorithm is a recommendation for a set of virtual borders $\mathcal{R} = \{V_1, V_2, ..., V_n\}$. The algorithm creates a binary $mask$ for each interesting semantic s as basis for blob detection (l. 5f.). This allows us to create hierarchical recommendations, e.g. a water dish is placed on a carpet. Afterwards, the contour of each blob is transformed into map space and stored along with its semantic value and corresponding camera (l. 7f.). Since this step is performed for every incoming semantic image of a camera $C \in \mathcal{C}$, we obtain a set of potential virtual border points in map space that is independent of a camera. Thus, we combine observations from multiple camera views. In order to create a recommendation $\mathcal{R}$, the points from the other cameras are retrieved and added to the $points$ (l. 9f.). This set of points contains all potential virtual border points in map space for

a certain semantic s and independent of a camera. It can contain multiple virtual borders with the same semantic, e.g. multiple carpets, but also noisy points due to errors in the semantic segmentation. Furthermore, inaccurate data points can occur due to inaccuracies in the extrinsic camera calibration. To address these challenges of the set of points, we perform a density-based clustering (DBSCAN) to extract clusters with certain characteristics, e.g. appropriate size or expansion (l. 11). The validation concerning morphological characteristics is aimed to reduce false recommendations. Finally, each cluster d is thinned to account for inaccuracies, and a virtual border is extracted and added to the recommendations $\mathcal{R}$ if it does not already exist in the OGM (l. 12ff.). The recommendations are subsequently sent to the *Augmented Reality* module for visualization which closes the human-centered interaction loop. The technique for the combination of multiple camera data points and extraction of a virtual border was adapted from [33].

Algorithm 2: Recommendation step.

Input: I_C: semantic image of camera $C \in \mathcal{C}$
Output: $\mathcal{R}$: set of virtual borders for recommendation

```
1  Function createRecommendation(Input, Output)
2      R = ∅;
3      S = getInterestingSemantics (D);
4      foreach s in S do
5          mask = getMask (I_C, s);
6          blobs = blobDetection (mask);
7          points = transformIntoMapSpace (blobs);
8          store (s, C, points);
9          foreach c in C do
10             points = points ∪ get (s, c);
11         clusters = clustering (points);
12         foreach d in clusters do
13             d = thinning (d);
14             R = R ∪ extractBorder (d);
```

V. Validation

Before we evaluate our LSS concerning the interaction time, we first validate our system to check if the recommendations of the LSS achieve acceptable recognition rates and if they preserve the high accuracy of the standard workflow.

A. Setup & Procedure

To this end, we first created typical areas for virtual borders in the environment based on three categories, i.e. (1) carpets, (2) pets' water dishes and (3) kids' corners indicated by toy blocks (boxes). These categories are inspired by the use cases mentioned in the introduction. While an area of the former category consists of a single object, i.e. a carpet, the latter categories can be composed of multiple objects, e.g. a kid's corner can be composed of multiple objects. Due to this fact and different sizes of objects, i.e. dishes and boxes are smaller than carpets, there is an unbalanced number of objects in the dataset (see Tab. I). We also chose different object instances, e.g. dishes with different colors and sizes, to make the scenes more challenging. In a learning phase, a user defined multiple virtual borders for these areas using the standard workflow, i.e. explicit interaction through the AR interface without recommendations. Thus, the semantics of the user-defined virtual borders were stored in the transactions database D, but no recommendations $\mathcal{R}$ for the user were created. This learning phase was performed to add items to the transactions database D. After this phase, we validated our system based on the current content of the transactions database D in a supporting phase, i.e. the creation of recommendations for virtual borders. For this purpose, we created a dataset containing images from five different perspectives of an indoor environment with a resolution of 1920 $\times$ 1080 pixels. For each of these perspectives, we created a set of 20 different scenes. Each scene was unique, i.e. containing one or multiple typical categories for virtual borders. An exemplary scene for each perspective is depicted in Fig. 3, and the characteristics of our dataset are shown in Tab. I. We also varied the setting of each scene by adding or removing additional furniture, e.g. chairs or tables, or by changing the positions of the areas or furniture. For each scene, we also created a Ground Truth OGM (2.5 cm per pixel) containing the actual positions of the virtual borders to allow an assessment of the recommendations. For the second and third category of virtual borders, we additionally added a tolerance area (1-2 pixels in the OGM) around the objects to balance areas due to sampling on a grid map. Furthermore, this tolerance area was extended between objects of the same category if they are less than 30 cm away from each other. Overall, we acquired a total of 100 unique scenes containing 4.4 objects for virtual borders on average. The minimum and maximum number of virtual border objects per scene were one and 14.

B. Recognition Rate

We validated the applicability of our LSS by calculating the recognition rate of the recommendations. An object of an actual virtual border is considered T. The LSS can correctly recommend this object (TP) or falsely recommend another object which is actually not part of a virtual border area (FP). We count a TP if the recommendation area intersects with an object and its tolerance area by more than 50%, otherwise the recommendation is considered a FP. If objects of the same category are spatially close to each other, a single recommendation can cover multiple objects. The recognition results are shown in Tab. I. Overall, the LSS achieves a recall of 93.5% and a precision of 89.7%. This shows that the LSS can correctly recognize virtual border areas and that there are only a few false recommendations. However, since these are only recommendations, false recommendations do not need to be selected by a user and will not be integrated into the OGM. It is also apparent that some categories are better recognized than others, e.g. the recognition rate of carpets is

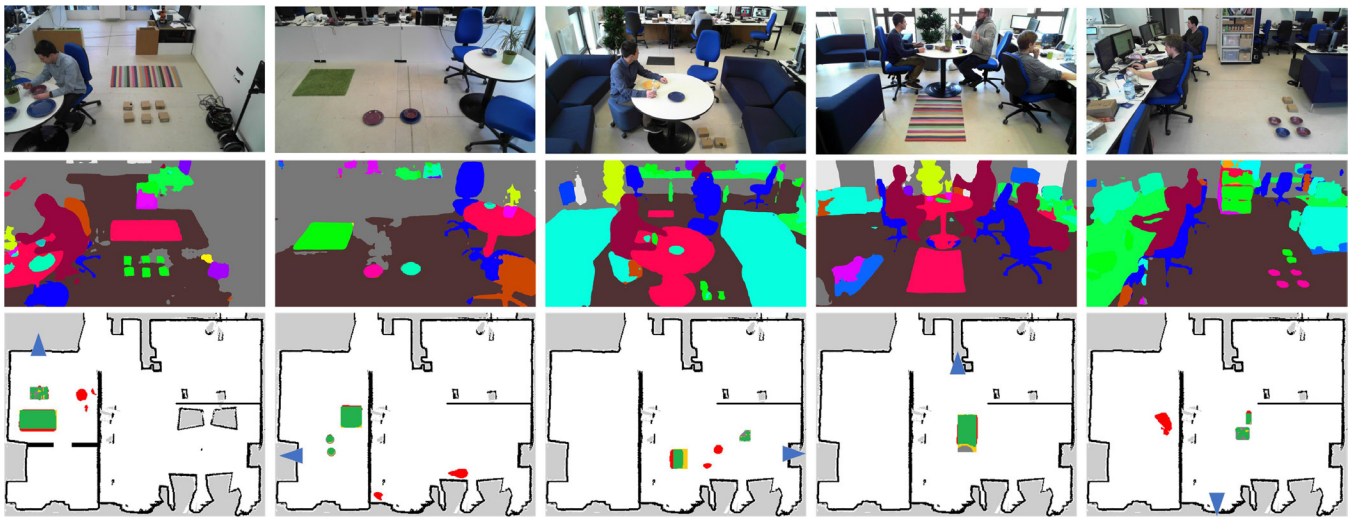

Fig. 3. First row shows exemplary RGB images from five different perspectives containing typical areas for virtual borders. Second row visualizes inference results from the *Scene Understanding* module where each color corresponds to a semantic category. The last row contains occupancy grid maps with Ground Truth (green and yellow) and recommendations of the system for virtual borders (green and red). Thus, green areas indicate the overlap between Ground Truth and recommendation areas. A blue triangle indicates the position of the camera.

higher than the recognition rate of dishes. This is due to the fact that dishes on the ground are often relatively small in image space compared to the size of an image. Moreover, a smaller precision is caused by false recommendations (FP) because there are actual objects of this category in the scene, but they are not part of a virtual border area, e.g. dishes on a table. This case is extremely rare for carpets resulting in a higher precision. Examples for false recommendations are visualized as red areas in the last row of Fig. 3. Finally, we observed that a category can encompass multiple different semantics. For example, a dish was recognized as *plate* or *ball* as indicated by two different colors in the second column of Fig. 3. The same applies for a carpet that was either recognized as *carpet* or *grass* (see first column of Fig. 3). Nonetheless, our LSS can cope with this ambiguity because interesting semantics are determined using frequent itemset mining. In case of an ambiguity, the learning phase would take longer because more user interactions would be necessary to consider a semantic as an interesting semantic.

TABLE I

CHARACTERISTICS OF THE DATASET AND RECOGNITION RESULTS.

Category	T	TP	FP	Recall	Precision	F1-Score
Carpet	72	69	1	0.958	0.986	0.972
Water dishes	152	133	21	0.875	0.864	0.869
Kid's corner	220	213	26	0.968	0.891	0.928
Sum - W. Avg.	444	415	48	0.935	0.897	0.915

C. Accuracy

After assessing the recognition rate, we determined the accuracy of the correct recommendations (TP). For this purpose, we defined the area of a recommendation as R and the Ground Truth area of an object as GT. The tolerance area of an object, an area that can, but does not need to be covered by a recommendation, was denoted as $GT_{TOL} \supseteq GT$. The accuracy of a recommendation is calculated based on the Jaccard similarity index:

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

Additionally, we compared the results with performances of the standard workflow without recommendations. These results are taken from [15] and were acquired from a multiple-user experiment ($N = 25$) where users defined virtual borders for the first time (*standard (inexperienced)*) and from a single-user experiment where a user defined different virtual borders 50 times (*standard (experienced)*). The results of the accuracy evaluation are shown in Fig. 4. We performed a Mann-Whitney U test with a significance level of $\alpha = 0.05$ to check if the accuracy of the LSS differs from the standard workflow (without recommendations from the LSS). This test was preferred to an unpaired t-test due to violations of assumptions concerning normal distribution and homoscedasticity. The Mann-Whitney U test showed that there is no significant difference in the accuracy between our LSS ($Mdn = 84.6\%$) and the standard workflow performed by an experienced user ($Mdn = 86.1\%$), $U = 4741.0$, $p = 0.194$. Compared to the performance of inexperienced users with the standard workflow ($Mdn = 66.9\%$), the LSS provides significant more accurate results, $U = 744.0$, $p < 0.001$. This shows that the recommendations of the LSS are at least as accurate as the virtual borders explicitly defined by a user with the standard workflow.

VI. EVALUATION

After showing the validity of our approach, we evaluated the interaction time as an indicator for the usability. To this end, we conducted a user study with 15 participants (13 males, 2 females; ages 18 - 34 years) recruited from the local

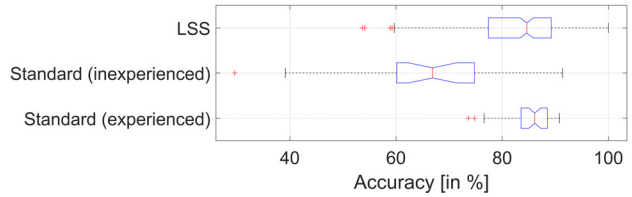

Fig. 4. Boxplots showing the accuracies of the LSS compared to the standard workflow performed by (in-)experienced users.

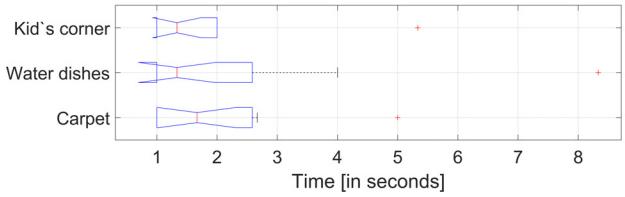

Fig. 5. Interaction time grouped by virtual border category.

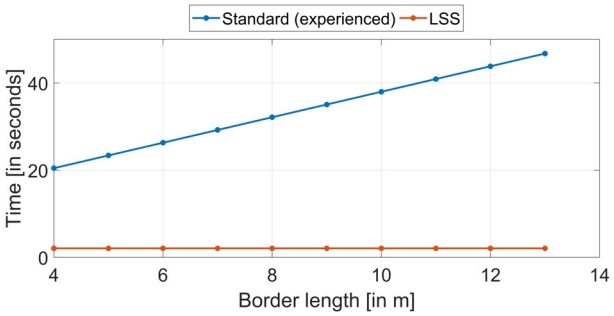

Fig. 6. Interaction time depending on border length.

environment. Their mean age was 26.6 years with a standard deviation of 4.6 years. Participants rated their experience with tablets on a 5-point Likert item with a mean of 4.0 and standard deviation of 1.0 (1=no experience, 5=experienced).

A. Setup & Procedure

For the user study, we set up a similar scenario as described in Sect. V. We created one area for each of the three virtual border categories in a 10 m $\times$ 8 m indoor environment. The areas had different border lengths and shapes. The environment additionally comprised different objects, such as tables, chairs, sofas, plants or displays. Some exemplary images and a map of the environment are shown in Fig. 3. To guarantee reproducibility of the experiment, we chose three images of our validation dataset from three different perspectives as input for the *Learning and Support* module. Each participant was first introduced to the LSS by an experimenter, i.e. description of the scenario and how to use the AR interface for the selection of virtual border recommendations. Afterwards, each participant filled a form with general information, such as age, gender and experience with tablets. Subsequently, a user was asked to define the three virtual borders by selecting the recommendations shown on the AR interface. The interaction device was a 7 inches Google Tango tablet. A participant could choose the order of the selections on his/her own, e.g. first a kid's corner, then water dishes and finally a carpet area, to avoid order effects. We measured the time between the selection of each recommendation and its final integration into the OGM. This procedure was performed three times per participant to consider the repeatability of the interaction results. The experiment took approximately ten minutes per participant.

B. Interaction Time

The results for the interaction time per virtual border category are visualized in Fig. 5. We calculated the mean value of the three performances of a participant for a category. Thus, each participant contributed a single data point to each category. Based on these data, we performed a repeated measures analysis of variance (ANOVA) with a significance level of $\alpha = 0.05$ to investigate the differences between the categories. Due to violation of the assumption of sphericity, a Greenhouse-Geisser correction was applied. The results show that there is no statistically significant difference in the interaction time between the virtual border categories $F(1.349, 18.881) = 0.201$, $p = 0.732$. This shows that the interaction time is independent of the category and the length of the virtual border. Hence, the LSS features a constant interaction time with a mean of $M = 2$ seconds. In contrast to this, the interaction time of the standard workflow is linear with respect to the border length due to the explicit definition of all virtual border components. A comparative visualization of both workflows is shown in Fig. 6. The data of the standard workflow are based on linear regression on the data taken from [15].

VII. Conclusions & Future Work

We proposed a novel LSS to support users in the interaction process of restricting a mobile robot's workspace. The system learns from user interactions employing semantic understanding of the scene acquired from multiple cameras integrated in the environment. Based on this, we extract knowledge about previous user interactions by formulating the problem as a frequent itemset mining task. The result is a set of frequent user interactions, that are used to create recommendations for similar interactions, i.e. recommendations for similar virtual borders. These are conveyed to the user through an AR interface. We showed that the LSS can robustly recognize a limited set of typical virtual borders, i.e. carpets, pets' water dishes and kids' corners, and that the accuracy of the recommended virtual borders is at least as accurate as standard HRI-based methods where a user defines all virtual border components on his/her own. Moreover, we experimentally showed that the interaction time can be significantly reduced to a constant because a recommendation for a virtual border can be simply selected and directly integrated into the map. This makes the interaction time independent of the length and shape of a virtual border, which is a drawback of the standard workflow that features a linear interaction time. A drawback of our LSS is its dependence on cameras integrated into the environment as

basis for scene understanding. However, due to the trend towards the integration of robots into smart environments, we think that this drawback will be mitigated in the future. Another point of criticism is that the cameras' fields of view must cover the areas for typical virtual borders to ensure the functionality of the novel workflow. For example, if an area is occluded by furniture or the cameras do not cover the whole environment, it is not possible to learn from user interactions and to create recommendations for virtual borders. In this case, the interaction time cannot be reduced, but the functionality can be guaranteed by the standard workflow as part of the LSS. In real-life scenarios, the actual time savings strongly depend on the environment and the camera setup. Thus, it will be a mix of the standard and novel workflow resulting in a constant time in the best case and a linear time in the worst case.

Future work should focus on considering the AR device as an additional sensor for the *Scene Perception* module. A challenge will be the real-time performance of the computational intensive *Scene Understanding* module on a hardware-limited device, such as a tablet without GPU support. However, this would overcome the limitations concerning camera view coverage and occlusions. Finally, the realization of a "track and adapt"-behavior is a work for the future. In this case, the system tracks virtual borders and adapts their location if they are moved.

REFERENCES

[1] M. Hagele, "Robots conquer the world [turning point]," *IEEE Robotics and Automation Magazine*, vol. 23, no. 1, pp. 120–118, 2016.

[2] A. Magassouba, K. Sugiura, and H. Kawai, "A multimodal classifier generative adversarial network for carry and place tasks from ambiguous language instructions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3113–3120, 2018.

[3] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, "Robot, organize my shelves! tidying up objects by predicting user preferences," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1557–1564.

[4] M. J. Chung, A. Pronobis, M. Cakmak, D. Fox, and R. P. N. Rao, "Autonomous question answering with mobile robots in human-populated environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 823–830.

[5] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[6] D. Sprute, R. Rasch, K. Tönnies, and M. König, "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)*, 2017, pp. 1175–1181.

[7] K. Charalampous, I. Kostavelis, and A. Gasteratos, "Recent trends in social aware robot navigation," *Robotics and Autonomous Systems*, vol. 93, pp. 85–104, 2017.

[8] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[9] H. Ahn, Y. Oh, S. Choi, C. J. Tomlin, and S. Oh, "Online learning to approach a person with no regret," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 52–59, 2018.

[10] N. Perez-Higueras, F. Caballero, and L. Merino, "Learning human-aware path planning with fully convolutional networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5897–5902.

[11] J. Doellinger, M. Spies, and W. Burgard, "Predicting occupancy distributions of walking humans with convolutional neural networks," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1522–1528, 2018.

[12] N. Wilde, D. Kuli, and S. L. Smith, "Learning user preferences in robot motion planning through interaction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 619–626.

[13] D. Sakamoto, Y. Sugiura, M. Inami, and T. Igarashi, "Graphical instruction for home robots," *Computer*, vol. 49, no. 7, pp. 20–25, 2016.

[14] D. Sprute, K. Tönnies, and M. König, "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)*, 2018, pp. 8574–8581.

[15] ——, "A study on different user interfaces for teaching virtual borders to mobile robots," *International Journal of Social Robotics*, vol. 11, no. 3, pp. 373–388, 2019.

[16] H. Liu, Y. Zhang, W. Si, X. Xie, Y. Zhu, and S. Zhu, "Interactive robot knowledge patching using augmented reality," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1947–1954.

[17] J. Elsdon and Y. Demiris, "Augmented reality for feedback in a shared control spraying task," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1939–1946.

[18] J. A. Frank, M. Moorhead, and V. Kapila, "Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2016, pp. 302–307.

[19] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. M. V. der Loos, and E. Croft, "Robot programming through augmented trajectories in augmented reality," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1838–1844.

[20] M. Zolotas, J. Elsdon, and Y. Demiris, "Head-mounted augmented reality for explainable robotic wheelchair assistance," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1823–1829.

[21] J. H. Kim, K. H. Lee, Y. D. Kim, N. S. Kuppuswamy, and J. Jo, "Ubiquitous robot: A new paradigm for integrated services," in *2007 IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 2853–2858.

[22] A. Sanfeliu, N. Hagita, and A. Saffiotti, "Network robot systems," *Robotics and Autonomous Systems*, vol. 56, no. 10, pp. 793–797, 2008.

[23] Y. Pyo, K. Nakashima, S. Kuwahata, R. Kurazume, T. Tsuji, K. Morooka, and T. Hasegawa, "Service robot system with an informationally structured environment," *Robotics and Autonomous Systems*, vol. 74, no. Part A, pp. 148 – 165, 2015.

[24] P. Simoens, M. Dragone, and A. Saffiotti, "The internet of robotic things: A review of concept, added value and applications," *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, pp. 1–11, 2018.

[25] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez, "A review on deep learning techniques applied to semantic segmentation," 2017, http://arxiv.org/abs/1704.06857.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS)*, 2012, pp. 1097–1105.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.

[31] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2018.

[32] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *2000 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2000, pp. 1–12.

[33] D. Sprute, K. Tönnies, and M. König, "Virtual border teaching using a network robot system," 2019, http://arxiv.org/abs/1902.06997.