

Design and Evaluation of a Flexible Interface for Spatial Navigation

Emily Tsang, Sylvie C.W. Ong, Joelle Pineau

School of Computer Science

McGill University

Montreal, Canada

emily.tsang@mail.mcgill.ca, song@cs.mcgill.ca, jpineau@cs.mcgill.ca

Abstract—The paper tackles the problem of designing intuitive graphical interfaces for selecting navigational targets for an autonomous robot. Our work focuses on the design and validation of such a flexible interface for an intelligent wheelchair navigating in a large indoor environment. We begin by describing the robot platform and interface design. We then present results from a user study in which participants were required to select navigational targets using a variety of input and filtering methods. We considered two types of input modalities (point-and-click and single-switch), to investigate the effect of constraints on the input mode. We take a particular look at the use of filtering methods to reduce the amount of information presented onscreen and thereby accelerate selection of the correct option.

I. INTRODUCTION

The design of user interfaces for the navigational command of a robot is an important problem, with application for a wide range of robotic devices. In this paper, we focus on the question of graphical user interface (GUI) design for controlling an intelligent wheelchair. Some of the challenges of this particular application include: strict limitations on the size of the display, need to accommodate individuals with a variety of impairments, need for accuracy and efficiency in the command selection.

The work described in this paper stems from the SmartWheeler project, an initiative aimed at the development of robotic technology for powered wheelchairs. It is one of a few smart wheelchairs that has undergone rigorous user testing in a controlled environment, including with a number of individuals with physical disabilities [1]. Navigational command of the wheelchair so far has been done primarily using voice commands. However as we prepare to move the wheelchair towards large public indoor environments, such as malls, museums, universities, airports, and others, it becomes imperative to develop a navigational command interface that is adapted to the noisy, crowded, and changing conditions of these environments. The role of that command interface is to allow the user to select specific navigational targets for the robot. For example, in a university setting, the user may wish to select a specific building and room number, or else to request navigation to the closest bathroom, or elevator.

There are many robotic challenges that arise when developing such a system. In this paper, we focus primarily

on the design and validation of the navigational command GUI. The design of navigational GUIs has received attention from the HRI and HCI communities recently [2], [3]. Some of the principles arising from this literature are applicable in our case. However we face additional challenges due to the nature of our target population. The design of navigational GUIs that support accessibility has received substantially less attention, though there are a few notable exceptions. There have been interfaces developed to control a robotic arm on a wheelchair to direct it towards an object one would like to grasp [4], [5]. Their interfaces, like ours, accommodate a variety of input types. However, their primary focus is on the control and navigation of the robotic arm within the *local* space visible through a camera mounted on the wheelchair. In other work, a GUI was designed for cognitively impaired users, which allows them to set local navigational targets for their wheelchair through a tactile screen [6]. This interface was later adapted to function with electroencephalogram (EEG) signal as input [7]. This work also focuses on the problem of local navigation, as the user can only set targets within the portion of the environment that is currently visible from the user's point of view (as shown on a generated 3D map). In contrast, our work focuses on using the GUI to achieve *global* navigational tasks within a large indoor environment. Users are not restricted to selecting destinations in their immediate surroundings, which would then require them to set multiple intermediate navigational targets to reach a desired goal. Rather, we assume the user is presented with a large set of possible navigational targets from the global map. We rely on several filtering techniques to reduce the set of targets such that the interface is manageable, and the interaction efficient.

We begin the paper by describing the SmartWheeler system. We then describe the design and development of the graphical user interface used for the navigational control of the wheelchair. One of the contributions of this paper is the description and empirical comparison of various filtering methods that allow the user to select from a large set of global navigational targets. We consider both category-based and spatial filters. We compare the efficiency and accuracy of these methods using different types of input (point-and-click, single-switch) to accommodate individuals with a variety of mobility disorders. We measure standard



Figure 1. The SmartWheeler platform.

performance indicators, including the time and number of clicks to selection, as well as the number of errors.

II. SMARTWHEELER PROJECT

According to a clinical survey, roughly a third of power wheelchair users struggle with steering tasks and find it particularly challenging to maneuver in crowded or constrained environments [8]. Clinicians surveyed think that many such users would benefit from an autonomous navigation system. The SmartWheeler project aims to develop a prototype of an intelligent wheelchair that can be used by individuals with mobility impairments, to assist in their daily displacements [9].

Our first prototype, shown in Figure 1, is built on top of a commercial electrical wheelchair. The chair has been outfitted with an additional onboard computer, front and back laser range-finders, an 8-inch touchscreen, and wheel odometers. The SmartWheeler was initially developed to process voice commands, with a complementary tactile interface. This system underwent a sequence of user testing according to the detailed Wheelchair Skills Test, which demonstrated that the intelligent system could correctly understand and carry out a variety of typical wheelchair driving tasks [1].

In the next phase of the project, the wheelchair will be deployed in an indoor mall environment, where it will be tasked with navigating this large, crowded space according to the commands of the user. In this setting, a dialog-based interface will be inadequate due to the noisy conditions [10]. Thus we have developed a new graphical interface to allow the user to select high-level navigational targets. It is important to note that in contrast with other smart wheelchairs, where the user's input is limited to the usual set of joystick commands leading to local navigational targets (e.g. forward and backward motion, left and right turns), our system is designed to allow the user to select global navigational targets (e.g. go to location X, find the nearest exit, etc.) To

allow the user to make maximal use of these capabilities, it is important that the navigational targets be easy to select for users with a wide range of motion impairments. The smart wheelchair's onboard computer system is equipped with standard robotic software (mapping, localization, path planning and obstacle avoidance, see [11] for details) allowing it to reach the selected targets.

III. GUI DESIGN

This section describes the design and implementation of a new graphical interface for achieving global navigation using a smart wheelchair.

A. Guidelines and constraints

We considered certain general usability principles [12] to help guide our design.

- 1) Learning time: Have a system that is easy to learn without an extensive training period.
- 2) Performance speed: Design the GUI to be responsive and to provide the user with quick ways to set targets.
- 3) Error rate: Minimize the impact of errors by making them easy to reverse.
- 4) Subjective satisfaction: Make an interface that is enjoyable to use and that minimizes possible sources of frustration.

Our design is also bound by certain constraints that arise because we are developing a system for disabled users.

- 1) Adaptability to various input types: The huge spectrum of disabilities may require users to resort to very different input methods.
- 2) Limited display size: The screen on which the GUI is presented to the user must be mounted on a wheelchair and, therefore, has a constraint on its dimensions.

B. GUI layout

The GUI, as shown in Figure 2, is divided into five panels. The central (and largest) panel shows a map of the wheelchair's environment. We show here a map of the particular mall in Montréal, Québec, where an upcoming deployment will take place. Only the second floor of the mall is presented at this stage, which is sufficient to evaluate the GUI design. The bottom panel contains a small space where user feedback is displayed; this communicates the state of the smart wheelchair to the user. We have found in previous studies that this is an important component for usability of this device [1]. The left panel contains a list of locations that may be selected as navigational targets. These locations are simultaneously displayed and labelled on the map. Currently, the user is only allowed to select from a fixed set of pre-programmed targets. The top-center panel contains buttons with various icons. These buttons allow the user to filter the targets displayed on the map and list according to their category. For instance, if the button with the *clothing* icon is selected, only apparel stores will be displayed, both on the

map (central panel) and in the list (left panel). Finally, the top-right panel contains buttons that allow zooming in and out on the map, thus allowing targets to be filtered according to their spatial location. These buttons function similarly to the category selection buttons – zooming allows the user to restrict targets belonging to a particular region for display, both on the map and in the list.

C. GUI features

We incorporated a set of features which took into account the design guidelines while working within the constraints. The major features are:

Visual-based & text-based navigational target selection

The user has the option of either *selecting the navigational target via the map* by choosing among the labelled locations displayed on the map, or *selecting the navigational target via the list* by choosing among the alphabetically-sorted buttons displayed in the list. Our conjecture is that providing both visual-based and text-based selection methods enhances the usability of the GUI. In particular, visual-based selection seems appropriate for interacting with a robot to specify navigational tasks. However text-based selection may also be desirable in cases where an individual is not very familiar with the global layout of the environment (as prescribed by the first usability principle above).

Adaptation to a range of input devices

In order to accommodate various input devices, the GUI is designed in two different versions – *point-and-click* and *single-switch*. The first can be used with any point-and-click device such as a mouse, touch-sensitive screen, or joystick; the second is suitable for single-switch input (e.g. push-button, sip-and-puff device, etc.). The various input devices supported cater to a large spectrum of motor impairments.

Using a switch as input is the equivalent of having a single action to interact with the interface. The use of switches is often the only viable option for users who lack the fine motor skills required to operate a joystick or to touch the screen directly. Among such users, some may be able to operate two or more switches. However, we chose to design our GUI to work with a single switch since this is the most basic input type. A single-switch interface is suitable for users with severe motor impairments and can be easily augmented to work with devices with more degrees of freedom.

The main difference between the two GUI versions is the method for selecting items on the display. The information presented on the panels is the same in both versions.

Target selection for single-switch input

Selecting items on the display—such as buttons from the list or locations on the map—is straightforward for point-and-click devices as the user has the freedom to point to any part of the display and ‘click’ for selection. The situation for single-switch devices is somewhat more complicated.

The challenge when using a single-switch input is to find an efficient way for the user to maneuver between

items. Many single-switch-adapted interfaces use automatic scanning, where items are sequentially highlighted and the user activates the switch when the desired item is highlighted [13]. There is software available for overlaying on existing applications to achieve switch-based mouse emulation, for example, WiVik®(<http://www.wivik.com>). However, when using such software, the pattern of scanning is not tailored to the specific application, so the result can be slow and cumbersome [14]. Therefore, to maximize efficiency and speed of item selection, we implemented a custom scanning pattern suited to our GUI display layout. In the panel-selection mode, the three panels with buttons (left panel, top-center panel, top-right panel) are scanned through. Once a panel has been chosen, the buttons within that panel are scanned through in the button-selection mode. Each panel has a button that allows the user to exit the current panel and return to panel-selection mode. The map in the central panel is not included in the panel-selection mode and thus cannot be selected. However, when the left panel is selected and the list of navigational targets are scanned through, the corresponding labelled locations on the map are also highlighted. Hence, with switch-input, there is no distinction between map and list-based navigational target selection: the two happen simultaneously.

Navigational target filtering

In our application, there is quite a large set of possible navigational targets – our map currently includes 37 pre-programmed navigational targets, and we view this as a minimal set that is likely to grow following the initial deployment. This, together with the limited display size, results in the map being fairly cluttered when the full set of targets are displayed. Similarly, the full list of buttons corresponding to the targets cannot be accommodated on the display, so the user needs to scroll through the list to see all the targets. This potentially affects accuracy and efficiency for both map and list-based navigational target selection. A practical solution is to allow users to focus their search by filtering out navigational targets that do not interest them. This is of particular importance to ensure efficient GUI interaction for the single-switch interface. Scanning through a lengthy list is slow, and furthermore, missing the desired selection may be frustrating because the user must then wait for the entire list to be traversed before it is highlighted again.

We provide the user with two ways to filter the list of navigational targets: *filtering by category* (via the top-center panel) and *filtering by region using the zoom* (via the top-right panel). To ensure compatibility with single-switch input, the zoom function is quadrant-based [4]. Each quadrant of the map is highlighted when the corresponding zoom-in button from the top-right panel is highlighted, allowing the user to activate the switch to enlarge that portion of the map. The user may zoom in at most twice, which results in displaying 1/16-th of the map.

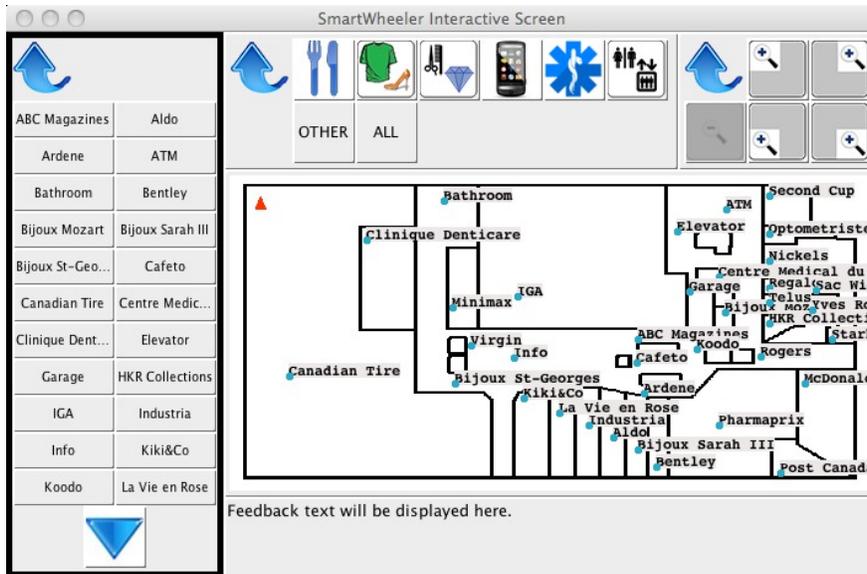


Figure 2. A screen shot of the GUI for single-switch input where the left panel is highlighted.

Error minimization and recovery

We implemented several features with the view of minimizing error and enhancing error recovery.

With point-and-click devices, accuracy may become an issue when the button size is too small. Due to the space constraints of our display, we put special consideration into button sizes. We wanted the buttons to be large enough to be selected easily when using a point-and-click device, but not so large as to impede on the space for the map. In particular, with regards to the category-selection buttons, we strived to achieve an efficient balance between the number of categories (i.e. the number of buttons, which indirectly affects the button size) and the number of targets per category (i.e. the number of targets to search through per category). There was an average of just over five navigational targets per category.

To aid error recovery, in the point-and-click version, we also added a feature whereby the map automatically zooms in when the user clicks an area of the map where there is no available target. This helps users who "miss" when trying to select a navigational target by enlarging the portion of the map they are considering and making selection easier on the second try.

Another important error recovery feature is the addition of a confirmation step that appears as a small popup window when a navigational target has been selected. This applies to both GUI versions. The pop-up window additionally presents the name of the store as well as the icon of the category the store belongs to. Informing the user of the category is helpful for finding the same store again, by using filtering by category. The concept of target validation has been used by other wheelchairs capable of autonomous navigation [7],

[6].

Last but not least, to make the GUI easier to learn, we avoided using a nested menu system and instead opted to make all the buttons visible to the user. The only exception is the list in the left panel, which may need to be scrolled down if there are too many navigational targets to be displayed.

IV. USER STUDY

The goal of our user study is to investigate the efficiency and intuitiveness of various ways of accessing navigational commands for the robot using the features provided.

A. Hypotheses

- Hypothesis 1. Users prefer selecting the navigational targets via the map, rather than selecting the navigational targets via the list, because it is a more visual way of setting goals.
- Hypothesis 2. Users appreciate filtering methods to help them select navigational targets.
- Hypothesis 3. Users prefer to filter targets with the categories, rather than by region with the zoom.

As was pointed out by an occupational therapist collaborating on the project, the input method used will be dictated by the user's available motor function. Therefore, we did not consider it pertinent to perform any direct comparisons between the single-switch and point-and-click input modes.

B. Participants

Thirty four participants were recruited to test the interface, 25 men and 9 women, between the ages of 18 and 35. All participants were university students, with no mobility impairment and without involvement in the project. Each

participant tested both the point-and-click and the single-switch versions of the GUI¹. The point-and-click version was implemented by tactile input to the display screen. The single-switch version was implemented using the space bar of a keyboard as the switch. The order in which each input mode was tested was randomized between subjects.

C. Task

The participants were required to interact with the GUI displayed on an 8-inch Lilliput touch-sensitive screen identical to the one mounted on the wheelchair. The touchscreen used for the testing was not connected to the wheelchair to minimize burden and risk. Navigation to the target was simulated via the interface only.

Participants were prompted to select nine navigational targets per input type. These destinations were presented as flash cards, as shown in Figure 3, displaying the store name, the type of store (category), as well as its relative location on the map. Providing all this information ensured that people without any prior knowledge about this particular mall and set of stores were not at a disadvantage. Once presented with a flash card, participants had to use the GUI and prescribed input method (either single-switch or point-and-click input) to achieve selection of the store listed on the card. All participants were given the same set of navigational targets; these were spread among different areas of the map and belonged to a variety of categories.



Figure 3. A sample flash card used to instruct the user study participants

As noted in Section III, the user can select a navigational target via the map or via the list when using the point-and-click version of the interface. The user can also filter the set of targets, using either the categories, or the zoom, or a combination of both. To explore participant preference for these functionalities, the participants were first instructed in how to use each of these four features. The order in which the different aspects of the GUI were shown to them was randomized so that preference was not influenced by order effect. Participants were then prompted to navigate towards a few destinations as practice. The data for these practice tasks do not figure in the results. They were simply a means to ensure that the participants had properly grasped how the GUI functioned. Finally, participants were given the nine test navigational targets to access and were free to use the map or the list, including alternating between them, to complete the task. The participants were also allowed to use one or both

¹There was a problem with the data collection for the switch-input interface for one of the participants. Therefore, the results only include the data for 33 participants for this version of the GUI.

Table I
METRICS RELATED TO THE TARGET SELECTION METHOD EMPLOYED USING TACTILE INPUT.

		Map	List	Both
Time (in s.)	mean	86.0	68.2	76.4
	std. dev.	(28.7)	(22.6)	(22.1)
Errors	mean	0.9	1.1	1.1
	std. dev.	(1.5)	(1.2)	(1.8)
Clicks	mean	32.5	27.3	27.8
	std. dev.	(9.6)	(2.4)	(4.2)
Sample size		11	11	12

Table II
METRICS RELATED TO THE FILTERING METHODS WITH TACTILE INPUT.

		Category	Zoom	Both	Neither
Time (in s.)	mean	70.2	108.5	80.6	62.5
	std. dev.	(18.2)	(29.1)	(26.8)	(22.2)
Errors	mean	0.7	2.7	1.0	1.0
	std. dev.	(1.0)	(2.1)	(1.7)	(1.7)
Clicks	mean	27.4	41.7	29.8	23.0
	std. dev.	(1.7)	(16.6)	(3.0)	(3.5)
Sample size		15	3	13	3

filtering features (categories and zoom) should they want to. The same procedure was applied to test the single-switch version of the GUI, except that only the category and zoom features were compared, since with the switch-adapted GUI, the sequential highlighting of the targets during the scanning phase happens simultaneously on both the map and the list.

D. Data collected

From automated logs, we gathered the time to task completion, the number of clicks (for point-and-click input) or switch activations (for single-switch input) required to complete the task, and the number of errors as well as their nature. Which actions were considered errors is detailed below.

We also collected questionnaire data from all participants. This was done after they had completed their interaction with the GUI using both input modes. We included open-ended questions as suggested by previous work [15]. We also collected observer notes, including transcription of comments uttered aloud by the participants as they were fulfilling the task.

V. RESULTS

We observed whether users selected the targets on the map or from the list as well as noted which filtering methods, if any, they employed. We compared mean values related to the metrics of the target selection via the map or the list, and of filtering by category or by zoom. The metrics considered were the time required to reach the nine navigational targets, the number of clicks or switch activations, and the number of errors. The data are presented in tables I through III.

Table III
METRICS RELATED TO THE FILTERING METHODS WITH SINGLE-SWITCH INPUT.

		Category	Zoom	Both
Time (in s.)	mean	207.5	197.6	289.7
	std. dev.	(33.1)	-	(52.4)
Errors	mean	3.2	3.0	4.2
	std. dev.	(2.8)	-	(2.8)
Clicks	mean	57.8	59.0	66.7
	std. dev.	(5.2)	-	(12.1)
Sample size		26	1	6

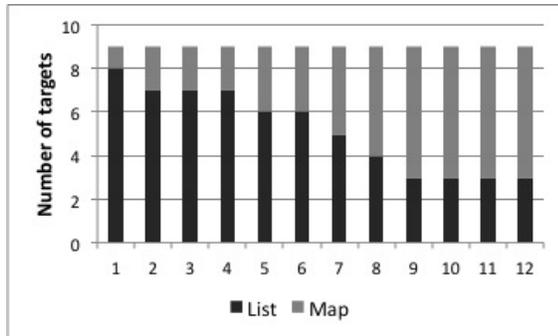


Figure 4. Number of targets selected using each method by the users who alternated between using the map and the list.

A. Target selection with tactile input

Interestingly, participants did not show a clear preference towards either of the target selection methods. The data presented in Table I show that of the users who chose a single target selection method for all of the test locations, half used the list while the other half interacted with the map. A third of the participants alternated between selecting targets directly on the map and from the list. Figure 4 illustrates that even among the users who used both the map and the list during the experiment neither of the two methods is used substantially more than the other.

Many participants who preferentially used the list explained their choice by pointing out that searching through an alphabetically-sorted list of names was easier than searching on an unfamiliar map. Furthermore, ten participants mentioned map clutter as a factor when deciding how to go about selecting the navigational targets. There was no clear trend in the justifications given for using primarily the map to select locations. Among the different reasons brought up were familiarity with the map and intuitiveness.

B. Filtering with tactile input

Most participants used at least one of the two filtering methods provided: the zoom and the categories. Indeed, as documented in Table II, only three participants did not use filtering to select any of the nine targets. Seven other people used neither the zoom nor the categories to get to at least one, and up to seven, of the targets, but used some filtering for the remaining ones. Therefore, over two thirds

of participants used filtering for every single target. This finding is particularly interesting given that the task could theoretically be completed faster and with fewer clicks when not using any filtering. The theoretical minimum number of clicks is based on the idea that participants require at least two clicks to select a target: clicking the location directly either on the map or from the list, and clicking to confirm the selection.

The categories were clearly used more than the zoom. Table II shows that fifteen participants made use of the category filtering but not of the zoom, while only three people used the zoom as their only filtering method. Thirteen subjects employed both filtering methods: they either alternated between them to get to different targets or combined category and zoom filtering to get to a given target. As illustrated in Figure 5, twelve of the thirteen used category filtering for more than half of the targets (this includes cases where participants used a combination of category and zoom to reach a given target) and nine used the categories (at least partially) for all targets. On the other hand, Figure 6 shows that the zoom was much less employed by those thirteen participants.

Certain reasons may explain why fewer participants used the zoom than the categories with the point-and-click version. Two people mentioned that they found the zoom buttons confusing, while three others maintained that it was difficult to determine in which quadrant certain target locations were in when using tactile input. Regarding the zoom functionality, six people expressed their dislike for having to avoid clicking on nearby stores when using the map. Furthermore, none of the participants who exclusively used list-based target selection used the zoom.

C. Filtering with single-switch input

As mentioned in the task description, only the filtering methods were compared for the single-switch interface because there is no distinction between map-based and list-based selection as targets are highlighted simultaneously on both map and list.

All participants used filtering, either zoom or category-based, for all nine targets of the task. With single-switch input, as was the case with tactile input, filtering by categories was the preferred choice of the participants. Table III (bottom row) shows that over 3/4 of participants exclusively used category filtering, whereas only a single person used the zoom exclusively. Six people used both filter features during the full experiment, although only one person used them in conjunction to get to a particular target. Preference for the categories was measured in users' active choice, and by Likert rankings, which averaged 1.4 for the category filtering and 2.1 for the zoom filtering. We used a five-point Likert scale where 1 was most positive and 5 was most negative. With single-switch input category-based filtering was also

significantly faster than using the zoom as shown by a two-tailed t-test (p -value = 0.02).

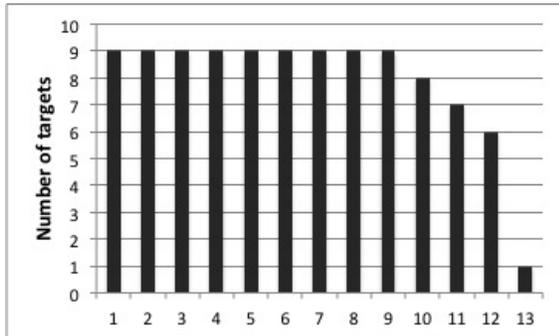


Figure 5. Number of targets chosen (at least partially) using category filtering by participants who used both filtering methods during the experiment. Subjects are sorted in decreasing order of prevalence of this behavior.

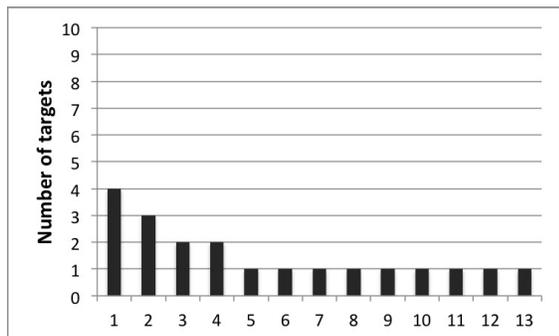


Figure 6. Number of targets chosen (at least partially) using zoom filtering by participants who used both filtering methods during the experiment. Subjects are sorted in decreasing order of prevalence of this behavior.

D. Errors

The single-switch input version of the GUI has a scanning interface. There are two main types of possible errors when using such an interface: selection errors, which come from choosing the wrong item, and timing errors, which involve missing an element the first time it is highlighted [16]. We only counted selection errors. Timing errors are apparent through an increased time to task completion, but were not otherwise quantified.

We took note of different types of selection errors that could occur in both the point-and-click and single-switch input versions. Apart from the obvious error of selecting the wrong target, we also considered the following selection errors: zooming into the incorrect portion of the map, zooming out then zooming into the exact same quadrant, selecting the wrong category, and reselecting the current category. Additionally, there are certain selection errors that are only applicable to single-switch input, including: entering and exiting a panel without making a selection, and leaving a panel then returning to it without making a selection

in between. In all cases, the incorrect selection with the necessary action to undo, if any, is counted as a single error. Hence, most errors result in a pair of additional clicks (or switch activations) rather than a single one.

Overall, we observed a low error rate for both versions of the interface. With the point-and-click version, on average there was one error while the mean number of clicks to complete the task was 29. The single-switch input incurred a slightly higher error rate with 3.3 errors for 60 switch activations on average. Since the single-switch input inherently has more potential for error, a slightly increased number of errors is acceptable.

VI. DISCUSSION

The results confirm that the interface is highly effective for selecting targets, using a variety of methods, and with little or no prior training or knowledge of the environment. This is particularly attractive for navigating in large spaces with which the user may or not be familiar. The results do not indicate a clear preference for map-based target selection over list-based selection, thereby contradicting our hypothesis 1. However certain factors may influence a user's predisposition to choose targets directly on the map or from the list. For instance, it is likely that users who are familiar with the map being used may be more tempted to use it directly instead of the list. On the other hand, an application limited to a very small area for a map or that has an extremely cluttered map may entice users to opt for selecting items from a list. Thus our results support the use of both selection mechanisms when designing interfaces for spatial navigation.

Participant concern for map clutter indicates that hypothesis 2, which concerns the utility of means to filter through the set of locations, is correct. Further evidence supporting this hypothesis is that all but three participants resorted to using a filtering feature at least once when using tactile input, even when given the choice not to, and regardless of whether they chose to select navigational targets via the map or the list. Indeed, both the categories and the zoom have the dual function of clearing some of the labels on the map and reducing the length of the list.

We were able to observe a few convincing trends, particularly the strong appreciation for filtering by category, both with point-and-click and single-switch input. This finding ties in with hypothesis 3. Of course, it is rather easy to sort stores into various categories in a mall setting. However, this result suggests that finding an intuitive classification system for possible navigational targets may be useful in other environments, even when categorizing the set of potential targets is not as obvious. Filtering by region with the zoom was not as popular with both versions of the GUI. A possible factor is that in some cases using the zoom may require first zooming out then zooming into a new region, hence more steps.

While the quadrant-based zoom currently provided on the GUI worked well for the single-switch interface, our implementation of the zoom function is suboptimal for a point-and-click device. Several participants expressed some level of dissatisfaction concerning the way the zoom functioned with tactile input. A future design may incorporate a more flexible way of zooming in, where the user may click anywhere on the map and the portion selected will double in size while centering on the spot of contact. This would require the user to first set the GUI into zoom mode, by clicking an extra button, for example. This alternative implementation would differentiate the act of zooming in intentionally and of selecting a navigational target. It also solves the problem of having to avoid navigational targets when clicking the map with the intention of zooming in, which some participants disliked. However, we will keep the feature where the map zooms in when the user accidentally clicks beside a target, rather than on it. This feature not only improves the user's chance of correctly selecting the target, but also provides visual feedback that the click was registered. Without such a visual cue, the user may believe the system is being unresponsive which could be a source of frustration.

It is important to remember that the participants were able-bodied, and that we may see different patterns of interaction with disabled users. Therefore, it will be essential to validate our system with the target population in future studies. Nonetheless, our results are a valid starting point, as they can be used as an estimate of an upper bound of performance [17]. They also offer some useful findings regarding the utility of filtering methods; we expect these results to hold with the target population.

Although we conceived our GUI for an upcoming deployment in a large indoor mall, the design can be used for other wheelchair navigational tasks, such as driving around a home or apartment. Many single-switch users currently require the aid of a caretaker to get around their home. Giving them a simple way to navigate their environment could provide these individuals with an unprecedented level of autonomy.

Finally, we believe that some of our findings may be generalized to guide the design of other navigational GUIs, including those not intended for disabled users. For instance, providing a way to filter potential targets with a set of categories or having the region of interest enlarged when the user is beside a possible target could be useful features for a multitude of navigational tasks. Further research in this area is especially pertinent as robots become more ubiquitous in human-centered environments and task domains.

ACKNOWLEDGEMENTS

The authors thank the co-investigators and collaborators of the SmartWheeler team, including Amin Atrash, Paul Cohen, Louise Demers, Robert Forget, Richard Gourdeau, Julieta Jakubowicz, Paula Stone, and Julien Villemure.

REFERENCES

- [1] J. Pineau, R. West, A. Atrash, J. Villemure, and F. Routhier, "On the feasibility of using a standardized test for evaluating a speech-controlled smart wheelchair," *International Journal of Intelligent Control and Systems*, 2011.
- [2] S. Burigat, L. Chittaro, and S. Gabrielli, "Navigation techniques for small-screen devices: an evaluation on maps and web pages," *International Journal of Human-Computer Studies*, vol. 66, no. 2, pp. 78–97, 2008.
- [3] A. H. Siyong and C. W. L. Kenny, "Evaluation of on screen navigational methods for a touch screen device," in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010, pp. 83–84.
- [4] K. Tsui and H. Yanco, "Simplifying wheelchair mounted robotic arm control with a visual interface," in *AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*, 2007, pp. 247–251.
- [5] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, "Development and evaluation of a flexible interface for a wheelchair mounted robotic arm," *Interfaces*, vol. 3, p. 11, 2008.
- [6] L. Montesano, J. Minguez, M. Diaz, and S. Bhaskar, "Towards an intelligent wheelchair system for users with cerebral palsy," *IEEE Trans Neural Syst Rehabil Eng*, vol. 18, 2010.
- [7] I. Iturrate, J. Antelis, A. Kubler, and J. Minguez, "A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation," *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 614–627, 2009.
- [8] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey," *J Rehabil Res Dev*, vol. 37, pp. 353–360, 2000.
- [9] J. Pineau and A. Atrash, "Smartwheeler: A robotic wheelchair test-bed for investigating new models of human-robot interaction," in *AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*, 2007.
- [10] L. P. Reis, R. A. M. Braga, M. Sousa, and A. P. Moreira, "IntellWheels MMI: A flexible interface for an intelligent wheelchair," in *RoboCup-2009, LNAI 5949*, 2010.
- [11] A. Atrash, R. Kaplow, J. Villemure, R. West, H. Yamani, and J. Pineau, "Development and validation of a robust speech interface for improved human-robot interaction," *International Journal of Social Robotics*, 2009.
- [12] B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley, 2004.
- [13] D. Harris and G. C. Vanderheiden, "Augmentative communication techniques," in *Non-Speech Language and Communication*, R. Schiefelbusch, Ed. University Park Press, 1980.
- [14] S. Carter, A. Hurst, J. Mankoff, and J. Li, "Dynamically adapting GUIs to diverse input devices," in *8th International ACM SIGACCESS conference on Computers and Accessibility (Assets)*, 2006.
- [15] K. Tsui, D. Feil-Seifer, M. Mataric, and H. Yanco, *Performance Evaluation and Benchmarking of Intelligent Systems*. Springer US, 2009, ch. Performance evaluation methods for assistive robotic technology.
- [16] S. Bhattacharya, D. Samanta, and A. Basu, "User errors on scanning keyboards: Empirical study, model and design principles," *Interacting with Computers*, vol. 20, May 2008.
- [17] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, "Experimental design for human-robot interaction with assistive technology," in *Proceedings of the HRI Workshop on Robotic Helpers: User Interaction, Interfaces and Companions in Assistive and Therapy Robotics*, 2008.