Navigating Blind People with a Smart Walker

Andreas Wachaja* Pratik Agarwal* Mathias Zink* Miguel Reyes Adame‡ Knut Möller‡ Wolfram Burgard*

***Abstract*— Navigation in complex and unknown environments is a major challenge for blind people. The most popular, conventional navigation aids such as white canes and guide dogs, however, provide limited assistance in such settings as they are constrained to interpret the local environment only. At the same time, they can hardly be combined with a walker required by elderly people with walking disabilities. Technologies developed in the field of robotics have the potential to assist blind people in complex navigation tasks as they can provide information about obstacles and reason on both global and local environment models. The contribution of this paper is a smart walker that enables blind users to safely navigate. It includes an innovative vibro-tactile user interface and a controller that takes into account human characteristics based on a user study. The walker has been designed to deal with the fact that humans can only sense and interpret a limited number of commands and have a delayed response. Our experiments validate our claim that the technique outlined in this paper guides a user to the desired goal in less time and with shorter traveled distance compared to a standard robotic controller.**

Fig. 1. The smart walker is an off-the-shelf walker retrofitted with an extension unit that contains computing capabilities and two laser scanners for perception. Additionally, two vibration motors are integrated into the handles for tactile feedback generation.

I. Introduction

One of the hardest tasks faced by blind people is navigating new environments safely. Conventional aids such as white canes or guide dogs help in detecting landmarks such as tactile pavings but they can only provide limited assistance in guiding the user towards a target location. For example, large exhibitions often provide human guides who assist blind people in finding their booth of interest. For blind people, navigating in such dynamic environments is often only viable with a non-blind assistant and hence limits their mobility. Additionally, this makes blind people dependent on others.

The task is even more challenging for blind people with walking disabilities. According to a recent report of the World Health Organization [1], 81.7% of all 39 million blind people worldwide are 50 years and older. These elderly blind people have an inherent risk towards walking disabilities. However, established navigation aids for the blind provide limited physical assistance as most of these devices are not designed for blind people with walking impairments. A conventional technique for such elderly blind people who depend on a walker is to regularly stop and monitor the environment with a cane stick. This is tediously slow and limits their radius of operation significantly.

*Department of Computer Science, University of Freiburg, Germany
‡Institute of Technical Medicine, Furtwangen University, Germany
We thank R. Broer from RTB GmbH & Co. KG Bad Lippspringe, Germany, for helpful comments and the exhibition space at SightCity 2014. This work has been partially supported by the German Federal Ministry of Education and Research (BMBF), contract number 13EZ1129B-iVIEW and by a grant from the Ministry of Science, Research and the Arts of Baden-Württemberg (Az: 32-7545.24-9/1/1) for the project ZAFH-AAL.

In this paper, we present a navigation system for blind people with walking impairments that uses intuitive vibro-tactile signals to guide users safely. Our approach is based on robotic techniques but applied for the purpose of navigating human users. Guiding users for such tasks requires additional considerations compared to robots. Humans take a longer time to react, i.e., a delay exists between the perceived navigation command in the form of a vibration signal and the resulting action. Additionally, humans are not as good as robots in accurately following commands. It is not possible to ask users to move or turn precisely. Our control system considers such human characteristics.

The contribution of this paper is twofold. First, we describe the setup of a smart walker system that is suitable for assisting elderly blind people, see Figure 1. The second contribution is a novel method to guide human users on a path. Our system is based on an off-the-shelf walker that is equipped with sensors, data processing capabilities, and one vibration motor in each handle. The smart walker is able to avoid collisions and guide blind people in complex environments to goal locations by leveraging recent advances in the field of robotic perception. Employing vibration motors instead of auditory signals allows blind users to use their hearing for other purposes. We derive a model of the human motion with the walker based on trajectories recorded with a motion-capture system and we use this model to design a novel controller. In an evaluation with multiple test subjects, we show that this controller guides a user to the desired goal in less time and less distance traveled compared to a standard controller.

II. Related Work

Our system belongs to the field of robot-assisted navigation. Such navigational techniques can be categorized by their level of autonomy. High-level systems track the user position and plan complete paths in order to guide the user along a specific route [2], [3]. These approaches provide functionalities that are far beyond the ones of conventional Electronic Travel Aids for the visually impaired, but they often reduce the autonomy of the user by incorporating the decision making process into the navigation process. The user is not a part of the decision making and all navigation decisions are made by the system. Devices with a medium level of autonomy propose a direction to avoid nearby obstacles but do not guide a user to a desired goal over large distances [4], [5]. Low-level approaches detect obstacles in the vicinity of the user and only inform about their positions [6], [7].

All the guidance techniques mentioned above rely on feedback mechanisms to either guide a user on a path or inform about hazardous areas. These feedback techniques range from vibration signals [8], [9], over audio output [6], [10] to force feedback [4], [11]. Blind users prefer not to receive guidance through audio signals as they rely on the auditory channel for a large amount of tasks [12].

Researchers have also evaluated novel strategies to improve the guidance of a human user. For example, Cosgun *et al.* use a vibration belt with distinct vibration patterns to communicate directional and rotational navigation commands [8]. Their main focus is the comparison of different navigation signals and the guidance of a user to a pre-defined goal location. While Cosgun *et al.* use a controller that models the human as a robot, we consider typical human characteristics in the design of our controller and also track the complete trajectory of the user during the guidance process. We will later show in the experimental section that by modeling human characteristics our controller improves the guidance process.

Related to our system, Yu *et al.* propose a motorized walker which aims to guide elderly people with cognitive impairments using shared-autonomy control [13]. Their walker can measure the driving performance of the user and varies the autonomy level between user-control and computer-control according to the observations. This enables the walker to adjust to the needs of the user. Bosman *et al.* propose a high-level system which guides users on a in complex indoor environments with tactile cues from vibration motors mounted on the wrist [14]. They show that their system can even be helpful for non-blind users, but also results in reduced location and orientation awareness.

The smart walker we present in this paper is related to the work of Yu *et al.* [13] as it is designed to provide an adjustable level of autonomy. It is unmotorized and hence cheaper to build. As our walker is passive, it allows the user to be in full control over the locomotion. Given the sensor setup, our system can identify both positive and negative obstacles such as downward leading stairs. Additionally, our system is also able to guide the blind user on a planned path. This enables a wide range of applications and potential user groups. We use vibro-tactile signals as feedback representation because this method does not overlay important sounds from the environment [12].

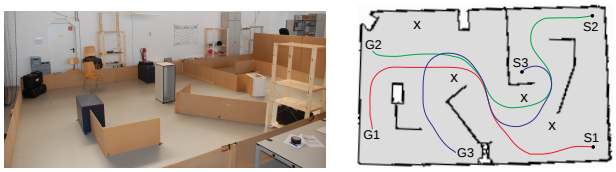


Fig. 2. One of our test environments and the corresponding map with planned paths from start positions `S` to goal locations `G`. Obstacles not present during the map creation process are marked with an `x`

III. Smart Walker Overview

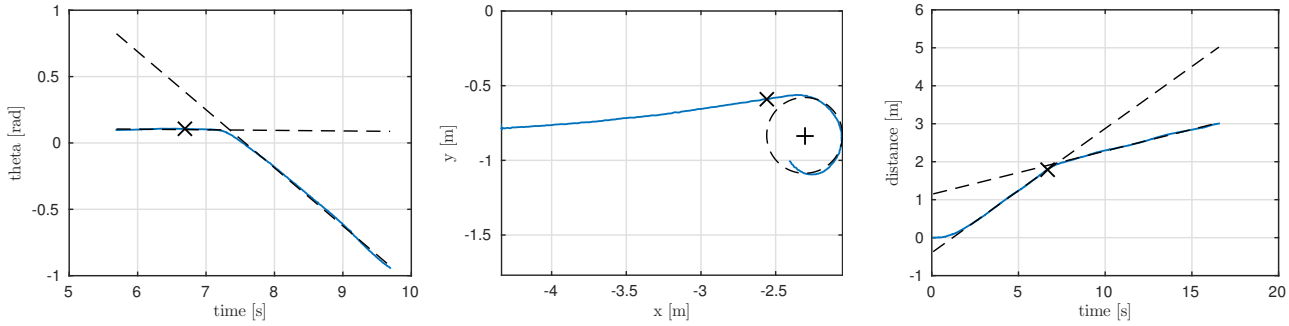
The smart walker system is an off-the-shelf walker, retrofitted with LIDARs for perception, data-processing capabilities and vibration motors in the handles for vibro-tactile feedback. The sensing and processing unit is built in a modular fashion, such that it can be easily mounted on different walker brands. An image of the smart walker is shown in Figure 1.

We use two planar laser range finders for perception and estimation of the egomotion. The first laser scanner is fixed with respect to the walker. We calculate the egomotion of the walker based on the measurements of this sensor by laser scan matching [15]. The second laser scanner is continuously tilted by a servo motor to sense the three-dimensional environment. We fuse the egomotion estimation with the measurements of the second scanner and data from its motor to obtain a dense three-dimensional point cloud.

Additionally, our approach leverages terrain classifiers from robotics to detect hazardous positive and negative obstacles from point clouds. Specifically, we modified the "height-length-density" (HLD) classifier, which is designed to determine safe and traversable cells in a planar grid map [16]. Our modification improves its suitability to human motion with a walker in tight narrow indoor spaces.

We use a publicly available module for path planning that is based on the Dijkstra algorithm. The path planner considers the map as well as detected obstacles. Figure 2 shows one of our test environments and the resulting map overlayed with planned paths. Typically, the map is created beforehand with a simultaneous localization and mapping (SLAM) algorithm, but our system is also able to handle the process of localization, mapping and path planning in parallel. In this case, the planner considers all available information for explored regions and uses heuristics for unexplored areas to estimate a path to the goal location.

The controller module on the walker guides the user for navigational tasks with four different vibration signals, namely *go straight*, *turn left*, *turn right* and *goal reached*. Each signal is repeated continuously until it is overwritten by another one. These navigation commands are generated by vibration motors which we integrated into the handles of



(a) Reaction time (b) Rotational radius (c) Velocities

Fig. 3. Results from one run of the system identification experiments for obtaining the human characteristics. In all of these runs, the user is directed to move straight and then turn when one of the handles vibrates. The point at which the turn-signal is sent to the user is marked with a cross sign. The solid blue line always depicts the actual user state while the dotted lines represent the best model fit based on k-means clustering and least square estimation. The left image illustrates that we compute the reaction time of a user by measuring the difference in time between the signal sent (cross) and the change in orientation (intersection of the dotted lines). The center image illustrates the rotational radius computed by fitting a circle (dotted lines). The right image illustrates the two distinct velocities.

the walker. In order to create signals which are intuitive and easily distinguishable, we vary the signal pattern in addition to the location of the signal (left/right):

- *Go straight:* Pulsed vibration on both vibration motors. After an on-cycle of 100 ms, the motors are turned off for 400 ms.
- *Turn left:* Continuous vibration on the left vibration motor. The right motor is turned off.
- *Turn right:* Continuous vibration on the right vibration motor. The left motor is turned off.
- *Goal reached:* Both motors are first enabled with 100% vibration intensity, then enabled with 50% and finally turned off. Each step is executed with a duration of 500 ms.

Our software is based on the ROS framework [17], which simplifies the use of existing, publicly available modules and increases its flexibility.

The smart walker has two caster wheels on the front that can rotate freely around the yaw-axis. In contrast, the two wheels on the back are mounted fixedly in parallel. Therefore, we can abstract the kinematic of our walker to a differential drive. The rotation axis of the walker is always on the line which goes through both contact points of the back wheels with the ground, as long as the wheels do not slip. The user of the walker adjusts to this kinematic constraint intuitively. We define the reference-frame with its origin in the middle between both contact points, the x-axis pointing in forward direction and the z-axis perpendicular to the ground.

IV. System Identification

An important prerequisite for guiding users on a path with our controller is to understand how humans react to navigation commands. In particular, we model in a user study the response of human users to changes in the navigation signal. We identified the following parameters to be important for human guidance:

- *Reaction time (t_R):* The time between the system sends a command and the user performs the desired action.
- *Rotational radius (r):* The radius of curvature for the trajectory employed by the user for turning.
- *Velocities ($v_{straight}$, v_{rot}):* The constant velocity of a user while moving straight and while turning.

In our user study we blindfolded the test subjects and asked them to move straight till a vibration-feedback on the handle directed the user to stop and turn either in the left or the right direction depending on the signal. The users were directed to turn on the spot to avoid generating trajectories with large radius to facilitate navigation in narrow passages. In total we observed 10 users. Initially, they were allowed to get familiar with the walker and vibration feedback. During this phase we did not record data. Once the users were familiar with the setup we collected 10 evaluation runs per user to model the human response to navigation commands. We used a motion capture system to track the position and orientation of the walker over time and fused this data with the navigation signals. Figure 3 shows the typical results of one run.

For calculating the reaction time per run, we consider the orientation of the walker over time. K-means clustering was used to fit two lines which approximate the rotation velocity before and during the turning process. The intersection point of these lines marks the point in time where the user reacts to the signal and actually turns. The reaction time is the difference between the signal sent to the user and the walker being rotated. This is further illustrated in Figure 3(a). The cross mark is the time at which the turn signal was sent to the user while the intersection of the dotted lines estimates when the user actually turned the walker.

To calculate the rotational radius we fit a circle to the rotational trajectory using a least square minimization technique. The radius of the circle is the approximated rotation radius. Figure 3(b) illustrates the circle fit to the trajectory. The velocity of the user in both situations, while going straight and while turning, is computed with a similar k-means method as the one we used to estimate the reaction time. The dotted lines in Figure 3(c) illustrate the straight

TABLE I

ESTIMATED PARAMETERS FOR THE PREDICTION-BASED CONTROLLER

Parameter	Mean (std-dev)
Reaction time t_R [s]	0.87 ($\pm$0.20)
Rotational radius r [m]	0.36 ($\pm$0.25)
Straight velocity $v_{straight}$ [m/s]	0.44 ($\pm$0.15)
Rotational velocity v_{rot} [m/s]	0.18 ($\pm$0.08)

and rotational velocities.

The results of the system identification experiments can be found in Table I. It can be seen that the reaction time of a human is almost 1 second. This is much more than that of a typical robot setup. Hence, we believe that incorporating the delay of a human user is a key for improving the guidance process. For example, if the delay is not considered, a user could overshoot the goal by more than half a meter based on the straight velocity. The rotational radius is also important for guiding a human on the desired path and to predict his position.

V. CONTROLLER

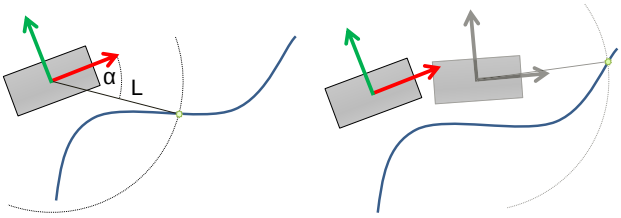
Our controller is based on a modified carrot-following approach. We extend the existing method described by Hogg *et al.* [18], as the basic algorithm cannot guarantee that the path is not followed in the wrong direction in case that the robot is oriented in the wrong way. The carrot-following approach estimates a waypoint on the path in a fixed lookahead distance L from the robot. This waypoint is called the carrot position. The algorithm calculates the intersections of a circle with the radius L around the reference-frame with all path segments as illustrated in Figure 4. We consider an intersection point as a potential carrot position in case that a direction vector which indicates the direction of the path at this position points away from the circle. This constraint avoids that the robot follows the path in the wrong direction. A new carrot position is only accepted if its distance from the old carrot position measured over the path length is below a threshold. This ensures that the robot follows a self-intersecting path all the way from the beginning to the end and does not choose a wrong carrot position at the intersection of two path segments.

Our controller monitors the angular difference α between the roll-axis of the walker and the line that connects the reference-frame and the carrot position. It sends a turn-command to the user as soon as $|\alpha| > \alpha_{thresh}$. To avoid oscillations for $\alpha \approx \alpha_{thresh}$ we introduce a hysteresis value α_{hyst}. After a turn-command was sent, the controller switches back to the go-straight-command as soon as $|\alpha| \leq \alpha_{thresh} - \alpha_{hyst}$.

We state three requirements to compute the lookahead distance L and the angular threshold α_{thresh}:

1) A robot driving on a straight line parallel to a straight path line commands the user to turn as soon as the distance between the two lines is above a maximum distance d_{max}:

$$L \sin(\alpha) \leq d_{max} \tag{1}$$



(a) Without prediction (b) With prediction

Fig. 4. Carrot planner for controlling the walker with (right) and without (left) considering the delay in the response time. The controller computes a reference point on the desired trajectory in a fixed lookahead distance L from the robot. The control commands are sent to the user based on the angular deviation α between the x-axis of the walker (red) and the line between the reference-frame and the carrot position. Our controller predicts the pose of the walker under consideration of the delay time and computes the navigation command based on the predicted pose (shown in gray).

2) A robot without control delay and with a fixed curve radius r driving on a straight path line towards a rectangular corner of the path should start turning as soon as the distance between the robot and the corner equals r:

$$L \cos(\alpha_{thresh}) = r \tag{2}$$

3) The lookahead distance should be maximized with respect to Eq. 1 and 2 in order to avoid oscillations.

From these requirements we can derive suitable values for L and α_{thresh}:

$$\alpha_{thresh} = \arctan\left(\frac{d_{max}}{r}\right) \tag{3}$$

$$L = \frac{r}{\cos(\alpha_{thresh})} \tag{4}$$

In its basic version, the carrot planner does not consider the reaction time t_R of the user. As we explained in Sec. IV, this delay can cause significant guidance errors. Therefore, we extend our carrot planner with a prediction module that predicts the pose of the walker at the time $t_0 + t_R$, where t_0 is the current point in time. This is a technique commonly used when handling delay times in systems, for example Engel *et al.* use this method to handle delay times caused by off-board computations [19]. We use a constant velocity model, the rotational radius determined in the user study and the navigation commands sent in the time range $[t_0 - t_R; t_0]$ to predict the specific pose of the walker which is relevant for the current navigation decision, see also Figure 4.

VI. EXPERIMENTS

In order to evaluate our smart walker and the controller we performed a second user study with 8 blindfolded participants. The goal of our evaluation is to compare the controller which considers the human characteristics against the standard controller without prediction. Based on the estimated system parameters in Table I we set the reaction time of the user t_R to 0.87 s and the rotational radius to 0.36 m. These values were used to determine the parameters for the carrot planner. Furthermore, we set α_{hyst} to 5° and

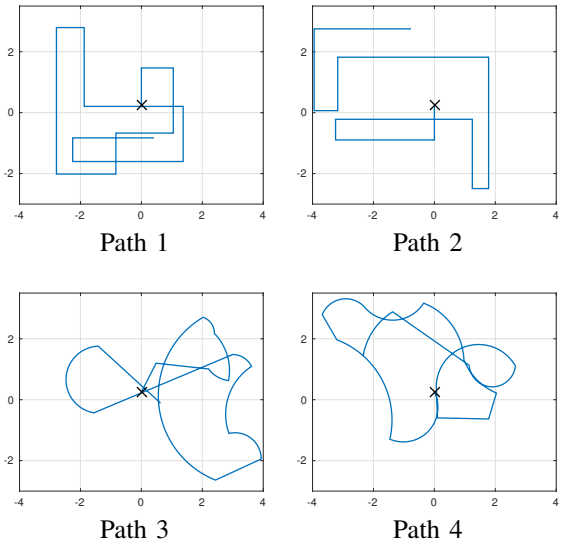


Fig. 5. The four randomly generated paths for our evaluation. The start point is marked with x. Path 1 and Path 2 contain straight lines joined by right angles. This simulates indoor environments. Path 3 and Path 4 consist of both straight lines and arcs. They simulate outdoor environments. Each path has a length of 30 m. All units are in meters.

d_{max} to 0.09 m. This yields a value of 14° for α_{thresh} and 0.35 m for the lookahead distance L (rounded).

For this evaluation, we generated 6 random paths. Half of the paths consist of lines connected with right angles simulating indoor environments. The other half of the random paths contain lines and arcs that are connected with varying angles. This simulates typical outdoor trajectories. Out of the 6 paths, we used 2 for training the users and the other 4 paths for evaluation. Figure 5 provides an overview of the paths which we used for evaluation.

We provided each test participant with a short introduction to the smart walker and the guidance commands. The test subjects completed both training paths in randomized order, one path with the standard controller and one path with our prediction-based controller. After the training, every test participant completed each of the remaining four paths twice, once with each controller. The order of paths and controllers was randomized to avoid any bias. For the purpose of quantitative evaluation, we tracked the trajectory of the test person using a motion capture system and recorded the navigation signals with a frequency of 50 Hz. For qualitative comparison, the test person graded the guidance style on a scale from 1 (very bad) to 10 (excellent) after every run.

Figure 6(a) illustrates the desired path that the user is guided along and the two trajectories of the test subject resulting from the different controllers. The standard robotic controller which does not consider the reaction time of the user oscillates around the ground truth path. This is also illustrated in Figure 6(b) where we compare the length of each trajectory pair per user and path by subtracting the distance traveled with the standard controller from the distance traveled with our controller. We can see that the length of the trajectories resulting from our controller which incorporates user prediction is shorter compared to the standard controller. A paired-sample t-test confirms that the trajectories with prediction are significantly shorter at a significance level of $\alpha = 0.01$. Additionally, the time required for the user to reach the goal on the desired path is also less with our controller. This is illustrated in Figure 6(c).

We also computed the mean deviation from the desired trajectory to compare the influence of the controllers. To compute the mean deviation we chose equidistant points on the desired trajectory and then calculated their distance to the path of the user for both controllers. We could not find significant differences between the pairs of mean deviations of the trajectory from the ground truth path at the significance level of $\alpha = 0.01$. The mean deviation over all experimental runs is 0.060 m with a standard deviation of 0.058 m.

The improved path guidance performance for the controller with prediction comes at the cost of a higher frequency of navigation signal changes, as can be seen in Figure 7(a). Also, the qualitative evaluation reveals that the users preferred the controller without prediction (see Figure 7(b)). As a few users stated, this was mainly due to the fact that the controller with prediction caused a high amount of signal changes which could not always be interpreted clearly. This result is in contrast to the improved path guidance performance in terms of resulting path length and trajectory execution time. Our results show that out of 32 runs, 78% of the runs were shorter with our controller and 65% of the runs required less time. We believe that these results which we acquired with healthy test persons can also be transferred to elderly people with walking disabilities. On one hand these people tend to walker slower but on the other hand they also suffer from decreased reaction times (compare [20]) that motivate the need for the prediction-based controller.

VII. Conclusions

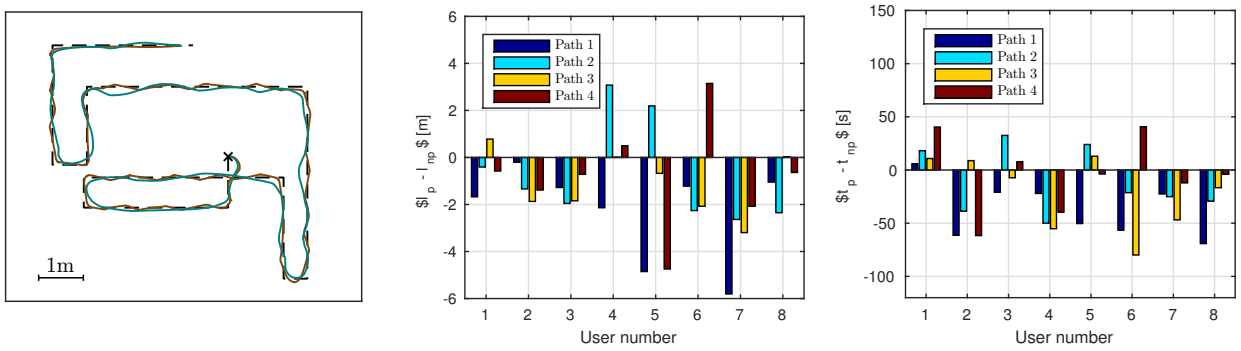
In this paper we outlined a smart walker designed to enable blind people with walking disabilities to navigate safely in unknown and complex environments. Our system leverages existing robotic technologies for guiding humans. We first proposed the system architecture for our walker and explained the various hardware and software modules. Secondly, we identified human characteristics which differ from robots for the task of precise navigation for path guidance. In practical experiments we showed that by modeling human characteristics, especially in terms of reaction time, users can be guided more accurately resulting in less time required and distance traveled to reach the desired goal. As a next step we plan to evaluate our controller in real-world everyday scenarios on the presented hardware platform.

References

[1] D. Pascolini and S. P. Mariotti, "Global estimates of visual impairment: 2010," *British Journal of Ophthalmology*, 2011.

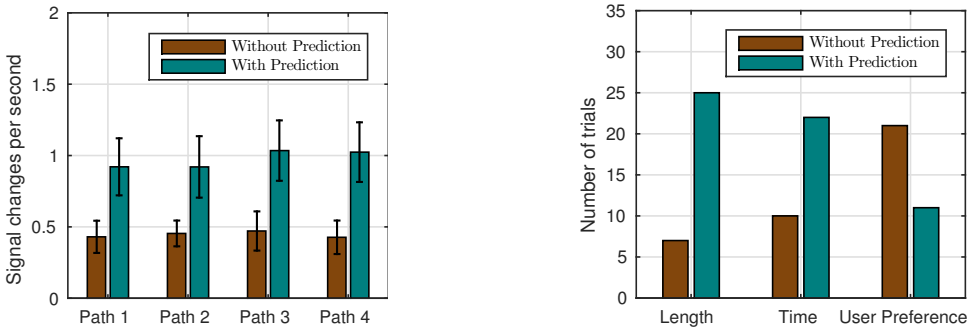
[2] V. Kulyukin, C. Gharpure, and J. Nicholson, "RFID in robot-assisted indoor navigation for the visually impaired," in *Int. Conf. on Intelligent Robots and Systems*, 2004.

[3] J. Glover, D. Holstius, M. Manojlovich, K. Montgomery, A. Powers, J. Wu, S. Kiesler, J. Matthews, and S. Thrun, "A robotically-augmented walker for older adults," Carnegie Mellon University, Computer Science Department, Tech. Rep., 2003.



(a) Path comparison (b) Difference in distance traveled (c) Difference in travel time

Fig. 6. The left figure overlays the ground truth path (dashed) with the trajectories of one user with both controllers. The start position is marked with x. The trajectory of the controller without prediction (brown) oscillates around the ground truth path. The prediction-based controller (turquoise) reduces this effect. The difference in path length (center) and the difference in the travel time (right) illustrate that the prediction-based controller yields in shorter trajectories and in reaching the goal faster. Negative values indicate shorter distance and less time for the prediction-based controller.



(a) Signal changes (b) Controller preference

Fig. 7. Comparison of the prediction-based controller with the standard controller without prediction. The left figure shows the mean number of navigation signal changes per second for each path. The controller with prediction relays a significantly higher frequency of navigation commands. Path 3 and Path 4 require a higher frequency of signal changes, which is a result of the more complex path geometry. The right figure shows the number of trials in which the prediction-based controller performs better than the standard one in terms of path length (shorter better), travel time (quicker better) and user-preference (higher score better). We assume that the users prefer the controller without prediction due to the lower frequency of navigation command changes.

[4] I. Ulrich and J. Borenstein, "The GuideCane—applying mobile robot technologies to assist the visually impaired," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Trans. on*, vol. 31, 2001.

[5] A. Rentschler, R. Simpson, R. Cooper, and M. Boninger, "Clinical evaluation of Guido robotic walker," *Journal of Rehabilitation Research & Development*, vol. 45, no. 9, 2008.

[6] A. Rodríguez, J. J. Yebes, P. Alcantarilla, L. Bergasa, J. Almazán, and A. Cela, "Assisting the visually impaired: obstacle detection and warning system by acoustic feedback," *Sensors*, vol. 12, 2012.

[7] A. Wachaja, P. Agarwal, M. R. Adame, K. Möller, and W. Burgard, "A navigation aid for blind people with walking disabilities," in *IROS Workshop on Rehab. and Assistive Robotics*, Chicago, USA, 2014.

[8] A. Cosgun, E. A. Sisbot, and H. Christensen, "Guidance for human navigation using a vibro-tactile belt interface and robot-like motion planning," in *Int. Conf. on Robotics & Automation*, 2014.

[9] K. Tsukada and M. Yasumura, "Activebelt: Belt-type wearable tactile display for directional navigation," in *UbiComp 2004: Ubiquitous Computing*. Springer, 2004.

[10] L. Ran, S. Helal, and S. Moore, "Drishti: an integrated indoor/outdoor blind navigation system and service," in *Annual Conf. on Pervasive Computing and Communications*, 2004.

[11] A. Morris, R. Donamukkala, A. Kapuria, A. Steinfeld, J. Matthews, J. Dunbar-Jacob, and S. Thrun, "A robotic walker that provides guidance," in *Int. Conf. on Robotics & Automation*, 2003.

[12] D. Dakopoulos and N. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans. on*, vol. 40, 2010.

[13] H. Yu, M. Spenko, and S. Dubowsky, "An adaptive shared control system for an intelligent mobility aid for the elderly," *Autonomous Robots*, vol. 15, no. 1, 2003.

[14] S. Bosman, B. Groenendaal, J.-W. Findlater, T. Visser, M. de Graaf, and P. Markopoulos, "GentleGuide: An exploration of haptic output for indoors pedestrian guidance," in *Human-computer interaction with mobile devices and services*. Springer, 2003.

[15] A. Censi, "An ICP variant using a point-to-line metric," in *Int. Conf. on Robotics & Automation*, 2008.

[16] R. Morton and E. Olson, "Positive and negative obstacle detection using the HLD classifier," in *Int. Conf. on Intelligent Robots and Systems*, 2011.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Int. Conf. on Robotics & Automation*, 2009.

[18] R. Hogg, A. Rankin, S. Roumeliotis, M. McHenry, D. Helmick, C. Bergh, and L. Matthies, "Algorithms and sensors for small robot path following," in *Int. Conf. on Robotics & Automation*, 2002.

[19] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Int. Conf. on Intelligent Robots and Systems*, 2012.

[20] G. Der and I. Deary, "Age and sex differences in reaction time in adulthood: results from the united kingdom health and lifestyle survey." *Psychology and aging*, vol. 21, no. 1, p. 62, 2006.