University of Nevada, Reno

Exploring the Use of Wearables to Enable Indoor Navigation for Blind Users

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of Doctor of Philosophy in
Computer Science and Engineering

by
Ilias Apostolopoulos
Eelke Folmer, Ph.D., Dissertation Advisor

December 2014

UNIVERSITY OF NEVADA RENO

THE GRADUATE SCHOOL

We recommend that the dissertation prepared
under our supervision by

ILIAS APOSTOLOPOULOS

entitled

Exploring the Use of Wearables to Enable Indoor Navigation for Blind Users

be accepted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

Eelke Folmer, Ph.D. – Advisor

George Bebis, Ph.D. – Committee Member

Kostas Bekris, Ph.D. – Committee Member

Sergiu Dascalu, Ph.D. – Committee Member

Daniel Cook, Ph.D. – Graduate School Representative

David W. Zeh, Ph.D. – Dean, Graduate School

December 2014

ABSTRACT

One of the challenges that people with visual impairments (VI) have to have to confront daily, is navigating independently through foreign or unfamiliar spaces. Navigating through unfamiliar spaces without assistance is very time consuming and leads to lower mobility. Especially in the case of indoor environments where the use of GPS is impossible, this task becomes even harder.

However, advancements in mobile and wearable computing pave the path to new cheap assistive technologies that can make the lives of people with VI easier. Wearable devices have great potential for assistive applications for users who are blind as they typically feature a camera and support hands and eye free interaction. Smart watches and heads up displays (HUDs), in combination with smart phones, can provide a basis for development of advanced algorithms, capable of providing inexpensive solutions for navigation in indoor spaces. New interfaces are also introduced making the interaction between users who are blind and mobile devices more intuitive.

This work presents a set of new systems and technologies created to help users with VI navigate indoor environments. The first system presented is an indoor navigation system for people with VI that operates by using sensors found in mobile devices and virtual maps of the environment. The second system presented helps users navigate large open spaces with minimum veering. Next a study is conducted to determine the accuracy of pedometry based on different body placements of the accelerometer sensors. Finally, a gesture detection system is introduced that helps communication between the user and mobile devices by using sensors in wearable devices.

This is dedicated to my family and Dimitra who have been with me through the good and bad.

ACKNOWLEDGEMENTS

This work was supported by a Google Faculty Research Award.

TABLE OF CONTENTS

Abstract i
Dedication ii
Acknowledgements iii
Table of Contents iv
List of Tables vi
List of Figures vii

1 Introduction **1**
1.1 Contribution 3
1.2 Publications 4

2 Navatar **6**
2.1 Description 6
2.1.1 System Overview and Challenges 7
2.2 Related Work 9
2.3 Approach 11
2.3.1 High-level: Closed loop 11
2.3.2 Cyber-representation 13
2.3.3 Interface 16
2.3.4 Localization 16
2.3.5 Adapting to Different Step Lengths 24
2.3.6 Direction Provision 26
2.4 Experiments 30
2.4.1 Setup 31
2.4.2 Feasibility Study and Evaluating Direction Provision 31
2.4.3 Online Operation with Integrated Localization and Direction Provision Modules 36
2.4.4 Experiments with users with VI 43
2.5 Discussion 50

3 Headlock **52**
3.1 Description 52
3.2 Related work 53
3.3 Design of Headlock 57
3.4 Implementation 60
3.4.1 Discovery 62
3.4.2 Guidance 63
3.5 Evaluation 66
3.5.1 Procedure 67
3.5.2 Results 69
3.6 Discussion & Future work 73

4 Pedometry **76**
4.1 Description 76
4.2 Related work 77
4.3 Experimental Setup 78
4.3.1 Instrumentation 79
4.3.2 Participants 80
4.3.3 Procedure 80
4.4 Results 81
4.4.1 Step counting algorithm 82
4.4.2 Error Extraction 82
4.4.3 Parameter Optimization 83
4.4.4 Accuracy Comparison 84
4.5 Discussion and Future work 85

5 Conclusion **88**

Bibliography **89**

LIST OF TABLES

2.1 Examples of model parameters 18
2.2 Parameters 33
2.3 Average distance between destination and the user's position upon completion (m). 34
2.4 Average path duration (sec). 34
2.5 Average error of dead reckoning in final location (m). 35
2.6 Average error of the proposed interactive localization process (m). . 36
2.7 Distance between ground truth and localization estimate upon completion (m). 40
2.8 Distance between location estimation and ground truth for three different systems (m) 42
2.9 Profiling data (msec) 44
2.10 Distance between ground truth and destination (m) 45
2.11 Profiling data for each path 46
2.12 Distance between destination and the user's position upon completion (m). 48

3.1 Feedback trial results values averaged across each user's 6 sonification trials and 6 TTS trials (* indicates outliers). 69

4.1 Step counting error (%) for individual subjects and positions 83
4.2 Optimal WPD parameter values for walking and running 84

LIST OF FIGURES

2.1 A legally blind individual testing our system. 7
2.2 The communication between the components of the system. 12
2.3 The accessible spaces map for the second floor of the engineering building. Purple indicates the accessible spaces, green indicates doors, and blue indicates elevators and staircases. 14
2.4 Example of a tile storing information about the closest doors and hallway intersections. 15
2.5 An example of pruning based on the direction of the landmark. The upper left particle is pruned because, while there is a door close to it, it is not on the right side. 21
2.6 Example of path correction. 30
2.7 Graph comparison of dead reckoning with the particle filter approach . . . 35
2.8 The map of the first floor 38
2.9 The map of the second floor 38
2.10 A sighted user, running experiments blindfolded wearing the ground truth equipment 39
2.11 Two different paths that lead to the same goal. The alternative path has a higher success probability because it uses less, but more distinctive landmarks. 41
2.12 A graph showing the localization error during the execution of a path. Each line represents a different estimation of the step length. 42

3.1 Finite State Machine describing the different modes and transitions between modes with landmark L. 58
3.2 (a) Without any tactile features to follow in an open space, cane users may to veer from their intended path towards a landmark. (b) Users scan the space for landmarks and select one as their target. (c) Audio feedback is provided to correct for veering during the user's navigation towards the landmark. (d) Updated distance information is provided while approaching the landmark. 59
3.3 Feedback is based on the position of the middle of the field of view (red vertical line in this cropped image) relative to the nearest edge of the bounding box surrounding the target landmark (green box). In this case, HEADLOCK would indicate for the user to go left toward the blue target orientation. 62
3.4 The veering cost function sums the measure of veering at each point along a user's path weighted by the magnitude of $C_{i,i-1}$. 71
3.5 Example traces of paths for both feedback modes. The path is marked as green when $\Theta(i) = 0$, yellow when $\Theta(i) < 10$, and red when $\Theta(i) > 10$. The blue lines represent the vectors A_0 and B_0 (see Figure 3.4). 74

4.1 Sample traces of acceleration magnitudes for a walking and running trial. The top traces are for walking with locations: a) head, b) hand, c) pocket. The bottom traces are for running with locations: d) head, e) arm, f) pocket. 84

CHAPTER 1
INTRODUCTION

Humans tend to navigate by either using path integration or landmark based navigation. Path integration is the process when an individual navigates relatively to a starting position using proprioceptive data. The location is estimated based on adding all the changes in position and orientation [2]. In this case a map of environment is not needed [51] as the individual keeps track of all the changes. With landmark-based navigation, the users find their way around based on their relative position to specific landmarks. Landmark-based navigation requires a map (either physical or mental) of the environment. It is also possible that humans use both path integration and landmark-based navigation [50].

The process of creating a mental map of the environment to help with landmark-based navigation is called cognitive mapping [38]. This process is heavily influenced by vision, which is used to commit to memory landmarks and the layout of the environment very fast [79, 33, 70]. However, since users with visual impairments cannot depend on vision, they use other senses to create a cognitive map. Touch is used to recognise the features in the environment (e.g. window, door, wall) while hearing is used to identify landmarks that create sounds (e.g. air-condition, water cooler). The process of cognitive mapping is similar with both indoor and outdoor spaces.

The first part of this work introduces an indoor localization and navigation system for people with visual impairments [6], called Navatar. This is a lightweight, inexpensive system, utilizing sensors that can be found in smart phones. It operates using virtual maps of the environment, thus deeming expensive physical infrastructure unnecessary. Navatar utilizes probabilistic algorithms, commonly

used in robotics, in order to calculate an estimation of the user's location. This helps compensate for the lack of GPS in indoor environments.

Three sets of experiments were performed in order to test Navatar's success. The reasoning for the first set of experiments was to test how feasible such a system is. In these experiments, users were asked to follow a set of paths while holding a smart phone. The data from the phone sensors were collected and analyzed in order to determine if it is possible to localize users from cheap sensor data.

The second set of experiments were performed by using an online version of the system. Users were asked to follow directions in order to reach a destination. Navatar would dynamically guide the users to a destination by providing turn-by-turn instructions. The localization and navigation accuracy was determined by these experiments.

The final set of experiments were performed on actual users with visual impairments. The process was similar to the previous experiments where users were asked to follow a set of intstructions provided by the phone. This time, users participated in a survey after using Navatar where they could express their opinion about the usability and ease of use of the system.

During the implementation of Navatar, more portable devices started becoming popular. Wearable devices have great potential for assistive applications for users who are blind as they typically feature a camera and support hands and eye free interaction.

Navatar provides accurate indoor navigation but it does not perform well in the case of open space navigation. Due to lack of features that users can sense, it is easy to veer off. In order to solve this problem, a new system was created to

deal with this issue. The new system, named Headlock, uses the Google Glass' camera to detect a landmark across the room and help users follow it. Headlock uses sonification and Text-to-Speech to help users stay in path when they start to veer off.

Another issue with indoor localization that had to be taken into consideration was the accuracy of the pedometry used. Counting the users' steps is very important for Navatar in order to more accurately estimate the position of the user. Brajdic et al [13] showed that, based on the placement of the smart phone (e.g. back pocket, bag), there can be considerable under-counting of the steps. Additionally, Navatar works by making sure that the phone is always held on hand by the user. Moving the phone around, though, can result in reduced step detection accuracy. For these reason, a user study was conducted to test the accuracy of other placements on the body. Due to the recent popularity of wearable devices (fitness trackers, smart watches, HUDs), many people carry additional accelerometers that are always placed in specific places on the body. We conducted a set of experiments in order to determine whether an accelerometer placed on the head (e.g. Google Glass) can be as accurate as an accelerometer placed on the arm (e.g. smart watch) and an accelerometer placed in the front pocket (e.g. smart phone).

1.1 Contribution

In prior research Navatar was developed, which allows for large-scale implementation, as it only uses 2D maps and relies on low-cost sensors available in smart phones. Navatar achieves accurate localization by exploiting the unique sensing capabilities fd blind users by having them confirm the presence of tactile landmarks along the provided path.

A limitation of this approach is that blind users are required to hold the phone in their hands, which impedes their ability to detect landmarks. By porting Navatar to Google GLass, we seek to improve the user interface experience.

This approach opens up new possibilities. Step detection can now move from the phone to Glass. Since its position is always fixed, it can replace the phone, thus freeing the users from carrying the phone in their hands. Additionally, this helps with the issue of navigating through large open spaces. Navatar works better when the user follows a wall. If the user is navigating through an open space, it is harder for the system to provide an accurate localization estimation. Google Glass provides a camera on the same level as the eyes, which can be used to track a landmark across the room. This is implemented by using computer vision algorithms to keep track of an object and by providing auditory cues that help users avoid veering from the desired path.

1.2 Publications

The work presented in chapter 2 has resulted in the following publications:

1. Ilias Apostolopoulos, Navid Fallah, Eelke Folmer, and Kostas E Bekris. Feasibility of interactive localization and navigation of people with visual impairments. In 11th IEEE Intelligent Autonomous Systems (IAS-10), Ottawa, CA, August 2010 2010.
2. Ilias Apostolopoulos, Navid Fallah, Eelke Folmer, and Kostas E Bekris. Integrated online localization and navigation for people with visual impairments using smart phones. ACM Transactions on Interactive Intelligent Systems

(TiiS), 3(4):21, 2014.

3. Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer. The user as a sensor: navigating users with visual impairments in indoor spaces using tactile landmarks. In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, CHI 12, pages 425-432, New York, NY, USA, 2012. ACM. [23% acceptance rate]

4. Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer. "Indoor human navigation systems: A survey." Interacting with Computers 25, no. 1 (2013): 21-33.

The work presented in chapter 3 has resulted in the following publications:

5. Alexander Fiannaca, Ilias Apostolopoulos, and Eelke Folmer. "Headlock: a wearable navigation aid that helps blind cane users traverse large open spaces." In Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility, pp. 323-324. ACM, 2014. [26% acceptance rate]

The work presented in chapter 4 has resulted in the following publications:

6. Ilias Apostolopoulos, Daniel Coming, and Eelke Folmer. "Accuracy of Pedometry Using a Head-mounted Display." CHI 2015. (awaits approval)

CHAPTER 2

NAVATAR

2.1 Description

Sighted people can navigate environments by primarily relying upon their visual senses to find their way. Individuals with visual impairments (VI) have to rely upon their compensatory senses (e.g., touch, sound) for way-finding, resulting in reduced mobility [12]. Unfamiliar environments are especially challenging as it is difficult to build a mental map of the surroundings using non-visual feedback [46].

To increase the mobility of individuals with VI various navigation systems have been developed. There are many solutions for outdoor navigation systems [52, 68] which depend on the Global Positioning System (GPS). Nevertheless, GPS signals cannot be received in indoor environments. Existing indoor navigation systems typically rely upon augmenting the physical infrastructure with identifiers such as RFID tags [42, 9, 3].

While RFID tags might be cheap, a large amount of them is required to cover a whole building. It is possible to install tags under carpets, but hallways or large open spaces with concrete floors or tiles make such installation challenging. Other solutions employ laser sensors [27, 26] or cameras [78]. While these methods have led to sophisticated state estimation algorithms, they can be expensive, cumbersome, and computationally demanding alternatives.

Figure 2.1: A legally blind individual testing our system.

2.1.1 System Overview and Challenges

This work describes an inexpensive, computationally minimalistic solution that does not require the augmentation of the physical infrastructure and depends on cheap, light-weight sensors, such as an accelerometer and a compass, that are already available on popular devices, such as smart-phones. The goal is to create a computationally lightweight navigation tool that can operate on an everyday hand-held device. The proposed system is a high-level navigation solution and not an obstacle avoidance approach, as people with VI can already accomplish this with the use of a guiding cane or a guide dog. The method utilizes interaction with the user through an audio/speech interface to provide directions using tactile landmarks common to every building and easily recognizable by individuals with VI, such as doors, hallway intersections, and floor transitions. The user con-

firms the presence of the landmarks along the provided path by interacting with a phone. This allows the system to track the user's location by using the sensor readings, knowledge of the indoor environment, and the user's landmark confirmations.

The proposed system has to deal with uncertainty at multiple levels of its operation. Uncertainty arises from the behavior of the user, e.g., how quickly does the person move, how accurately does one person turn when instructed to do so, how good is the person at identifying landmarks. For instance, users cannot easily distinguish landmarks of the same type. When a user confirms a door, and there are multiple doors close to each other, it is possible that the user did not confirm the correct one. The system has to take into consideration the possibility that the user confirmed an adjacent landmark of the same type. Furthermore, the system has to deal with uncertainty due to the sensors, as portable devices usually make use of low accuracy modules. An additional complexity is that the proposed system can only utilize landmark observations infrequently, so as to keep the cognitive load of the human user at an acceptable level. This is in contrast to localization solutions in other domains, such as robotics, where the system can frequently gather observations.

The approach described in this work takes into consideration the above sources of uncertainty and provides an integration of localization and path planning primitives. Different variations of filtering algorithms are considered and evaluated in terms of their ability to adapt simultaneously the step length of the user as well as provide an accurate localization estimate. To provide directions to the user, a path is created from the start to goal using the A* algorithm. Then, incremental directions are provided based on the location estimation of the user generated by

the localization module. Experimental results show that this system is capable of successfully locating and guiding the user to the desired destination when a path with sufficiently identifiable landmarks is provided.

2.2 Related Work

Certain assistive devices focus on local hazard detection to provide obstacle avoidance capabilities to users with VI [72, 82]. Navigation systems, however, must be able to locate the user and provide directions to a user-specified destination. To surpass this issue, alternative localization techniques have been developed:

A) Dead-Reckoning techniques integrate measurements of the human's motion. Accelerometers [14] and radar measurements [80] have been used for this purpose. The problem, though, is that without any external reference, the error in dead-reckoning grows unbounded.

B) Beacon-based approaches augment the physical space with identifiers. Such beacons could be retro-reflective digital signs detected by a camera [78], infrared [68] or ultrasound identifiers [64]. A popular solution involves RFID tags [42, 9, 3]. Nevertheless, locating identifiers may be hard, as beacons may require line of sight or close proximity to humans. Other beacons, such as wireless nodes [66, 45, 44], suffer from multi-path effects or interference. Another drawback is the significant time and cost spent installing and calibrating beacons.

C) Sensor-based solutions employ sensors, such as cameras [39], that can detect pre-existing features of indoor spaces, such as walls or doors. For instance, a multi-camera rig has been developed to estimate the six degrees of freedom (DOF)

pose of people with VI [16]. A different camera system matches physical objects with objects in a virtual representation of the space [28]. Nevertheless, cameras require good lighting conditions, and may impose a computational cost prohibitive for portable devices. An alternative makes use of a 2D laser scanner [27, 26]. This method achieves 3D pose estimation by integrating data from an inertial measurement unit (IMU), the laser scanner, and knowledge of the 3D structure of the space. While this setup has led to sophisticated algorithms for 3D pose estimation and laser scanners that can robustly detect low-level features, it depends on relatively expensive and heavy sensors, which might impede mobility of users with VI.

The proposed approach is also a sensor-based solution. It employs the user as a sensor together with information from light-weight, affordable devices, such as an accelerometer and a compass. These sensors are available on smart phones and it is interesting to study the feasibility of using such popular devices to (i) interact effectively with a user with VI; and (ii) run localization primitives in real-time given their limited resources. Indoor localization using mobile devices has been reported for robots [40]. Despite the fact that the current work has to deal with a more challenging system, it is possible to achieve high accuracy. To achieve this objective under the minimalistic and noisy nature of the available sensors, this work utilizes probabilistic tools that have been shown to be effective in robotics and evaluates their efficiency for different forms of direction provision.

Bayesian methods for localization work incrementally, where given the previous belief about the agent's location, the new belief is computed using the latest displacement and sensor reading. An important issue is how to represent and store the belief distribution. One method is the Extended Kalman filter (EKF) [35, 74], which assumes normal distributions. While Kalman filters provide a compact rep-

resentation and return the optimum estimate under certain assumptions, a normal distribution may not be a good model, especially for multi-modal distributions. An alternative is to use particle filters [21, 77, 47, 65, 62, 76], which sample estimates of the agent's state. Particle filters are able to represent multi-modal distributions at the expense of increased computation. Such distributions arise often in the target application, such as when a door is confirmed, where the belief increases in front of all of the doors in the vicinity of the last estimate. Thus, particle filters appear as an appropriate solution. This work shows that it is possible to achieve a sufficient real-time solution with particle filters.

The user's step length changes dynamically and these changes have to be taken into consideration during path execution. To achieve this, this project is building upon work in multi-model state estimation [58, 59, 48, 73]. Multi-model estimation is commonly used to calculate both the state and the model of the system when the latter is changing.

2.3 Approach

2.3.1 High-level: Closed loop

Tactile landmarks, such as doors, hallway intersections, or floor transitions, play an important role in the cognitive mapping of indoor spaces by users with VI [33, 79]. By incorporating the unique sensing capabilities of users with VI, the system aims to provide guidance in spaces for which the user does not have a prior cognitive map. The system assumes the availability of a 2D map with addressing information (room numbers) and landmark locations. Then, it follows these steps:

1. The user's initial location is provided with some noise. The desired destination is also provided to the system.
2. Given landmarks identifiable by users with VI on the map, the system computes the shortest path using the A* algorithm and identifies landmarks along the path. The path is translated into a set of directions.
3. Then, incremental directions are automatically provided to the user to reach the next identifiable landmark. The user presses the screen on the phone after successfully executing each direction.
4. Directions are provided iteratively upon the confirmation of each landmark, or automatically when the user is not following the appropriate path according to the localization algorithm. The phone's speech output interface is used for the communication of the direction.

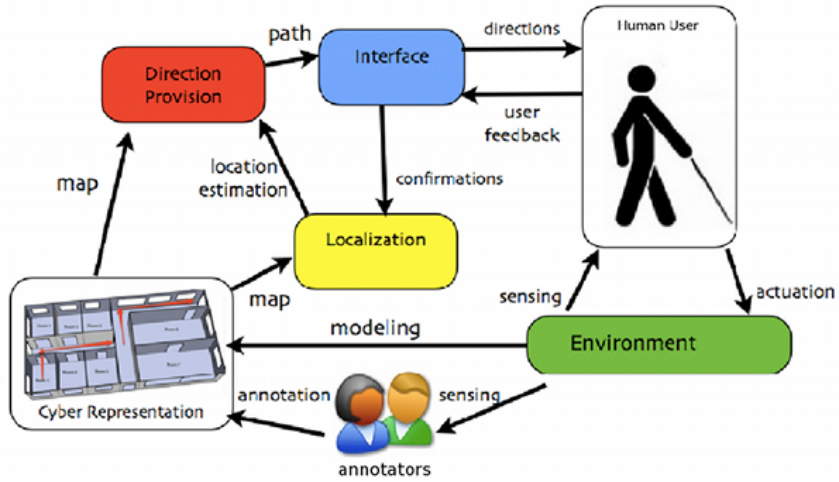


Figure 2.2: The communication between the components of the system.

Figure 2.2 lists a high-level overview of the four different components of the system: (1) the cyber-representation component stores annotated models of indoor environments; (2) the localization component estimates the location of the user with VI; (3) the direction provision component generates directions towards a

user specified destination; and (4) the interface component interacts with the user. The focus of this work is primarily on the localization and direction provision components.

The landmarks considered by the system are features that can be found in most indoor environments: doors, hallway intersections, floor transitions, water coolers, ramps, stairs, and elevators. These landmarks are easily recognizable from users with VI by using touch and sound, thus limiting the need for additional physical infrastructure.

2.3.2 Cyber-representation

Because of the limited processing power of portable devices, some of the most computationally expensive operations were implemented during an offline process. This procedure is used to create appropriate map information and integrate additional data in order to be easily accessible in a timely manner during the execution of the system. Maps can be easily augmented with landmark information by a sighted user without the need for additional sensors. For instance, the map can be created by overlaying an architectural blueprint. There are also options to add landmarks on the map by selecting the appropriate type and draw it on the map. This is an easy process and was used to create the maps for the experiments.

During the offline process, the following three steps are executed:

1. Given the map M and some accessible points $\{a_1, \ldots, a_n\}$, the offline process finds the accessible spaces $A = \{A_1, \ldots, A_m\}$ (purple area in Fig. 2.3) of the map by using an eight neighbor flooding algorithm. The accessible spaces A

typically represent hallways and spaces where the user can navigate through. In the experimental process for this work, only corridors where considered. The accessible spaces A are represented as bitmaps with ones for accessible points and zeros for inaccessible ones.

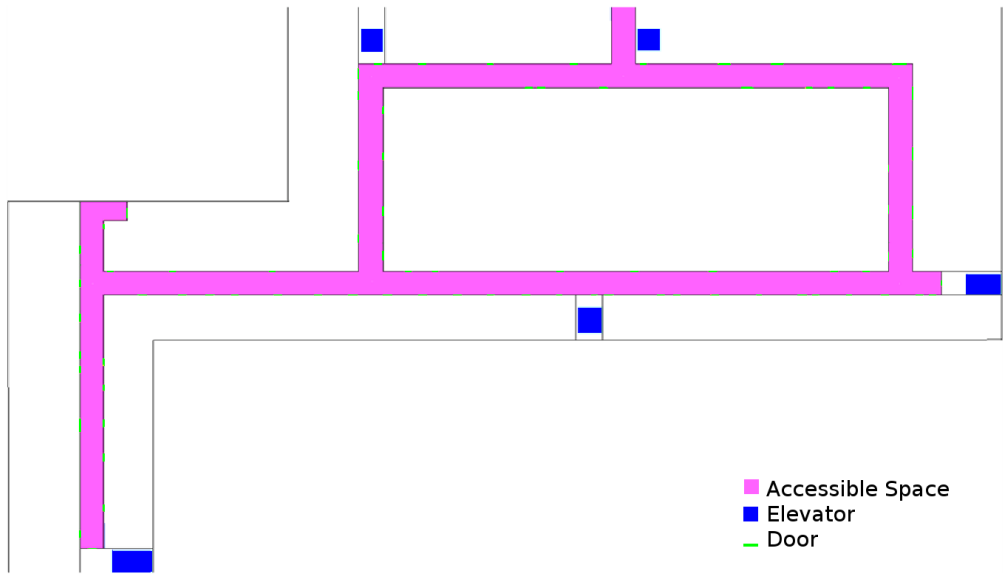


Figure 2.3: The accessible spaces map for the second floor of the engineering building. Purple indicates the accessible spaces, green indicates doors, and blue indicates elevators and staircases.

2. The accessible spaces map M_A is used in order to create a set of pre-stored positions $r_i(l_i)$ in the vicinity of each landmark l_i in the accessible space (i.e., within the observation range of the landmark). The positions are randomly sampled around each landmark l_i and assigned to this landmark in the map data structure M.

3. A lower resolution map m (i.e., a mini-map version of M_A) is created by dividing the accessible spaces map M_A into non-overlapping tiles m_{pq} where each one has a size of one square meter. p represents the row the tile is on and q represents the column. Only tiles m_{pq} that overlap with accessible areas

A are maintained. Each accessible tile m_{pq} holds information for all types of landmarks $\{L_1, \ldots, L_k\}$ close to it (Fig. 2.4). For each landmark type L_j, up to five landmarks in the vicinity of a tile (up to 18m away in the experiments) are stored in each m_{pq}. Weights $W = \{w_1, \ldots, w_i\}$ inversely proportional to the distance of landmark l^i to the center of the tile m_{pq} are also stored together with each landmark. These weights are used during the online process when the user confirms a landmark. The weights represent the probability of the user to be close to any of these landmarks. When all accessible tiles m_{pq} are created, they are stored in the map data structure M.

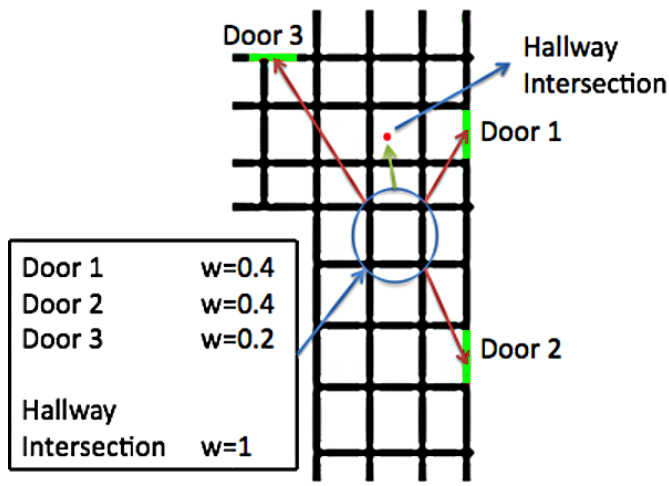


Figure 2.4: Example of a tile storing information about the closest doors and hallway intersections.

By storing the specified offline information, the online process avoids searching for landmarks when required to reset the states of the particles as this might become computationally expensive. The proposed offline process may be computationally heavy for a smart phone but it can be executed on a desktop computer, where it takes less than a second for environments considered in this work.

2.3.3 Interface

The application provides directions through text to speech using the smartphone's speaker and the user confirms the execution of a direction (e.g., "Turn left") by tapping the screen. Because touch screens are notoriously inaccessible to users with visual impairments a successful tap was indicated using a vibrotactile cue. Two types of directions were tested in order to make the application more accessible, landmark and metric-based. In landmark-based directions, the user is asked to find and confirm a type of landmark, while in metric-based directions the user is asked to walk a number of steps before proceeding to the next direction. Additionally, the distance that each direction covers was tested in order to find out what kind of directions are faster to execute and guide the user more successfully to the desired destination. In some settings, the user was asked to confirm a landmark that was a few meters away, while in others, the instructions guided the user from one hallway intersection to the next.

2.3.4 Localization

Consider a planar system moving among n static landmarks. The planar system is a human with VI, and the landmarks correspond to tactile features of indoor spaces. Let $\xi = (x^i, y^i, \theta)$ denote the state of the system and i represent the floor the user is on. The map m of the world is available and stores a function, which returns whether each point (x^i, y^i) is occupied by an obstacle or not. The map also stores the n landmarks present in the world. The landmarks belong to k different types $\{L_1, \ldots, L_k\}$, such as doors, hallway intersections or floor transitions (most often $k < n$). Landmarks l_j in the same class L_j are indistinguishable to the human

user.

The data $d_T = (o(0{:}T), u(0{:}T-1))$ available to the system up to time T are transitions $u(0{:}T-1)$ and observations $o(0{:}T)$. A transition $u_t = (u_t^f, u_t^\theta)$ at time t corresponds to a motion where the agent acquires the global orientation u_t^θ and moves forward u_t^f. This transition determines the kinematic transition model of the system:

$$\xi_{t+1} = (x_t^i + u_t^f \cdot cos(u_t^\theta), y_t^i + u_t^f \cdot sin(u_t^\theta), u_t^\theta) \tag{2.1}$$

In the case that the user changes floor then the transition is denoted as:

$$\xi_{t+1} = (x_{t+1}^j, y_{t+1}^j, u_t^\theta) \tag{2.2}$$

where j represents the new floor and (x_{t+1}^j, y_{t+1}^j) is the point from where the user enters the floor.

In this application the translation is measured with an accelerometer and the orientation with a compass. An observation o_t^j of a landmark type L_j from state $\xi_t = (x_t, y_t, \theta_t)$ implies:

$$\exists\, l_j \in L_j : \|(x_t, y_t), l_j\| < R_{obs}. \tag{2.3}$$

The above observation model specifies that a user can sense a landmark type L_j in one's vicinity, only if such a landmark $l^i \in L_j$ is within a predefined observation distance R_{obs} from the current coordinates of the system (x_t, y_t).

The objective is to be able to incrementally estimate the user's state ξ_T at time T. The general Bayes filter computes a belief distribution $B_T = P(\xi_T|d_T)$ at time T over ξ_T given the data d_T. The computation requires:

a) an initialization B_0,

Table 2.1: Examples of model parameters

Transition Model	**Observation Model**
Average Step Length	Landmark Identification Accuracy
Step Detection Accuracy	Distance from Landmark upon Confirmation
Turning Accuracy	Confirmation Efficiency

b) a transition model $P(\xi'|u, \xi)$, which describes the probability the user is at ξ' if it was at ξ and transitioned by u, and

c) the observation model $P(o|\xi, m)$ which describes the likelihood of observing o when the user is at ξ and given the map m. The map is assumed static and correct in this work.

The input to these models corresponds to raw data from the accelerometer and compass, which are then filtered to detect when the user has made a step and/or changed direction. Table 2.1 provides examples of potential parameters that need to be extracted from the accelerometer or the compass in order to build appropriate transition and observation models. It is important for the models to be able to adapt to different users. This is especially critical in this application, as different users will also have different types and degrees of visual impairments. There is also information provided by the user when a landmark is confirmed.

Then given a normalization factor η the belief distribution can be updated as follows:

$$B_T = \eta \cdot P(o_T|\xi_T, m) \int P(\xi_T|u_{T-1}, \xi_{T-1}) \cdot B_{T-1} \cdot d\xi_{T-1}$$

The computational cost of integrating over all states renders the explicit computation of the above equation inefficient. Most online algorithms simplify the problem by approximating the above equation. This work follows a Particle Filter approximation.

Particle Filter It is possible to represent B_T through a set of particles $p^i = (\xi^i, w^i)$ ($i \in [1, N]$). Each particle stores a state estimate ξ^i together with a weight w^i, representing the probability of ξ^i being the true state. As the number of particles increases, the better the particle filter represents the belief distribution. In order to update the particle filter given a new transition and an observation, this work follows an approach similar to importance sampling [21]. At each time step T, given a particle population $\{p_T^1, \ldots, p_T^P\}$, a transition u_T and an observation o_{T+1}, the following steps are executed:

A. For each particle $p_T^i = (\xi_T^i, w_T^i)$

- **i.** Sample the transition model $P(\xi_{T+1}^i | u_T, \xi_T^i)$ to acquire: ξ_{T+1}^i.
- **ii.** Sample the observation model to acquire the new weight $w_{T+1}^i = P(o_{T+1} | \xi_{T+1}, m)$.

B. Sample a new population of particles p^i given the weights w_{T+1}^i

Transition Model The approach collects all the pedometer and compass readings that have been produced during the last time step. During one iteration of the algorithm, the pedometer typically returns that either zero or one step has been executed. This value has to be translated into a distance estimate. If a step is taken, the distance that was covered is the average step length that is calculated for the user (Alg. 1). In case a step was not detected, some noise is introduced in order to scatter the particles around the location estimation of the user. This happens in order to improve the probability of having a good location estimation of the user. Because the compass is highly erroneous due to potential metal structures in buildings, the system does not use the raw compass readings. Instead it categorizes them into eight discrete directions based on which one is closer to the

raw reading. The eight discrete directions are multiples of 45 degrees. The system, also, takes into consideration the noise from the compass by introducing a 10% probability that the user is actually facing to one of the adjacent directions of the one that the compass shows. This is implemented in line 11 of Alg. 1 with *PickDirection* function.

Algorithm 1: Transition Model

Input: ξ_t particle state, u_t transition, sl step length, stDev step standard deviation
Output: ξ_{t+1} new state

1: $steps \leftarrow u.steps$
2: **if** $steps > 0$ **then**
3: $\quad dist \leftarrow 0$
4: $\quad$ **for** $i = 1$ to $steps$ **do**
5: $\quad\quad dist \leftarrow dist + random() * stDev + sl$
6: **else**
7: $\quad dist \leftarrow random() * stDev$
8: **if** $dist < 0$ **then**
9: $\quad dist \leftarrow 0$
10: $\xi_{t+1}.\theta = PickDirection(u_t^{\theta}.direction)$
11: $\xi_{t+1}.x = \xi_t.x + dist * cos(\xi_{t+1}.\theta)$
12: $\xi_{t+1}.y = \xi_t.y + dist * sin(\xi_{t+1}.\theta)$
13: **return** ξ_{t+1}

Observation Model Initially, a check is performed regarding the validity of the new state of the particle. In case that a particle has reached an inaccessible state, like passing through a wall, a weight of zero is assigned in order to eliminate this particle. After that, there are two cases for computing the weights w_{T+1}^i of the particles. If there was no landmark confirmation by the user during the last step, then all particles keep their old weights. Otherwise, the system expects a specific type of landmark to be in the vicinity of the user. If the user confirmed the presence of a landmark of type L_j, then the approach searches for every particle the closest landmark of this type and prunes all the particles that are not close to such a particle.

The particles are further pruned based on the direction of the landmark relative to the user's estimated orientation. The directions provided to the user specify in which direction the landmark will be found (e.g., "a door on your right"). The particles that have the landmark on the wrong side of the wall based on the particle's estimated orientation get a zero weight (Alg. 2).

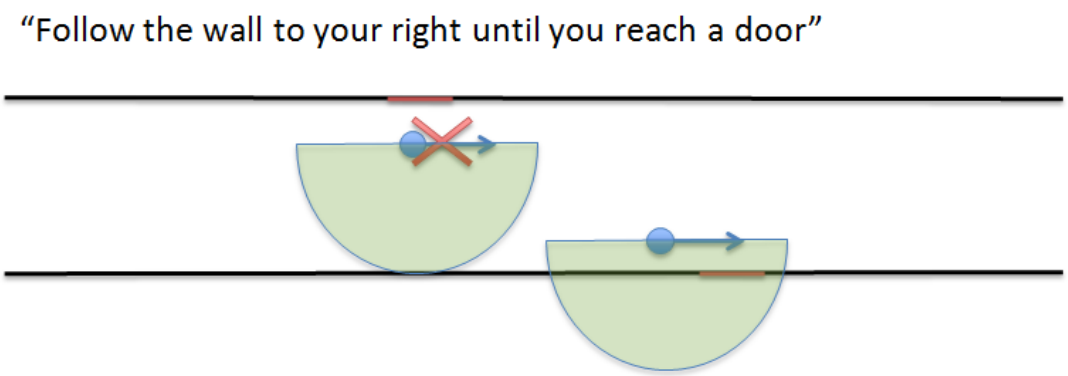


Figure 2.5: An example of pruning based on the direction of the landmark. The upper left particle is pruned because, while there is a door close to it, it is not on the right side.

Algorithm 2: Observation Model

Input: ξ_{t+1} new state of particle, l landmark confirmed, M_A accessible map
Output: w^{i+1} weight of particle

1: **if** $M_A.isValid(\xi_{t+1}) = FALSE$ **then**
2: **return** 0 {New state is inaccessible}
3: **if** $l = 0$ **then**
4: **return** 1 {No landmark confirmation on this transition, leave all weights as they are}
5: $closestLandmark \leftarrow M_A.getClosestLandmark(\xi_{t+1}, l)$ {Find closest landmark of type l closest to state ξ_{t+1}}
6: $dist \leftarrow distance(closestLandmark, \xi_{t+1})$
7: $landmarkDir \leftarrow direction(\xi_{t+1}, closestLandmark)$ {Returns left or right}
8: **if** $landmarkDir = l.dir$ **then**
9: **return** $1 - dist/2$
10: **return** 0 {The closest landmark is more than 2m away}

Sampling During the sampling process, every new particle is sampled from the previous population according to the probability distribution defined by the weights

of the particles. If at some point all particles die after a landmark confirmation, the offline data are used to reset the particles. In this case, it is necessary to reset the particles in the vicinity of landmarks of the type L_j that was confirmed [4]. When a specific particle p_i needs to be resampled close to a landmark of type L_j, the system first checks the current location of p_i. It will then find the corresponding tile m_{pq} in the lower resolution map m that p_i is currently located in and retrieve the list with the landmarks of type L_j assigned to this tile. A landmark l_j from the list is randomly selected with a probability proportional to the weights W. The weights W for each landmark from the list represent the probability that the user might actually be close to any of these landmarks instead of the particle's location estimate. The location estimate of the particle is replaced randomly by one of the positions (from step 2 above) in the vicinity of the selected landmark l_j (Alg. 3). If all the particles died when there was no landmark confirmation, the particles backtrack to their previous state.

Location Estimation After the sampling process is complete, a k-means algorithm [24] is used in order to estimate the user's current location over all particle estimates. Based on the cluster with the highest concentration of particles. Thus, particles that don't have a correct step length estimation or particles that have been isolated from the rest due to noise, are not taken into consideration. The k-means algorithm (Alg. 4) classifies the particles p^i into clusters $C = \{c_1, ..., c_k\}$. The number of clusters k changes dynamically based on how concentrated the particles are. The k-means algorithm starts with three clusters in the implementation accompanying this work. After this initial classification, the algorithm checks the distances of the centers of the clusters pairwise. If the centers of two clusters $c_i, c_j (i \neq j)$ are closer than a threshold δ_{min} then the two merge into one cluster c_i. On the other hand, if a

Algorithm 3: Sampling

Input: p set of particles, l landmark confirmed, M_A accessible map

```
1:  w^1 ← p^1.w
2:  for i = 2 to size(p) do
3:      w^i ← p^i.w + w^(i−1)
4:  if w^size(p) = 0 then
5:      if l > 0 then
6:          for i = 1 to size(p) do
7:              closestLandmark ← M_A.getClosestLandmark(p^i.ξ_t, l) {In case that all
                particles die, they get replaced with the particles stored in the
                closest landmark}
8:              p^i ← closestLandmark.particles[random()]
9:      else
10:         for i = 1 to size(p) do
11:             p^i.ξ_t ← p^i.ξ_(t−1) {In case that all particles die and there was no
                landmark confirmation}
12: else
13:     for i = 1 to size(p) do
14:         selection ← random()
15:         j ← 1
16:         while selection < w^j do
17:             tmpParticle ← p^j
18:             j ← j + 1
19:         p^i ← tmpParticle
```

cluster's radius gets bigger than a given threshold $delta_{max}$ then the cluster c_i splits into two clusters c_i, c_{k+1}. Thus, if the particles are concentrated in one place, only one cluster can be used, while if they are dispersed throughout the environment more clusters are used. After classifying all the particles, the algorithm selects the cluster c_i with the highest accumulated weight and returns the center for this cluster c_i as a location estimate.

Algorithm 4: K-Means Algorithm

Input: p set of particles
Output: C set of clusters, M set of clusters' centers

```
K ← 3
converged ← 0
for i = 1 to K do
    M_i ← random(p)
while converged ≠ K do
    for i = 1 to sizeOf(p) do
        closestCluster ← getClosestCluster(p^i)
        closestCluster ← {closestCluster, p^i}
    converged ← 0
    for i = 1 to K do
        newMean ← calculateCenter(C_i)
        if radius(C_i) > d_max then
            p* ← findFurthestParticle(C_i)
            C_i ← {C_i}\p*
            C_{K+1} ← {p*}
            C ← {C, C_{K+1}}
            K ← K + 1
        closestCluster ← getClosestCluster(C_i)
        if distance(C_i, closestCluster) < d_min then
            C_i ← {C_i, closestCluster}
            K ← K − 1
        if distance(newMean, M_i) < converge_thres then
            converge ← converge + 1
        M_i ← newMean
return C, M
```

2.3.5 Adapting to Different Step Lengths

It is highly desirable for the localization process to be able to simultaneously estimate the user's step length as it computes a location estimate. This is important since the step length is used for the transition model and might be different from user to user, from path to path, or even within the same path.

To accomplish that, three different variations of particle filtering techniques were implemented:

A. A simple benchmark particle filter where the standard deviation for the noise in the step length estimate was high enough so as to include all possible step lengths for different users. This corresponds to the algorithm described in the previous section.

B. A particle filter which includes a step length estimate as part of the particle's state, which is allowed to alter between consecutive steps. The standard deviation for how much the step length parameter changes between transitions is set to a small value. The particles are initialized so that they cover all possible step lengths for different users. In this case, each particle advances based on the local estimate it keeps internally for the user's step length. The step length estimation is being updated so as to allow the approach to detect changes in the user's step length during the execution of a path. After a landmark confirmation by the user, only the particles that estimate the user's location to be in the vicinity of such a landmark will survive. In this way, only a subset of the user's step length estimations survives. If all particles die, they get resampled as described in section 2.3.4 under *sampling*. In this case, the particles' step length values are set again to the initial values.

C. A technique that utilizes ten particle filters, where the total number of particles is equal to the approaches above. Each particle filter has a different mean estimate for the step length and the standard deviation is somewhere in between the above two methods. The particles in each particle filter advance based on the step length estimation of the particle filter they belong to. After a landmark confirmation by the user, the particle filters with the correct step length estimation will survive while the rest will be replaced with a variation of the particle filters that survived. If all the particles of a filter die, they are replaced with the particles of the most successful filter. This is the filter that

has the highest sum of particles' weights. The filter's step length changes to the step length of the most successful particle filter with a small Gaussian noise added. If all particles die, they get resampled as described in section 2.3.4 under *sampling*. The particle filters' step length values are set again to the initial values (Alg. 5).

The objective with the last two alternatives is that the system will build an increasingly more accurate estimation of the user's step length. In particular, for the third method, the particle filter with the highest sum of particle weights is chosen at the beginning of the sampling process. This is computed by summing the weights of all particles in each filter and choosing the filter with the maximum accumulated weight. Within each particle filter, the particles with the higher weights are the ones sampled. In case that all particles in a particle filter die, the whole particle filter copies the position estimates from the particle filter with the highest weight. It also copies the mean of the step length distribution and alters it slightly given a small standard deviation, in order to better estimate the user's step length. The exact parameters used for the experiments accompanying this work are provided in the experimental section.

2.3.6 Direction Provision

The A* algorithm is used for path planning purposes. A* is using the minimap m, which is a grid representation of the environment, to plan the path. The path can be planned based on different criteria, such as the shortest travel time or the probability of success. Additional or alternative objectives can also be considered for the path. In particular, the path should be easy to follow by a person with VI,

Algorithm 5: Sampling 10 Particle Filters

Input: P set of particle filters, l landmark confirmed, M_A accessible map, stDev step standard deviation

```
bestWeight ← 0
bestPF ← ∅
for i = 1 to 10 do
    Weights_i ← 0
    for j = 1 to size(P^i) do
        Weights_i ← P^i.p^j.w + Weights_i
        if Weights_i > bestWeight then
            bestWeight = Weights_i
            bestPF = P^i
for i = 1 to 10 do
    if bestPF = ∅ then
        for j = 1 to size(P^i) do
            if l > 0 then
                closestLandmark ← M_A.getClosestLandmark(P^i.p^j.ξ_t, l)
                P^i.p^j ← closestLandmark.particles[random()]
            else
                P^i.p^j.ξ_t ← P^i.p^j.ξ_{t−1}
    else if Weights_i = 0 then
        P^i = bestPF
        P^i.step = bestPF.step + random() ∗ stDev
    PFWeights_1 ← P^i.p^1.w
    for j = 2 to size(P^i) do
        PFWeights_j ← P^i.p^j.w + PFWeights_{j−1}
    for j = 1 to size(P^i) do
        if P^i.p^j.w = 0 then
            selection ← random()
            k ← 1
            while selection < PFWeights_k do
                tmpParticle ← P^i_j
                k ← k + 1
            P^i.p^j ← tmpParticle
```

reduce localization uncertainty, and ensure the user's safety.

After the shortest path is calculated, it is then translated into directions. Each direction represents either a straight segment on the path or a turn. The closest landmark to the end of a segment is chosen as the landmark to be found by the user.

The number of landmarks and the frequency with which commands are provided can affect the performance and usability of the system. It might appear that requiring the user to confirm all the landmarks along a path can result in a more accurate localization. Nevertheless, an initial set of experiments indicated that the uncertainty can be reduced by using more distinctive and unique landmarks, such as hallway intersections rather than doors, which can easily be missed. Furthermore, planning a path that guides the user along walls and avoids open areas will help the user to maintain one's orientation and lower uncertainty. To ensure safety, the path avoids obstacles or hazardous areas such as ramps with no handrail. The landmarks specified in the directions must be safe to touch as the user will confirm their existence using a cane or a hand.

In case the user didn't follow the directions correctly and walked away from the prespecified path, the system is able to detect this and issue new directions. The system calculates the location estimate after every step and checks the distance from the path. If the distance is longer than a threshold (two meters in the experiments) then the system informs the user with audio feedback, calculates a new path based on the latest location estimate and provides to the user a new direction. Fig. 2.6 describes such a scenario. The user did not turn on the second hallway intersection, but instead continued straight. After a few steps, the system detected that the user was walking away from the path, informed him and provided new

directions leading to the desired destination. The set of directions provided to the user was:

- "Follow the wall to your right until you reach a hallway intersection"
- "Turn right"
- "Follow the wall to your left until you reach a hallway intersection"
- "Turn left"

Here the user fails to follow the direction and does not turn

- "Follow the wall to your right until you reach the second door"

The user keeps walking straight. After a few meters the phone detects that the user is not on the path and provides new directions.

- "Off course. Recalculating the path"
- "Turn around"
- "Follow the wall to your right until you reach a hallway intersection"
- "Turn right"
- "Follow the wall to your right until you reach the second door"
- "You have reached your destination"

In the end, the user completed the path successfully.

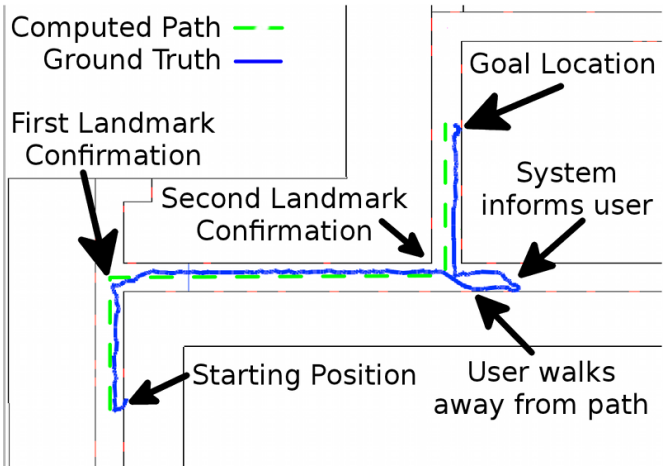


Figure 2.6: Example of path correction.

2.4 Experiments

Three major sets of experiments were executed in order to guide the development of the proposed system. The first set of experiments was executed in order to determine if it is feasible to implement such a navigation tool for a portable device and what type of direction provision yielded the best results. After confirming that it is feasible to implement such a system, many improvements were done on the localization module. A second set of experiments was executed in order to find out if it is possible to successfully localize and guide a user online and integrate it with dynamic direction provision. In these experiments, different localization techniques were tested in order to determine which one can adapt to the user's step length better. Finally, after determining that it is possible to successfully guide a user with VI to a desired destination, a third set of experiments was performed with the help of users with VI.

2.4.1 Setup

The system was implemented as a Java application for the open-source Google Android smart phone (ver. 1.6). A map of a buildings floor on the engineering building of the University of Nevada, Reno was created in the Keyhole Markup Language (KML) using Google SketchUp. A 2D map is extracted from the 3D model and loaded to the application (Fig. 2.9). The map was manually augmented with the following landmarks: (i) three water coolers, (ii) one floor transition marked by a metal strip, (iii) three hallway intersections, (iv) two hallway turns and (v) 72 doors. Prior to each study the users followed one path so that they would become familiar with the phone and the direction provision process.

2.4.2 Feasibility Study and Evaluating Direction Provision

The purpose of these experiments was to determine the feasibility of guiding a user with VI through an indoor environment using a minimalistic and interactive sensing approach achievable on a smart phone.

Setup Five different paths were defined along the corridors of the building. For each path, there are two alternatives for directions, with three levels of granularity each, as specified in Sec. 2.3.3. Overall, six different ways to provide directions were tested per path. During these experiments, the algorithm was not running on the phone. Instead, the data were collected during the execution of a path and were later processed offline. The directions provided to the user were fixed for each path instead of providing commands based on the user's location estimation. Finally, the user's step length was calculated before the experiments by executing a test

path with prespecified length. The system calculated the user's stride length by dividing the distance of the path with the number steps the user took to complete it.

Participants Ten volunteers were involved in the experimental session. Users held the phone in their hand while holding a cane in their other (Fig. 2.1). One of the volunteers was legally blind. This individual pointed out landmarks, such as a metal strip on the floor, which sighted people typically ignore. Nine more volunteers were involved that were sighted users and who were blindfolded during the experiments. Some of the users had visited the building in the past and were aware of its structure, while others didn't. This discrepancy did not appear to considerably influence the efficiency of users in reaching the desired destination. Each user executed ten traversals, which corresponded to two traversals per path using different types of directions.

Ground Truth To measure the true position of the user, an observer was recording the users motion. This was achieved by placing markers on the floor every two meters. Every time the user was crossing a marker, the observer was recording the time on a second smart phone. To recreate the true path, the assumption was that the user moves with constant speed between markers. Thus, the resolution of the ground truth was two meters for this set of experiments.

Parameters The following table provides the parameters of the experimental process. A relatively high standard deviation for the orientation parameter in the transition model was chosen because of the unreliable nature of the compass. A very small number of particles (20) was used to see if it possible to localize the

user accurately and deal with the low processing power of the portable devices. Recording the status of the application (e.g., saving all the measurements, landmark confirmations and the particle filter state) took three times longer than the actual estimation by the particle filter.

Table 2.2: Parameters

Number of Particles P	20
Landmark radius R_{obs}	1 meter
Standard Deviation in Orientation σ_θ	30°
Standard Deviation in Forward Motion σ_f	0.2 meters
Maximum Number of Tries To Find a Collision Free Transition	5

Success Ratio of Direction Provision Table 2.3 provides the average distance between the destination and the actual position achieved by the user over all experiments of the same type. This table shows that most of the paths were completed successfully. In particular, in 84% of the experiments the distance between the desired destination and the achieved position was less than 2 meters, which was the resolution of the ground truth. In 92% of the experiments the error was less than 3.5 meters. It also turns out that landmark-based directions resulted in smaller errors and higher success rates. Table 2.4 provides the average duration of a path until completion. The users were able to complete paths quicker when they were not asked to confirm a large number of landmarks, which was the expected result ("No Max" case in direction provision).

Localization Accuracy Tables 2.5 and 2.6 provide the errors for dead reckoning and the proposed particle filter based approach. Dead reckoning is the accumulation of sensor input without any filtering. In particular it specifies the average error in meters between the final true location of the user and the estimate by the

Table 2.3: Average distance between destination and the user's position upon completion (m).

Distance from Destination	Path 1 (98.14m)	Path 2 (69.49m)	Path 3 (72.54m)	Path 4 (67.66m)	Path 5 (54.25m)
Landmark No Max	0.46	1.83	0	2.44	1.83
Landmark 9 Meters	0	1.22	0.46	2.19	1.83
Landmark 15 Meters	0.91	0.91	1.83	1.83	2.29
Metric No Max	2.74	0.61	0	2.29	1.83
Metric 9 Meters	3.05	2.74	1.22	0.91	1.22
Metric 15 Meters	4.57	0	0	2.74	1.83

Table 2.4: Average path duration (sec).

Path Duration	Path 1 (98.14m)	Path 2 (69.49m)	Path 3 (72.54m)	Path 4 (67.66m)	Path 5 (54.25m)
Landmark No Max	155.75	123.67	135.67	119.67	111.25
Landmark 9 Meters	201.33	177.00	212.00	192.50	138.75
Landmark 15 Meters	265.00	155.25	156.50	226.67	110.50
Metric No Max	136.25	180.00	137.50	129.50	108.50
Metric 9 Meters	242.67	252.75	173.67	219.00	169.00
Metric 15 Meters	264.00	173.67	247.00	180.00	147.33

corresponding technique. The estimate from the particle filter corresponds to the particle which was closest to the average state of all particles at the last iteration. It is important to note that in most cases there were particles closer to the actual position of the user than the "average" particle.

The comparison between the two tables shows that the particle filter approach considerably reduces errors when compared against the integration of sensor readings. The improvement ranges from a factor of ten to a factor of two for different paths and types of direction provisions. This despite the very small number of particles employed in this initial set of experiments. The important point, however, is the considerable effect that the direction provision process has on the efficiency of the particle filtering algorithm. The average error in meters in the final location for

the "Landmark 9 meters" approach is approx. 9.5 meters, while it goes down to 2.1 meters for the "Landmark 15 meters" approach, which also appears to be the best solution to the problem. The errors were lower for paths that contained distinctive landmarks such as hallways (in the order of 1.2-2.5m) and considerably higher for paths that corresponded to long straight line paths where all the landmarks were the same (doors). Figure 2.7 provides an error graph for a specific path/direction provision combination for dead reckoning and the particle filter approach.

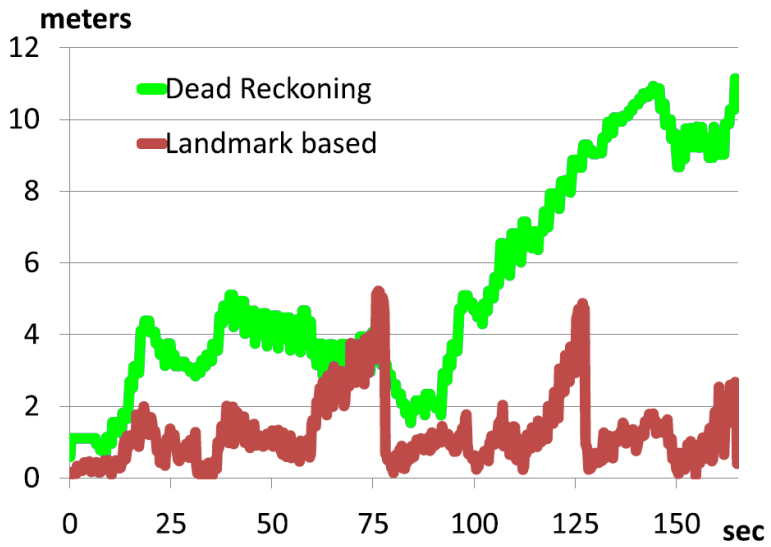


Figure 2.7: Graph comparison of dead reckoning with the particle filter approach

Table 2.5: Average error of dead reckoning in final location (m).

Dead-Reckoning	Path 1 (72.54m)	Path 2 (69.49m)	Path 3 (98.14m)	Path 4 (67.66m)	Path 5 (54.25m)
Landmark No Max	10.83	25.53	20.79	9.82	13.79
Landmark 9 Meters	17.87	32.50	10.19	8.59	13.43
Landmark 15 Meters	16.29	26.89	26.81	13.12	8.89
Metric No Max	15.49	25.69	14.91	13.41	11.65
Metric 9 Meters	23.84	28.99	18.00	5.89	7.53
Metric 15 Meters	31.44	20.95	19.55	3.88	7.90

Note that the errors in tables 2.3, 2.5 and 2.6 are not comparable, since the first corresponds to how close to the desired destination the user reached and the last

Table 2.6: Average error of the proposed interactive localization process (m).

Interactive Localization	Path 1 (72.54m)	Path 2 (69.49m)	Path 3 (98.14m)	Path 4 (67.66m)	Path 5 (54.25m)
Landmark No Max	1.32	15.51	18.80	1.14	5.31
Landmark 9 Meters	1.34	25.95	10.12	3.02	7.35
Landmark 15 Meters	3.03	4.02	5.47	3.63	2.33
Metric No Max	3.98	11.33	12.95	3.75	5.38
Metric 9 Meters	0.84	11.92	3.20	3.05	4.24
Metric 15 Meters	3.06	5.25	10.43	1.48	3.87

two tables correspond to localization accuracy. The directions did not depend on the localization process and this explains why it was possible for the error in localization to be higher than the distance between the true user location and the desired one.

Discussion This set of experiments indicated the feasibility of guiding a user with VI through an indoor environment using a minimalistic and interactive sensing approach achievable with a smart phone. The results suggest that the time to execute a path and the accuracy is proportional to the number of directions provided to the user. The results also show that the best direction provision method is Landmark No Max threshold as it yields good localization results and guides the user to the desired destination faster than the other methods.

2.4.3 Online Operation with Integrated Localization and Direction Provision Modules

In this set of experiments, the system was executed online on a smart-phone. Additionally, in order to address the issue of computational limitations on the phone,

the new preprocessing step described in section 2.3.2 was utilized.

This implementation employed dynamic direction provision. The most successful type of direction provision identified by the previous set of experiments (infrequent confirmation of highly identifiable landmarks) was used. The directions were generated dynamically based on the latest location estimation. The dynamic direction provision is also able to correct the user's path in case one doesn't follow the directions correctly.

This set of experiments allows to evaluate the different methods for estimating user's step length on the fly. Furthermore, the final location estimation process utilizes the k-means approach.

Setup A new map of the first floor, in addition to the second floor map, of the engineering building of University of Nevada, Reno was created (Figure 2.8). The new map added the following landmarks: *(i) 139 doors, (ii) 10 hallway intersections, (iii) 3 water coolers, and (vii) one ramp*. Additionally, two elevators and eight sets of stairs were added to connect the two floors. For the experiments, ten paths were tested from which two included multiple floors.

Eight volunteers participated in the experiments. Some of the participants had visited the building before while others had not. The users were blindfolded and used a white cane and the phone. Before the actual experiments the blindfolded users executed a training path to familiarize themselves with the cane and the inability to see.

A commercial robot localization system called Hagisonic StarGazer was used to capture the ground truth. The product consists of (1) set of passive landmarks

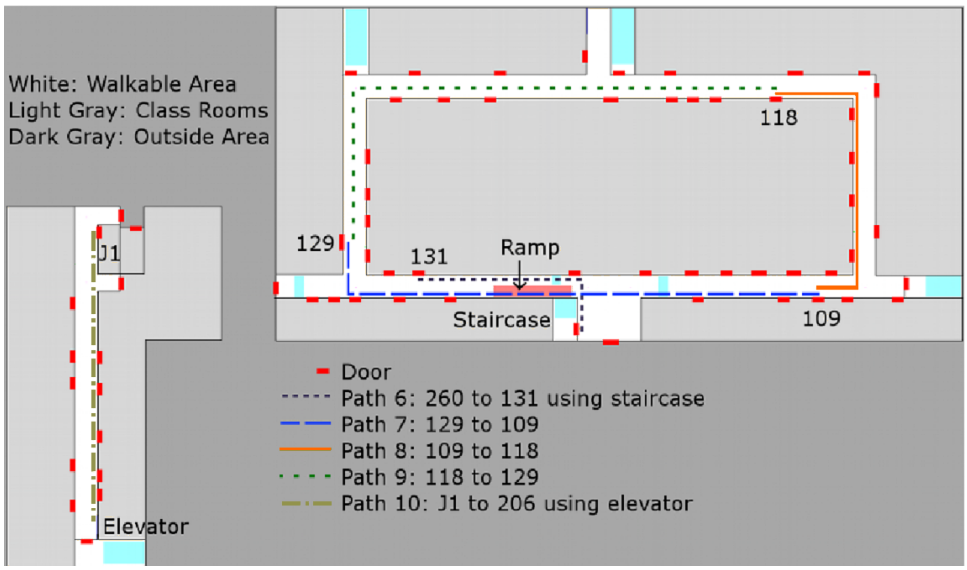


Figure 2.8: The map of the first floor

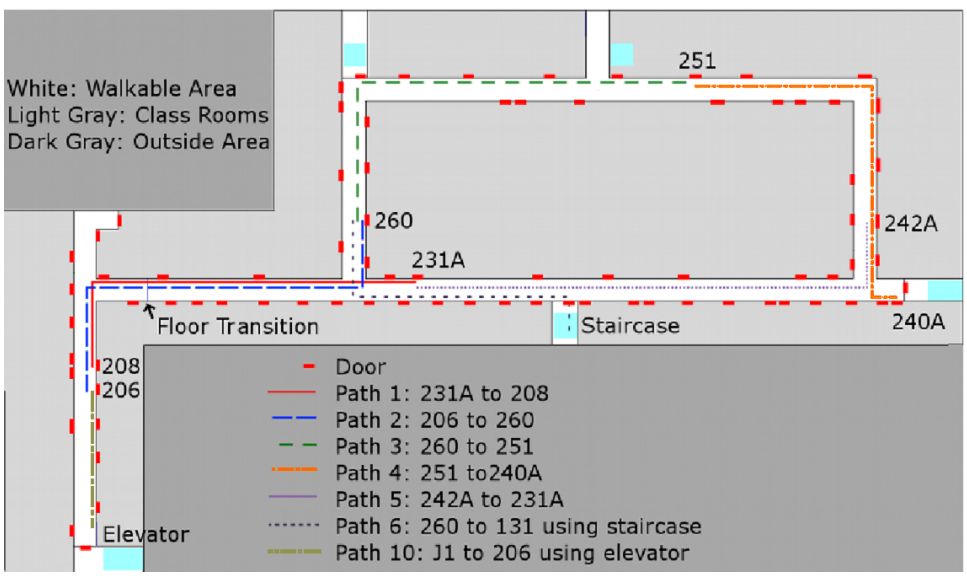


Figure 2.9: The map of the second floor

with unique ID mounted on the ceiling; (2) an infrared camera that outputs ID, distance, and angle of the closest landmark; and (3) an open source library to operate the camera. A map of landmarks is created using software specifically developed for this purpose. To use the system for human tracking, the camera and a battery are installed on a belt that the user wears during experiments. The user carries a

Figure 2.10: A sighted user, running experiments blindfolded wearing the ground truth equipment

tablet (Fig. 2.10) that records the landmark information received from the camera and calculates the user location from the map based on the relative landmark location. The accuracy of the camera is 2cm when installed on a flat surface but costs more than an order of magnitude compared to the proposed method and requires carrying a heavy sensor; but the camera is sensitive to tilt and might result in incorrect readings when installed on a belt. None of the participants felt the belt and backpack impeded their mobility in any way. To reduce noise, the ground truth is smoothed using an outlier detection algorithm.

As mentioned in Section 2.3.5, there were three different approaches evaluated for their ability to estimate the user's step length. The first approach has one particle filter that updates the position estimates given a Gaussian distribution for the step length, which has a mean at 0.54 meters and a standard deviation of 0.08 meters. The second approach has one particle filter where the step length is stored together with the state estimate. Initially, the particles' internal estimate of the step length is distributed according to the same probability distribution as in the first approach (mean: 0.54, st. dev. of 0.08 meters). Each particle advances based on the

Table 2.7: Distance between ground truth and localization estimate upon completion (m).

Distance from Ground Truth	Mean	STDEV
Path 109-118	1.31	0.48
Path 118-129	4.25	8.27
Path 129-109	2.61	1.53
Path 206-260	1.11	0.38
Path 231A-208	1.23	0.78
Path 242A-231A	13.95	7.95
Path 251-240A	1.29	0.33
Path 260-131	4.05	3.04
Path 260-251	1.18	0.23
Path J1-206	2.16	1.09

local value it stores for the step length. In every step the length estimation changes in order to be able to detect small changes in the stride of the user. A Gaussian distribution is used again with a st. dev. of 0.007 meters in order to allow small variations in the step length. The third alternative uses 10 particle filters. The step length mean value starts at 0.37 meters for the first particle filter and is increased by 0.06 meters for each consecutive particle filter. The standard deviation of these distributions is 0.02 meters.

Results Table 2.7 provides the average distance between the ground truth and the localization estimate when the path was completed. The path 242A-231A was failing most of the times for many of the users, because it required the counting of six doors in a row to reach the desired destination without a unique landmark confirmation in between. For this reason, the probability of the user reaching the goal is decreased considerably as the particles are scattered. In order to deal with this issue an alternative path was introduced as shown in Figure 2.11. Although this path is longer, it might have higher probability of leading the user to the desired destination because it has less, but more distinctive landmarks along the way. This

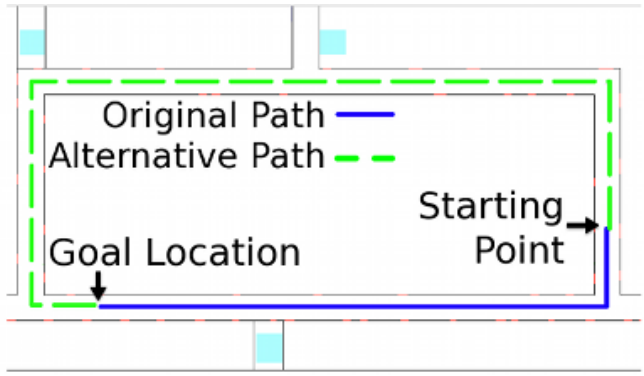


Figure 2.11: Two different paths that lead to the same goal. The alternative path has a higher success probability because it uses less, but more distinctive landmarks.

motivates the future development of a path planning process, which calculates alternatives that don't belong in the same homotopic class and take into consideration the probability of each path to lead the user to the desired destination based on the quality of the landmarks.

Fig. 2.12 shows the localization error during the progression of a path. In this figure one can notice that, for some particle filters the localization error increases during the execution of the path while for the ones that have a step length that is close to the actual step length of the user the error remains low. After the first confirmation of a landmark, though, the particle filters that have an incorrect step length estimation are replaced with the ones that have a better estimation and the error decreases. Nevertheless, one landmark confirmation is not always enough to estimate an accurate value for the user's step length. That is why after the first landmark confirmation, the error starts growing again for some particle filters. After the second confirmation, the particles converge to similar step lengths. When the user reaches the goal all the values were close to 0.5-0.6 meters step length.

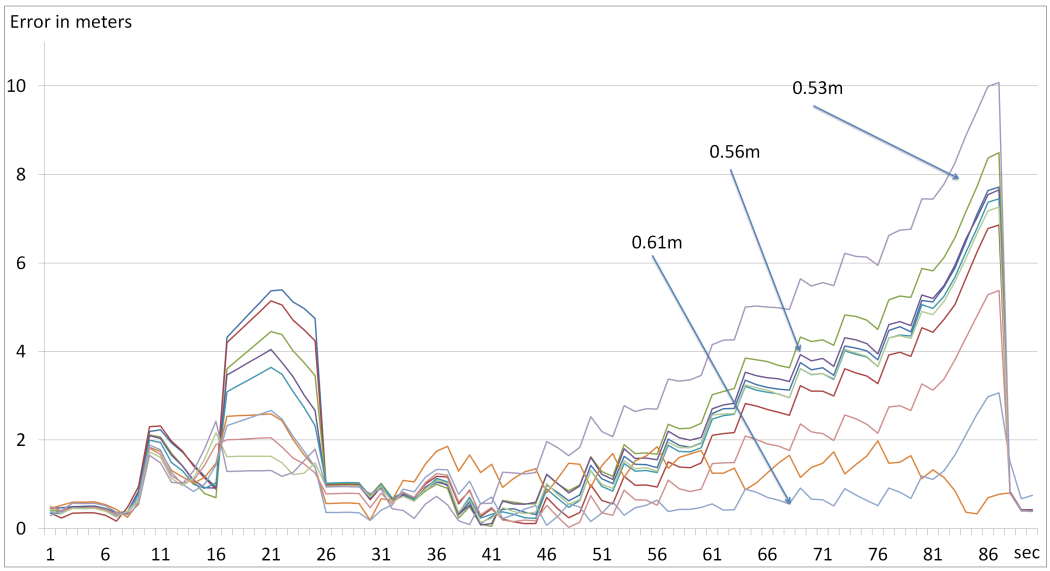


Figure 2.12: A graph showing the localization error during the execution of a path. Each line represents a different estimation of the step length.

Table 2.8: Distance between location estimation and ground truth for three different systems (m)

Distance from Destination	Mean	STDEV
1 PF with 1000 particles	2.31	2.30
1 PF with step length in state	1.02	1.12
10 PFs with 100 particles each	0.02	0.05

Adaptation to the User's Step Length The three different variants of particle filters described in Section 2.3.5 were tested in order to conclude which one is more suitable for adapting to the user's step length. Table 2.8 holds the mean localization error and standard deviation for the three different methods. In each case, six experiments were executed by three different participants. The two methods that do aim to estimate the user's step length perform better than the straightforward particle filter with large standard deviation. Overall the approach that utilizes multiple particle filters calculates a better estimation of the user's step length.

Computational overhead Table 2.9 holds performance data obtained from profiling. The system managed to run all the experiments smoothly without running into issues due to the limited computational power of the phone. In this implementation it was possible to use 1000 particles because of the preprocessing step.

Success Ratio of Direction Provision Table 2.10 shows the average distance of the ground truth (i.e., the true location of the user) from the desired destination. The majority of the paths were completed successfully. In particular, in 76% of the experiments the system guided the user to the desired destination within an average distance of 1.5 meters. In 11% of the experiments the error the system guided the user to a location adjacent to the desired destination (e.g., next door) with an average distance of 2.75 meters.

These results suggest that the new improved system can successfully localize a user in an indoor environment and adapt on the fly in the case that the user failed to follow the directions successfully. They, also, suggest that the multiple particle filter approach is better fit to localize the user with higher accuracy.

2.4.4 Experiments with users with VI

Though preliminary user studies were conducted with blindfolded participants, this study specifically focused on users who were visually impaired.

Setup Six users were recruited (three female, average age 51.16, SD=17.66) to participate in a user study. Participants were recruited through the local National Federation of the Blind chapter. All participants were visually impaired (three

Table 2.9: Profiling data (msec)

Profiling	109-118	118-129	129-109	206-260	231A-208	242A-231A	251-240A	260-131	260-251	J1-206
Avg total time	2808.13	4332.13	5237.13	3444.13	3114.75	4387.88	3397.88	4861.86	3953.5	3684.38
Avg time per step	39.14	37.72	39.88	39.23	40.01	39.41	39.02	41.25	40.39	38.69
Time st. dev.	22.11	21.47	23.03	23.40	22.29	23.152	22.04	24.16	23.24	21.79
Max step time	95.13	99.75	106.25	104.88	106.38	114.13	99.5	118	105.13	92.5
Avg steps	71.63	114.75	131.5	87.88	77.25	111.375	86.38	117.86	97.13	95.5
Avg time for transition	1281.75	1970.88	2220.38	1556.75	1409.38	2007.5	1490.25	2118.57	1811.75	1676.13
Avg time for observation	2048.13	3116.75	4012.63	3764.75	2241.63	3158.5	2358.38	3549.57	2883.75	2629
Avg time for sampling	39.88	58.25	64.00	35.25	44.38	40.5	43.25	69.714	46.88	39.13

Table 2.10: Distance between ground truth and destination (m)

Distance from Destination	Mean	STDEV
Path 109-118	1.17	0.5
Path 118-129	3.47	5.39
Path 129-109	1.78	0.95
Path 206-260	1.15	0.35
Path 231A-208	0.83	0.24
Path 242A-231A	4.08	2.92
Path 251-240A	0.51	0.25
Path 260-131	2.05	2.34
Path 260-251	0.74	0.32
Path J1-206	1.21	0.77

completely blind, two legal, one low vision) and used a cane for navigation. None had any self-reported mobility or hearing impairments. One user did not own a phone, four owned a cell phone and one owned a smart-phone. None of the participants had been in the Engineering building prior to the user studies. For the experiments, eleven paths were tested; and these paths were created so as to have approximately equal lengths and minimize the amount of overlap between them. Two paths involved transitions between floors. Specific characteristics of each path, such as its length and number of landmarks present, can be found in Table 2.11. Initially, ten paths were created but in a previous study one path (242A-231A) was found to lead to large errors in location estimation as it involves traveling a relatively large distance with only doors as landmarks (see Figure 2.9). The original 242A-231A path was failing most of the times for many of the users, because it required the counting of six doors in a row to reach the desired destination without a landmark confirmation in between. For this reason, the probability of the user reaching the goal is decreased considerably as the particles are scattered. In order to deal with this issue an alternative path was introduced as shown in Figure 2.11. Although this path is longer, it has higher probability of leading

Table 2.11: Profiling data for each path

Path	231A-208	206-260	260-251	251-240A	242A-231A	260-131	129-109	109-118	118-129	J1-206	*242A-231A*	Avg	St.Dev
Path Length (m)	43.89	49.38	51.21	45.72	57.61	57.61	58.52	36.58	64.01	50.29	95.10	55.45	15.26
# of Directions	5	7	6	7	5	11	10	7	5	7	9	7.18	2.04
# of Words	33	47	45	47	33	84	76	47	33	51	61	50.64	16.94
Landmarks	2	4	4	4	7	12	7	3	1	6	5	5.0	3.0
- Hallway Inters.	1	2	2	2	1	2	2	2	1	1	3	1.73	0.65
- Doors	1	2	2	2	6	8	3	1	1	4	2	2.91	2.26
- Stairs	0	0	0	0	0	1	1	0	0	0	0	0.18	0.40
- Ramp	0	0	0	0	0	0	1	0	0	0	0	0.09	0.30
- Floor transition	-	-	-	-	-	Stairs	-	-	-	Elev.	-	-	-
Success Rate	0.83	0.33	0.50	0.83	0.17	0.50	0.67	0.67	0.67	0.50	1.00	0.61	0.24
Average time(s)	72.29	115.31	107.34	79.73	100.81	227.39	135.20	80.21	95.10	136.97	162.93	119.39	45.32
Time stdev	12.28	44.42	14.12	16.88	31.47	69.82	35.29	33.10	26.59	34.21	33.12	-	-
Average # steps	81	107.83	92.5	82.17	111	148.17	132	76.50	117	106.83	164.33	110.85	31.07
Steps stdev	15.07	34.30	34.62	13.69	23.86	56.76	35.66	17.11	21.01	26.50	63.15	28.27	16.31

the user to the desired destination. This is confirmed also from the error values extracted from the experiments on this alternative path.

All users were right handed and navigated with the cane in their right hand. Users were asked to hold the phone in their left hand but hold it relatively still as to allow for detecting steps with higher accuracy.

User studies were conducted over a weekend with few students present in the building to avoid interference. Hallways were cleared of any obstacles and all doors to offices were closed. For each path, participants were led to the start location upon which the observer would select the right path and activate the direction provision. After the user study participants were interviewed using a questionnaire to collect qualitative experiences.

Results Table 2.11 shows the success rate, the average time and corresponding standard deviation as well as and the average number of steps and the standard deviation for each path. 85% of the paths were completed successfully. Though the use of multiple particles in a previous study with blindfolded users [5] was able to mitigate individual differences between step lengths, for some users with visual impairments this turned out to be difficult to achieve. Table 2.12 lists the distance between the target destination and the user's position upon completion of the path. There was no observation of users unable to confirm a landmark, but system failures occur when the system thinks the user has arrived at their destination when they actually have not arrived yet. Paths were considered a success when the user was able to navigate within 2.0 meter of the target location. A length of 2.0 meter was chosen, as this is approximately the range that can be sensed using a cane held in the hand and to correct for actual user location as the user was

Table 2.12: Distance between destination and the user's position upon completion (m).

Distance from Destination	User 1	User 2	User 3	User 4	User 5	User 6	Avg	St.Dev.
Path 231A-208	1.36	0.58	0.48	0.38	0.83	0.99	0.77	0.37
Path 206-260	1.94	1.09	0.71	0.31	0.72	0.51	0.88	0.58
Path 260-251	1.85	1.04	1.68	0.23	0.33	10.11 **F**	2.54	3.77
Path 251-240A	0.76	0.55	0.54	0.55	0.24	0.33	0.50	0.19
Path 242A-231A	14.77 **F**	1.32	15.15 **F**	1.22	2.81 **F**	1.83	6.18	6.82
Path 260-131	4.00 **F**	0.46	4.97 **F**	0.96	2.78 **F**	1.48	2.44	1.78
Path 129-109	1.22	1.60	0.76	0.83	2.14 **F**	1.68	1.37	0.53
Path 109-118	0.90	0.69	0.83	0.86	4.89 **F**	1.55	1.62	1.63
Path 118-129	1.11	1.73	0.99	1.59	1.28	1.89	1.43	0.36
Path J1-206	1.25	1.84	0.67	1.21	0.77	1.30	1.17	0.42
Path 242A-231A(Alt)	2.07 **F**	0.98	0.99	0.87	1.14	2.73 **F**	1.46	0.76
Success rate	0.91	1.00	0.82	1.00	0.64	0.91	0.73	0.82

wearing the camera on their back. Failures are indicated using an **F** in the table 2.12. For all paths and users an average error of 1.85 meter (SD=2.74) was found.

Paths that involve counting many doors, (e.g., 242A-231A, 260-131, 129-109) have lower success rates and it seems users are scuttling more to detect the doors, which makes it harder for the system to pick up steps. Similar to the earlier study, path (242A-231A) has a low success rate (50%). The alternative path for (242A-231A) on average took longer to complete (163 seconds) than the original (101 seconds) but has a significant higher success rate (83%) and a lower average error (6.18 versus 1.46 meter). This may indicate that planning longer paths that

include more distinguishable landmarks may increase accuracy. The completion rate for all paths improves from 85% to 88% and the average error reduces to 1.42 (SD=1.50) if results for the original path are excluded.

Non-directed interviews were used with open-ended question to collect qualitative experiences. None of the participants had used a navigation system before but all users had previously experienced getting lost in outdoor as well as indoor environments. Depending on the space all users engage in cognitive mapping when this space is unfamiliar to them. Primarily tactile landmarks, such as doors or windows are used in cognitive mapping, but one user noted using sounds as well, such as the sound of a water cooler or an air conditioning unit. Cognitive mapping was typically limited to smaller spaces such as rooms. None of the participants said they had freely explored large indoor environments out of fear of getting lost. Participants typically used a sighted guide when navigating in unfamiliar environments, which would help with working out a route. Three users preferred routes with many landmarks but would avoid routes with too many turns as that was found to be confusing.

A 5-point Likert scale was used to measure the usability of the system. All users found that the directions were easy to follow (M=4.66, SD=0.58) and that Navatar navigated them efficiently (M=4.66, SD=0.58). They also agreed this system could minimize the chance that they would get lost in unfamiliar environments (M=4.66, SD=0.58). Overall they liked using the system (M=4.66, SD=0.58). Despite Navatar only being able to guide user 5 for five paths this user was still very positive about the system. Upon further inquiry the user stated: “I have never used such a system before and I think it could be really helpful, despite its current shortcomings”. Suggestions for improvement included: (1) improve the accuracy; (2) being able to

repeat directions; (3) offer Braille output; (4) not hold the phone in the hand but in a pocket and (5) for directions involving stairs indicating whether one has to go up or down. This last suggestion was caused by the system providing directions dynamically so depending on the estimate of the user location the system would direct *"go down the stairs"* but if the system's estimate were beyond or on the stairs this direction would be omitted. One user preferred to not hold the phone in the hand because the user preferred to use this hand to protect their face. Repeating directions was already implemented as directions are calculated dynamically based on the system's current estimate but was not conveyed to participants to keep the study simple and to avoid false confirmations.

2.5 Discussion

This chapter presented a navigation system that can guide a user with VI through an indoor environment using a minimalistic and interactive sensing approach achievable with a smart phone. The sensors used in the experiments are inexpensive and available on popular portable devices. Nevertheless, they are also highly erroneous. For instance, compass sensors, especially cheap ones, perform very poorly in indoor environments due to metal structures and electro-magnetic noise. The experiments presented in this section were performed in a building with such issues. Despite this challenge, it was still possible to track multiple human users, who did not have any visual feedback, with sufficient accuracy through the interactive localization process described.

This line of research opens the door to exciting new applications for artificial intelligence methods in the area of human-centered and accessible systems. It is in-

teresting to investigate how to automatically plan alternative paths that lead along a larger number of distinguishable landmarks, such as hallway intersections. Such planning under uncertainty tools may significantly boost chances of the user successfully arriving at the destination and the localization estimate being more accurate. Furthermore, it would be interesting to investigate whether it is possible to develop more robust methods for automatically detecting changes in the user's speed during the execution of a path. Similarly, it is possible to make use of more realistic models of human motion [7] instead of the unicycle-like system employed in this work.

2D maps of an indoor environment, as used here, can be acquired from architectural blueprints. Nevertheless, it may be more useful to use richer types of representations. 3D virtual models can be employed to more accurately represent indoor environments with multiple levels and features like low ceilings, ramps, uneven floors and rails, which are impediments to people with VI. It is interesting to investigate how to extract landmarks, such as doors or staircases, automatically from the geometry of such models in order to utilize them in navigation and localization tools for individuals with VI. 3D models may be easier to annotate than 2D maps as annotators can more easily visualize the space. Crowdsourcing tools could be utilized so that 3D models of popular environments can be dynamically updated when they change so as to support tools such as the one described here.

CHAPTER 3
HEADLOCK

3.1 Description

Vision, the human ability to sense the size, shape, movement, and orientation of objects or people in space, plays a dominant role in spatial perception [20]. Spatial perception is an essential skill for efficiently navigating spaces, i.e., navigating without running into obstacles or veering from an intended path. Navigating spaces is quite a challenge for users who are blind as they largely have to rely on their hands and ears for spatial perception. Spatial perception using the tactile sense is significantly slower. Haptic modality is largely redundant when combined with visual modality [61]. Where vision can identify objects at a distance, the sensing range of the hands is limited by the length of a user's arm. This range can only be extended through the use of a cane, at the cost of only being able to perceive depth information from a single point. Hands can distinguish shapes and textures, such as Braille, but they cannot sense color, which is an important feature for recognizing objects [56], such as doors. Though a small number of blind people are capable of echolocation by tapping their cane or making clicking noises with their mouths, environments generally don't contain many persistent sound sources that can be used for spatial perception. Consequently, blind people often rely on sighted people to describe spaces for them or to help them navigate spaces. This dependency on others reduces their mobility [36].

Computer vision has been explored to offer various types of assistive perceptual functionality to blind users (see [54] for an overview). Because smartphones feature cameras, various assistive navigation apps have been developed that can

help read signs [71] or detect the presence and orientation of crosswalks [31]. Various smartphone based indoor navigation systems [18, 67] have been explored. Though smartphones use is increasing among blind users [11], for navigation the use of smartphones is subject to various challenges: (1) People who are blind may find it challenging to aim a camera independently [32]; (2) existing apps only correctly identify objects that are up close but not far away [30]; and (3) because users typically hold a cane in one hand, holding a smartphone in the other impedes their ability to open doors, recognize landmarks and press buttons [18].

Once only the focus of academic research labs, wearable computing has garnered significant public interest [23]. Of specific interest are Optical Head-mounted Displays (OHMD), such as Google Glass [1]. Because OHMDs typically feature a camera and support hands and eyes free interaction, it has been argued that this platform has great potential for assistive applications for users who are blind [23]. This paper makes the following contributions:

1. We present HEADLOCK, a large open space navigation aid for a soon-to-be commercially available OHMD (Google Glass).
2. We identify what type of audio feedback is most effective and usable to minimize veering; and
3. We report qualitative experiences of people who are blind using HEADLOCK.

3.2 Related work

Wearables. One of the earliest wearable navigation systems was that of Ertan et.

al. (1998) [17], in which navigation directions in the form of haptic feedback are conveyed through an array of vibrational motors sewn into a vest worn by a blind user. This system requires installing infrared transceivers in a space for localizing the user. Hub et. al. (2003) [29] presents a system that provides navigational instructions through a cane augmented with a stereo camera, a simple keypad, and a speaker. This system relies on wireless local area network (WLAN) localization, which takes advantage of the fact that many indoor locations already have wireless access points. The augmented cane detects objects and retrieves information regarding these objects from a 3D world model. Marston et. al. (2007) [57] present a wearable navigation aid utilizing a GPS antenna and laptop to provide navigational guidance through haptic and audio feedback. Ross and Blasch [69] evaluated three different wearable interfaces for visually impaired users including sonification, speech, and haptic feedback to minimize veering when crossing a street. The interfaces were evaluated with 15 elderly visually impaired users and no significant difference in accuracy was found, though the haptic tapping interface was found to be most usable. Unfortunately, each of these approaches has a reliance on either a priori knowledge in the form of maps or requires prohibitively expensive instrumentation of environments with beacons [18]. Several wearable systems require the user to carry expensive sensors and supporting computing equipment in a backpack, which may impede their mobility, as blind users already carry several assistive devices with them, such as a cane or a braille reader.

Smartphone Apps. Computer vision could be a better technology for developing assistive navigation aids because it is the natural technological parallel of the human vision system which normally handles wayfinding (and thereby navigation) problems [54]. Computer vision based approaches don't rely on installed beacons for localization, but instead take advantage of preexisting visual features

in spaces for localization. In recent years, mobile devices such as smartphones have advanced significantly in computational capabilities and available internal sensors, such that smartphones are currently considered one of the biggest assistive devices since the introduction of Braille [11]. Various apps are available that offer functionality to recognize currency, faces, or street signs. A few apps offer functionality related to navigation. The Crosswatch smartphone application [31] identifies crosswalks from a camera phone image and conveys their location and orientation using speech. A similar application performs real-time sign recognition [71] to locate signs and conveys what is on the sign using speech. Manduchi (2012) [53] presents a mobile computer vision system designed to detect and guide users towards artificial landmarks (i.e. fiducials). While this system requires the installation of a set of artificial landmarks, Manduchi argues that this system may detect natural landmarks, such as, elevator buttons or signs. This approach only senses landmarks up to a maximum distance of 3.5 meters. VizWiz::LocateIt [10] allows blind users to take a picture of a scene (e.g. a picture of a shelf of different cereals in a grocery store) using their smartphone and then sends users feedback guiding them towards a nearby target (e.g. a box of Wheaties on the shelf). This approach employed both sonification and text-to-speech (TTS) interfaces to guide users towards objects. User studies with blind users found the sonification and TTS feedback useful for finding objects. Smartphone based approaches are subject to a number of limitations. It is a challenge for blind people to aim a camera independently [32]. Smartphones have a limited field of view (FOV), which makes it easy to miss objects that are close by [53]. If an object is not visible to the camera and the user doesn't have a notion where the object could be, searching for an object may be a slow and cumbersome task [53]. For target finding using a smartphone a user study found that users would prefer wearing the camera over hold-

ing the camera in their hand [53]. Assistive smartphone apps typically require their users to hold the smartphone close to an object or person to avoid poorly focused photos and having to solve hard image segmentation problems [10]. Though these apps help understand what something is, they don't convey where something is [30]. Probably the most important limitation of current smartphones is that they are unable to acquire depth information, which could be useful for recognizing objects and landmarks during navigation.

Previous Work. The authors developed an indoor navigation system called NAVATAR [18]. This system is minimal in terms of installation cost, such to enable large scale implementation, as it only relies on an annotated 2D map and relies on dead-reckoning localization using sensors available in smartphones. To improve localization accuracy NAVATAR requires blind users to confirm the presence of anticipated tactile landmarks along a provided path. This approach offloads the computationally hard-to-solve problem of recognizing landmarks to a cognizant agent, which seamlessly integrates with how blind users already navigate familiar spaces. User studies with six blind users in a university environment found a localization accuracy of 1.85 m. GIST [37] is a wearable gestural interface that allows blind individuals to sense color, distance and the presence of humans using different hand gestures, where spatial information is provided depending on where the user's arm is pointed. GIST uses a commercially available depth sensor (Kinect) that the user wears on their chest with a small tablet carried in a backpack. Though GIST is somewhat bulky in design, a user study with 8 blind users demonstrated the feasibility of GIST to perform spatial interaction tasks, including discovering and navigating towards objects and persons of interest.

3.3 Design of Headlock

When navigating a new space, blind people rely on their spatial image in working memory [43] to memorize salient tactile landmarks they encounter (i.e., doors, hallway intersections, water coolers, stairs, floor transitions, etc.) and which helps them build a cognitive map of the space [34]. This cognitive map is essential for navigation as blind individuals rely on the identification of anticipated tactile landmarks [75] along their path to determine their current location (e.g. *follow the wall on my right until there is a water-cooler*). Williams et. al. (2013) [81] note that this navigation can be challenging for blind users when in large open indoor spaces. Within this context, our previous research [18, 37] identified two problems:

- **Veering.** Traversing large open spaces is a challenge for cane users, as either there aren't any tactile features to be followed or the distance between landmarks is large, which cause users to veer from their intended path (see Fig 3.2:a).
- **Handsfree**. Because cane users already use one hand to hold a cane, it is desirable to leave the other hand free to allow for identification of landmarks (doors, elevators, railings) and to allow for interaction with the environment (e.g., open doors, press a button, hold handrail).

To address these issues, we developed HEADLOCK, a navigation app for optical head-mounted displays (OHMD). A benefit of using an OHMD is that the camera is closely aligned with the user's field of view (FOV), giving the user an accurate indication of the direction in which the camera is aimed, which helps with orientation. Users don't have to carry heavy supporting computing equipment as

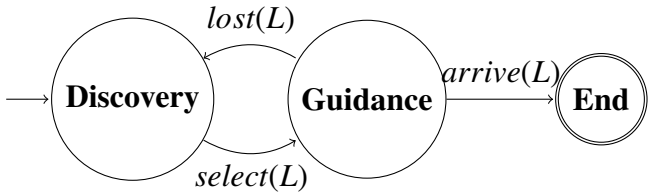


Figure 3.1: Finite State Machine describing the different modes and transitions between modes with landmark L.

OHMDs are lightweight, and only have to be tethered to a smartphone. OHMDs are non-obtrusive, which is important, as a recent study shows blind users prefer "small, easily accessible, and discreet" forms of wearables [23].

Approach. HEADLOCK allows users to remotely sense the presence of a landmark, and then lock onto this landmark to efficiently navigate towards it while minimizing veering. Similar to Manduchi [53], HEADLOCK distinguishes a *discovery* and a *guidance* mode.

Discovery. Computer vision is employed to parse the video stream from the OHMD's camera allowing users to scan a space by moving their head horizontally in order to detect salient landmarks (see Fig 3.2:b). Particular landmarks may have a known size, which allows HEADLOCK to also acquire an estimate of the distance to the landmark. Once the desired landmark has been found, the user instructs HEADLOCK to lock onto this landmark, at which point HEADLOCK transitions to the guidance phase (see Fig 3.1).

Guidance. The goal of the guidance mode is to provide feedback so that the user navigates efficiently to the landmark. To detect veering, we track the position of the landmark relative to the center of the camera's FOV. When it diverges from the center by a certain threshold, feedback is provided to indicate the direction and magnitude of the error allowing the user to correct their course (see Fig 3.2:c). In

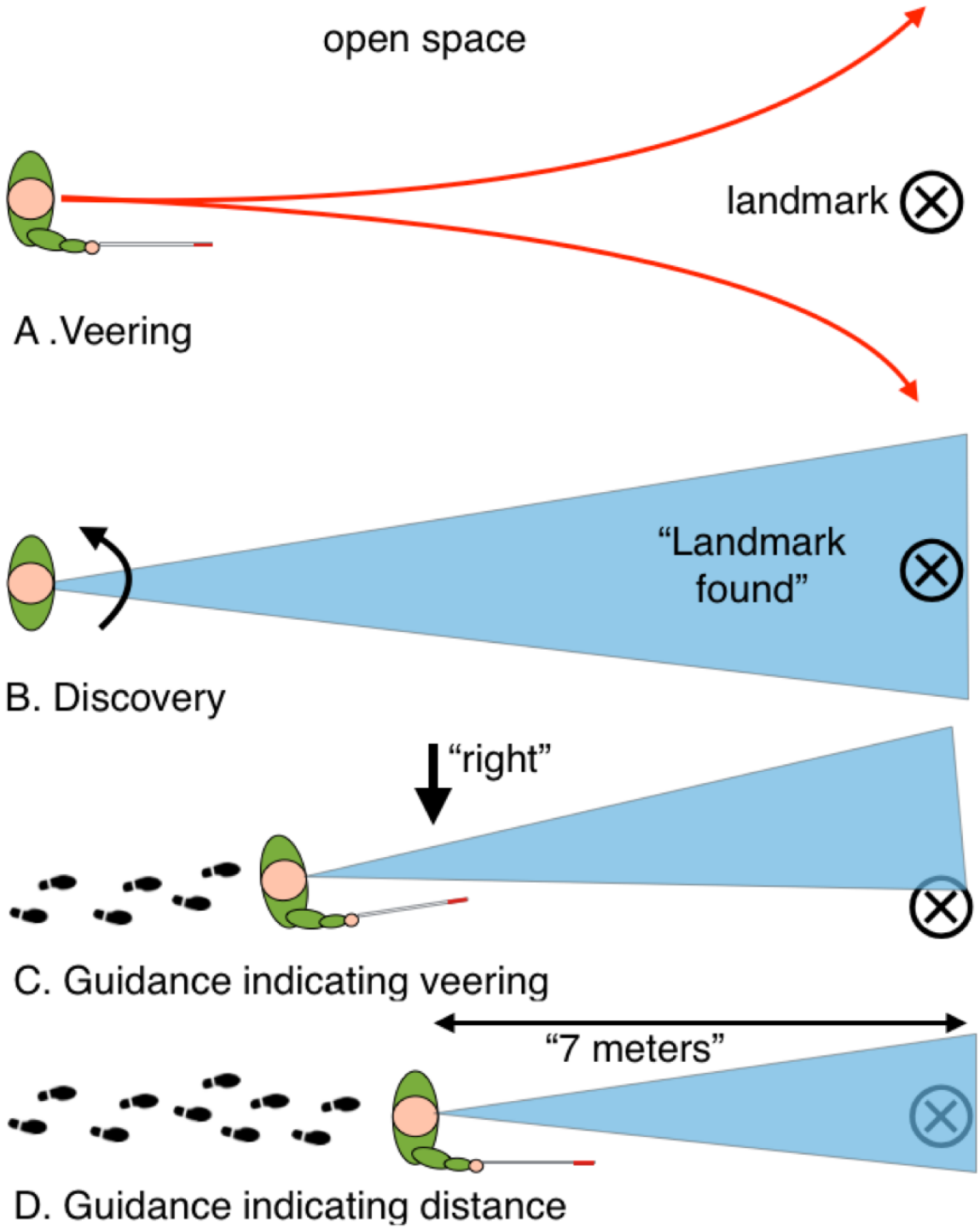


Figure 3.2: (a) Without any tactile features to follow in an open space, cane users may to veer from their intended path towards a landmark. (b) Users scan the space for landmarks and select one as their target. (c) Audio feedback is provided to correct for veering during the user's navigation towards the landmark. (d) Updated distance information is provided while approaching the landmark.

the case HEADLOCK loses visual track of the target landmark, we inform the user of this event and indicate the direction in which the user should turn to bring the landmark back into view. The discovery mode is automatically restarted, allowing for a landmark to be relocated quickly (see Fig 3.1). To convey progress, distance information is provided while the user navigates towards the landmark (see Fig 3.2:d). Upon reaching a landmark, the navigation task is considered complete and feedback is provided accordingly.

Research. A challenge that needs to be investigated is what type of feedback most efficiently allows a cane user to navigate to a landmark. Because OHMDs are typically not capable of providing haptic feedback, we explore two different types of audio feedback for efficiently guiding users towards a landmark: text-to-speech (TTS) and sonification. TTS takes advantage of the intuitiveness and rich vocabulary offered by human language, where sonification may be harder to learn but can convey multidimensional data efficiently and has previously been explored for guiding users towards objects [10].

3.4 Implementation

To understand what type of audio feedback is most effective and usable, we implemented HEADLOCK for one specific type of landmark: doorways. Doorways are typically used in the cognitive mapping of spaces by blind individuals [34] as they often have various subtle but distinguishable tactile features (e.g. direction it opens, type of door handle, sound it makes, etc.). Large open spaces often feature doorways, and a reasonable navigation scenario could include crossing large open spaces with the end goal of finding a particular doorway (e.g crossing a foyer

to find the elevator door). Since one of the objectives of HEADLOCK is to facilitate navigation of unknown spaces, rather than navigating towards a particular door, a doorway is a useful ubiquitous landmark that is present in many indoor spaces. Doorways are also a useful landmarks for HEADLOCK as they are relatively easy to detect due to their uniform color and shape (within the context of a single building).

Instrumentation. We use Google Glass [1] as our OHMD. Glass is a wearable voice-controlled Android 4.0.3 device that resembles a pair of glasses and only weighs 50 grams. Glass has various sensors including a camera (54.8° x 42.5° FOV), a microphone, and 3-axis accelerometer among others. Users interact with Glass using both natural language voice commands and the touchpad located on the frame. Audio is either provided through a bone conduction transducer or through an external headphone. The 5 megapixel camera on Glass is located on the right side of the frame. A user wearing Google Glass has a natural understanding of the direction in which the camera is aimed based on how their head is oriented.

Door Detection. HEADLOCK was implemented in Java using the Android Development Toolkit targeting Android 4.0.3. The door detection algorithm is a simple OpenCV-based blob detector which searches for blobs within the HSV color range of the doors in our test building. The algorithm processes frames with a scaled-down video resolution of 240x160 at 30 FPS providing HEADLOCK with bounding boxes for all recognized doors. This decreased resolution is utilized in order to achieve a reasonable performance on Glass without overheating the device. Preliminary trials showed that doors could be accurately detected at this resolution up to 12 meters away.

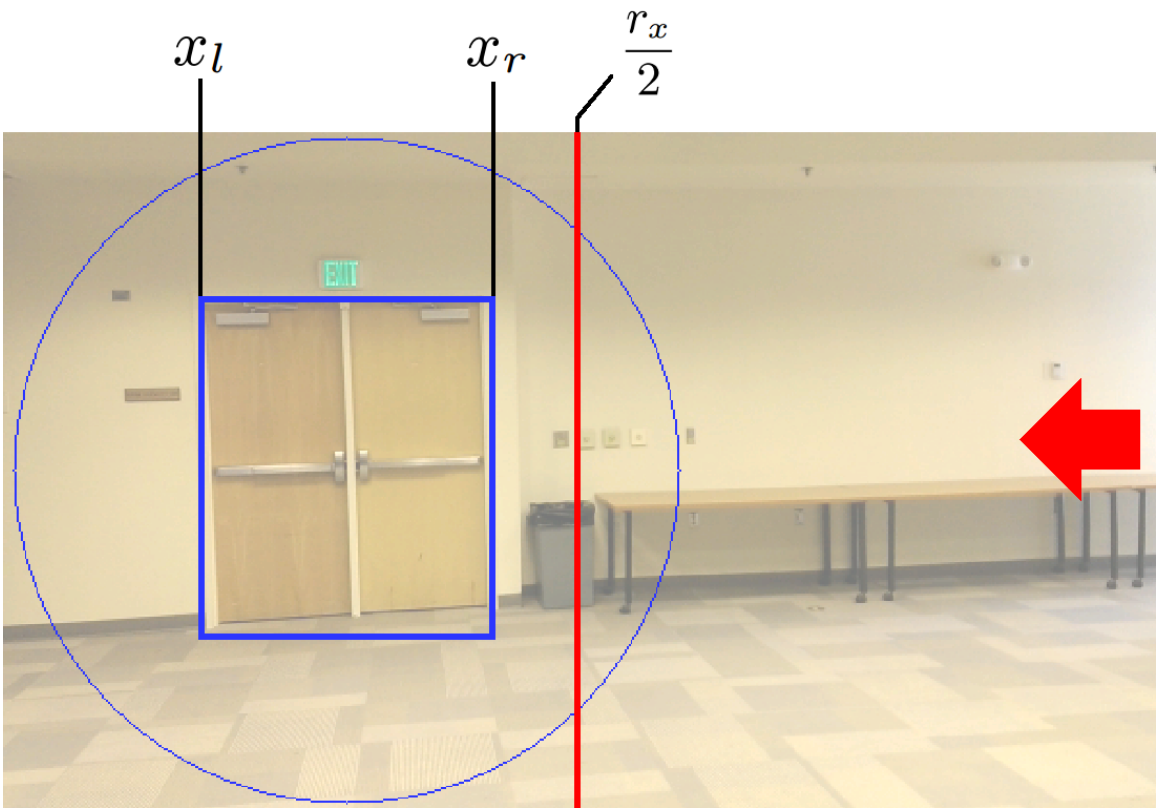


Figure 3.3: Feedback is based on the position of the middle of the field of view (red vertical line in this cropped image) relative to the nearest edge of the bounding box surrounding the target landmark (green box). In this case, HEADLOCK would indicate for the user to go left toward the blue target orientation.

3.4.1 Discovery

HEADLOCK provides feedback every 2,000 ms indicating whether or not a door has been detected. In the sonification mode, we designed an audio icon consisting of three high pitch beeps to indicate a door and three low pitch beeps if no door is visible. In the TTS mode, the feedback consists of the TTS phrases *"Door found"* or *"No door found"*. Upon receiving feedback indicating the presence of a door, users select the door by tapping on the Google Glass touch pad.

3.4.2 Guidance

Guidance output is generated using a veering metric (v) and a distance-to-landmark metric (d). The veering metric simply measures the distance from the closest edge of the tracked door to the middle of the field of view:

$$v(x_l, x_r) = max\left(\frac{r_x}{2} - x_r, x_l - \frac{r_x}{2}\right)$$

where x_l and x_r are the X coordinates of the left and right edges (respectively) of the door's bounding box, and r_x is a constant representing the X resolution of the input image (see Fig 3.3). The distance metric approximates the percentage of the navigation task the user has completed:

$$d(x_l, x_r) = 0.67 \log\left(100\frac{(x_r - x_l) - w_i}{r_x - w_i} + 4\right) - 0.27$$

where w_i is a constant value set for each tracked door representing its width in pixels when first observed. This equation was determined experimentally to represent a relative measure of the approximate distance traveled during the navigation task based on the observed width of the door. This metric could easily be replaced with a related true distance approximation rather than a relative distance approximation, but this would require a priori knowledge of the actual width of the observed door. Since a goal of the HEADLOCK application is to limit the amount of required a priori knowledge, a relative distance metric was chosen.

Sonification Feedback. Since the data being conveyed to the user is quantitative data representing X (veering) and Z (progress) values, it is well suited to being conveyed as an auditory graph [60]. For the auditory graph sonification scheme employed by HEADLOCK, pulse delay (the length of the silent period between each

beep) was chosen to indicate changes in X-axis veering data:

$$p_s(x_l, x_r) = \begin{cases} 200 + 800v(x_l, x_r) & : x_l > 0.5r_x, x_r < 0.5r_x \\ 0 & : x_l \leq 0.5r_x \leq x_r \end{cases}$$

When a user veers (first case), $p_s(x_l, x_r)$ varies the delay between beeps linearly from 200 ms to 1000 ms with respect to $v(x_l, x_r)$, indicating the magnitude of veering to the user. When the user isn't veering (second case), $p_s(x_l, x_r)$ returns a constant value of zero (a continuous tone, no beeping). In auditory graphs, frequency is often used as an analog for changes in Y-axis data [60]. In the case of HEADLOCK, this corresponds with the Z-axis data:

$$f(x_l, x_r) = \begin{cases} 1710 & : x_l > 0.5r_x, x_r < 0.5r_x \\ 130 + 920d(x_l, x_r) & : x_l \leq 0.5r_x \leq x_r \end{cases}$$

All pitches set by $F(x_l, x_r)$ were chosen experimentally and verified to be within the frequency range of human hearing [25]. When the user is veering off course (first case), $f(x_l, x_r)$ sets the output pitch to the constant value of 1710 Hz. Due to the fact that a single high pitch beep is utilized for all veering feedback, users must perceive the direction of their veering by listening to the change in pulse delay over several time steps. In the case that the user is not veering (second case), $f(x_l, x_r)$ varies the pitch of the feedback tone linearly from 130 Hz to 1050 Hz with respect to $d(x_l, x_r)$. This frequency range is significantly lower than the frequency chosen for veering feedback, making it easy for users to distinguish between veering and non-veering feedback pitches.

TTS Feedback. The TTS feedback scheme can be described in terms of pulse delay, $p_t(x_l, x_r)$, and a TTS readable version of the distance-to-landmark metric,

$d_{out}(x_l, x_r)$. The only difference between $p_t(x_l, x_r)$ and $p_s(x_l, x_r)$ is in the second case:

$$p_t(x_l, x_r) = \begin{cases} 200 + 800v(x_l, x_r) & : x_l > 0.5r_x, x_r < 0.5r_x \\ 2000 & : x_l \leq 0.5r_x \leq x_r \end{cases}$$

The 2000 ms pulse delay in the second case reduces the feedback redundancy that would be present if there were continuous TTS feedback similar to the continuous sonification feedback described for the second case of $p_s(x_l, x_r)$. Note that, rather than beeps, the system either generates the synthetic speech *"Left"* or *"Right"* in the first case of $p_t(x_l, x_r)$, and *"Straight"* in the second case, indicating the direction the user needs to move in order to proceed to the target landmark. In the second case, the system also generates a speech version of the distance-to-landmark metric $d_{out}(x_l, x_r) = 100 \times d(x_l, x_r)$ indicating the users progress towards the door.

Error & Task Completion. In the case that the door moves outside of the camera's FOV, the user is provided with TTS feedback (during both TTS and sonification modes) indicating the direction which they must turn in order to bring the door back into the FOV, and the discovery phase is automatically restarted. The navigation task is considered complete when the ratio of the perceived width of the door (in pixels) to the X resolution of the image exceeds 70% (i.e. when the door fills 70% of the field of view). This value was determined experimentally. This assumes that the doorway is somewhat large and near the eye level of the user upon completion of the guidance phase. Regardless of the feedback mode, HEADLOCK plays the same "success" sound effect to signal to a user that they have completed their navigation task.

3.5 Evaluation

We performed a quantitative evaluation to determine what type of audio feedback was most effective and a qualitative evaluation to assess the usability and utility of HEADLOCK.

Participants. Eight blind participants were recruited from the local chapter of the National Federation of the Blind (3 Female, average age 44.1, SD = 11.1). Five participants were totally blind and three participants were legally blind with a small amount of residual light perception. The legally blind participants wore a blindfold during the user studies in order to ensure that their performance was not unduly affected by their ability to perceive small amounts of light. None had any self-reported impairments in mobility or hearing. All participants were right-handed and used canes for navigation.

Instrumentation. HEADLOCK was implemented on Google Glass as previously described. Ground truth to measure veering was gathered using a commercially available beacon based localization system (Hagisonic StarGazer) that offers a localization accuracy of ±2 cm. The StarGazer camera was worn on a subject's back using a belt with supporting computing equipment (Windows Surface Tablet) in a backpack. Experiments were performed in a large open space (a conference room ≈ 10 X 12 m) on the UNR campus that was instrumented with ceiling tags required for the StarGazer system. The room had a single double door on the south wall (see Fig 3.3). During the user study, the left door was covered with green paper to ensure that HEADLOCK only recognized the right door.

3.5.1 Procedure

The user study was organized into four parts:

1. An unaided navigation task without HEADLOCK.
2. A HEADLOCK tutorial.
3. Trials with the first and second feedback modes.
4. A survey eliciting qualitative feedback.

All participants were randomly divided into two groups (A and B). For group A, the first feedback mode was sonification and the second was TTS, while group B had TTS first and sonification second. This counterbalancing ensured that the results of the study were not biased towards one feedback mode due to interference effects.

Unaided Navigation. To establish a baseline for comparison between navigating with and without HEADLOCK, subjects first performed an unaided navigation task. Subjects were taken into the conference room, positioned on one end of the room, 10 meters from the door, and asked to locate the door on the other side of the room. In a real-world scenario, it is unlikely that blind people who are faced with the task of having to cross a large open space would initially be oriented in the exact direction they needed to move unless they enlisted the aid of a sighted individual. Therefore, subjects were positioned facing the correct side of the room, but were not oriented in the exact direction to the target doorway. Subjects used their cane for navigation. Qualitative observations of the manner in which users accomplished this goal were recorded. Additionally, users were asked

several open-ended questions regarding how they typically deal with similar navigation scenarios.

Tutorial. The tutorial familiarizes users with Glass and the usage of HEADLOCK . Prior to each trial with a specific feedback mode, users were instructed on how the mode for that trial worked. Subjects were allowed to ask questions regarding the operation of the HEADLOCK interface. After the tutorial, subjects could try HEADLOCK until they felt comfortable using it.

Feedback Trials. To begin each trial, subjects were guided by a sighted observer to five randomly selected points in the room before being guided to the starting location on the north end of the room. The point of this process was to disorient the subjects, ensuring that they did not start the discovery phase of HEADLOCK with an a priori understanding of where the target doorway was located. Users started in the same position as in the unaided navigation task, 10 meters away from the door. The starting position was the same for all subjects and trials. They were positioned facing either the east, or west side of the room (randomly selected), while the target doorway was located on the south wall of the room, ensuring that the user would have to scan the room in order to discover the doorway. The HEADLOCK application was then started by the observer and the user was instructed to discover the doorway and then follow the guidance to navigate to it. The time required for the user to complete each phase was recorded. Each user completed six trials (three trials starting with the user facing east and three with the user facing west) for each of the two feedback modes.

User	Discovery (s)		Guidance (s)		Veering (◦)	
#	Son	TTS	Son	TTS	Son	TTS
1	14.75	8.65	29.50	16.62	12.84	7.42
2	13.43	12.60	38.46	21.02	28.74	13.77
3	25.81	14.65	28.86	18.82	16.43	25.19
4	21.00	12.52	29.95	31.86	6.06	10.39
5	20.30	14.96	23.52	20.94	10.95	11.49
6	14.95	9.79	35.74	19.08	29.00	15.03
7	16.97	17.52	31.61	14.25	7.89	8.07
8	43.56	11.98	*76.84**	*60.29**	20.95	28.02
Mean	21.34	12.83	31.09	20.37	16.61	14.92
σ	9.86	2.86	4.87	5.60	8.89	7.69

Table 3.1: Feedback trial results values averaged across each user's 6 sonification trials and 6 TTS trials (* indicates outliers).

3.5.2 Results

Unaided Navigation. All subjects were observed to follow very similar strategies to find the doorway located on the opposite side of the conference room. Each user began their task by setting out in the general direction of the doorway, using their canes to sense landmarks (e.g. walls). Several subjects veered significantly and found the east wall first, while the remainder of the subjects veered less significantly and found the south wall first. After finding a wall, all subjects followed the wall until they detected the doorway with their cane. One subject commented that wall following was her default strategy for finding a particular store in a mall. Three subjects commented that when faced with situations such as this, they use both sounds (e.g. echoes off of walls and sounds from air vents) and smells (e.g. the scent of the food court in a mall) to help orient themselves in large open spaces.

Feedback Trials. Table 1 lists the average results for each user over the six trials. In order to understand the efficiency of the sonification mode as compared to the TTS mode, the average time required to complete the discovery and guidance

phases for each mode was calculated for each user, and the averages were compared using a one-way ANOVA. A Grubbs' test found the average times for one subject to be a statistically significant outlier for TTS ($\boldsymbol{Z} = 2.38, N = 8, p < .05$) as well as sonification ($\boldsymbol{Z} = 2.32, N = 8, p < .05$). It is interesting to note that this subject was the only subject to report no previous experience with mobile technologies, possibly explaining the user's slower performance. It was found that TTS required 12.83 s for discovery (39.9% faster than sonification) and 20.37 s for guidance (34.5% faster than sonification). Both of these results were found to be statistically significant ($\boldsymbol{F}_{1,14} = 5.496$, $p < 0.05$ and $\boldsymbol{F}_{1,2} = 14.608$, $p < 0.005$ respectively). Next, since HEADLOCK was designed to prevent blind subjects from veering during navigation, veering was analyzed across all subjects' navigation paths for each feedback mode. In order to measure veering, two gradients were defined at every point along each user's path: the vectors between the every point and the points of the left and right sides of the doorway (labeled as vectors A_i and B_i for point i in Figure 3.4 respectively). The current trajectory at every point of the subjects' navigation path was calculated as the vector between point i and point $i - 1$ (labeled vector $C_{i,i-1}$ in Figure 3.4). These vectors were then used to calculate the subjects' degree of veering, $\Theta(i)$, at each point along their paths. If $C_{i,i-1}$ was angled between A_i and B_i ($\angle AC < \angle AB$ and $\angle BC < \angle AB$), the section of the path between point i and $i - 1$ did not exhibit veering (i.e. $\Theta(i) = 0$). On the other hand, if $C_{i,i-1}$ was angled outside of the region between A_i and B_i, then the degree of veering was accounted for as the angle between $C_{i,i-1}$ and the closest of either A_i or B_i (i.e. $\Theta(i) = min(\angle BC, \angle AC)$). This measurement of veering was used to create a cost function assigning a value to each path, representing the average veering which

occurred along each path P:

$$Cost_{veer}(P) = \frac{\sum_{i=1}^{n} \|C_{i,i-1}\| \, \Theta(i)}{\sum_{i=1}^{n} \|C_{i,i-1}\|}$$

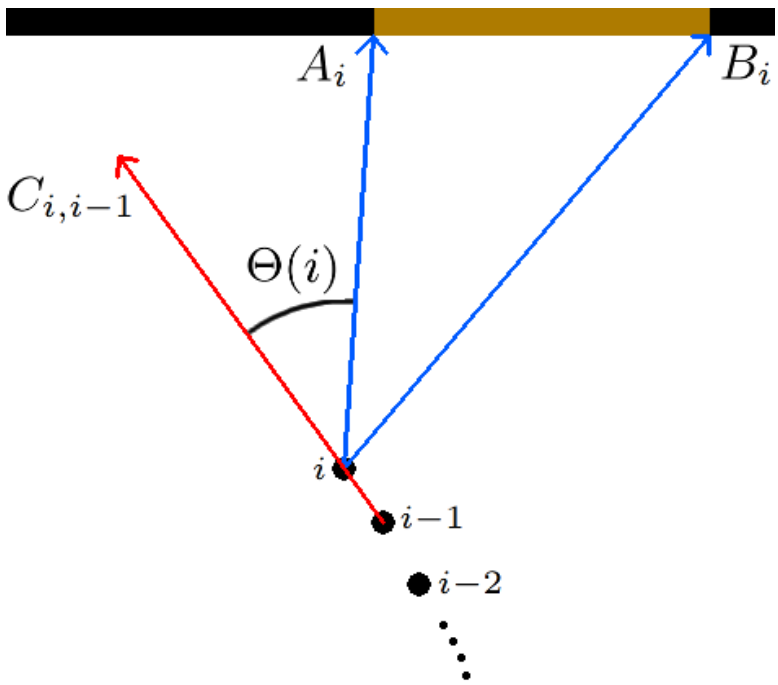


Figure 3.4: The veering cost function sums the measure of veering at each point along a user's path weighted by the magnitude of $C_{i,i-1}$.

The cost of all paths were calculated and averaged for each feedback mode for each user. These average cost values were compared between feedback modes with a one-way ANOVA. No significant difference in veering was found between feedback modes ($\boldsymbol{F}_{1,14} = 0.164$, $p > 0.05$).

Qualitative Feedback. After the feedback trials, non-directed interviews with open-ended questions were used to collect qualitative experiences as to understand the usability of HEADLOCK and to identify areas for improvement. The us-

ability and effectiveness of the app was evaluated using a 5-point Likert scale that ranged from 1 (strongly disagree) to 5 (strongly agree). Users agreed that HEADLOCK allowed them to navigate effectively to a doorway (M = 3.94, SD = 0.63). Additionally, users expressed neutral feelings that the system would minimize the chances they would become lost (M = 3.50, SD = 1.12) but users liked the system overall (M = 4.00, SD = 0.83). With respect to the feedback modes, users expressed neutral feelings about the sonification mode being easy to understand (M = 3.75, SD = 1.48) in addition to neutral feelings about the sonification feedback being insufficient for navigating towards doorways (M = 2.50, SD = 1.58). The TTS feedback was found to be easy to understand (M = 4.31, SD = 0.97) and sufficient to navigate towards a doorway (M = 4.19, SD = 1.27). There was no significant difference in rankings between feedback types ($p > 0.05$). Users agreed that HEADLOCK would allow them to navigate large open spaces more independently (M = 3.88, SD = 0.93).

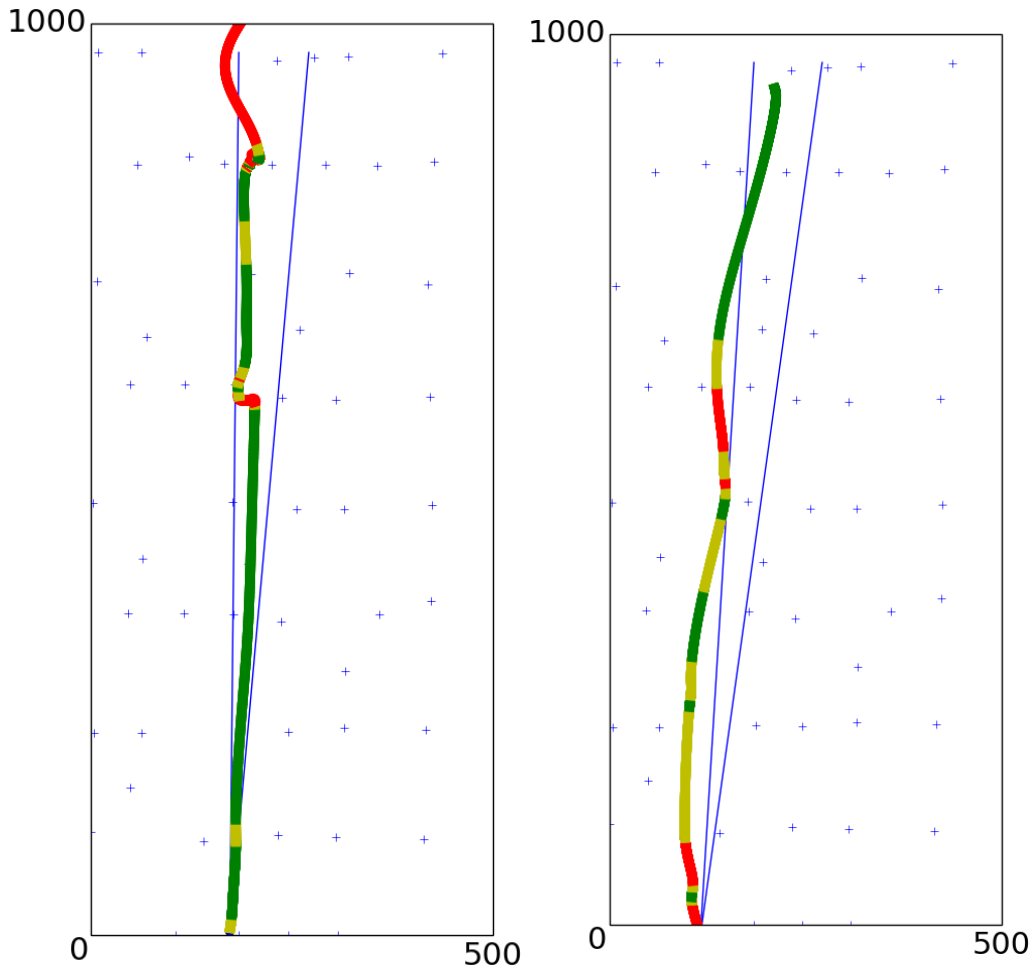
In terms of general suggestions for improvement, two users suggested that HEADLOCK should include obstacle detection as a feature. Two users suggested that the pulse delay for TTS feedback should be adjustable to allow for users to receive feedback more or less frequently based on personal taste. This was reflected in another user who said that sonification feedback felt too precise and therefore said the precision should be adjustable. Finally, two users commented that the physical interface of Google Glass should be made more accessible by adding bumps to the touchpad so that users can more easily determine where to tap the touchpad for interactions.

3.6 Discussion & Future work

Our study demonstrated the feasibility of HEADLOCK and allowed us to identify the following research issues.

Feedback Tradeoffs. TTS allows for more efficient navigation to doors than when using sonification though no significant difference in the degree of veering was found. Subjects expressed a preference for TTS (non significant). From our analysis it appears that the sonification feedback during veering often caused subjects to pause in order to find the correct direction to the door. Using TTS, subjects were observed to correct their course while continuing to navigate to the door leading to more efficient navigation with respect to time (see Figure 3.5). This could be explained that the sonification feedback was harder to interpret and more difficult to learn than using TTS. Several subjects indicated that they found following the increasing pitch of sonification to be less obtrusive than receiving TTS distance updates. Determining the direction in which to turn to correct for veering is easier with TTS than with sonification, as users have to figure out the direction of veering error, where TTS immediately conveys this (e.g. *"Left"* vs. *"Right"*). Based on these results, it seems feasible to explore a hybrid feedback approach where we convey veering error using TTS and distance to the landmark using sonification. Pitch could also be used in conjunction with TTS feedback to convey the magnitude of the veering error (e.g. a high pitch *"Right"* indicating a large turn to the right). Future user studies will evaluate how this feedback will affect HEADLOCK 's efficiency.

Head-Mounted Displays. Subjects were very excited to use Glass. Several subjects stated they found it hard to imagine that Glass was a fully functioning



(a) Sonification **(b)** TTS

Figure 3.5: Example traces of paths for both feedback modes. The path is marked as green when $\Theta(i) = 0$, yellow when $\Theta(i) < 10$, and red when $\Theta(i) > 10$. The blue lines represent the vectors A_0 and B_0 (see Figure 3.4).

wearable computer due to its light weight and small size. One minor accessibility issue with the use of Glass was identified. Because all subjects used their cane in their right hand for navigation, they had to temporarily place the cane in their left hand to be able to select the landmark with their right hand using the touchpad. To allow for more efficient interaction with Glass while navigating the touchpad should be on the left side. Voice recognition could be used to enable hands-free input, but we did not implement this as for this study we wanted our input to be most robust as voice recognition may be difficult to use in noisy environments.

No subjects expressed any difficulty in hearing the feedback from Glass's bone conduction speaker, which is good as this type of speaker can be used in noisy environments.

Obstacle Detection. Two subjects expressed that obstacle detection should be a feature of HEADLOCK . Due to the position and direction the camera is pointing it may be difficult to implement ground level obstacle detection on an OHMD that has the fidelity, accuracy and reliability of a cane. However, OHMDs may be better suited for over-head obstacle detection due to the camera position, which would address a current limitation of a cane.

Limitations. Our study was limited in that only doors were used as landmarks. We did not evaluate whether OHMDs are a better platform than smartphones for navigation aids. We did not evaluate the maximum distance at which a landmark can be identified nor how distance affects a cane user's ability to efficiently navigate to a landmark. We plan to investigate these issues in future research.

Landmark Detection. Future work will focus on expanding the set of landmarks that can be recognized, which could include signs, staircases or large windows. The environment in which our user study was conducted only had single door but HEADLOCK should allow its users to make a selection from multiple landmarks that are visible. Hallways in airports or schools often feature pattern or lines, which could also be detected and be used for following without cane users having to follow a wall instead. We plan to merge the open space navigation benefits of HEADLOCK with the high level indoor navigation features of NAVATAR [18].

CHAPTER 4

PEDOMETRY

4.1 Description

Wearable computing has garnered significant public interest, with a number of systems currently available. Of specific interest are head-mounted displays (HMD), such as Google Glass, which are no more obtrusive than wearing a pair of glasses. HMDs are considered more socially transparent than smartphones, as they enable their users to quickly access information without having to look away from a conversational partner. Wearables typically support eyes- and hands-free interaction, allowing them to be used in active contexts, such as walking or running. Pedometry, i.e, step counting, is used in various mobility related applications, such as physical activity tracking [63] or infrastructure-free indoor navigation [18]. Pedometry is implemented using wearable accelerometers, e.g., Fitbit, or using inertial sensors, which have become ubiquitous in smartphones [13] and HMDs. Because wearables can be used in active contexts, they are an especially good fit for pedometry-based applications. Several studies have found the accuracy of pedometry to vary depending on where an inertial sensor is placed on the human body [49, 13] with a higher accuracy achieved for inertial sensors placed closest to the feet [19]. This paper investigates the accuracy of pedometry on HMDs for both walking and running; something that has not been addressed by previous work.

4.2 Related work

The accuracy of pedometry for various sensor placements has been investigated in the following work. Foster et al. [19] compared the accuracy of a waist-worn pedometer with two ankle-worn pedometers. Their results show that the latter setup offers a significantly higher accuracy but does require wearing two pedometers. Graser et al. [22] studied the accuracy of pedometers in various locations on the waistline, including navel, thigh (front/back), side (right), and back. Their study found no significant difference in accuracy between locations. A longitudinal study by Ling et al. [49] explored three different placements (bra strap, waist, shoe) for pedometers on women. Step counts from the bra strap- and shoe-placed pedometers varied more than 10% from the waist step counts.

Though wearable pedometers are popular, they are not nearly as ubiquitous as smartphones. Brajdic & Harle [13] evaluated the accuracy of pedometry using sensors available in smartphones. Their study varied the location at which the smartphone was placed (front pant pocket, back pocket, backpack, handbag, handheld). Various step counting algorithms are evaluated. Results found no significant difference in step counting accuracy between positions, though with the phone worn in the back pocket, the likelihood of undercounting steps increases, due to extra oscillations caused by the relaxation of the gluteus maximus during the walking cycle. A related study found smartphones to be more accurate for energy expenditure prediction than wearable accelerometers [63]. Smartphones feature barometers which can be appropriated to distinguish between walking up or down stairs which generate different amounts of energy expenditure.

Specifically related to HMDs is the following work. Manohar et al. [55] demon-

strates the feasibility of an earpiece with an internal accelerometer to accurately classify various activities (sitting, standing, walking). Their study did not involve pedometry nor were comparisons made with other locations on the body. Atallah et al [8] evaluated different locations (ear, chest, arm, wrist, waist, knee and ankle) of an accelerometer for classification of every day activities. These activities were classified by their intensity: low (eating/reading), medium (walking/vacuuming) and high level (running/cycling). Results showed that for low level activities, a waist worn sensor yielded the the best performance. For medium level activities, the chest and wrist locations performed best. For high level activities, wearing the sensor at the ear yields the best performance, as this location can more accurately sense the change in body posture compared to other positions. Pedometry was not evaluated. Cinaz & Kenn [15] implement pedometry on a head-mounted device, but no comparisons with other body locations are made.

4.3 Experimental Setup

For this experiment, we evaluate the accuracy of pedometry on a HMD with pedometry on a smartphone. This choice is motivated by that wearable devices feature the same inertial sensors as smartphones. HMDs and smartphones are devices that can run apps where wearable accelerometers only offer limited user interaction. HMDs and smartphones are also heavier than wearable accelerometers. Our experiments involve running in addition to walking as this is a common activity for physical activity tracking and no studies exist that evaluate the accuracy of step counting for running using smartphones. We vary the location of the smartphone.

4.3.1 Instrumentation

We use Google Glass as our HMD. Google Glass is a wearable voice-controlled Android device that resembles a pair of glasses. It displays information in the user's field of vision using a 640x360 prism projector. Glass has various sensors: a camera, a microphone, 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, ambient light, and proximity sensors. Users interact with Glass using natural language voice commands or through the touchpad located on the frame. Audio is provided through a bone conduction transducer or using an external headphone. Glass weighs 50 grams and runs on the Android OS. For our smartphone we used a Samsung I9300 Galaxy S III, as this phone features the same 3-axis accelerometer (Invensense MPU6050) as Google Glass, minimizing differences in results due to differences in hardware. This smartphone weights 133 grams and measures 137 x 71 x 9 mm and runs on Android OS. We developed a single Android 4.0.3 app to collect acceleration data. The same app ran on the smartphone as well as on Glass to mitigate any differences in performance due to differences in code or OS. When the app is running, data collection starts as soon as the smartphone screen or the Glass touchpad is tapped. The app creates a log file for each trial with each line holding a sample and each sample containing a timestamp and the accelerations for the X, Y, and Z axis. If the screen or touchpad is tapped again, data collection stops. Experiments found that Glass records samples with a frequency of 200 Hz and the smartphone at 100 Hz. In order to synchronize the traces prior to each trial, each app sends a time request to a Network Time Protocol server at the beginning of each trial. The response from the server is then used as the offset from the device's native clock and is added to every timestamp we record from the sensor. Experimental results showed such time requests have an accuracy of a few tens of

milliseconds, which we deemed acceptable for our experiment.

4.3.2 Participants

We recruited 16 students (8 females and 8 males, average age 29.69, SD=7.14, average height 172cm, SD=9.85cm) to participate in our experiment. One subject was left handed. None of subjects self-reported any non-correctable impairments in perception or limitations in mobility.

4.3.3 Procedure

Subjects were equipped with a Google Glass device. For walking, subjects held one smartphone in their dominant hand, and placed a second smartphone in their right front pant pocket. These locations are the most common location to place a smartphone. As a prior study already explored different smartphone locations for pedometry [13], we limit our study to these two locations and they should yield the same performance. Prior to equipping the subjects, an observer activated each app to start the data collection, first starting the smartphones and then Glass. Subjects were instructed not to touch the smartphone screens and Glass touchpad during the trial. Subjects were then asked to walk 40 steps in a straight line. Experiments were performed in an indoor environment that consisted of a long hallway that was approximately eight feet wide with floors made out of linoleum tiles. This environment was free of any obstacles and people. Once subjects completed 40 steps, the subject tapped the touchpad on the Glass device to end the trial and then the observer stopped the data collection on the smartphones. To gather data

on running, each subject wore Glass, attached one smartphone to the bicep of their dominant arm using an armband, and placed a second smartphone in their right front pant pocket. Subjects ran 30 steps and followed the same procedure as for walking. 30 steps running was approximately the same distance as 40 steps walking. Data collection on the Glass is ended first then on the smartphones. Subjects were asked to perform each walking and running trial three times. Subjects were split into two groups with the first group first walking and then running and the other group the other way around. Each trial was video recorded. Subjects were instructed to verbally count out their steps. An observer also counted the number of steps each subject took in each trail.

4.4 Results

After running the experiments, the raw acceleration data from all three devices were gathered and analyzed. Each application execution represents a trace. Each entry in a trace includes accelerations in the three major axes with their corresponding timestamps. To analyze this data, the acceleration magnitude was calculated for each entry. The data was then smoothed using a centered moving average. The traces were matched with each other and synchronized based on their timestamps. As the Glass app was activated last and ended first, the start and end timestamps on the Glass trace was used to trim the smartphone traces to avoid picking up unintentional accelerations associated with touching and attaching the smartphones. A video review verified the step count for each trial.

4.4.1 Step counting algorithm

Brajdic & Harle [13] evaluated the performance of various step counting algorithms on smartphones. The best results were found for windowed peak detection WPD, hidden Markov model (HMM), and continuous wavelet transform (CWT) algorithms (see [13] for references). Of these three techniques WPD is simplest to implement. WPD first smooths the acceleration magnitude using a centered moving average window of size (*MovAvg*). WPD detects steps using a sliding window of a fixed size (*PeakDet*) to detect a single peak. The algorithm finds the maximum value in the window and then shifts the window. If, after the shift, a new max is found, the previous peak is replaced if it is still present in the window. We use WPD for our experiments, as a previous study [13] found WPD to offer the same performance as more complex algorithms. WPD is computationally inexpensive to perform and therefore optimizes battery life on mobile devices.

4.4.2 Error Extraction

The error in our experiments is defined as the absolute difference of the number of steps calculated by the algorithm from the ground truth:

$$e_{ij}(k) = \frac{|steps_{ij}(k) - \widehat{steps}_{ij}(k)|}{steps_{ij}(k)} \tag{4.1}$$

where i represents a specific position, k is the trial, $\widehat{steps}_{ij}(k)$ is the approximation of steps subject j took according to WPD, $steps_{ij}$ is the ground truth observation for the number of steps subject j took, and $e_{ij}(k)$ is the error for user j for a specific position i, for a specific trial k. Each subject performed three trials for walking and running. The error for each trial is defined as:

Subject	Walk			Run		
	hand	head	pocket	arm	head	pocket
1	2.5	3.33	1.67	2.22	1.11	3.33
2	3.33	3.33	2.5	0	3.41	2.3
3	0.81	5.73	0	0	0	0
4	5	3.33	3.33	1.08	0	1.08
5	0.83	1.67	4.17	0	0	3.23
6	0	0	1.67	1.11	0	0
7	3.33	0	0	1.08	1.08	1.08
8	0.83	0	1.67	0	0	1.08
9	0	1.67	1.67	3.41	0	1.15
10	7.5	2.5	5	2.19	0	0
11	0	0.83	0.83	0	1.08	1.08
12	1.69	0.85	1.69	2.19	0	0
13	0	2.5	2.5	0	0	2.22
14	1.67	0	5	1.08	3.19	3.18
15	2.5	1.67	2.5	1.08	0	1.11
16	3.33	3.33	2.5	2.12	1.08	2.12
Average	2.08	1.92	2.29	1.1	0.68	1.4
St.Dev	2.02	1.59	1.46	1.04	1.09	1.13

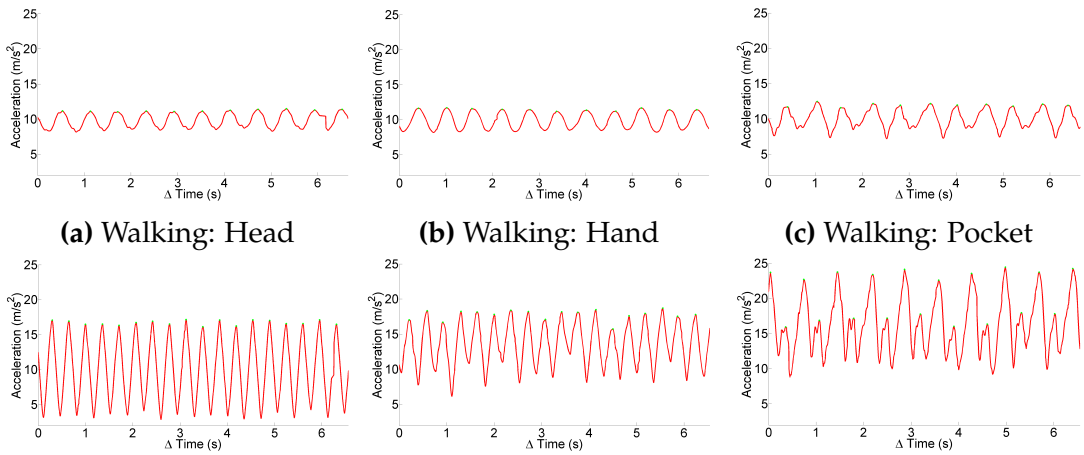
Table 4.1: Step counting error (%) for individual subjects and positions

$$e_{ij} = \frac{\sum_{k=1}^{n} e_{ij}(k)}{n} \tag{4.2}$$

where e_{ij} represents the average error for subject j at for position i, and n represents the number of trials ($n = 3$).

4.4.3 Parameter Optimization

Because running and walking can be accurately classified [8], we use different parameter values for walking and running to optimize the performance of the WPD algorithm. An analysis shows that step counting errors are either caused by: (1) the selection of the window size (*PeakDet*); or (2) noise from the user (*MovAvg*). Too much smoothing eliminates valid peaks and not enough smoothing causes errors



(a) Walking: Head **(b)** Walking: Hand **(c)** Walking: Pocket

(d) Running: Head **(e)** Running: Arm **(f)** Running: Pocket

Figure 4.1: Sample traces of acceleration magnitudes for a walking and running trial. The top traces are for walking with locations: a) head, b) hand, c) pocket. The bottom traces are for running with locations: d) head, e) arm, f) pocket.

due to noise. We optimized WPD parameters, but improvement of the step detection algorithm was not an objective of this research. To determine the optimum values for these parameters, an exhaustive search was performed of all reasonable values to minimize the overall average step counting error for all users and positions. Table 4.2 shows the selected values.

Experiment	Parameter Values
Walking	$MovAvg = 280ms$, $PeakDet = 430ms$
Running	$MovAvg = 175ms$, $PeakDet = 230ms$

Table 4.2: Optimal WPD parameter values for walking and running

4.4.4 Accuracy Comparison

Table 4.1 lists the average step counting error over the three trials for running and walking for the three different positions explored. For the walking trials, the head position achieved the best results with an average error of 1.92% (SD=1.59%). Similarly, for the running trials, the head position yielded the best results with an average error of 0.68% (SD=1.09%). A one-way ANOVA found no statistically signifi-

cant difference in average step counting error rate between head, hand and pocket for walking ($\boldsymbol{F}_{8,39}$ = .968, p = .475) nor running ($\boldsymbol{F}_{2,45}$ = .269, p = .765). For walking and running, no significant difference between over, under and exactly counting of steps was found. A one-way ANOVA found a significant difference in error rate between walking and running for the head and pocket positions ($\boldsymbol{F}_{1,61}$ = 16.863, p = .000). The hand and arm error rates were excluded from this analysis as these positions cannot be directly compared.

4.5 Discussion and Future work

Our experiments found no significant difference in step counting accuracy between a HMD and smartphones worn in the user's pant pocket or held in the hand/arm. This result is contrary to what we expected, e.g., we anticipated step counting to be worse for the head location as accelerations would be more damped, since the head is more distant from the feet than the other locations we explored. Our assumption was also corroborated by related studies [19, 49] that found the accuracy of step-counting to improve the closer the sensor is located to the feet. Figure 4.1 shows six traces of the observed acceleration for the three different locations for a running and walking trial. Though the amplitudes of the peaks for the pocket location are slightly higher than for the head and hand locations, this difference does not significantly affect the performance of the WPD algorithm.

The step counting error for running was significantly lower than for walking. When running, the heel strikes the ground with a higher velocity than when walking, so the magnitude of the observed accelerations is larger. The amplitude and frequency of the peaks are significantly higher for running than for walking. Some

notable differences between traces for different locations of the accelerometer can be observed. The head (Fig. 4.1a & 4.1d) and hand (Fig. 4.1b) traces resemble smooth sinusoids. The arm (Fig. 4.1e) and pocket (Fig. 4.1c & 4.1f) traces look different and resemble unbalanced sinusoids with peaks that alternate in amplitude, e.g. a high peak is followed by a low peak. Because the pocket and arm locations are right above the leg, they pick up accelerations from a heel strike from one leg more strongly than the other. Given these results it seems reasonable to assume that the closer an accelerometer is placed to the sagittal plane, i.e., the vertical plane dividing the body into a left and right half, the more balanced the sinusoid of the acceleration signal will be. For running, the arm picks up additional accelerations from swinging the arms back and forth, which causes the amplitudes of these peaks to be slightly higher than for the head location (see Fig.4.1e & 4.1d).

The Samsung smartphone weighs nearly three times as much as Glass. This difference in mass could affect the ability to accurately pick up accelerations (with heavier devices sensing accelerations more accurately). The larger distance from the feet makes it harder for Glass to pick up accelerations and noise is more falsely classified as a step. However, Glass is pretty firmly attached to the user's head as it rests on the nosebone where the skin is very thin. A human head also weighs approximately 8% of the total body weight and therefore this setup allows for detecting step accelerations accurately. Though we did not use wearable accelerometers like Fitbit for comparison, we do believe our results extend to ear worn accelerometers such as available in Bluetooth headsets.

Wearable devices, such as a HMD or smartwatch, are typically tethered to a smartphone for its mobile data feed. Future work could explore fusion of acceleration information from both devices to create a more accurate pedometer, though

this could be detrimental to battery life. More accurate step counting algorithms could be developed that can exploit the asymmetrical shapes of the pocket and arm location traces. Runners that strike the ground with their forefoot instead of their heel, are less prone to develop knee injuries [41]. These types of strikes could be classified according to their unique acceleration signatures. While running an HMD could convey such information, as well as asymmetries in strikes between feet, so a runner can adjust the way they run in real-time.

CHAPTER 5
CONCLUSION

Advancements in mobile and wearable devices have paved the way to new, exciting technologies that can help users with visual impairment in their every day lives. This dissertation has presented two new systems that can help users navigate in indoor spaces. The first one, Navatar, is able to localize users in a building and provide them instructions in order to reach their destination. The second one, Headlock, can help users navigate large open spaces with minimum veering from the desired path. Additionally, a study have been conducted which proves that heads up displays, like Google Glass, can be used to accurately count the user's steps.

This research provides an exciting new insight to the many possible ways mobile and wearable devices can be used. This work sets the ground to many possible applications that take advantage of the sensory input of wearable devices which can lead to new and innovative ways of using our devices. This has the potential to greatly help people with visual impairments with their every day lives.

BIBLIOGRAPHY

[1] Google glass, `http://www.google.com/glass/start/`.

[2] Cognitive strategies dependent on the hippocampus and caudate nucleus in human navigation: Variability and change with practice. *The Journal of neuroscience*, 23(13):5945–5952, July 2003.

[3] Tomohiro Amemiya, Jun Yamashita, Koichi Hirota, and Michitaka Hirose. Virtual leading blocks for the deaf-blind: A real-time way-finder by verbal-nonverbal hybrid interface and high-density rfid tag space. In *Proceedings of the IEEE Virtual Reality*, Washington, DC, USA, 2004.

[4] E. Apostolopoulos, N. Fallah, E. Folmer, and K. E. Bekris. Feasibility of interactive localization and navigation of people with visual impairments. In *11th IEEE Intelligent Autonomous Systems (IAS-10)*, Ottawa, CA, August 2010 2010.

[5] E. Apostolopoulos, N. Fallah, E. Folmer, and K. E. Bekris. Integrated localization and navigation of people with visual impairments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA 12)*, 2011 (submitted).

[6] Ilias Apostolopoulos, Navid Fallah, Eelke Folmer, and Kostas E Bekris. Integrated online localization and navigation for people with visual impairments using smart phones. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(4):21, 2014.

[7] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. An optimality principle governing human walking. In *IEEE Transactions on Robotics*, volume 24, 2008.

[8] Louis Atallah, Benny Lo, Rachel King, and Guang-Zhong Yang. Sensor placement for activity detection using wearable accelerometers. In *Proceedings of the 2010 International Conference on Body Sensor Networks*, BSN '10, pages 24–29, Washington, DC, USA, 2010. IEEE Computer Society.

[9] M. Bessho, S. Kobayashi, N. Koshizuka, and K. Sakamura. Assisting mobility of the disabled using space-identifying ubiquitous infrastructure. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, pages 283–284, Halifax, Nova Scotia, Canada, 2008. ACM.

[10] J.P. Bigham, C. Jayant, A. Miller, B. White, and T. Yeh. Vizwiz::locateit - enabling blind people to locate objects in their environment. In *Computer Vision*

and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pages 65–72, 2010.

[11] Nick Bilton. New york times blog. disruptions: Visually impaired turn to smartphones to see their world, `http://bits.blogs.nytimes.com/2013/09/29/disruptions-guided-by-touch-screens-blind-turn-to-smartphones-for-sight/`.

[12] Alexander A. Black, Jan E. Lovie-Kitchin, Russell L. Woods, Nicole Arnold, John Byrnes, and Jane Murrish. Mobility performance with retinitis pigmentosa. *Clinical and Experimental Optometry*, 80(1):1–12, January 1997. The contents of this journal from 1998 to 2005 can be freely accessed online via the journal's web page (see link).

[13] Agata Brajdic and Robert Harle. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 225–234, New York, NY, USA, 2013. ACM.

[14] F. Cavallo, A. M. Sabatini, and V. Genovese. A step towards gps/ins personal navigation systems: real-time assissment of gait by foot inertial sensing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1187–1191, Edmonton, Canada, August 2-6 2005.

[15] Burcu Cinaz and Holger Kenn. Headslam - simultaneous localization and mapping with head-mounted inertial and laser range sensors. In *Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers*, ISWC '08, pages 3–10, Washington, DC, USA, 2008. IEEE Computer Society.

[16] F. Dellaert and S. Tariq. A multi-camera pose tracker for assisting the visually impaired. In *1st IEEE Workshop on Computer Vision Applications for the Visually Impaired*, June 20-25 2005.

[17] S. Ertan, C. Lee, A. Willets, H. Tan, and A. Pentland. A wearable haptic navigation guidance system. In *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, pages 164–165. IEEE Comput. Soc, 1998.

[18] Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer. The user as a sensor: navigating users with visual impairments in indoor spaces using tactile landmarks. In *Proceedings of the 2012 ACM annual conference on Human*

Factors in Computing Systems, CHI '12, pages 425–432, New York, NY, USA, 2012. ACM.

[19] Randal C Foster, Lorraine M Lanningham-Foster, Chinmay Manohar, Shelly K McCrady, Lana J Nysse, Kenton R Kaufman, Denny J Padgett, and James A Levine. Precision and accuracy of an ankle-worn accelerometer-based pedometer in step counting and energy expenditure. *Prev Med*, 41(3-4):778–83, 2005.

[20] T.G. Ghirardelli and A.A. Scharine. *Helmet-mounted displays: Sensation, perception and cognition issues*, chapter Auditory-Visual Interactions, pages 599–618. US Army Aeromedical Research Lab, 2009.

[21] N. J. Gordon, D. J. Salmond, and A. F. M. Smithm. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(2):107–113, 1993.

[22] Susan Vincent Graser, Robert P Pangrazi, and William J Vincent. Effects of placement, attachment, and weight classification on pedometer accuracy. *J Phys Act Health*, 4(4):359–69, Oct 2007.

[23] Ye Hanlu, Malu Meethu, Uran Oh, and Leah Findlater. Current and future mobile and wearable device use by people with visual impairments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2014.

[24] J. A. Hardigan. Clustering algorithms. 1975.

[25] Henry E Heffner and Rickye S Heffner. Hearing ranges of laboratory animals. *Journal of the American Association for Laboratory Animal Science*, 46(1):20–22, 2007.

[26] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis. A 3d pose estimator for the visually impaired. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2716–2723, Oct. 11-15 2009.

[27] J. A. Hesch and S. I. Roumeliotis. An indoor localization aid for the visually impaired. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3545–3551, Roma, Italy, April 10-14 2007.

[28] A. Hub, J. Diepstraten, and T. Ertl. Design and development of an indoor navigation and object identification system for the blind. In *Proceedings of the*

6th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS), pages 147–152, Atlanta, GA, USA, 2004. ACM.

[29] Andreas Hub, Joachim Diepstraten, and Thomas Ertl. Design and development of an indoor navigation and object identification system for the blind. *ACM SIGACCESS Accessibility and Computing*, (77-78):147, September 2003.

[30] Ali Israr, Olivier Bau, Seung-Chan Kim, and Ivan Poupyrev. Tactile feedback on flat surfaces for the visually impaired. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, CHI EA '12, pages 1571–1576, New York, NY, USA, 2012. ACM.

[31] V Ivanchenko, J Coughlan, and H Shen. Crosswatch: a camera phone system for orienting visually impaired pedestrians at traffic intersections. *Lecture notes in computer science*, 5105:1122–1128, 2008.

[32] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P. Bigham. Supporting blind photography. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '11, pages 203–210, New York, NY, USA, 2011. ACM.

[33] A. A. Kalia, G. E. Legge, and N. A. Giudice. Learning building layouts with non-geometric visual information: the effects of visual impairment and age. *Perception*, 37(11):1677–99, 2008.

[34] Amy A Kalia, Gordon E Legge, and Nicholas A Giudice. Learning building layouts with non-geometric visual information: the effects of visual impairment and age. *Perception*, 37(11):1677–1699, 2008.

[35] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(35-45), 1960.

[36] Gertrudis I J M Kempen, Judith Ballemans, Adelita V Ranchor, Ger H M B van Rens, and G A Rixt Zijlstra. The impact of low vision on activities of daily living, symptoms of depression, feelings of anxiety and social support in community-living older adults seeking vision rehabilitation services. *Qual Life Res*, Nov 2011.

[37] Vinitha Khambadkar and Eelke Folmer. GIST: a gestural interface for remote nonvisual spatial perception. In *Proceedings of the 26th annual ACM symposium*

on User interface software and technology, UIST '13, pages 301–310, New York, NY, USA, 2013.

[38] Rob Kitchin and Scott Freundschuh. *Cognitive mapping past, present and future*. Routledge, 11 New Fetter Lane, London EC4P 3EE, 2003 edition, 2000.

[39] O. Koch and S. Teller. Wide-area egomotion estimation from known 3d structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, June 17-22 2007.

[40] Nisarg Kothari, Balajee Kannan, and M. Bernardine Dias. Robust indoor localization on a commercial smart-phone. , The Robotics Institute, Carnegie-Mellon University, 2011.

[41] J. P. Kulmala, J. Avela, K. Pasanen, and J. Parkkari. Forefoot strikers exhibit lower running-induced knee loading than rearfoot strikers. *Med Sci Sports Exerc*, 45(12):2306–2313, Dec 2013.

[42] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran. Rfid in robot-assisted indoor navigation for the visually impaired. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1979–1984, Sendai, Japan, Sept. 28-Oct.2 2004.

[43] Simon Lacey, Rebecca Lawson, JackM. Loomis, RobertaL. Klatzky, and NicholasA. Giudice. *Representing 3D Space in Working Memory: Spatial Images from Vision, Hearing, Touch, and Language*, pages 131–155. Springer New York, 2013.

[44] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, and D. S. Wallach. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, 20(3):555–559, June 2004.

[45] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless ethernet. In *Eight ACM International Conference on Mobile Computing and Networking (MOBICOM 2002)*, Atlanta, GA, September 2002. ACM Press, ACM Press.

[46] O. Lahav and D. Mioduser. Multisensory virtual environment for supporting blind persons' acquisition of spatial cognitive mapping – a case study. In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 1046–1051, 2001.

[47] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, April 2000.

[48] X. R. Li, X. Zhi, and Y. Zhang. Multiple-model estimation with variable structure, part iii: model-group switching algorithm. In *IEEE Transactions on Aerospace and Electronic Systems,*, volume 35, pages 225–241, 1999.

[49] Catherine G Ling and Sheila O Smith. Alternative placements for women wearing pedometers. *J Am Acad Nurse Pract*, 22(5):264–9, May 2010.

[50] J M Loomis, R L Klatzky, and R G Golledge. Navigating without vision: basic and applied research. *Optom Vis Sci*, 78(5):282–9, May 2001.

[51] J. M. Loomis, R. L. Klatzky, R. G. Golledge, and J. W Philbeck. Human navigation by path integration. *Wayfinding: Cognitive mapping and other spatial processes . R. G. Golledge, Ed. Johns Hopkins University Press, Baltimore, MD*, pages 125–151, 1999.

[52] Jack M. Loomis, Reginald G. Golledge, and Roberta L. Klatzky. Navigation system for the blind: Auditory display modes and guidance. *Presence: Teleoperators and Virtual Environments*, 7(2):193–203, 1998.

[53] R. Manduchi. Mobile vision as assistive technology for the blind: An experimental study. In *13th International Conference on Computers Helping People with Special Needs (ICCHP'13).*, 2012.

[54] Roberto Manduchi and James Coughlan. (computer) vision without sight. *Commun. ACM*, 55(1):96–104, January 2012.

[55] Chinmay Manohar, Shelly McCrady, Ioannis T Pavlidis, and James A Levine. An accelerometer-based earpiece to monitor and quantify physical activity. *J Phys Act Health*, 6(6):781–9, Nov 2009.

[56] Daniela Mapelli and Marlene Behrmann. The role of color in object recognition: Evidence from visual agnosia. *Neurocase*, 3(4):237–247, 1997.

[57] James R Marston, Jack M Loomis, Roberta L Klatzky, and Reginald G Golledge. Nonvisual route following with guidance from a simple haptic or auditory display. *Journal of Visual Impairment & Blindness*, 101(4):203–211, 2007.

[58] P. S. Maybeck. *Stochastic Models, Estimation and Control*, volume 2 of *Mathematics in Science and Engineering*. Academic Press, New York, N.Y. ; London ; Paris, 1982.

[59] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34:103–123, 1998.

[60] Michael A Nees and Bruce N Walker. Listener, task, and auditory graph: Toward a conceptual model of auditory graph comprehension. In *Proceedings of the International Conference on Auditory Display (ICAD2007)*, pages 266–273, 2007.

[61] Fiona N. Newell, Andrew T. Woods, Marion Mernagh, and Heinrich H. Bülthoff. Visual, haptic and crossmodal recognition of scenes. *Experimental Brain Research*, 161:233–242, 2005. 10.1007/s00221-004-2067-y.

[62] S. M. Oh, S. Tariq, B. Walker, and F. Dellaert. Map-based priors for localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[63] Amit Pande, Yunze Zeng, Aveek K. Das, Prasant Mohapatra, Sheridan Miyamoto, Edmund Seto, Erik K. Henricson, and Jay J. Han. Energy expenditure estimation with smartphone body sensors. In *Proc. of the 8th International Conference on Body Area Networks (BodyNets'13)*, pages 8–14, Brussels, Belgium, 2013.

[64] L. Ran, S. Helal, and S. Moore. Drishti: An integrated indoor/outdoor blind navigation system and service. *IEEE International Conference on Pervasive Computing and Communications*, page 23, 2004.

[65] I. M. Rekleitis. A particle filter tutorial for mobile robot localization. TR-CIM-04-02, Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada, 2002.

[66] T. H. Riehle, P. Lichter, and N. A. Giudice. An indoor navigation system to support the visually impaired. In *30th Annual International Conference of the IEEE Conference on Engineering in Medicine and Biology Society (EMBS)*, pages 4435–4438, Aug. 2008.

[67] Timothy H Riehle, Shane M Anderson, Patrick A Lichter, William E Whalen,

and Nicholas A Giudice. Indoor inertial waypoint navigation for the blind. *Conf Proc IEEE Eng Med Biol Soc*, 2013:5187–90, 2013.

[68] D. A. Ross and B. B. Blasch. Development of a wearable computer orientation system. *Personal and Ubiquitous Computing*, 6(1):49–63, 2002.

[69] David A. Ross and Bruce B. Blasch. Development of a wearable computer orientation system. *Personal and Ubiquitous Computing*, 6(1):49–63, 2002.

[70] S. K. Semwal. Wayfinding and navigation in haptic virtual environments. *Multimedia and Expo, IEEE International Conference on*, 0:143, 2001.

[71] H. Shen and J. Coughlan. Towards a real-time system for finding and reading signs for visually impaired users. In *13th International Conference on Computers Helping People with Special Needs (ICCHP '12)*, Linz, Austria., 2012.

[72] S. Shoval, J. Borenstein, and Y. Koren. Auditory guidance with the navbelt - a computerized travel aid for the blind. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):459–467, August 1998.

[73] S. Skaff, A. Rizzi, H. Choset, and M. Tesch. Context identification for efficient multiple-model state estimation of systems with cyclical intermittent dynamics. *IEEE Transactions on Robotics*, 27(1), February 2011.

[74] R. Smith, M. Self, and P. Cheeseman. *Estimating Uncertain Spatial Relationships in Robotics*. Springer Verlag, 1990.

[75] Brian J Stankiewicz and Amy A Kalia. Acquisition of structural versus object landmark knowledge. *J Exp Psychol Hum Percept Perform*, 33(2):378–90, Apr 2007.

[76] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, September 2005.

[77] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[78] B. Tjan, P. Beckmann, N. Giudice, and G. Legge. Digital sign system for indoor wayfinding for the visually impaired. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - Workshop on Computer Vision Applications for the Visually Impaired*, San Diego, CA, June 20-25 2005.

[79] B. Tsuji, G. Lindgaard, and A. Parush. Landmarks for navigators who are visually impaired. In *Proceedings International Cartography Conference*, 2005.

[80] P. Van Dorp and C. A. Groen, F. Human walking estimation with radar. *IEE Proceedings, Radar, Sonar and Navigation*, 150(5):356–365, 2003.

[81] Michele A Williams, Amy Hurst, and Shaun K Kane. Pray before you step out: describing personal and situational blind navigation behaviors. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 28. ACM, 2013.

[82] D. Yuan and R. Manduchi. Dynamic environment exploration using a virtual white cane. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 243–249, San Diego, CA, June 20-25 2005.