

UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

Easy Return

Permalink

<https://escholarship.org/uc/item/0jz7c81x>

Journal

PROCEEDINGS OF THE 2018 CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI 2018)

ISBN

978-1-4503-5620-6

Authors

Flores, German

Manduchi, Roberto

Publication Date

2018-04-19

DOI

10.1145/3173574.3173591

Easy Return: An App for Indoor Backtracking Assistance

German Flores

Dept. of Computer Engineering
University of California, Santa Cruz
ghflores@ucsc.edu

Roberto Manduchi

Dept. of Computer Engineering
University of California, Santa Cruz
manduchi@soe.ucsc.edu

ABSTRACT

We present a system that, implemented as an iPhone app controllable from an Apple Watch, can help a blind person backtrack a route taken in a building. This system requires no maps of the building or environment modifications. While traversing a path from a starting location to a destination, the system builds and records a path representation in terms of a sequence of turns and of step counts between turns. If the user wants to backtrack the same path, the system can provide assistance by tracking the user's location in the recorded path, and producing directional information in speech form about the next turns and step counts to follow. The system was tested with six blind participants in a controlled indoor experiment.

Author Keywords

Wayfinding; Spatial accessibility; Inertial sensing; Step counting.

ACM Classification Keywords

K.4.2 Social Issues: Assistive technologies for persons with disabilities.

INTRODUCTION

Navigating through a previously unvisited environment can be exceedingly challenging (and potentially unsafe) for people who are blind. Blind travelers cannot recognize visual landmarks at a distance, cannot preview visible portions of a route, and cannot access visual maps. In order to learn the spatial layout of a place while traversing a route, blind travelers must rely heavily on path integration, a mechanism that is known to produce systematic errors [24]. While some blind individuals are able to build fairly precise spatial representations from direct locomotion experience [32], others can develop only a limited, one-dimensional understanding of the environment during route traversal [17]. Technological solutions that can support safe blind

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada
© 2018 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-5620-6/18/04...\$15.00
<https://doi.org/10.1145/3173574.3173591>



Figure 1. An example of an indoor path. A path can be described as a sequence of straight segments separated by left or right turns. For example, this path (way-in) could be represented as: 7-L-3-R-35-R-7-L-52-L-5-R-10, where each number represents the length of the corresponding segment in steps, and L/R indicates left/right turn. The return path has similar representation, but in reverse: 10-L-5-R-52-R-7-L-35-L-3-R-7.

wayfinding may thus be very attractive for increased mobility and independence.

In this work, we are primarily interested in indoor navigation, where a signal from the GPS satellites is too weak for reliable positioning. Indoor traveling is important for independent living. Blind individuals need to be able to navigate through office buildings, hospitals, schools, shopping malls, subways, train stations, airports, and more. Unfortunately, most technologies for indoor self-positioning require some infrastructure in place, such as infrared or radio beacons. Those technologies that don't require an infrastructure (computer vision, inertial sensing) still require access to a map of the building to be visited. While the availability of indoor maps is certainly increasing, thanks to commercial efforts such as Google Indoor Maps, it would be unrealistic to assume that accurate maps of all public buildings will be made accessible in a standard format, possibly annotated with iBeacon RSSI fingerprints [2] or with the location of distinctive visual features.

Our system concept is inspired by the observation that many blind persons are able, through proper orientation and mobility (O&M) training, to move independently in a building once they familiarize themselves with the building's layout. However, when visiting a building for the first time, self-orientation without sight may be very challenging, and blind travelers in these situations normally rely on a sighted human guide [18,23]. We thus aimed to develop a software application that, installed on a regular

smartphone, can provide some level of assistance when a blind traveler is lost or unsure about how to walk back to a specific location, and human assistance is unavailable.

The ability to backtrack one's path is very useful in several situations, as shown by the fact that multiple commercial GPS devices designed for blind outdoor navigation (e.g., Humanware's Trekker Breeze) already support it. One example is the case of a blind person who is led by a sighted human guide to a location inside a building (for example, a doctor's office, a hotel room, or the restroom in a public place), and then wishes or needs to go back to the starting point by himself or herself, perhaps because a human guide is no longer available. Another example is the case of a new blind student (or newly hired employee) who is shown different locations in one or more buildings (in the school or workplace) by a sighted guide. This person will need to memorize the routes taken, in order to be able to re-visit the same locations at some other time. Finally, consider the case of blind individual who, without the help of a sighted guide, decides to explore a building to learn its layout. In all of such cases and more, the proposed backtracking system may enable or facilitate backtracking one's route, saving the traveler from potentially annoying, embarrassing, or unsafe situations.

Our system does not rely on existing maps, and therefore it is functional even when maps are not available. It runs on a regular smartphone (iPhone 6), which users can conveniently keep in their pants pocket. In this way, it does not interfere with any mobility aid (long cane or guide dog) the traveler may be using. In our implementation, the system is controlled by a paired Apple Watch. While the Watch is not necessary for system functioning, it provides easy access for app control.

Our easy return system uses the inertial sensors in the iPhone to count steps and detect turns during traversal of a path in a building. Since most buildings have corridors intersecting at right angles, we only measure turns by $\pm 90^\circ$; including angles of $\pm 45^\circ$ would certainly be possible. While a walker traverses a path, the system builds and records a simple representation of the path as a sequence of left/right turns, as well as of the step count between turns (see Figure 1). The return path is simply the "way-in" path scanned in reverse. When the user walks back to the starting point, the system tracks the user's location against the recorded path, and produces directions (in speech form) in terms of the remaining turns and step counts. The system is designed to handle situations such as step under- or overcounting, or turns mistakenly taken ahead of time or too late, requiring a route correction.

A user study was conducted with six blind walkers (five using a long cane and one using a guide dog). Participants walked through eight different indoor routes; after traversing each route with a sighted guide, they were asked to backtrack the same path, back to the starting point. In half of the routes, participants could only rely on their spatial memory; in the

other half, they were asked to use our backtracking system. Quantitative data was collected under two metrics: *completeness* (the ability to reach the starting point), and *efficiency* (the ability to follow the path in the reverse direction without introducing unnecessary deviations.) Exit surveys were conducted, designed to evaluate the perceived utility and ease of use of the system.

RELATED WORK

There is increasing interest in technology that can support independent wayfinding in indoor, GPS-denied environments for people who are blind. Some of these technologies involve some sort of infrastructure modifications, such as the placement of infrared beacons [7], RFID tags [9], or Bluetooth Low Energy beacons (iBeacon or Eddystone standards) [2]. Unfortunately, these methods invariably require additional costs for installation, calibration (e.g., RSSI fingerprinting for iBeacons [21]) and/or maintenance, which may hamper their wide diffusion. Technologies that require no environment modifications include mobile computer vision [16,26,31,25] and inertial sensing [8,10,12,29,30,15]. Mobile computer vision for wayfinding require use of a camera, either from a smartphone held by hand or attached to the user's garment, or as part of a wearable device (e.g., Google Glass). These systems may recognize features such as doors or signs, may



Figure 2. Three of our participants controlling the system via the Apple Watch.

be trained to recognize a specific environment, or may be used (possibly in conjunction with a depth sensor) to estimate and track the location of the user via SLAM. Compared to computer vision systems, inertial navigation [20,33] relies on sensors that, embedded in a smartphone or other device, can be worn discreetly, require no aiming, and are unaffected by adversarial visual conditions (bad lighting, motion blur, occluders). Positioning by inertial sensors requires time integration of sensory data (dead reckoning), an operation that invariably results in drift increasing over time. Hence, when used for pedestrian navigation, dead reckoning requires some sort of periodic "reset" such as zero-velocity updates [5] (a technique that is only possible for shoe-mounted sensors), or user-initiated updates when a

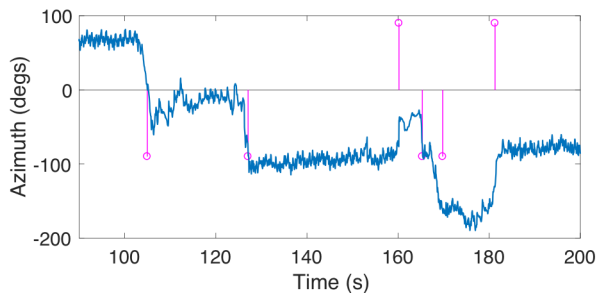


Figure 3. Azimuth time series collected from a blind walker using a cane (from the WeAllWalk data set [13]). The turns (90° or -90°) detected by the algorithm of [14] are shown by the pink stems.

known landmark is reached [12]. A popular alternative is to use the inertial sensors only for step counting [34] and orientation estimation [28,14], then integrate this information to estimate the user's position. In benign situations (users walking with consistent stride length, corridors intersecting at right angles), this simple technique can be quite effective. Note that step counting does not seem to be greatly affected by the exact location of the sensor on the walker's body [3].

Virtually all wayfinding systems require prior knowledge of the map of the building to be visited, with the exception of SLAM-based approaches that build a map during traversal. Rather than a full metric map, our system generates a map of a specific path in terms of turns and step counts, to be used for assisted back-tracking without any knowledge of the building's map. We are aware of only one similar existing backtracking system, the PathGuide Android app [27], which uses inertial and magnetic sensors, and allows for pushing reference tracks to the cloud. PathGuide was designed for sighted users, who can easily manage system errors and inaccuracies (e.g., an incorrect number of steps before the next turn) via visual access. Blind walkers, however, need a higher level of directional accuracy and system resilience to errors. For example, a situation with the system advising a walker to make a right turn in five steps, when the turn is actually a few more steps away, would be perfectly acceptable for a sighted walker. In contrast, a blind user, upon realizing that there is no opening in the wall where expected, would need to start exploring the right wall back and forth, until an opening is found. In the meantime, the system, which is tracking the user's motion, must be able to correctly interpret the inertial data collected during this (often uncoordinated) exploratory phase. As we have found in our experiments, this is a relatively common situation, which needs to be managed by a properly designed algorithm.

METHOD

Participants

Six participants (three females, three males) tested our easy return system. All participants were blind, except for some remaining light perception. Their ages ranged between 22 and 69 (mean: 47; median: 52). One participant (P3) used a

guide dog, while the remaining ones used a long cane. All participants were expert travelers. All participants except for P3 (who didn't use a smartphone) were iPhone users. None of them owned an Apple Watch, one of the reasons being that, at the time of the test, Apple Watch did not support Bluetooth for VoiceOver.

Apparatus

System

Our prototype consists of two devices: an iPhone 6, which the participants carried tucked in one of their pants pockets; and an Apple Watch, paired with the iPhone. Data from the iPhone's inertial sensors was used to detect and count steps taken by the participants, as well as to detect turns (left or right). In addition, the iPhone was in charge of running the path matching algorithm and synthesizing guidance messages. Participants operated the Apple Watch during the trial for two tasks: (1) *system control* (e.g., specifying when to start and end tracking, and when to start and end guidance), and (2) *information access* (querying the system to get updated directions). Note that use of the Apple Watch is not necessary in principle – similar functionalities could have been implemented by other interface mechanisms. However, by enabling control of the system via the Watch, we empowered participants to be in charge of the experiment, with no need for intervention or assistance by the experimenters. The exit survey showed that the participants almost universally enjoyed using the Watch for this purpose.

Step Counting

An essential component of our system, the step counting algorithm detects individual heel strikes as peaks in the accelerometer data measured by the iPhone carried by the user. The step counter parameters were optimized on the WeAllWalk data set [13]. WeAllWalk contains inertial data from 10 blind and 5 sighted participants, who walked along 10 different paths (75 to 300 meters long, containing multiple 45°, 90°, and 180° turns). The inertial data was collected by two smartphones, carried by the participants in different locations of their choice on their garments. Inertial time series in WeAllWalk were manually annotated with ground truth data, indicating whether the walker was on a straight segment of the path, or undertaking a turn.

Turn Detection

Turns by 90° or -90° taken while walking are detected by analysis of the azimuth data provided by the CMAttitude object in the iOS' Core Motion framework. Azimuth (heading) is the angle formed by the heading direction with the Y axis of a fixed reference frame whose Z axis points downwards. We used the turn detection algorithm described in [14], which explicitly tracks drift while also detecting turns. Note that, if unaccounted for, drift (as due to time integration of any residual bias in the gyro data) may result in gross errors in the estimated azimuth angle. The algorithm defines a Hidden Markov Model (HMM), with states representing the difference between the measured azimuth

values at two consecutive time points, as due to either drift or to a turn. More specifically, the state at any point represents one of three possible situations: (1) no change in heading; (2) azimuth drift; (3) a variation in heading due to a turn. Azimuth drift can take value equal to d or to $-d$, where d is an appropriate small constant learned from training data. The azimuth measured at a certain point in time can thus be modeled as the cumulative sum of prior azimuth differences as due to drift or to turns, with additive noise. A variation of the Viterbi algorithm [14] was used to estimate the optimal sequence of states (drift variations and turns). As in the case of our step counter, the parameters of the turn detector were learnt from the WeAllWalk data set [13]. An example of an azimuth time series with the results of the turn detector is shown in Figure 3.

Path Matching

Use of our system comprises two distinct phases. During *way-in*, the user walks from point A to point B without any assistance from the system. The walker may be helped by a human guide; may rely on verbal directions; may have consulted a tactile map; may remember a prior traversal of the same path; or may simply be exploring the place. The *return* phase involves returning to point A. Although there may be more than one path from B to A, our system will only help the user re-trace the original path.

During the way-in phase, the iPhone app is in *tracking* mode. This involves counting steps and detecting turns, and recording the list of turns along with the number of steps in between any two turns. We indicate by $T^{\text{in}}[k]$ the k -th turn amount (90° or -90°) in the way-in, and by $S^{\text{in}}[k]$ the number of steps between the $(k-1)$ -th and the k -th turn ($S^{\text{in}}[1]$ is the number of steps from the starting point¹). If there are N turns before arriving at a destination, $S^{\text{in}}[N+1]$ represents the number of steps between the last turn and the arrival point. The whole path is thus represented as an array: $\text{Path}^{\text{in}} = [S^{\text{in}}[1], T^{\text{in}}[1], S^{\text{in}}[2], \dots, S^{\text{in}}[N], T^{\text{in}}[N], S^{\text{in}}[N+1]]$ (see Figure 1 for an example.)

The return phase begins with the user located at the end point of the way-in path, facing the opposite direction he or she came from. The “ideal” return path is the same as the way-in path, but in reverse order. It is represented as: $\text{Path}^{\text{ret}} = [S^{\text{ret}}[1], T^{\text{ret}}[1], S^{\text{ret}}[2], \dots, S^{\text{ret}}[N], T^{\text{ret}}[N], S^{\text{ret}}[N+1]]$, where $S^{\text{ret}}[i] = S^{\text{in}}[N-i+2]$ and $T^{\text{ret}}[i] = T^{\text{in}}[N-i+1]$. During the return phase, the system is in *guidance* mode. The system still counts steps and detects turns; in addition, it tries to match the current location of the user within the way-in reverse path. Specifically, upon detecting a turn, the path matcher attempts to identify the correct index k of that turn in Path^{ret} . Based on this information, the user can be notified about where he or she stands in the path, and about the remaining waypoints to the destination.

If the user were to walk the return path carefully, taking the same sequence of turns in reverse, and the same number of steps between corresponding turns, path matching would be trivial. In practice, though, our path matching module must be able to manage non-ideal situations. Let $T^{\text{det}}[i]$ denote the i -th detected turn in the return path, and let $S^{\text{det}}[i]$ denote the number of steps counted between the $(i-1)$ -th and the i -th detected turns. The system may detect an incorrect sequence of turns (e.g., $T^{\text{det}}[i] \neq T^{\text{ret}}[i]$), due for example to the user turning too soon, and then having to correct his or her route (see example in Figure 4). Even if turns are detected correctly, the system may measure a different number of steps between corresponding turns (e.g., $S^{\text{det}}[i] \neq S^{\text{ret}}[i]$); this may be due to differences in walking pattern between the way-in and the return phase, or to step counting inaccuracies.

In order to deal with these situations, we devised a path matching algorithm inspired by the longest common subsequence (LCS) problem [6]. Specifically, given the current detected sequence $\text{Path}^{\text{det}} = [S^{\text{det}}[1], T^{\text{det}}[1], \dots, S^{\text{det}}[K], T^{\text{det}}[K]]$ of measured step counts and turns (with K turns detected so far), the algorithm computes two consistent matching subsequences of turns, one from Path^{ret} and one from Path^{det} . “Consistent” subsequences are such that the walker’s orientation is matched after matching turns, where the ideal and measured walker’s orientation after the i -th turn in the return path are defined as: $O^{\text{ret}}[i] = \sum_{j \leq i} T^{\text{ret}}[j]$, $O^{\text{det}}[i] = \sum_{j \leq i} T^{\text{det}}[j]$. Let I^{ret} and I^{det} represent ordered equal length subsequences from $(1, 2, \dots, N)$ and from $(1, 2, \dots, K)$, respectively (the equal lengths of these subsequences is indicated by $\|I^{\text{det}}\|$). The equal orientation constraint for the selected subsequences can be expressed as: $O^{\text{ret}}[I^{\text{ret}}[k]] = O^{\text{det}}[I^{\text{det}}[k]]$ for $k=1, 2, \dots, \|I^{\text{det}}\|$. In addition, we impose that the currently measured orientation ($O^{\text{det}}[K]$) be equal to the orientation after the final turn in the matching subsequences ($O^{\text{det}}[I^{\text{det}}[\|I^{\text{det}}\|]]$).

Among all possible matching sequences satisfying these requirements, we select one minimizing the following cost, which penalizes discrepancies between step counts in intervals between matching turns, as well as deletions in Path^{det} and (initial) deletions in Path^{ret} .

$$C(I^{\text{det}}, I^{\text{ret}}) = \sum_{i=1}^{\|I^{\text{det}}\|} \left(\left(\sum_{j=I^{\text{det}}[i-1]+1}^{I^{\text{det}}[i]} S^{\text{det}}(I^{\text{det}}[i]) \right) - \left(\sum_{j=I^{\text{ret}}[i-1]+1}^{I^{\text{ret}}[i]} S^{\text{ret}}(I^{\text{ret}}[i]) \right) \right) + C_{\text{skip}}(K + I^{\text{ret}}(\|I^{\text{ret}}\|) - 2 \cdot \|I^{\text{det}}\|)$$

Here, C_{skip} is the cost assigned to each deletion of a turn in Path^{det} or Path^{ret} . We set $C_{\text{skip}}=7$ in our study, based on initial trial-and-error experiments.

A (not necessarily unique) optimal matching subsequence is easily computed using dynamic programming, which must be run for each new detected turn. Note that, given the typically small number of turns, computational time is

¹ The superscripts ⁱⁿ and ^{out} indicate turn amounts and step counts during way-in path and “ideal” return path,

respectively. The superscript ^{det} refers to the turn amount or number of steps actually detected during the return phase.

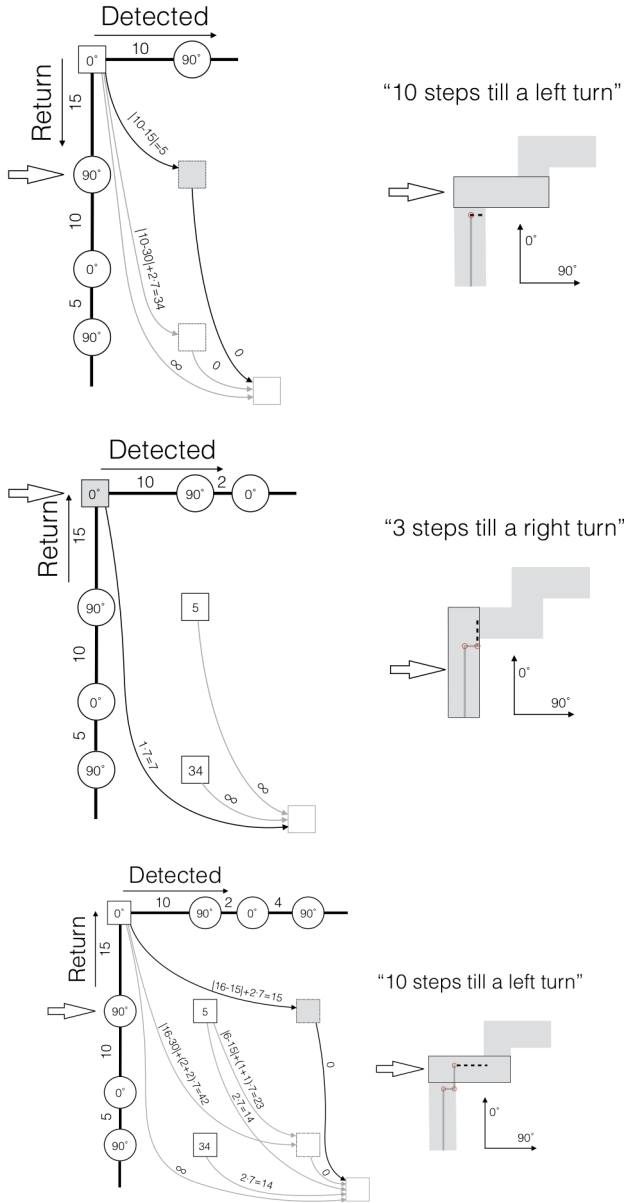


Figure 4. An example with three snapshots during a return path. Left: State graph construction and path cost determination. Right: The return path $Path^{ret}$ (equal to the way-in path $Path^{in}$ in reverse). See text for explanation of the symbols used.

negligible. A simple example of matching subsequence computation is shown in Figure 4 using an oriented graph representation. Each node in the graph represents a potential match between a turn in $Path^{ret}$ and a turn in $Path^{det}$, with a subsequence of turns represented by a path in the graph. Each node is assigned an index (i,j) , where i is the index of a turn in $Path^{ret}$ and j is the index of the matching turn in $Path^{det}$. Edge costs combine both penalties for step count discrepancies and deletion costs. Two dummy nodes are included: an initial node with orientation of 0° , and a final node. All paths must go from the initial to the final dummy

nodes. The edge from a node (i,j) to the final dummy node has cost proportional to the number of terminal turns deleted from $Path^{det}$ when the orientation at this node is consistent with the current orientation ($O^{ret}[i]=O^{det}[K]$), or infinite cost otherwise (thus inhibiting sequences with inconsistent final orientation). The return path $Path^{ret}$ (equal to the way-in path $Path^{in}$ in reverse) is shown by thick gray lines, while the currently detected path $Path^{det}$ is shown by a black line with red circles indicating turns (current location is on the dashed line). The user's orientations $O^{ret}[i]$, $O^{det}[j]$ are shown together with the step counts between turns $S^{ret}[i]$, $S^{det}[j]$ on the vertical axis (Return) and horizontal axis (Detected) of the state graph. Compatible nodes (i,j) with $O^{ret}[i]=O^{det}[j]$ are shown by squares, with the associated optimal sub-path cost inscribed. Nodes associated with newly detected turns are shown with dashed contour. Edges are marked with their associated edge cost. Optimal paths terminating at the dummy end node (shown with gray contour) are marked in black. The last node in the optimal path is filled in in gray.

In the example of Figure 4, the walker took a right turn before its time, ending at a wall. Then, the walker turned left before taking the correct right turn. The system incorrectly matched the first turn detected with the first right turn in $Path^{ret}$, thus producing an (incorrect) direction ("10 steps till a left turn"). However, as soon as the user turned left to correct the route, the system correctly deleted the first right and left detected turns. The direction produced at this time ("3 steps till a right turn") refers to the first segment in $Path^{ret}$, and accounts for the 12 steps already taken by the user. Finally, as the user makes a right turn onto the second segment in the path, the system matches this second turn in $Path^{ret}$, producing the correct direction ("10 steps till a left turn"). An example of successful path matching in a challenging real-world condition is shown in Figure 6.

User Interface

Participants were in full control of the system via the Apple Watch interface. Users could scroll a menu of control items by rotating the Digital Crown in the Watch; items were presented sequentially in speech form, and could be selected by double-tapping the watch's screen. The following items were available: *Track me as I walk*, to be selected at the beginning of the way-in phase; *Stop tracking me*, to indicate arrival at the way-in path destination; *Guide me back*, to initiate the guidance phase; *Stop guiding me*, to end the guidance phase; *Restart system*, to reinitialize the iPhone 6 and Apple Watch for the next tracking phase; *Get last instruction*, to listen to remaining step counts and turns. This option was also actionable by a "swipe up" gesture on the Watch, allowing for faster access.

Speech output was generated by the system in five different situations: (1) At the beginning of the guidance phase; (2) Right after each detected turn; (3) When the user was estimated to be at 7 steps (or less) of the distance from the next turn or destination; (4) Upon estimated arrival to a destination; (5) Whenever the user "swiped up" the Watch's

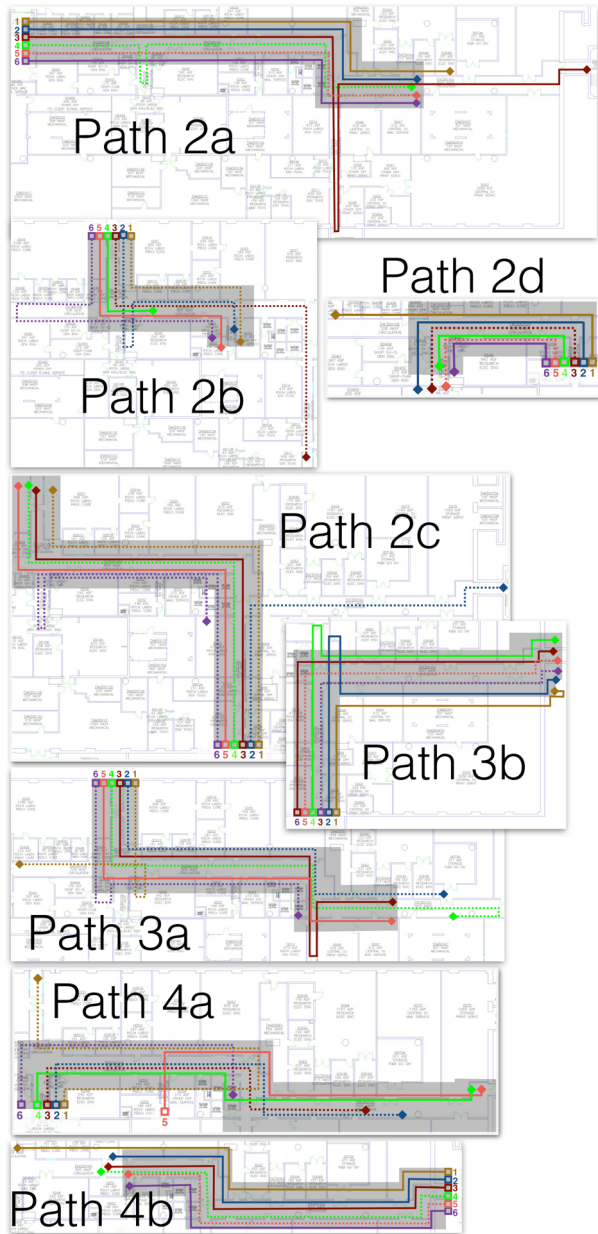


Figure 5. The paths traversed by our participants. The way-in path is shown by a gray stripe. Each return path started at the location marked by a hollow square □ and ended at the location marked by a diamond ◆. Solid line: Modality 1. Dashed line: Modality 2. The number at the beginning of each return path indicates the participant's ID.

screen. In cases (1)–(3), a sentence is generated informing the user of the number of steps till the next waypoint (left or right turn) or, in the last leg of the path, till destination. Note that for case (3), the number of steps in the synthesized sentence is actually reduced by two (the system announces “Five steps till a left/right turn”). This is because this sentence is normally produced while the user is walking, and by the time the sentence is completed, the user typically has already taken a couple more steps. In case (5) (interaction prompted by the user), the system utters the remaining list of

step counts and turns till the destination, in addition to the remaining number of steps till the next waypoint.

Audio output from the system was produced by the iPhone and by the Watch in different contexts. Due to the relatively low volume of sound generated by the Watch, system-prompted notifications, which are likely to be generated while walking, were produced by the iPhone kept in the user's pocket. For all our participants, the volume level was sufficient for correct understanding (note that the experiments were conducted in a very quiet environment). In a more realistic environment, with substantial background noise, earphones or bonephones would be in order. For user-prompted notifications (case (5)), as well as during browsing of the audio menu, speech was produced by the Watch itself. This was acceptable (in spite of its low speech volume) as in these cases users already moved their wrist close to their faces to wake the Watch up and to operate it, and so were able to hear the audio from the Watch.

Experiments

Experiments were conducted individually between May and July of 2017. We chose to run the tests during weekends or in the evening, to minimize the chance of encountering other people in the corridors. Before starting the experiment, and after signing the IRB-approved consent form, each participant was first explained the general concept of the experiment and the functioning of the system. He or she was then handed the iPhone and the Watch, and shown the correct way to interact with the Watch. Then, participants were asked to rehearse use of the system for at least two times on a simple path with two turns until they felt comfortable with its interface and with the tracking/guidance mechanism. At this point, participants were walked to the basement of one of the buildings in our campus (see floor plan in Figure 1), which was the area selected for the experiments. The walkable areas included both wide and narrow corridors containing several turns. All participants were tested with the same sequence of eight paths, shown in Figure 5. Path lengths ranged between 37 and 79 meters (mean: 57 meters). Four paths contained two turns (Path 2a–2d); two paths contained three turns (3a, 3b); and two paths contained four turns (4a, 4b). The participants used our easy return system (*Modality 1*) only in two of the 2-turn paths, as well as in one each of the 3-turn and 4-turn paths. In the remaining paths, participants were asked to backtrack the path without system assistance (*Modality 2*). In this way, each participant ran four trials in *Modality 1* and four in *Modality 2*. We originally designed the sequence of assignments path–modality in such a way that (a) each path would be traversed three times in *Modality 1* and three in *Modality 2* (each time by a different walker), and that (b) each participant would walk an equal number of equal difficulty paths (as measured by the number of turns) with either modality. Due to an oversight, participant P2 was tested with the wrong sequence; this resulted in half of the paths traversed (cumulatively) twice with *Modality 1* and four times with *Modality 2* (thus condition (a) was not

	2a (63m)			2b (37m)			2c (68m)			2d (30m)			3a (60m)			3b (59m)			4a (79m)			4b (57m)		
P1	<4m	✓	109s	<2m	✓	59s	<2m	✓	98s	>6m	–	55s	>6m	–	101s	<4m	–	213s	>6m	–	145s	>6m	✓	168s
P2	<2m	✓	72s	<2m	–	69s	>6m	–	88s	<2m	✓	33s	<6m	✓	79s	<2m	–	148s	>6m	✓	85s	<2m	✓	70s
P3	>6m	–	148s	>6m	–	57s	<2m	✓	60s	<2m	✓	34s	<2m	–	51s	<2m	✓	50s	>6m	✓	54s	<2m	✓	55s
P4	<2m	✓	85s	>6m	✓	44s	<2m	✓	69s	<2m	✓	36s	<6m	–	110s	<2m	–	119s	<2m	✓	80s	<4m	✓	62s
P5	<2m	✓	76s	<2m	✓	53s	<2m	–	136s	<2m	✓	44s	<2m	✓	92s	<2m	✓	75s	<2m	✓	72s	<2m	✓	69s
P6	<2m	✓	67s	<2m	–	67s	>6m	–	122	<2m	✓	27s	>6m	–	58s	<2m	✓	80s	>6m	✓	51s	<2m	✓	73s

Table 1. Performance of each participant (rows) at each path (column). Each cell contains three values: (1) the closest higher threshold (2, 4 or 6 meters) to the distance D_0 (e.g., ‘<4m’ means: 2 meters $< D_0 \leq 4$ meters); (2) the efficiency of the return path (✓ means efficient); (3) the completion time (in seconds) of the return path. Cells with a gray background indicate paths traversed in Modality 1 (using the easy return system).

satisfied). Condition (b) was still respected, meaning that total number of paths with equal difficulty traversed with Modality 1 or Modality 2 was the same. In addition, while walking on Path 4a, P5 was mistakenly guided on a shorter route (still containing 4 turns). Figure 5 shows all of return paths taken by our participants, in Modality 1 (solid line) or Modality 2 (dashed line).

At the beginning of each trial, participants switched the system to tracking mode, by selecting the appropriate entry in the Watch menu, and began walking along the path. During the way-in phase, they received verbal directions from the experimenter (who was walking in front of them) about each turn to take. Upon arrival at destination, participants stopped the tracking mode and turned around. If the path was selected for assisted backtracking (Modality 1), participants switched the system to guidance mode, thus beginning to receive directions from the system. Otherwise, they simply attempted to backtrack the path on their own. Note that for each trial, participants were not informed about the modality that would be used for the return path until the end of the way-in path. At the beginning of the experiment, participants were encouraged to interrogate the system (via swipe-up on the Watch’s screen) as often as they wanted.

In each way-in path traversal (regardless of the return modality), participants were momentarily stopped halfway through the path, and asked to answer a simple question (such as “What did you have for breakfast today?” or “Do you prefer cats or dogs?”). This disruption was meant to momentarily distract participants, with the goal to interfere with memorization of the path being traversed. The goal was to mimic a real-world situation that would cause the walker to temporarily suspend attention to the primary task (walking and memorizing the path) [4]. Interruptions were executed during way-in in both modalities; there were no such disruptions during the return traversals.

At the end of the trials, participants were asked to answer a short questionnaire, with five Likert scale survey questions, and two open-ended questions. The questions and a summary of the answers are shown in Tables 2 and 3.

We should note that a prior study was conducted with three participants using an earlier version of the system. This version ran a much simpler path matching algorithm, with

unsatisfactory results. This prompted us to carry out a major revision the system before conducting the study described here. The results of the prior study are not reported here. None of the participants to the prior study participated in the study described here.

Quantitative Results

We defined two metrics for comparative assessment of the ability of our system to help blind walkers backtrack their path. Both metrics are binary. The first one, *completeness*, determines whether the user was able to reach the end point of the return path within a distance D_0 (we considered the cases $D_0 = 2, 4$ and 6 meters). The second metric, *efficiency*, assesses whether the return path taken by the walker correctly matches (in reverse) the way-in path, or if additional turns were taken that had to be corrected. If the return path extended beyond the reversed way-in path, or vice-versa, only the overlapping sub-path is considered when computing efficiency (e.g., the return paths of P2, P3 and P6 for Path 4a in Figure 5, or the return path of P2 for Path 3a.) In these cases, the return path is efficient but incomplete. In other cases (e.g., the return paths of P2 and P6 for Path 2b), the path may be complete but inefficient.

The completeness and efficiency values are shown together with time to completion in Table 1. Their average over all paths and over the subset of more challenging paths (containing 3 or 4 turns) are shown in Table 2. Table 2 also shows whether the null hypothesis of equal average completeness or efficiency between the two modalities was rejected; this was obtained by first averaging the measured values over the chosen set of paths for each participant, then running a paired-samples t-test. A significant difference (at $\alpha=0.05$) of mean completeness between Modality 1 and 2 was found for the paths containing three or four turns. No significant difference between the two modalities was found in the mean efficiency.

Qualitative Results and Observations

Table 3 reports the results of the exit survey questions on a Likert scale. All participants agreed that the system was easy to use (only P5 gave a Likert score of 4), and all but for P4 (who gave a Likert score of 3) enjoyed controlling it with the

	D_0	Av. completeness %		Av. efficiency %	
		M1	M2	M1	M2
All paths	2m	75	50	71	67
	4m	83	54		
	6m	83	62		
Paths with 3-4 turns	2m	83	25	67	67
	4m	92	33		
	6m	92	50		

Table 2. Average completeness and efficiency for Modality 1 (M1) and 2 (M2). Cases with significant difference ($\alpha=0.05$) are shown in boldface.

Watch-based interface. Participants gave mixed scores to statements about the system giving correct or useful information. P2, P3 and P4 (scores ≥ 4) were more positive than P1, P2 and P6 (scores ≥ 3). Scores were all over the chart for the statement about the system making it easier to find one's way back. P2 and P3 gave a score of 5, while P5 gave it a 1. It should be noted that P5, an expert traveler, was able to almost perfectly re-trace all of his paths by himself. He admitted that, during the trials with the easy return system active (Modality 1), he was for the most part monitoring how the system worked, rather than relying on it for assistance. This may explain his low Likert score: he simply didn't feel that he needed the system for these tests.

Summarized answers to the three open-ended questions are presented in Table 4. Several participants (P1, P5, P6) envisioned real-world situations in which this system could be useful: shopping malls, open spaces, exploring buildings, going to class. P2 and P3 emphasized the importance of "validation" – having the system confirm that one is on the right track. P3 noted that wayfinding requires concentration: the system could help when one is not constantly concentrated on the path. Various desiderata emerged, including the ability to detect 45° and 180° turns (P4, P6), the ability to mark and label multiple routes (P4), and the ability to pause/resume the system (P5). All participants but for P4 found the system useful. P4, on the other hand, found it only marginally useful, commenting that this is "incremental" technology (as opposed to "must have"). Part of the reason for this low rating may be the fact that P4 is an expert traveler with an outstanding sense of orientation. He therefore may not feel as strong a need for a travel assistant as the other participants.

Direct observation of the trials provided useful insights into the way participants used the system, and allowed us to highlight the system's weaknesses. The participant using the guide dog (P3) often (but not always) seemed to simply follow the dog in the Modality 2 return paths. This was effective most of the times, but not in Path 2b. Use of the guide dog proved very beneficial in those situations where a corridor had a 90° bend (in which case, the dog simply followed the bend). Cane users, on the other hand, typically needed to cautiously explore the corridor's walls before ascertaining where to turn. Likewise, when following

directions from the system (Modality 1), she "passed on" a turn directive produced by the system to the dog (e.g., by stopping the dog then pushing the harness in the desired direction); the dog was then able to correctly negotiate the junction. This was particularly useful when, due to incorrect step counting, a turn direction was issued too soon or too late, a situation that was often challenging for cane users.

Most participants quickly learnt to stop or slow down when they heard the "turn in 5 steps" advisory, and to start exploring for a bend or an opening. Unfortunately, the system sometimes overcounted or, more often, undercounted steps in the return path. It appears that the participants might have walked slightly differently when guided by the experimenter in the way-in, than when walking by themselves or assisted by the system in the way back. Participants may have walked more gingerly in the return path (resulting in some missing step detections), or with a different stride length. In either case, the system ended up sometimes producing a direction a couple of steps before or after the correct time. For cane users, this often meant bumping into a wall, or not finding an expected opening, then having to spend some time looking for the right place to turn; a task that sometimes required moving back and forth, potentially confusing the inertial system. For example, Figure 6 shows an example with participant P2 along Path 3d receiving a right turn direction after the correct point, thus having to explore a portion of space looking for an opening (often hitting a desk, which was placed against the wall). This resulted in a number of "spurious" turns detected by the system. Eventually, she was able to find the opening and started walking in the ensuing corridor. Thankfully, our path matching algorithm worked correctly, and the system was able to continue production of meaningful directions.

DISCUSSION

Our easy return system by and large worked as expected, even considering weaknesses such as step over- or under-counting, and occasional confusion in situations with multiple "spurious" turns. We believe that the user studies conducted with the current prototype demonstrated the potential of this technology. It is encouraging that almost all participants were enthusiastic about the concept and envisioned different situations in which it would be beneficial. The participants were also very happy with the system's ease of use. All except for P4 definitely enjoyed controlling it with the Watch-based interface.

Clearly, more work is still needed before this becomes a functional, robust application ready for everyday use. For example, incorrect step counting sometimes resulting in directions that are given too late or too early. We are planning to experiment with more reliable step counters (e.g. [11]) in future work. In practice, though, even the best of systems will produce some errors, such as under- or overcounting steps, resulting in occasional incorrect directions. For the system to be useful, the user needs to understand these limitations, and find an acceptable way to

use information from the system while ultimately being in charge of taking decisions. This is by no means specific to our easy return system. For example, Abdolrahmani et al. [1] discuss how unavoidable errors in a computer vision wayfinding system may affect the final user experience. Each one of our participants quickly found a way to juggle the (possibly incorrect) information from the system with their own (possibly inaccurate) sense of orientation.

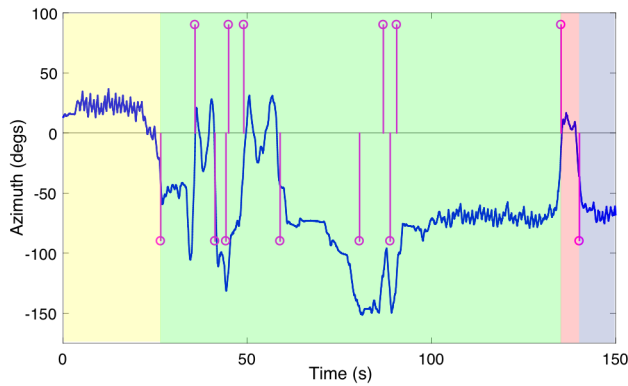


Figure 6. Top: P2 trying to find an opening to the right in Path 3b, following directions from the system. Bottom: The ideal return path (way-in path in reverse) was: 40-R-63-L-6-R-11. The detected path was: 39-R-1-L-3-R-3-R-1-L-1-L-1-R-1-R-1-L-1-R-1-L-52-L-3-R-11.

Our backtracking system enabled a substantially higher rate of return paths completed in the experiment, but this difference was significant only for the more difficult paths (with three or four turns). It seems that the 2-turn paths (which represented one half of the total routes) were easily managed by our participants. This suggests that our system may find most utility in challenging, stressful, or distracting situations, when walkers may have a hard time memorizing and executing the return route. These are precisely the type of situations of perceived utility of our system as mentioned

by our participants in the answers to the first open-ended survey question (Table 3). The two modalities resulted in similar efficiency: even using our system, walkers sometimes took incorrect turns that had to then be corrected.

The ability of the system to recover from “errors” (either by the user, who may take a wrong turn or miss a turn, or by the system) is critical. This was made apparent to us by experience with a prior prototype (as mentioned earlier), which ran a simpler and less effective version of the Path Matching algorithm. When this system was tested with sighted users, who could follow directions while also using visual feedback, it worked flawlessly. However, tests with blind walkers resulted in incorrect directions generated as soon as a path was not perfectly executed. This prompted us to go back to the drawing board, and to design the more refined Path Matching algorithm presented here.

We would like to emphasize that our backtracking system was designed to only work in “corridor networks” inside buildings, with corridors intersection at 90°. While it could easily be extended to the case of 45° intersections [14], it would not be usable in situations including curved paths, or paths bending at arbitrary angles (e.g., in typical open space traversal). In addition, our system is currently unable to provide corrective directions or alternate routes if a user misses a turn or takes a wrong turn.

CONCLUSIONS

We presented a novel system that can assist a blind person attempting to backtrack a path taken in a building. Our easy return system was implemented as an iPhone app, controlled by an Apple Watch. Users don’t need to interact with the iPhone while walking (which could be problematic when handling a long cane or a guide dog), but can keep the phone safely in their pocket. We tested our prototype system in a controlled environment with six blind participants, who walked along eight indoor paths of increasing difficulty with a sighted guide, then, for each path, attempted to return to the starting point, either by themselves, or with assistance from our backtracking system. Use of our system was shown to increase the rate of complete return path traversal, but only for the more difficult routes (containing three or four turns). All of our participants except for one found great potential in the easy return concept, and were enthusiastic about the user interface designed around the Apple Watch.

	P1	P2	P3	P4	P5	P6	μ	σ
<i>I think that the system gave me correct information</i>	3	5	5	4	3	4	4	0.9
<i>I think that the system gave me useful information</i>	3	5	5	5	3	3	4	1.1
<i>I think that the system made it easier for me to find my way back</i>	3	5	5	3	1	3	3.3	1.5
<i>I think that the system was simple to use</i>	5	5	5	5	5	4	4.8	0.4
<i>I enjoyed using the Apple Watch to control the system</i>	5	5	5	3	5	5	4.7	0.8

Table 3. Exit survey – Likert scale questions (1: I don’t agree. – 5: I completely agree). μ = mean, σ = std. dev.

<i>Do you think that an easy return system that works as well or better than the one you tested today would be useful in your daily life, or in the daily life of a blind person you know?</i>	
P1	Absolutely. But it must work much better – it must be trusted and I did not trust it from the beginning. But it did tell me where to turn. Useful for shopping, walking in new areas, situations when you get lost, e.g. how to get out of a parking lot.
P2	Yes. I can imagine (using it) when someone guides you somewhere, and you have it already in the watch, and you can reverse it. For me even if I know the route pretty well it's difficult to turn it around in my head without getting the turns mixed up. And even if you know how to get back, in that way it can tell you 'yes, you are going the right way'. It's a combination of your skill and of the system working together (validation)
P3	Yes. Would be useful for me for sure. I often have a clear picture of where I want to go but if I don't concentrate constantly, I lose. And I don't want to concentrate constantly. There may be ambient noise. You don't always feel you are at the top of your game.
P4	Marginally useful. You won't become rich. Tech is incremental – on their own most apps are nice to have, not "have" to have. I think that dog users walk different, would introduce noise. Mostly supplemental. Would be great if I could mark/label routes then select them.
P5	I do think so. In certain situations (shopping centers, malls, open spaces, when there are lots of twists/turns). Lots of opportunities.
P6	I think so. I don't know if I would use it every day, probably only occasionally, but some people I know would use it when they visit buildings or go to class.
<i>What functionalities would you like to be added to this system?</i>	
P1	Needs to support Bluetooth (earphones) – for when there is background noise, people talking, etc.
P2	If you take a wrong turn, it should put you back in the right track. In a longer route, it should tell you if you are on the right way (not just 'five steps away'). The swipe up functionality is easier here, but in the real world, where there are pedestrians etc., it would be better if it was automatic.
P3	If it could give the names of the office doors, elevator etc. (as I am walking by)
P4	Store multiple routes. Calibrate to learn your gait so you can turn to feet/meters. I am not a step counter, I reason in distances.
P5	I would like the ability to pause/resume tracking or guidance. E.g. if I stop to talk with someone. It should also be robust for situations such as bumping onto someone. Also, I would like the ability to calibrate the device for the particular stride of the user.
P6	In the finished version, I'd like to have a setting where every 15 seconds or so the system would give me the next instruction (like an auto-repeat). Also, I'd like it to identify 45 degrees turns.
<i>Any additional comments/feedback?</i>	
P1	–
P2	It's so fun and I want it! I could have used it yesterday [in a situation in which, due to a fire test alarm in the building, she was not able to call the elevator, and was not sure that she was in the right place]
P3	–
P4	Interesting start. Would like it to work in open hall. Would want 180 and 45 degs turns.
P5	Good and easy to use, especially when Bluetooth will be supported with Voiceover on the Watch.
P6	Good start. Seems to not be very precise in the number of steps it computes. It's good that you don't need cellular data, as sometimes there is no connectivity. Does it need the Watch? Because not everyone has one. The finished version should have Braille support for people who are deaf/blind.

Table 4. Exit survey – Open-ended questions and individual answers (summarized).

REFERENCES

1. Abdolrahmani, A., Easley, W., Williams, M. A., Ronquillo, E., Branham, S., Chen, T., & Hurst, A. (2016). Not all errors are created equal: Factors that impact acceptance of an indoor navigation aid for the blind. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*.
2. Ahmetovic, D., Gleason, C., Ruan, C., Kitani, K. M., Takagi, H., & Asakawa, C. (2016). NavCog: A navigational cognitive assistant for the blind. In *Proc. MobileHCI*.
3. Apostolopoulos, I., Coming, D. S., & Folmer, E. (2015). Accuracy of pedometry on a head-mounted display. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*.
4. Bailey, B. P., & Konstan, J. A. (2006). On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in human behavior*, 22(4), 685-708.
5. Beauregard, S. (2007). Omnidirectional pedestrian navigation for first responders. In *4th IEEE Workshop on Positioning, Navigation and Communication (WPNC'07)*.
6. Bergroth, L., Hakonen, H., & Raita, T. (2000). A survey of longest common subsequence algorithms. In *Seventh IEEE International Symposium on String Processing and Information Retrieval*.
7. Brabyn, J., Crandall, W., & Gerrey, W. (1993). Talking signs: A remote signage, solution for the blind, visually impaired and reading disabled. *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.
8. Chen, D., Feng, W., Zhao, Q., Hu, M., & Wang, T. (2012). An infrastructure-free indoor navigation system for blind people. *Intelligent Robotics and Applications*, 552-561.
9. Chumkamon, S., Tuvaphanthaphiphat, P., & Keeratiwintakorn, P. (2008). A blind navigation system using RFID for indoor environments. In *5th IEEE International Conference on Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology*.
10. Dias, M. B., Steinfeld, A., & Dias, M. B. (2015). Future directions in indoor navigation technology for blind travelers. In Hassan A. Karimi (Ed.), *Indoor Wayfinding and Navigation*, CRC Press.
11. Edel, M., & Köppe, E. (2015). An advanced method for pedestrian dead reckoning using BLSTM-RNNs. *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*.
12. Fallah, N., Apostolopoulos, I., Bekris, K., & Folmer, E. (2012). The user as a sensor: Navigating users with visual impairments in indoor spaces using tactile landmarks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*.
13. Flores, G. H., & Manduchi, R. (2016). WeAllWalk: An annotated data set of inertial sensor time series from blind walkers. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*.
14. Flores, G. H., Manduchi, R., & Zenteno, E. D. (2014). Ariadne's thread: Robust turn detection for path back-tracing using the iPhone. *Proc. IEEE Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*.
15. Flores, J. Z., & Farcy, R. (2014). Indoor navigation system for the visually impaired using one inertial measurement unit (IMU) and barometer to guide in the subway stations and commercial centers. In *Proc. Computers Helping People with Special Needs*.
16. Gleason, C., Guo, A., Laput, G., Kitani, K., & Bigham, J. P. (2016, October). VizMap: Accessible visual information through crowdsourced map reconstruction. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*.
17. Golledge, R. (1999). Human wayfinding and cognitive maps. In Golledge, R. (Ed.) *Wayfinding behavior: Cognitive mapping and other spatial processes*. JHU Press, 1999.
18. Jacobson, W.H. (2012). Orientation and mobility. In R. Manduchi, S. Kurniawan (Eds.), *Assistive Technology for Blindness and Low Vision*, CRC Press, Boca Raton, FL.
19. Kalia, A. A., Legge, G. E., Roy, R., & Ogale, A. (2010). Assessment of indoor route-finding technology for people with visual impairment. *Journal of visual impairment & blindness*, 104(3), 135.
20. Kang, W., & Han, Y. (2015). SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal*, 15(5), 2906-2916.
21. Kriz, P., Maly, F., & Kozel, T. (2016). Improving indoor localization using Bluetooth low energy beacons. *Mobile Information Systems*, 2016.
22. Kulyukin, V., & Nicholson, J. (2012). Toward blind travel support through verbal route directions: A path inference algorithm for inferring new route descriptions from existing route descriptions. *The Open Rehabilitation Journal*, 5:22-40.
23. Long, R.G., & Giudice, N.A. (2010). Establishing and maintaining orientation for mobility. In Wiener, W.R., Welsch, R.L., & Blasch, B.B., (Eds.) *Foundations of orientation and mobility*. AFB Press, NY.

24. Loomis, J.M., Klatzky, R.L., Golledge, R.G., Cicinelli, J.G., Pellegrino, J.W., & Fry, P.A. (1993). Nonvisual navigation by blind and sighted: Assessment of path integration ability. *Journal of Experimental Psychology: General*, 122(1):73-91.
25. Manduchi, R., & Coughlan, J. (2012). (Computer) vision without sight. *Communications of the ACM*, 55(1), 96-104.
26. Manduchi, R., Kurniawan, S., & Bagherinia, H. (2010). Blind guidance using mobile computer vision: A usability study. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*.
27. Microsoft Research Blog. Path Guide: A New Approach to Indoor Navigation. July 14, 2017. <https://www.microsoft.com/en-us/research/blog/path-guide-new-approach-indoor-navigation/> (accessed September 7, 2017).
28. Mohssen, N., Momtaz, R., Aly, H., & Youssef, M. (2014). It's the human that matters: Accurate user orientation estimation for mobile computing applications. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*
29. Pressl, B., & Wieser, M. (2006). A computer-based navigation system tailored to the needs of blind people. In *Proc. Computers Helping People with Special Needs*.
30. Riehle, T. H., Anderson, S. M., Lichter, P. A., Whalen, W. E., & Giudice, N. A. (2013). Indoor inertial waypoint navigation for the blind. In *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.
31. Tian, Y., Yang, X., Yi, C., & Ardit, A. (2013). Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. *Machine vision and applications*, 24(3), 521-535.
32. Tinti, C., Adenzato, M., Tamietto, M., & Cornoldi, C. (2006). Visual experience is not necessary for efficient survey spatial cognition: Evidence from blindness. *The Quarterly Journal of Experimental Psychology*, 59(7):1306-1328.
33. Yang, Z., Wu, C., Zhou, Z., Zhang, X., Wang, X., & Liu, Y. (2015). Mobility increases localizability: A survey on wireless indoor localization using inertial sensors. *ACM Computing Surveys (CSUR)*, 47(3), 54.
34. Yang, C. C., & Hsu, Y. L. (2010). A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10(8), 7772-7788