An Assistive Indoor Navigation System for the Visually Impaired in Multi-Floor Environments

J. Pablo Muñoz[1], *Student Member, IEEE*, Bing Li[2], Xuejian Rong[2], Jizhong Xiao[3*], *Senior Member, IEEE*, Yingli Tian[3], *Senior Member, IEEE*, and Aries Arditi[4]

***Abstract*—This paper presents an innovative wearable system to assist visually impaired people navigate indoors in real time. Our proposed system incorporates state-of-the-art handheld devices from Google's Project Tango and integrates path planner and obstacle avoidance submodules, as well as human-computer interaction techniques, to provide assistance to the user. Our system preprocesses a priori knowledge of the environment extracted from CAD files and spatial information from Google's Area Description Files. The system can then reallocate resources to navigation and human-computer interaction tasks during execution. The system is capable of exploring complex environments spanning multiple floors and has been tested and demonstrated in a variety of indoor environments.**

I. INTRODUCTION

More than 285 million people in the world have some form of visual impairment [1]. The visually impaired face significant challenges when navigating indoors, particularly in unfamiliar environments. We have successfully developed an innovative wearable system to confront this particular challenge. Our context-aware system, Intelligent Situation Awareness and Navigation Aid (ISANA), helps the user navigate to a desired indoor destination. Based on the user's current location, the system produces step-by-step audio instructions, while continuously correcting deviations from the planned path.

Our contributions are threefold: First, we have designed and implemented a working wearable system that successfully helps a visually impaired person navigate indoors in multi-floor environments by using robust data structures and incorporating modules for high-level semantic localization, path planning, obstacle avoidance and human-computer interaction. The system is capable of responding to much of the uncertainty produced by the user's actions and by sudden environmental changes, and the system responds accordingly in real time. Second, we have developed a motion planning algorithm to handle multi-floor navigation challenges. Third, we have implemented a high-level localization layer to hierarchically manipulate semantic information that can better assist the user.

This paper is organized as follows: Section II provides an overview of related work. Section III provides an overview of the system. Section IV describes the procedure for creating the data structures used during path planning and navigation and explains how our system handles navigation in environments spanning multiple floors. Section V covers the system's ability to detect and avoid obstacles. Section VI describes how the system organizes semantic information and its human-computer interaction capabilities, and Section VII provides an evaluation of the system.

II. RELATED WORK

Researchers have directed substantial efforts to address the challenge of assisting visually impaired people navigate indoors and outdoors. But technologies that are successfully used outdoors, e.g., Global Positioning Systems (GPS), cannot be used indoors. For indoor environments, researchers have relied on other methods for localization and navigation, including RGB-D cameras [2] [3] [4] [5] [6] [7], stereo and fisheye cameras [8] [9] [10], iBeacons, laser range finders [11] [12], and inertial measurement units (IMUs) [12] [13]. Some have taken approaches that require altering the infrastructure such as placing digital signs [14] or tactile landmarks [15] inside the building.

Wearable sensors have been used in assistive technologies since the 1990s [16]. In the past decade, researchers have proposed assistive systems that integrate smartphone capabilities [17] and floor plan information [18]. As mobile devices have become more prevalent, *context-awareness* [19] and its integration in assistive navigation technologies have also become an active area of research [20] [21] [22] [23] [24]. Recently, researchers have also explored the use of 3D glasses to assist visually impaired people [25].

We have also considered previous studies that assess the effectiveness of information provided to the visually impaired user [26]. Our system has been carefully designed to keep the human engaged with the system, which has been successfully

This work was supported in part by U.S. Federal Highway Administration (FH-WA) grant DTFH 61-12-H-00002, National Science Foundation (NSF) grants CBET-1160046, EFRI-1137172 and IIP-1343402, National Institutes of Health (NIH) grant EY023483. The devices used in this research were provided by Google's Project Tango. Dr. J. Xiao thanks Google Project Tango for providing a grant to CCNY Robotics Lab as well as free Tango devices and technical support. Dr. J. Xiao thanks the Alexander von Humboldt Foundation for providing the Humboldt Research Fellowship for experienced researchers and Prof. Jianwei Zhang at University of Hamburg for hosting the research stay in summers of 2013-2015. The authors acknowledge Ms. Barbara Campbell for her valuable feedback and suggestion for ISANA. The authors would like to thank Dr. Ivan Dryanovsky, Dr. Chucai Yi, Dr. Samleo L. Joseph, Xiaochen Zhang, Mohammed Amin, Patrick Centeno, Luciano C. Albuquerque, Norbu Tsering for their contributions to this research.

[1]J. Pablo Muñoz is a Ph.D. student of Computer Science, City University of New York - Graduate Center, 365 5th Ave, New York, NY 10016, USA jmunoz2@gradcenter.cuny.edu

[2]Bing Li and Xuejian Rong are Ph.D. students of Electrical Engineering, City College of New York, 160 Convent Ave, New York, NY 10031, USA

[3]Jizhong Xiao and Yingli Tian are faculty of the Department of Electrical Engineering, City College of New York, 160 Convent Ave, New York, NY 10031, USA. *Corresponding Author: jxiao@ccny.cuny.edu

[4]Aries Arditi is principal scientist at Visibility Metrics LLC, 49 Valley View Road, Chappaqua, NY 10514, USA

demonstrated at the U.S. Department of Transportation [27] and IJCAI'16 [28].

III. SYSTEM OVERVIEW

Current technology allows for centimeter accurate localization in indoor environments. We exploit this localization capability by building our system on the Google's Project Tango [29] platform. Project Tango devices localize themselves in an environment by recognizing visual features stored in an Area Description File (ADF). We generate the ADF in advance and then *align* its coordinate system to an occupancy image obtained from extracting semantic information from CAD files or floor plan pictures.

One component of our system is a standalone application that outputs occupancy images and other data structures used by our software. We call this the *Map Editor*, which runs on the Android platform. The *Map Editor* application parses CAD files and extracts semantic information contained in these files. This application can also perform further processing of raw images and output a data structure containing semantic information about the indoor environment and an occupancy grid. Our *Path Planning* and *Navigation* modules make use of these data structures when assisting a visually impaired user. In addition to these data structures, the *Path Planning* and *Obstacle Avoidance* modules use a navigation graph generated by the system (Section IV) and depth information. The results of the *Path Planning* and *Obstacle Avoidance* modules are passed to the *Navigation Assistant*, which is in charge of the high-level decision-making of the system and the coordination of the interaction with the user.

Finally, we have implemented a *Human-Computer Interaction* module that allows the user to add new destinations while exploring an unfamiliar environment, request assistance to reach a destination or request information about her location at any time. The system uses text-to-speech and audio capabilities to indicate to the user the recommended actions that she should take. We expand the explanation of these modules in the following sections of this paper.

IV. PATH PLANNING AND NAVIGATION

A. Navigation graph

Navigation in our system starts with a navigation graph. This weighted graph is based on an occupancy grid image generated from a CAD file or a floor plan picture of the environment.

Our system extracts semantic information from the CAD file that will later be used for high-level localization and assistance to the user. In the case of an unlabelled floor plan picture, our system allows for the input of labelling information in advance and during execution. CAD files often contain text labels that are placed within areas of interest. Our *Map Editor* standalone application uses this information to automatically associate these labels to nodes in the navigation graph. (We discuss the details of how our system handles semantic information in Section VI.) Regardless of the input representation of the environment mentioned above, our system is able to parse the input files and create a grayscale occupancy image I, by applying an adaptive threshold that separates occupied and navigable areas. We use a grayscale format because this will allow us to represent several different kinds of navigable areas and improve the quality of the information provided to the visually impaired user.

The navigation graph is created using a procedure that performs a single pass over the image to create the set of vertices, V, of the navigation graph, based on pixel intensity, $I(i,j)$, and location i,j,

$$\begin{aligned} V = \{[i,j] \mid & \, i \bmod separator == 0, \\ & j \bmod separator == 0, I(i,j) > occupiedTh\}, \end{aligned} \tag{1}$$

where $occupiedTh$ is the adaptive threshold value that determines whether an area is occupied or not. Parameter $separator$ controls the number of pixels that separate each pixel location associated with a vertex. We have determined the value of $separator$ experimentally to be in image space the equivalent of 20 cm in the world frame. The system is sensitive to the value of $separator$. Smaller values for this parameter significantly increase the number of nodes generated in the navigation graph, increasing the time needed to generate and optimize paths. Greater values make the system less precise when guiding the visually impaired person.

B. Path planning

Using the navigation graph described previously, our system can proceed with an informed search to find the best path to a destination. Section VI-B describes how the user would select a destination. By default, we use A* [30] for path finding. A* is complete and produces an optimal path, as was proved by Dechter and Pearl [31], when its heuristic is admissible, i.e., heuristic cost is less than or equal to the lowest possible cost to the destination. Our default heuristic is the L_2 distance in image space. However, the modularity of our system allows for an easy replacement of the heuristic or base algorithm used for path planning. We have added data structures on top of the original A* algorithm to allow for faster reinitialization and replanning.

Initially, execution of the path planner module will result in a path, $P = \{v_1, v_2, ..., v_k\}$, a sequence of k points, that might not be very convenient for the user. For instance, when navigating in a straight hall, the path planner might suggest several intermediate waypoints, which might cause user annoyance due to the number of instructions to reach each of these waypoints. Some of these waypoints can be safely removed in order to create a better experience for the user. We implemented a pruning procedure that reduces the number of waypoints along the path and creates temporary edges in the navigation graph, resulting in an optimized path for a more natural assistance to the user. Assuming that each pair of neighboring vertices in the original path P have a weight w, equivalent to the Euclidean distance in image space for each pair of vertices, the new and improved path P is obtained by the following procedure:

procedure PATHSMOOTHING($P = \{v_1, ..., v_k\}$)
$i \leftarrow 1$
$j \leftarrow k$
$r \leftarrow 1$

while $j - i > 1$ **do**
▷ Check if we can *see* waypoints that are later in the path
if $\lambda(v_i, v_j)$ **then**
$R \leftarrow \{v_l \mid l > i, l < j\}$ ▷ Set of waypoints to be removed from the path
$P \leftarrow P - R$ ▷ Path update
$i \leftarrow 1$
$j \leftarrow \mid P \mid -r$
$r \leftarrow r + 1$
else
$i \leftarrow i + 1$
end if
end while
return P
end procedure

where λ is the boolean function,

$$\lambda(v_i, v_j) = \begin{cases} true & \text{if } I(nint(r.x), nint(r.y)) > h, \\ & r = v_i + qd, \\ & q = v_j - v_i \\ & d = 0 \text{ to } 1,\ step = \frac{1}{\|v_j - v_i\|_2}, \\ false & \text{otherwise} \end{cases} \quad (2)$$

where v_i, v_j are the pixel coordinates of each vertex on the occupancy image that is a waypoint in the path, r is a ray from waypoint at vertex v_i to waypoint at vertex v_j in the path. $nint$ is the nearest integer function. We check the pixel intensity at r, i.e., we check for occupied areas in between waypoints, and vector q is the vector with magnitude equivalent to the distance of the each pair of waypoints in image space, and oriented towards waypoint, vertex v_j. h is the same adaptive threshold described in Equation 1 that decides whether a pixel represents an occupied area or not.

Thus, a ray is traced from the waypoint at image location v_i to the waypoint at image location v_j, determining whether the path can be smoothed or not.

C. Automatic replanning of the path to improve user experience

In instances where the visually impaired person deviates from the planned path, our system keeps track of the user actions and is able to recover in *O(1)* time, producing a new instruction to the user that reflects the new state of the system. If the user deviates from the path, the system brings her back to an optimal position based on the position of the next waypoint and the current distance of the path. To handle these kinds of situations, our system first computes the aggregated distance of all waypoints in the path,

$$d_w = w(u_L, v_1) + \sum_{i=1}^{k} w(v_i, v_{i+1}), k = |P| - 1, \quad (3)$$

where u_L is the current position of the user, and v_1 to v_k are the vertices of nodes (waypoints) in the path, and w is again the weight of the edge, i.e. the Euclidean distance in image space between two vertices. Since the only possible change in each iteration of the system loop is the distance to the next waypoint, v_1, we can safely cache the path total distance between the first waypoint (user immediate destination) and the other waypoints in the path, so it is not unnecessarily computed over and over. This simplifies the computation of d_w to,

$$d_w = w(u_L, v_1) + c_w, \quad (4)$$

where c_w is the cached value. The next step is to compute the optimal distance considering the recent user behavior, i.e., after the user has moved, checking if the system is able to observe a waypoint later in the path. As described previously, the λ function from Equation 2 can be used to check whether the system is able to observe a particular waypoint. First we create a new path by appending u_L, the current position of the user, to P,

$$P_o = \{u_L\} \cup P \quad (5)$$

Next, we execute the PATHSMOOTHING procedure on P_o, and compute the optimal distance,

$$d_o = \sum_{i=1}^{k} w(v_i, v_{i+1}), k = \mid P_o \mid -1, v_1 = u_L \quad (6)$$

The system then proceeds to compare both distances, d_w and d_o, and updates the path if necessary,

$$d_o < d_w \Rightarrow P \leftarrow P_o - v_1, v_1 = u_L \quad (7)$$

D. Multi-floor navigation

Our robust prototype can navigate multi-floor environments by creating a higher level of abstraction that describes the navigation through floors in the building. The system creates a graph that represents the 3D relations of gateway points or areas in each floor. Thus, when navigating to a destination, the system keeps track of the traversal of two different kinds of navigation graphs:

- Navigation in building graph.
- Navigation in floor graph.

If the final destination is on a different floor, the user will be guided to the closest gateway node. When the user arrives at this gateway point, the system will release previous navigation information, reconnect to the localization service using the configuration for the new floor, and after relocalization, guide the user to the next intermediate destination, or final goal, if the user is already on the floor of the final destination. When transitioning between floors, the system currently does not prioritize elevators over stairs. Future work includes an investigation of the trade-off between the comfort to the user in travelling to the closest node, e.g. stairs, versus taking a longer route to the closest elevator. The decision could be taken based on configurable parameters, for instance, when the profile of the user indicates so form of mobility impairment.

To allow for multi-floor navigation, each destination target has the following information: the floor on which it is located, i and j coordinates in image space, a label associated with this node, and whether other areas, e.g., floors, can be accessed through this node.

E. Automatic vs. User-trigger map switch

An earlier version of our prototype switched the maps automatically when the user was within a certain distance of a gateway point if the user was being assisted to reach

a destination on a different floor. After several iterations of development, we have found that letting the user trigger the change of map has many benefits. For instance, when the user is travelling in an elevator, the system remains idle until reaching the appropriate floor.

V. OBSTACLE DETECTION AND AVOIDANCE

We detect obstacles in the user's path with a modified version of the Vector Field Histogram (VFH) [32] [33] based on the depth perception capabilities of the hardware; that is, the Project Tango tablet provides us with a point cloud obtained from the RGB-D camera. Depth information is projected to the floor to establish the obstacle position in the occupancy data structure, and the same point cloud is also projected forward to establish the object's relative height in relation to the user [34]. A beep train is emitted by the system with an interval that represents the proximity of the obstacle. An increase in frequency of beeps means that the user is approaching the obstacle. Unlike the white cane, our system is capable of detecting the height of the obstacle, thus also helping the user to avoid obstacles that are above the level of the torso. The Obstacle Avoidance module alerts the user only when the obstacle detected occupies a previously empty space in the occupancy image.

Point cloud data is received approximately every 200 milliseconds and depending on the detection resolution, the obstacle avoidance module requires between 27 to 56 ms to return valuable information to the user.

A. Obstacle detection and path replanning

Once an obstacle is detected, our path planner proceeds to update the path, taking into account the obstacle dimensions. The replanning due to obstacle detection is handled by the navigation module as follows: First, nodes are removed from the navigation graph based on the point cloud data,

$$P_W = \{[x, y] \mid [x, y, z] \in \text{Point cloud}\}, \tag{8}$$

where P_W is the set of 2D points projected from all the points in the original point cloud. Next, obstacle points, i.e., the points that are within a distance range from the user are transformed to image coordinates,

$$P_I = \{[i, j] \mid Ap, p \in P_W\}, \tag{9}$$

where A is the transformation matrix from world coordinates to image coordinates obtained from the alignment of the Tango's Area Description File and our occupancy data structure. Next, we change the properties of the occupancy image, and the navigation graph,

$$\begin{aligned} &\forall p_i \in P_I \forall p_j \in V \{\|p_i - p_j\|_2 \leq \frac{separator}{2} : \\ &\underbrace{I(p_i.x, p_i.y) = 0}_{New\ occupied\ cells}, \quad \underbrace{V \leftarrow V - p_j}_{Vertex\ removed\ from\ graph} \quad \}, \end{aligned} \tag{10}$$

where P_I is the set of points in image coordinates that the detected obstacle occupies. The system updates the occupancy image and matches all these points with the closest vertices that are then removed from the navigation graph. The criteria for matching are that they have to be at maximum $\frac{separator}{2}$ distance in image space. $separator$ is the parameter described in Section IV for the pixel separation between vertices in the navigation graph. Finally, we replan the path to a destination.

B. Restoring the navigation graph after avoiding an obstacle

Due to the dynamic nature of many obstacles, our system needs to be able to restore the navigation graph so the user is able to visit areas where obstacles previously were but are no longer present. In the case of dynamic obstacles, e.g., people crossing in front of the user, the system is able to restore the navigation graph and re-plan in less than 8 milliseconds, making it imperceptible to the user, thanks to an efficient caching mechanism that keeps track of the real time changes to the graph.

VI. CONTEXT AWARENESS AND HUMAN COMPUTER INTERACTION

A. Organization of Semantic Information

Each node object in the navigation graph has a $label$ field. We use this label to name small areas in the environment. In the case of bigger areas, e.g., narrow halls, elevator banks, we have created an $area$ structure that encapsulates the properties and methods related to these spaces. To organize the associated semantic information, we build a hierarchical tree of areas. Once the tree is built, the user can query the system for its area location. The system will search the tree using the position of the user and will output the corresponding information when it has reached a leaf in the tree.

B. Human-Computer Interaction

We have enabled voice recognition and gesture detection capabilities for receiving commands from the user. Voice recognition is activated using the wake-up keyword "Isana". Gesture detection has proved to be very useful in noisy environments. We are currently using simple gestures, e.g., swiping left to right, to help the visually impaired person learn the available gestures in a short time. In addition to voice recognition and gesture detection, the user can enable the system to receive commands by listening to touch events on the device's screen. Text-to-speech and auditory feedback have been implemented to guide the user. The user can also mute speech output commands at any time.

Once the system is running, the initial step is for the user to select the destination via speech recognition or gesture detection. The system proceeds to load all the information required for effective assistive navigation. First, the map, occupancy grid and a list of destinations are loaded. If the user wants to add a new destination while the system is running, it can do so by indicating the corresponding label that should be assigned to the closest node to the current location. After the resources have been loaded, the next step is to use the Tango's capabilities to localize the system in the selected environment. Once the system has successfully localized itself, the user can proceed to select a destination. The system will generate a path based on the user's preferences and after searching for the best possible trajectory.

As the user moves in the environment, the system keeps track of the next waypoint and decides whether a waypoint has to be

removed from the path. We do not expect the user to follow the path with great accuracy. Thus, using the algorithm described in Section IV, the system will notify the user if it detects that she is deviating from the path. Taking into account the unpredictable behavior of the user, the system continuously checks whether it can better guide the user if a further waypoint is detected. It does so by using the λ function described in Equation 2. The system continuously checks for obstacles and alerts the user if any has been detected. If for some reason the system gets lost in the environment, e.g., failure in localization, an assistant module is activated to guide the user's actions and recover the system's capabilities.

Navigation instructions are given based on the heading direction to the next waypoint. The system also takes into account the distance to the waypoint and it is capable to provide this information to the user. Heading category values have been determined experimentally to ensure a pleasant experience to the user.

In multi-floor environments, once the user has reached a gateway area, the system is capable of warning or guiding the user in her transition to the new area. For instance, if the transition areas are escalators or an elevator, the system can inform the user that she has to proceed with caution or where the exact location of the closest elevator is.

User preferences are read from a configuration file. Among the properties that can be customized by the user are the estimated stride length of the user, the preferred distance to walls, and the intervals between receiving new instructions.

To manage possible conflicting instructions to the user, e.g., when the path requires the user to go straight but there is an obstacle in front of her, the system has a built-in functionality for setting the priority of the information that is given to the user.

VII. EXPERIMENTS

A. Setup and Results

We evaluated different aspects of our prototype using Google's Project Tango's Yellowstone tablets. We tested the system with several blindfolded users to refine our implementation. We also randomly generated an environment spanning an area of 641.98 m^2 (29.26m x 21.94m). We tested 147 multiple trajectories, with a maximum trajectory length of 325.45 m.

We experimented with different spacing of the nodes to construct the navigational graph. Experimentally, we found that setting $separator$ equal to 7 and making it equivalent to 20cm in the world frame yielded the best experience for the user. This can be easily done by modifying the size of the occupancy image. In the case of the testing environment, it took an average of 400 milliseconds for the system to create the navigation graph with this configuration.

As the distance to a destination increases, the length of the path (number of nodes) increases linearly with A*. Depending on the characteristics of the environment and using the smoothing procedure, the system is able to keep the number of nodes in the path to a minimum. For the longest trajectory to a destination, applying the path smoothing procedure resulted in a path with 119 nodes, while without executing this procedure, the number of nodes in the path was 1475. For the maximum trajectory length in our testing environment, 325.45 m., the system only required 21 additional milliseconds to smooth the path.

As was expected, the total distance that the user had to travel was reduced by executing the path smoothing procedure. The additional path smoothing procedure reduced the number of waypoints, thus, creating more straight lines in the path and resulting in a saving of 14.93 meters in the longest trajectory to the furthest destination in our testing environment (325.45 m). Even more important than path shortening, path optimization makes the path simpler for the visually impaired user.

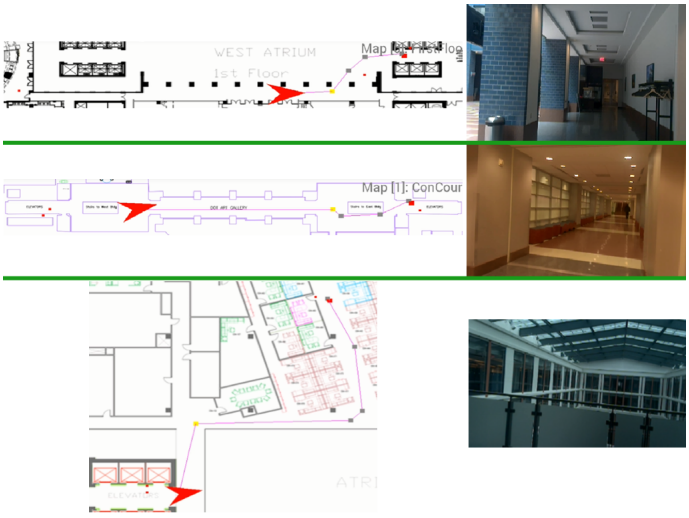

Fig. 1. Three example areas during the demonstration at the Department of Transportation. Top: The user is in an area with columns and big open space on the left. Several obstacles are present and the user has to carefully navigate between the columns up to a point in which she is guided towards the elevators. Middle: The user is being guided through a hallway with very similar visual features. At the end of the hallway there are stairs and columns that present additional challenges to the system Bottom: The user is about to turn and traverse a narrow area before entering a floor with multiple rows of cubicles. The system allows space for the user to turn and navigate in this narrow area.

B. Demonstration

We successfully demonstrated the system at the Department of Transportation (DOT) in Washington, D.C. in March 2016 [27]. The demonstration consisted of assisting a visually impaired person navigate from the main entrance of DOT headquarters to an office located on the eighth floor of an adjacent and interconnected building. To reach the destination, the system had to assist the person in three different environments: narrow hallway followed by a big open space; an environment with few distinctive marks and endpoints with similar appearances; on a floor full of cubicles with similar characteristics and with several obstacles (including columns) that were located in the path of the user. Figure 1 shows the system providing assistive navigation at the Department of Transportation headquarters. In the first example, the user traverses an area with multiple columns that divide a big open space. In the second example, the user is a long hallway that contains very similar visual features. Finally, in the third

example, the user is turning in a narrow area outside of the elevators bank.

VIII. CONCLUSIONS

In this paper, we have described an innovative wearable system that advances the goal of efficiently helping visually impaired people navigate indoors in unfamiliar environments. We use state-of-the-art devices from Google's Project Tango, combining techniques from Robotics and implementing Human-Computer Interaction functionality consistent with blind user's abilities, to provide an efficient, seamless, and pleasant experience to the user. We extract semantic information from CAD files and floor plans to provide navigation information in a user-friendly format. We are constantly improving the functionality for the user to provide additional semantic information to the system during execution. The path provided to the visually impaired person has been efficiently pruned to minimize the number of instructions provided to the user. We have implemented data structures and procedures that reduce the required computation and complexity of the assistive navigation problem. We successfully demonstrated the system in public venues and in environments spanning multiple floors. Videos of the system are available at: http://www.assistiverobot.org/ain/.

REFERENCES

[1] World Health Organization, "Towards Universal Eye Health: a global action plan 2014-2019," Tech. Rep., 2013. [Online]. Available: http://www.who.int/blindness/AP2014_19_English.pdf?ua=1

[2] Y. H. Lee and G. Medioni, "RGB-D camera Based Navigation for the Visually Impaired," *RSS 2011 RGBD: Advanced Reasoning with Depth Camera Workshop*, pp. 1–6, 2011.

[3] ——, "RGB-D Camera Based Wearable Navigation System for the Visually Impaired," *Comput. Vis. Image Underst.*, vol. 149, no. C, pp. 3–20, aug 2016.

[4] H. He, Y. Li, Y. Guan, and J. Tan, "Wearable Ego-Motion Tracking for Blind Navigation in Indoor Environments," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1181–1190, oct 2015.

[5] M. Zöllner, S. Huber, H.-C. C. Jetter, H. Reiterer, M. Zollner, S. Huber, H.-C. C. Jetter, and H. Reiterer, "NAVI: A Proof-of-concept of a Mobile Navigational Aid for Visually Impaired Based on the Microsoft Kinect," in *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part IV*, ser. INTERACT'11, vol. 6949 LNCS, no. PART 4. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 584–587.

[6] M. Brock and P. O. Kristensson, "Supporting Blind Navigation Using Depth Sensing and Sonification," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, ser. UbiComp '13 Adjunct. New York, NY, USA: ACM, 2013, pp. 255–258.

[7] A. Aladrén, G. López-Nicolás, L. Puig, and J. J. Guerrero, "Navigation Assistance for the Visually Impaired Using RGB-D Sensor With Range Expansion," *IEEE Systems Journal*, vol. 10, no. 3, pp. 922–932, sep 2016.

[8] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet, "A Generic Fisheye camera model for robotic applications," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 1, no. c, 2007, pp. 1683–1688.

[9] T. Schwarze, M. Lauer, M. Schwaab, M. Romanovas, S. Böhm, and T. Jürgensohn, "A Camera-Based Mobility Aid for Visually Impaired People," *KI - Künstliche Intelligenz*, vol. 30, no. 1, pp. 29–36, 2016.

[10] V. Pradeep, G. Medioni, and J. Weiland, "Robot vision for the visually impaired," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, pp. 15–22.

[11] J. A. Hesch and S. I. Roumeliotis, "An indoor localization aid for the visually impaired," *Proceedings - IEEE International Conference on Robotics and Automation*, no. April, pp. 3545–3551, 2007.

[12] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis, "A 3D pose estimator for the visually impaired," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 2716–2723, 2009.

[13] L. H. Chen, E. H. K. Wu, M. H. Jin, and G. H. Chen, "Intelligent Fusion of Wi-Fi and Inertial Sensor-Based Positioning Systems for Indoor Pedestrian Navigation," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 4034–4042, nov 2014.

[14] G. E. Legge, P. J. Beckmann, B. S. Tjan, G. Havey, K. Kramer, D. Rolkosky, R. Gage, M. Chen, S. Puchakayala, and A. Rangarajan, "Indoor navigation by people with visual impairment using a digital sign system." *PloS one*, vol. 8, no. 10, p. e76783, 2013.

[15] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, "The User As a Sensor: Navigating Users with Visual Impairments in Indoor Spaces Using Tactile Landmarks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 425–432.

[16] A. R. Golding and N. Lesh, "Indoor navigation using a diverse set of cheap, wearable sensors," in *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*. IEEE, 1999, pp. 29–36.

[17] I. Apostolopoulos, N. Fallah, E. Folmer, and K. E. Bekris, "Integrated Online Localization and Navigation for People with Visual Impairments Using Smart Phones," *ACM Trans. Interact. Intell. Syst.*, vol. 3, no. 4, pp. 21:1—-21:28, 2014.

[18] K. C. Lan and W. Y. Shih, "Using smart-phones and floor plans for indoor location tracking - Withdrawn," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 2, pp. 211–221, apr 2014.

[19] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," in *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, ser. WMCSA '94. Washington, DC, USA: IEEE Computer Society, 1994, pp. 85–90.

[20] I. Afyouni, C. Ray, and C. Claramunt, "Spatial models for context-aware indoor navigation systems: A survey," *Journal of Spatial Information Science*, vol. 4, no. 4, pp. 85–123, 2012.

[21] I. Afyouni, "Knowledge Representation and Management in Indoor Mobile Environments," Ph.D. dissertation, 2013.

[22] W. S. Gribble, R. L. Browning, M. Hewett, E. Remolina, and B. Kuipers, "Integrating Vision and Spatial Reasoning for Assistive Navigation," in *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, no. 003658. London, UK, UK: Springer-Verlag, 1998, pp. 179–193.

[23] F. Lyardet, J. Grimmer, and M. Mühlhäuser, "CoINS: Context sensitive indoor navigation system," in *ISM 2006 - 8th IEEE International Symposium on Multimedia*, 2006, pp. 209–216.

[24] F. Lyardet, D. W. Szeto, and E. Aitenbichler, "Context-Aware Indoor Navigation," in *Ambient Intelligence, Proceedings*, vol. 5355, 2008, pp. 290–307.

[25] S. Mattoccia and P. Macri, "3D Glasses as Mobility Aid for Visually Impaired People," in *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III*, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Cham: Springer International Publishing, 2015, pp. 539–554.

[26] E. Havik, M. K. Steyvers, and J. J. Aart C ; Frank, "The Effectiveness of Verbal Information Provided by Electronic Persons," *Journal of Visual Impairment & Blindness*, vol. 105, no. 10, pp. 624–638, 2011.

[27] "http://www.assistiverobot.org/ain/," 2016.

[28] J. P. Muñoz, B. Li, X. Rong, J. Xiao, Y. Tian, and A. Arditi, "Demo : Assisting Visually Impaired People Navigate Indoors ," in *IJCAI International Joint Conference on Artificial Intelligence*, 2016, pp. 4260–4261.

[29] J. Lee and R. Dugan, "Google Project Tango." [Online]. Available: https://www.google.com/atap/projecttango/#project

[30] P. E. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems, Science and Cybernetics*, vol. 4, no. 2, pp. 100 – 107, 1968.

[31] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality af A*," *Journal of the ACM*, vol. 32, no. 3, pp. 505–536, 1985.

[32] J. Borenstein and Y. Koren, "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 535–539, aug 1991.

[33] ——, "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[34] B. Li, J. P. Muñoz, X. Rong, J. Xiao, Y. Tian, and A. Arditi, "ISANA: Wearable Context-Aware Indoor Assistive Navigation with Obstacle Avoidance," in *Fourth International Workshop on Assistive Computer Vision and Robotics (ACVR) in conjunction with ECCV 2016*, 2016.