ASSIST: Personalized indoor navigation via multimodal sensors and high-level semantic information

Vishnu Nair[1], Manjekar Budhai[1], Greg Olmschenk[1,2], William H. Seiple[3], and Zhigang Zhu[1,2]

[1] Department of Computer Science, The City College of New York, NY, USA
[2] Department of Computer Science, CUNY Graduate Center, NY, USA
[3] Lighthouse Guild, New York, NY, USA
{vnair000,mbudhai000}@citymail.cuny.edu, golmschenk@gradcenter.cuny.edu, WSeiple@lighthouseguild.org, zhu@cs.ccny.cuny.edu

Abstract. Blind & visually impaired (BVI) individuals and those with Autism Spectrum Disorder (ASD) each face unique challenges in navigating unfamiliar indoor environments. In this paper, we propose an indoor positioning and navigation system that guides a user from point A to point B indoors with high accuracy while augmenting their situational awareness. This system has three major components: *location recognition* (a hybrid indoor localization app that uses Bluetooth Low Energy beacons and Google Tango to provide high accuracy), *object recognition* (a body-mounted camera to provide the user momentary situational awareness of objects and people), and *semantic recognition* (map-based annotations to alert the user of static environmental characteristics). This system also features personalized interfaces built upon the unique experiences that both BVI and ASD individuals have in indoor wayfinding and tailors its multimodal feedback to their needs. Here, the technical approach and implementation of this system are discussed, and the results of human subject tests with both BVI and ASD individuals are presented. In addition, we discuss and show the system's user-centric interface and present points for future work and expansion.

Keywords: Indoor positioning, environmental & situational awareness, Bluetooth beacons, Google Tango

1 Introduction

Assistive technologies aim to open access to skills and opportunities that are often inaccessible to those with disabilities. Considering that there are 285 million blind & visually impaired (BVI) individuals worldwide [22] and that people with Autism Spectrum Disorder (ASD) often lack the ability to develop cognitive maps of places they have been to [8], a need was identified for an assistive technology that can aid these individuals in indoor navigation. In light of this, we propose a specialized, full-fledged, multisensor system called ASSIST ("Assistive

Sensor Solutions for Independent and Safe Travel") with the goal of promoting independent and safe travel within complex indoor environments for BVI and ASD individuals. ASSIST is centered around an Android mobile application that relies on the use of Bluetooth Low Energy (BLE) beacons alongside the area learning, motion tracking, and localization capabilities provided by Google Tango. In addition to providing turn-by-turn indoor navigation, we introduce provisions for situational and environmental awareness, including people detection/recognition and static environment information. These capabilities are combined and presented in a flexible and user-friendly application ("app") which can be operated using either touch or voice inputs and can be configured as needed by varying the type and level of feedback, allowing for a unique experience for each user. Our main goal with this system is to improve the quality of life of our users by promoting confidence and independence when it comes to daily indoor navigation. To this end, our work has the following four unique features:

1. **A multi-level recognition mechanism for robust navigation**: (a) Location recognition by improving our previously-created hybrid BLE-Tango system [10] to ensure robustness; (b) object recognition by utilizing a wearable camera to provide reliable alerting of dynamic situational elements (such as people in the user's surroundings); and (c) semantic recognition by using map annotations to provide alerting of static environmental characteristics.
2. **User-centric multimodal interfaces**: The ASSIST app provides a user-centric interface that features multimodal feedback, including a visual interface, voice input and feedback, and vibration reminders. Users can also customize the interfaces for various metrics (steps, meters, feet) and modalities (visual, audio, tactile) based on their challenges (i.e., BVI or ASD).
3. **Near real-time response and zero training**: The ASSIST system is optimized such that information is provided to the user in near real-time. Next to no training is needed for a user to use the app and system. We have also performed user-centric, real-world tests with the overall system to determine its usability to people with disabilities, including BVI and ASD individuals (the results for which are presented).
4. **Modular hardware/software design**: We formulate a hardware/software workflow to produce a working system and open avenues for future work. A modular implementation is targeted to allow for easy adding/upgrading of features.

2 Related Work

2.1 Indoor map learning and localization

Research into accurate indoor positioning and navigation has proposed the use of various technologies, including but not limited to the use of cameras on smartphones [9], RFID tags [3], NFC signals [13] and inertial measurement unit (IMU) sensors [16]. Bluetooth Low Energy (BLE) beacons have been a popular technology of interest; perhaps the most relevant project is NavCog, a smartphone-based "mobility aid" which solely uses BLE beacons to provide turn-by-turn

navigation and information about nearby points-of-interest and accessibility issues [1]. Another BLE-based system proposed the use of beacons as part of a system to provide the visually impaired with information about the topology of an approaching urban intersection [2]. However, these BLE-based systems have relatively low localization accuracy (up to meters) and, thus, cannot work well in crowded or cramped indoor environments. Google Tango has also been of interest with the most relevant project being ISANA, a context-aware indoor navigation system implemented using Tango, which parses CAD files to create indoor semantic maps which are then used in path planning alongside other assistive features such as sign reading and obstacle alerting using the onboard camera [7]. However, limited real-world tests have been performed.

Our own previous work proposed a method of indoor localization that involves combining both BLE beacon localization and Google Tango map learning to create a highly accurate indoor positioning and navigation system [10]. The work we present here extends the work in [10] by providing a multi-layer recognition mechanism and generalizing coverage of the modeling and navigation across multiple floors of a building. In addition, new interfaces are created, and human subject tests are also performed for both BVI and ASD users; whereas, our previous work only tested the system with BVI users.

2.2 Object detection and recognition

Object detection is an integral part of providing situational awareness. Detecting and classifying local persons or objects, within real-time speeds, is a key point of research that can improve safety for users. YOLOv2 is a convolutional neural network (CNN) that was built with the goal of being able to detect a large number of classes and having fast detection speeds by applying the network to an entire image, as opposed to localized areas [15]. Using a smartphone as the main mode for detecting objects and alerting users is another main point of research. [20] use the Lucas-Kanade algorithm, in addition to other optical flow methods, to identify and track potential obstacles. Attempting to improve the detection performance, as well as providing vibrotactile and audio alerts for their users, [14] limit the total number of pixels needed for performing detection, and only analyze the floor-area immediately in front of the phone's camera.

2.3 Methods of environmental understanding

Much research has been done on giving those with cognitive and visual disabilities a greater understanding of their surrounding environment. Visually impaired individuals usually use a cane to detect obstacles in their immediate vicinity. Some studies have attempted to put sensors on canes to preemptively warn the user about upcoming obstacles [18]. Other projects have taken a more vision-based approach. A project called "SoundView" uses a mini-CCD camera to detect objects tagged with barcodes and relay information about the presence of these objects to a visually impaired user via an earpiece [11]. Another project

developed a sensor module that acted like a barcode scanner that a user could use to obtain information about the characteristics of an object of interest [5].

The system we propose is targeted toward users who have difficulties in developing cognitive maps of complex (and often unfamiliar) environments. To this end, several works have been published that use a wearable camera to recognize locations and localize within an environment. Furnari et al. propose a method to segment egocentric (first-person view) videos based on the locations visited by the holder of the camera [4]. Ortis et al. then extend this work to automatically connect the habitual actions of users with their locations [12]. Finally, Spera et al. extend this automatic recognition of locations in egocentric videos to localize shopping carts within a large retail store [17]. Our work utilizes a hybrid sensor approach so that the system may continue to work even if one sensor modality (such as the camera) fails to work properly.

3 System Sensory Components

ASSIST consists of three primary components: location recognition via hybrid sensors, real-world person and object detection via a body-mounted camera, and map-based semantic recognition of the user's environment. These three components interact with each other to provide a user with sufficient information to move them successfully to their destination while augmenting their understanding of the environment around them. With regards to the initial setup of a location, it is worth noting that, other than the initial installation of BLE beacons at strategic positions hidden from view, no manipulation of the visible environment is required for the system (including the camera portion) to work correctly.

3.1 Location recognition via hybrid sensors

Two methods of indoor positioning were of particular interest to us: Bluetooth Low Energy (BLE) beacons and Google Tango. (Note that, although we have continued to use it for our tests and development, Tango was deprecated by Google in the first half of 2018. Future work will focus on integrating Tango's successor, ARCore, into the system, once development of ARCore adds features to the platform such that it can act as a replacement for Tango, specifically, after the implementation of a substitute for Tango's "area learning" feature.)

A main consideration with using BLE beacons for localization is that received signal strength (RSS) values are often volatile. We found that BLE signals are extremely noisy, because they are easily attenuated by materials commonly found in a building [6]. Thus, without the use of complex probabilistic algorithms, fine localization using BLE is difficult. In [10], we found that, even with a relatively dense placement of one beacon every 3-5 meters placed out of sight just above ceiling tiles, beacons were only accurate enough by themselves to *approximate* a user's position (i.e., determine a coarse location). Yet, some users, especially BVI, require highly accurate (fine) positioning to avoid collisions with obstacles.

Thus, we looked into using Google Tango, which utilizes an Android-integrated RGB-D camera with capabilities of 6-degrees-of-freedom VIO (visual-inertial odometry) and feature-based indoor localization to allow for device pose (orientation and position) estimation in a 3D environment [7]. Tango makes use of Area Description Files (ADFs), which are feature maps of an indoor environment, and its onboard sensors to determine a device's position within an ADF down to a few centimeters [10]. However, due to limitations in the Tango SDK, it is a known issue that the loading of larger ADFs (usually with a size above 60 MB) can often trigger an internal timeout/crash within the Tango SDK. The areas mapped in our testbed ranged in size from 600 to 1000 sq. ft. and produced ADFs that ranged in size from 15 to 40 MB (depending on the features in the environment). Although suitable for our specific testbed, this is not practical for an area such as a large public transportation hub, where a single "floor" could be much larger and have many features. Thus, we require multiple ADFs to cover an expansive area. However, this requires that the appropriate ADF be selected automatically based on the user's current position. (ADFs are aligned with the area's floor plan/map as described in our previous work, via an affine transformation of the Tango-returned coordinates from the ADF's coordinate space to the map's coordinate space [10].)

To account for these respective strengths and weaknesses, we utilize a hybrid system that uses BLE beacons to figure out the approximate area that the user/device is located in. The area selected by BLE beacons is represented by a specific ADF that Google Tango uses to get the user/device's exact position.

Hybrid localization For the coarse localization component, the phone searches for all beacons it can detect in a one second interval. Of the beacons it detects in this interval, the three strongest beacons are taken and run against a prebuilt database of "fingerprints" for all of the areas in question. Each fingerprint represents a specific (x, y) position in the map coordinate space and consists of 1) the three strongest beacons (in terms of their RSS) that can be detected at that position and 2) the (general/coarse) area in which that (x, y) position lies. A simple matching algorithm then matches each real-time capture with the database entries and selects the general region associated with the matched fingerprint. Each coarse region is associated with a specific Tango ADF. When the BLE component successfully selects a new general region, Tango is restarted with the ADF of this region and locks onto this new region within a few seconds.

An important consideration is the switching of ADFs when navigating on the same floor. Since scans for BLE beacons are done at intervals of several seconds, the system may not respond fast enough when trying to switch between areas on the same floor. It is thus necessary to work around this delay and find a faster method for switching in these situations. We introduce an additional mechanism to compensate for this.

Boundary-based ADF switching for hybrid localization We can rely on map-based labeling of borders between ADFs. When the device approaches the

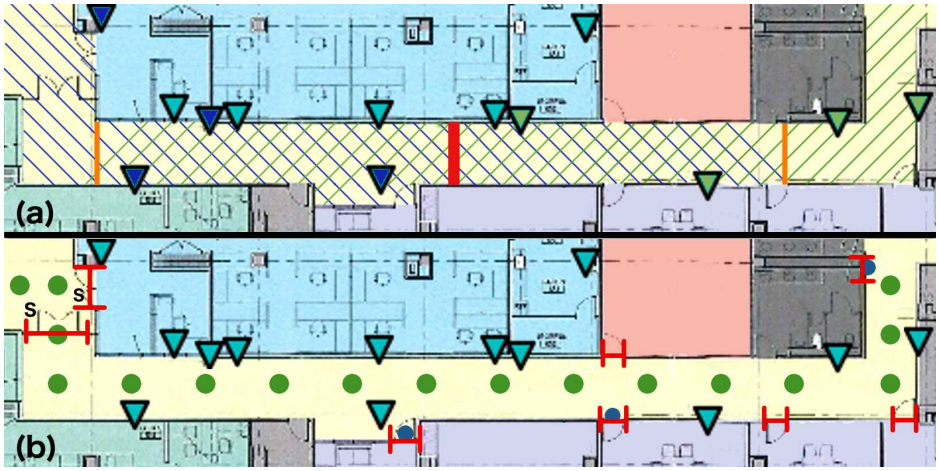

Fig. 1. Visualization of map annotations on the floor plan of a long corridor. *Top (a)*: ADF and beacon annotations. Diagonal lines represent coverages of respective ADFs (one blue and one green). Area where diagonal lines overlap represents overlap between both ADFs. Thick red line in center represents "primary" ADF border, where respective BLE/coarse localization areas meet each other. Thin orange lines to either side represent "secondary" ADF borders where overlaps between both ADFs end. Triangles represent installed beacons. Dark blue triangles are beacons representing area of blue-lined ADF; green triangles/beacons represent area of green-lined ADF. Lighter-blue triangles/beacons are irrelevant to this example (i.e., they represent another area above). Since these beacons are located on the other side of the wall from the hallway and Bluetooth signals are known to be attenuated by materials commonly found in a building [6], it is highly unlikely that that area will be selected. *Bottom (b)*: Environment and navigation annotations. Triangles represent all beacons. Green dots represent navigational nodes. Smaller, dark blue dots represent checkpoints (i.e., points of interest). Red "H"-like symbols represent all doorways annotated. Letter "S" next to each of both doors on far left represents annotation for a "security" door (i.e., one that requires a key card to open)

border between two ADFs (i.e., a "primary" border), the system can preemptively restart Tango with the approaching ADF so that when we do reach it, Tango will have already localized into the new ADF and can continue. We also make use of "secondary" borders that act as fallbacks in situations where primary border switching fails (e.g., when the device is close to the border with another ADF and BLE localization locks the device into the other ADF). Secondary borders make use of overlaps between adjacent ADFS such that Tango localization will still be successful even if we have selected the wrong ADF. Figure 1a visualizes an example of beacon placement and ADF switching logic.

ADFs are mapped strategically in the offline phase to optimize this mechanism. During an ADF switch, Tango is restarted; however, it may take up to several seconds to lock onto a position in the new ADF. During this "deadlock,"

positioning capabilities are suspended; thus, the interface does not update. It is of paramount importance then that this deadlock not interfere with normal navigation activities (for example, a turn cannot come up during the possible deadlock period). To account for this, during mapping, primary ADF borders are placed in areas where little or no navigational turns are available (e.g., in a long corridor). Via this strategic border placement, the system is provided a buffer during which it can lock on to its position while ensuring that the user will not need its guidance during this period (i.e., the user simply needs to continue walking forward during this period).

Via this modified hybrid method, we are able to ensure that our positioning system is not limited by the size of the area. We are also able to ensure that we can provide the highest accuracy possible (especially for BVI individuals who require it) and that a failure of the BLE/coarse location system can be handled in a timely manner. Furthermore, map-based ADF border marking can ensure that the system responds as quickly as possible to general location changes. In the end, this approach combines the coarse yet expansive location recognition of beacons with the fine yet limited-scale location recognition of Tango.

3.2 Body camera-based recognition and alerting of variable situational elements

As part of the modular implementation of ASSIST, people within the locality of the user can be detected via an on-person body camera. Currently, the system utilizes a YOLOv2 CNN model that is trained to detect people's heads. It was trained utilizing the Hollywood Heads dataset, consisting of 224,740 annotated movie frames including actors' heads [21]. This model, when running on a server (in our case, an Amazon EC2 p2.xlarge instance running an Nvidia K80 GPU) as opposed to a phone, has detection speeds of approximately 30 ms.

CNN-based head detection on the mobile device Our work originally attempted to perform both head detection and tracking on the Tango device. This was done by modifying a sample Tensorflow Android application which allowed for a Tiny YOLO model to be loaded and utilized to perform detection. The Tiny YOLO model, which aims to run very quickly at the cost of accuracy, consists of only 16 layers and utilizes a 416x416 input image size [15]. As such, the application read RGB images sized at 640x480 pixels, minimizing the need for image resizing. Our model was trained using a subset of 50,000 images from the Hollywood Heads dataset. When running on the Lenovo Phab 2 Pro, we achieved detection speeds of approximately 800ms. The advantages to performing detection on the mobile device include having access to tracking capabilities and the availability of depth/point cloud information provided by Tango. The sample application implemented tracking by executing the Lucas-Kanade algorithm. The points utilized for tracking were identified via a Harris filter, used both inside and outside of detected bounding boxes. The resulting application is capable of tracking up to 6 persons, while maintaining the detection speed of

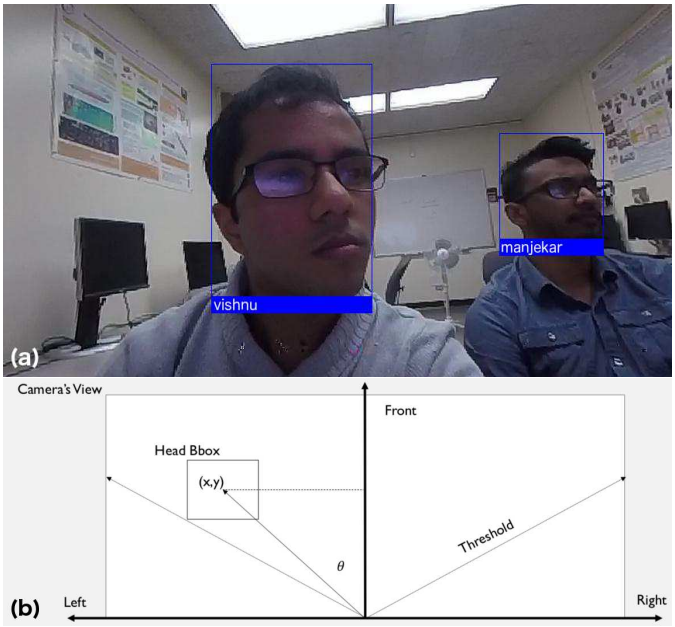

Fig. 2. *Top (a)*: Detection server-annotated image showing detected heads (from YOLOv2 CNN) and facial recognition outputs (from facial recognition model used). Image was taken using our test body camera (a GoPro Hero5 Session). *Bottom (b)*: Mobile camera view utilized for determining orientation of detected person with respect to the user's point of view

800ms mentioned previously. (These values were attained by setting an internal class confidence level of 0.01, an ultimate detection confidence of 0.25, and keeping a record of 200 frames of optical flow deltas.)

Utilizing Tango's point cloud generator, we can expand our 2D RGB detections to include specific 3D information. Via a series of frame transformations, we were able to evaluate a depth value for the centers of generated detection bounding boxes. By transforming a point in the 640x480 frame (in which the detected bounding boxes are placed) into a point in the 1920x1080 frame (used for the point cloud buffer), we can then grab relative depth information via a Tango method (which utilizes bilateral filtering on the most recently-saved point cloud). Furthermore, by dividing the RGB frame as shown in Figure 2b, we can relay a relative orientation of the detected person with respect to the user.

This application, though successful as a standalone implementation, proved to be a challenge when it came to merge it with the remainder of the system. Because the detection and tracking requires a great deal of computing power, finding an approach for scheduling these functionalities within the full application proved to be difficult and was subsequently abandoned.

CNN-based head detection on an external server For our current implementation, we run a YOLOv2 model on an external server dedicated to detection. An external server was chosen because of 1) the relative ease with which modular vision-based functionality could be implemented or removed and 2) the fact that the mobile application does not need to be continuously updated with every such server change provided that the interface between the mobile application/camera and the server remains the same.

As part of a proof-of-concept, we used a GoPro Hero5 Session as our camera. (This extra camera was selected, in part, to offload computations from the mobile device. Future work will focus on an optimized onboard implementation using the mobile device's camera.) The GoPro is connected via a WiFi dongle and accessed through a Python script. From here, the recorded images are compressed, encoded, and sent to an external server. On the server, the received package is decoded, decompressed, and passed through the neural network. The detection results are then sent to a dedicated navigation server, and ultimately to the phone, where the corresponding information can be relayed to the user.

The model sitting on the server is more extensive than the model that originally ran on the phone. It is a YOLOv2 model, consisting of 32 layers, also utilizing a 416x416 input image size. By running detection on a server, we have access to more processing resources, and can thus utilize larger images which can be resized. In order to maximize speed, however, the GoPro was set to read in images sized at 864x480 pixels. The model was trained for 15,000 iterations, utilizing the entire Hollywood Heads dataset. The increase in processing power will allow for more than 6 people to be both detected and tracked, while maintaining real-time speeds. Currently, we do not have tracking implemented on the server, but it can be done via a similar process to the mobile implementation.

The modular nature of the server-based detection system ensures that we can add or remove functionality. For example, we have tested the addition of a pre-trained facial recognition model[4] with which we can relay the identity of known, detected persons to the user. Figure 2a shows an example of face detection and recognition using a body camera.

3.3 Map-based semantic recognition and alerting of static environmental characteristics

Our system is heavily dependent on having pre-existing floor plans/maps of the area in question. Map-based pixel coordinates are used to mark the map on the interface and perform related calculations, such as distance measurements. We also label the map with navigational nodes and checkpoints. However, we can use these floor plans further to our advantage by explicitly annotating the map with various static characteristics of the environments represented on the map (e.g., the locations of doorways and elevators, as shown in Figure 1b). We can then use these annotations to alert the user of these static elements and incorporate them into navigation. This concept is further prominently used in our system in

[4] https://github.com/ageitgey/face_recognition

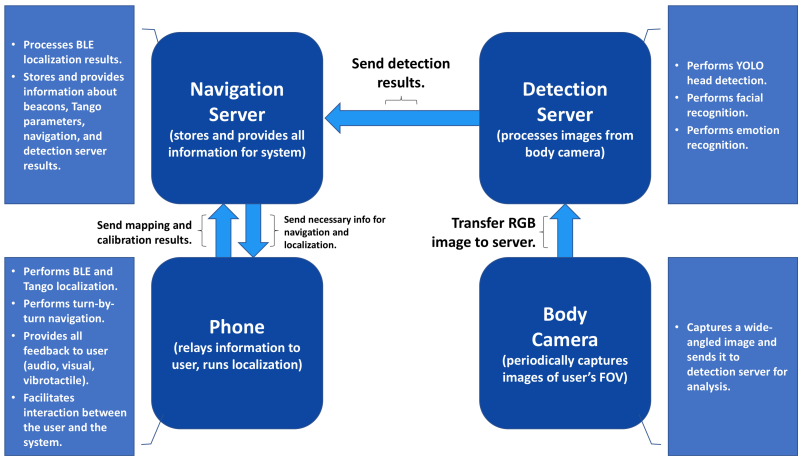

Fig. 3. System implementation pipeline

the recognition of elevators, including the identification of the specific elevator that the user has entered and subsequent start of navigation from the front of this same elevator door on the destination floor.

Through this map-based semantic recognition and alerting via a heavily annotated graph, our system can update a user in real-time about any pre-established environment characteristics. The operation is a simple and lightweight one that involves recognizing the environment (via the user's current position), searching for the appropriate information in the database, and communicating it to the user. The amount of information that will be communicated will depend on the preferences of the user. It should be noted that such an annotated map can be generated automatically as shown in our previous work [19].

4 Architecture and Interfaces

4.1 System architecture

The full system has been implemented using a client-server architecture (Figure 3) due to size, speed, and scaling concerns. Although many of these operations could theoretically be performed on a phone, doing so would not be ideal for a large facility, because the size and scale of these operations would consume processing power, battery life, and storage space if done on the phone itself. Thus, the total system contains two servers in addition to the phone and body camera. One server (called the "detection server") receives images from the body camera and processes them, detecting and recognizing faces, emotions, and other objects it has been trained to detect. This particular server is equipped with a graphics processing unit (GPU) and can thus perform these detections in mere milliseconds. The results of this processing of the images are sent to another

server. This second server (called the "navigation server") forms the system's "brain" and contains all information about all aspects of the system, including but not limited to information about the map, Tango ADFs, coordinate transformations, installed BLE beacon characteristics, and visual processing results from body camera images. Because of this, the phone is in constant contact with the server and exchanges the necessary information with it as needed.

4.2 Real-time response and fallback plans

This information exchange occurs very quickly. A full trip of the body camera data (from the moment the photo is taken to the announcing of the results) takes approximately 1.1 seconds. Information exchange between the navigation server and the phone takes only about 100-300 ms (depending on the size of the data exchanged). These times are more than sufficient to provide a near real-time reactive experience for the user. The navigation system and its characteristic responsiveness can be seen in the system demo video (link provided at the end).

The recovery procedure for this system should a failure occur (e.g., loss of Internet connection) is a main point of future work. In the case of a loss of Internet, Google Tango utilizes onboard SLAM (simultaneous localization and mapping) and VIO techniques to provide accurate positioning. Provided that ADFs have been pre-downloaded, Tango localization will not be interrupted. However, BLE beacon localization currently does require an Internet connection, because a cloud-based beacon database can be shared by all instances of the app. In this case, it may become necessary to fall back to onboard tracking using an IMU. However, the accuracy of the use of the IMU (and, similarly, BLE beacon readings themselves) may be improved via the use of an Extended Kalman Filter, thus creating another point of future work. An alternative approach may involve downloading the beacon database onto the user's phone when an Internet connection is readily available and thus locally performing BLE-based localization.

4.3 User-centric navigation experience

Our system employs a user-centric navigation interface by promoting configurability. Both the type (audio, visual, and vibrotactile) and level (information density and vibration intensity) of feedback can be adjusted to suit varying levels of disabilities. The system also utilizes a conversation-style voice engine, implemented using Google's DialogFlow, to enable voice input. This voice engine can be currently used to initiate navigation and will be expanded to allow for changing application settings and asking for additional route and situational information during navigation. In addition, the system also provides modular integration for smartwatches for additional forms of feedback. The design of the system is such that next to no real "training" is required for a user to safely use the application. Figure 4 shows some interface screens for the mobile application. The system demo video (link provided at the end) shows a BVI individual using the voice engine to start feedback, the audio feedback during navigation, and specialized visual and vibrotactile feedback for individuals with ASD.

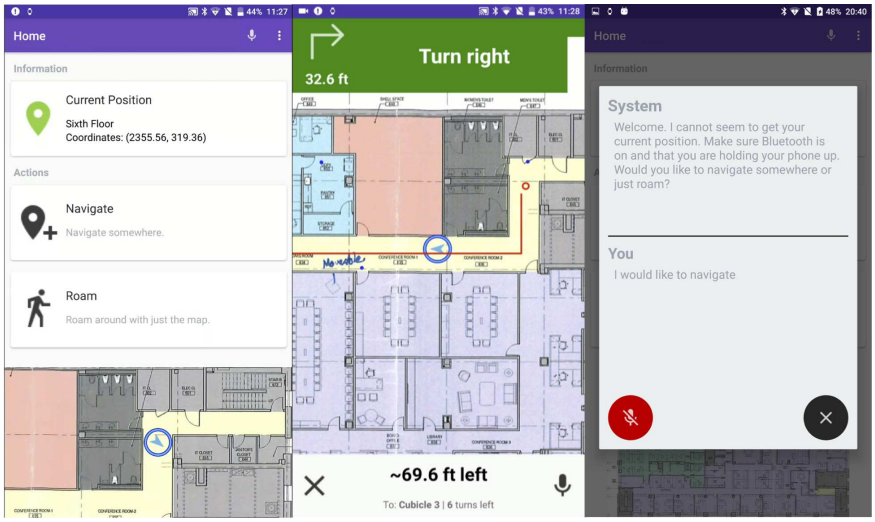

Fig. 4. Interface screens for mobile application. *From left to right*: (a) home screen, (b) navigation interface, and (c) voice engine interface

5 User Testing

Our previous work compared navigation using solely BLE beacons with navigation using a hybrid BLE-Tango system [10]. Human subjects tests involved numerical evaluations of runs to record statistics such as total interventions, trip duration, and total bumps. In that study, we found that when subjects used hybrid navigation, they required significantly fewer interventions and less assistance when compared to their runs with BLE navigation. For comparison purposes, each path covered a single area, such as a corridor or a group of cubicles.

In this study, our goal was to evaluate the high-level usability of the entire system and to allow users to travel across areas and floors during a single test, thus requiring them to peruse doors and/or elevators. To this end, we performed human subjects tests on both BVI individuals and those with ASD. For these experiments, we used the Lenovo Phab 2 Pro, a Tango device. These tests were performed at Lighthouse Guild, a vision rehabilitation center in New York City, and evaluated the experiences of users when using the app and assessing subjects' impressions of the app in guiding them between points safely and accurately. The system demo video (link provided at the end) shows some runs.

5.1 Procedures

We conducted two separate tests (one each for BVI and ASD) of the system and mobile application, where we asked subjects to traverse pre-selected paths using the guidance provided by the system. To evaluate our subjects' experiences, we administered both a pre-experiment survey (which asked for demographics) and

a post-experiment survey (which assessed subjects' impressions of the application and its various components). The BVI test had a convenience sample of 11 individuals (ranging from low vision to totally blind) use the application to navigate themselves on three separate paths that brought them across floors. There were **8** participants 55 years old or older, **1** participant 45-54 years old, **1** participant 35-44 years old, and **1** participant 18-24 years old; there were **7** males and **4** females. The ASD test had a small convenience sample of five **male** individuals with medium-low- to high-functioning forms of ASD try two to three separate paths on a single floor (depending on the focus of the subject). There were **2** participants aged 25-34 years and **3** participants aged 18-24 years.

5.2 Results

According to the surveys, subjects generally had a very favorable impression of the system. (Results are reported as **means**.) For the ASD tests, all five subjects agreed to strongly agreed that using the app was easy (**4.6/5**), that they felt safe while using the app (**4.6/5**), and that they could easily reach a destination with the app (**4.4/5**). The subjects also found the app helpful to extremely helpful (**4.6/5**) and were moderately to very satisfied (**3.4/5**) with it. Those individuals who used the smartwatch to receive supplementary vibrotactile cues found them moderately helpful (**3.75/5**).

Similar results were recorded for our tests with BVI individuals. The subjects agreed to strongly agreed (**4.5/5**) that using the app was easy, agreed (**4.2/5**) that they felt safe while using it, agreed (**4.3/5**) that they could easily reach a destination using it, and generally found the app helpful (**4.3/5**). The subjects almost universally agreed that the voice feedback provided by the app was extremely helpful (**4.8/5**). We also tested other features with our BVI subjects. With regards to the voice assistant which allowed them to initiate navigation, almost all of our subjects who used it found it extremely helpful (**4.9/5**). The app would also issue guidance on corrective turns if the user was not facing the correct direction; users found them moderately to extremely helpful (**4.6/5**).

5.3 Discussion of Results

The app was generally very well-received by all subjects. BVI subjects approved of the voice feedback provided by the app as well as the simplicity of the voice assistant. ASD subjects expressed favorable opinions on the visual and vibrotactile cues provided by the app and also liked the addition of a smartwatch to keep their attention. However, we noted that BVI and ASD subjects each gave very different feedback in what they would like to see in such a system. Feedback gathered from our ASD tests centered mostly on optimizing the interface and feedback provided by the app for ASD individuals (e.g., simpler instructions for medium-low functioning ASD individuals). In contrast, feedback gathered from our BVI tests mostly centered around fine-tuning and then augmenting the experience provided by the app (e.g., expansion of the voice assistant and possible use of smart glasses with built-in cameras). This difference in feedback highlights the

importance of personalizing the navigation experience to each disability. Thus, offering users the choice to turn on or off certain features and pre-establishing some of the assistance to be given based on the user's disability can make the navigation experience much more comfortable for the user.

6 Conclusion

Through our work, we have created and tested a system that would not only guide a person indoors with high accuracy but would also augment that same user's understanding of his/her environment. This system consists of highly accurate, BLE-Tango hybrid navigation coupled with a body camera for the alerting of high-priority situational elements and a pre-built database of map annotations for the alerting of high-priority environmental characteristics. Our system provides a complete picture of the user's surroundings in a user-centric way by incorporating varying modes of feedback for different disabilities.

Evaluations have shown that such a system is welcomed by both BVI and ASD individuals. These tests have also opened many avenues for future work with which we could further improve and optimize this system. Additional evaluations are also required for the testing of the visual body camera-based alerting system which would play a pivotal role, especially for BVI users. However, in the end, our work has established a solid base from which we can expand our current system into a more full-fledged assistive application that can both effectively navigate a person and augment their understanding and awareness of their environment.

System Demo

A system demo can be viewed here: https://youtu.be/Hq1EYS9Jncg.

Acknowledgments

This research was supported by the U.S. Department of Homeland Security (DHS) Science & Technology (S&T) Directorate, Office of University Programs, Summer Research Team Program for Minority Serving Institutions, administered by the Oak Ridge Institute for Science and Education (ORISE) under DOE contract #DE-AC05- 06OR23100 and #DE-SC0014664. This work is also supported by the U.S. National Science Foundation (NSF) through Awards #EFRI-1137172, #CBET-1160046, and #CNS-1737533; the VentureWell (formerly NCIIA) Course and Development Program (Award #10087-12); a Bentley-CUNY Collaborative Research Agreement 2017-2020; and NYSID via the CREATE (Cultivating Resources for Employment with Assistive Technology) Program. We would like to thank the staff at Goodwill NY/NJ for their invaluable feedback and for recruiting subjects for our tests with autistic individuals. We would especially like to thank all of our subjects for their participation and cooperation as well as for providing extremely helpful feedback in improving our system.

References

1. Ahmetovic, D., Gleason, C., Ruan, C., Kitani, K., Takagi, H., Asakawa, C.: Navcog: a navigational cognitive assistant for the blind. Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '16 (2016)
2. Bohonos, S., Lee, A., Malik, A., Thai, C., Manduchi, R.: Universal real-time navigational assistance (urna): an urban bluetooth beacon for the blind. Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments - HealthNet '07 (2007)
3. Chumkamon, S., Tuvaphanthaphiphat, P., Keeratiwintakorn, P.: A blind navigation system using rfid for indoor environments. 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (2008)
4. Furnari, A., Battiato, S., Farinella, G.M.: Personal-location-based temporal segmentation of egocentric videos for lifelogging applications. Journal of Visual Communication and Image Representation (52), 1–12 (2018)
5. Hub, A., Diepstraten, J., Ertl, T.: Design and development of an indoor navigation and object identification system for the blind. Assets '04 Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility (2004)
6. Kara, A., Bertoni, H.: Blockage/shadowing and polarization measurements at 2.45 ghz for interference evaluation between bluetooth and ieee 802.11 wlan. IEEE Antennas and Propagation Society International Symposium. 2001 (2001). https://doi.org/10.1109/aps.2001.960112
7. Li, B., Munoz, P., Rong, X., Xiao, J., Tian, Y., Arditi, A.: Isana: Wearable context-aware indoor assistive navigation with obstacle avoidance for the blind. In: Lecture Notes in Computer Science, vol. 9914. Springer (2016)
8. Lind, S., Williams, D., Raber, J., Peel, A., Bowler, D.: Spatial navigation impairments among intellectually high-functioning adults with autism spectrum disorder: Exploring relations with theory of mind, episodic memory, and episodic future thinking. Journal of Abnormal Psychology **122**(4), 1189–1199 (2013)
9. Mulloni, A., Wagner, D., Barakonyi, I., Schmalstieg, D.: Indoor positioning and navigation with camera phones. IEEE Pervasive Computing **8**(2), 22–31 (2009)
10. Nair, V., Tsangouri, C., Xiao, B., Olmschenk, G., Seiple, W.H., Zhu, Z.: A hybrid indoor positioning system for the blind and visually impaired using bluetooth and google tango. Journal on Technology & Persons with Disabilities **6**, 61–81 (2018)
11. Nie, M., Ren, J., Li, Z., Niu, J., Qiu, Y., Zhu, Y., Tong, S.: Soundview: An auditory guidance system based on environment understanding for the visually impaired people. 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2009)
12. Ortis, A., Farinella, G.M., D'Amico, V., Addesso, L., Torrisi, G., Battiato, S.: Organizing egocentric videos of daily living activities. Pattern Recognition (72), 207–218 (2017)
13. Ozdenizci, B., Ok, K., Coskun, V., Aydin, M.N.: Development of an indoor navigation system using nfc technology. 2011 Fourth International Conference on Information and Computing (2011)
14. Peng, E., Peursum, P., Li, L., Venkatesh, S.: A smartphone-based obstacle sensor for the visually impaired. Ubiquitous Intelligence and Computing **6406**, 590–604 (2010)

15. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)
16. Ruiz, A.R.J., Granja, F.S., Honorato, J.C.P., Rosas, J.I.G.: Accurate pedestrian indoor navigation by tightly coupling foot-mounted imu and rfid measurements. IEEE Transactions on Instrumentation and Measurement **61**(1), 178–189 (2012)
17. Spera, E., Furnari, A., Battiato, S., Farinella, G.M.: Egocentric shopping cart localization. International Conference on Pattern Recognition (ICPR) (2018)
18. Strumillo, P.: Electronic interfaces aiding the visually impaired in environmental access, mobility and navigation. 3rd International Conference on Human System Interaction (2010)
19. Tang, H., Tsering, N., Hu, F., Zhu, Z.: Automatic pre-journey indoor map generation using autocad floor plan. Journal on Technology & Persons with Disabilities **4** (2016)
20. Tapu, R., Mocanu, B., Bursuc, A., Zaharia, T.: A smartphone-based obstacle detection and classification system for assisting visually impaired people. The IEEE International Conference on Computer Vision (ICCV) Workshops pp. 444–451 (2013)
21. Vu, T., Osokin, A., Laptev, I.: Context-aware cnns for person head detection. The IEEE International Conference on Computer Vision (2015)
22. World Health Organization: Vision impairment and blindness (2017), http://www.who.int/mediacentre/factsheets/fs282/en/