IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 27, 2017, accepted October 9, 2017, date of publication October 25, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2766579

Visual and Infrared Sensor Data-Based Obstacle Detection for the Visually Impaired Using the Google Project Tango Tablet Development Kit and the Unity Engine

RABIA JAFRI[1], RODRIGO LOUZADA CAMPOS[2], SYED ABID ALI[3], AND HAMID R. ARABNIA[4]
[1]Department of Information Technology, King Saud University, Riyadh 12371, Saudi Arabia
[2]Department of Computation and Civil Construction, Federal Technological Center of Minas Gerais, Timoteo 35180-008, Brazil
[3]Araware LLC, Wilmington, DE 49418 USA
[4]Department of Computer Science, University of Georgia, Athens, GA 30602 USA

Corresponding author: Rabia Jafri (rabia.ksu@gmail.com)

This research project was supported by a grant from the ''Research Center of the Female Scientific and Medical Colleges'', Deanship of Scientific Research, King Saud University.

ABSTRACT A novel visual and infrared sensor data-based system to assist visually impaired users in detecting obstacles in their path while independently navigating indoors is presented. The system has been developed for the recently introduced Google Project Tango Tablet Development Kit equipped with a powerful graphics processor and several sensors which allow it to track its motion and orientation in 3-D space in real-time. It exploits the inbuilt functionalities of the Unity engine in the Tango SDK to create a 3-D reconstruction of the surrounding environment, then associates a Unity collider component with the user and utilizes it to determine his interaction with the reconstructed mesh in order to detect obstacles. The user is warned about any detected obstacles via audio alerts. An extensive empirical evaluation of the obstacle detection component has yielded favorable results, thus, confirming the potential of this system for future development work.

INDEX TERMS Visually impaired, blind, obstacle detection, obstacle avoidance, navigation, Unity, Project Tango, assistive technologies, multimodal sensors.

I. INTRODUCTION

One of the major challenges faced by visually impaired (VI) individuals while navigating independently in indoors environments is detecting and avoiding obstacles or drop-offs in their path - the inability to do so causes them emotional distress, undermines their autonomy, and exposes them to injury [1]–[3]. For instance, a recent survey by the Royal Institute for the Blind found that 95% of the respondents had experienced a collision outside of the home in the last three months leading to physical injury and loss of confidence [4]. Though white canes, guide dogs and human caregivers are usually utilized to assist with this task, each of these solutions has its own limitations: The white cane is highly conspicuous, has limited reach dependent on its length, cannot sense obstacles above the waist level and requires contact with obstacles in order to sense them (which may not be practical - e.g., for detecting people or fragile objects) [1], [5]. Moreover, some VI people, suffering from multiple disabilities, may not have the physical strength or motor skills to use a cane effectively [15]. Guide dogs are expensive, require extensive training, have a useful life of about five years and require appropriate care which VI individuals, especially elderly ones, may find difficult to provide [5]. A sighted caregiver may be available to help out some of the time but it is not possible or desirable for such a human aide to be available at all times (indeed, according to a recent report, 26% of blind adults in the United States live alone [6]) [7].

This delineates a compelling need to develop technological solutions to assist VI individuals in detecting and avoiding obstacles in their path. In recent years, infrared-enabled depth sensor-based and visual sensor-based systems on mobile devices have emerged as some of the most promising solutions for addressing this issue [8]–[12]. However, both these kinds of solutions incur high computation costs, especially when dealing with multiple obstacles. Since mobile platforms are currently limited in terms of both the processing power required to achieve real-time performance for executing computationally intensive code and the battery life needed to support intense processing for extended periods of time, most of these solutions use the mobile device simply as a frontend for receiving input and delivering output while all the actual processing of the data is done on a remote server equipped with powerful processors. This, in turn, introduces additional constraints and challenges: the user's device has to be connected to a remote system, the inevitable communication overhead may negatively affect the real-time performance and the absence of or failure of a network connection would render the system useless (network connectivity is an even more pertinent issue in developing countries where 90% of the VI population resides [13]) [14]. Furthermore, prototype devices attempting to combine camera images and depth data to improve the accuracy of the detection tend to have difficulty in precisely synchronizing their various components and sensors with each other (though recently introduced commercial sensor technologies such as Kinect [41] have integrated RGB-D sensors but these are not designed as mobile or wearable devices – please refer to the related work section for further discussion about this). Consequently, currently such systems fall short in terms of accurately localizing the user and providing real-time feedback about obstacles in his path [15].

The Project Tango Tablet Development Kit [16], recently introduced by Google, is an Android device, equipped with a powerful processor (NVIDIA Tegra K1 with 192 CUDA cores) and various sensors (motion tracking camera, 3D depth sensor, accelerometer, ambient light sensor, barometer, compass, GPS, gyroscope), which allow it not only to track its own movement and orientation through 3D space in real time using computer vision techniques but also enable it to remember areas that it has travelled through and localize the user within those areas to up to an accuracy of a few centimeters. Its integrated infrared based depth sensors also allow it to measure the distance from the device to objects in the real world providing depth data about the objects in the form of point clouds [15], [17], [18]. Moreover, being compact, lightweight, relatively discreet and affordable renders it aesthetically appealing, socially acceptable and accessible for VI users [14].

We have, therefore, developed an application for the Project Tango tablet to assist VI users in detecting obstacles in their path during navigation in an indoors environment. The system is focused on micro-navigation in previously unmapped surroundings. It exploits the inbuilt functionalities of the Unity engine in the Project Tango SDK to create a 3D reconstruction of the surrounding environment to detect obstacles in real-time and also provides the user with audio alerts about any detected obstacles [18]. Our aim is to exploit the main strengths of the Tango tablet – its range of custom sensors and extensive inbuilt software which enable and facilitate 3D reconstruction of and interaction with the surrounding environment and its capacity for performing computationally expensive operations in real-time on the device itself without the need to connect to an external server or rapidly draining the battery. Since the Tango platform is swiftly being expanded and also being integrated into other mobile devices (e.g., recently released smartphones such as Lenovo Phab 2 Pro [19] and Asus Zenfone AR [20]), this further validates our selection of this platform in view of future development work. Furthermore, we are utilizing a widely used cross-platform game industry software, Unity [21], which, to the best of our knowledge, has not been employed for building assistive navigation applications for VI individuals so far. We aim to demonstrate that the functionalities of a popular game engine that developers may already be familiar with can readily be exploited to build assistive applications for disabled individuals in real-world scenarios given that this engine is now available on a mobile device with enough computational power and embedded sensors to make this feasible.

The main contributions of our research are as follows: The development of a novel real-time assistive stand-alone application for VI users on a cutting-edge aesthetically appealing mobile device equipped with one of the most powerful processors available to date on a consumer-level mobile platform [22], which allows them to detect obstacles independently in possibly unfamiliar indoor surroundings; the innovative use of the functionalities of the inbuilt Unity engine on the Tango tablet to design an obstacle detection mechanism in a real-world context specifically targeted towards individuals with visual impairments; the extensive empirical evaluation of the obstacle detection system yielding favorable results and, thus, confirming the potential of this application in particular and the platform in general for further development work in this area.

The rest of the paper is organized as follows: Section II provides an overview of existing assistive systems for the VI utilizing visual and infrared sensors for obstacle detection and delineates their strengths and limitations. Section III describes the proposed application explaining the system setup and data acquisition, the obstacle detection process and the feedback mechanism. Section IV expounds upon the various tests for empirically evaluating the system. Section V reports and discusses the results of the tests. Section VI highlights some directions for future work and section VII concludes the paper.

II. RELATED WORK

In recent years, several infrared-enabled depth sensor-based and visual sensor-based systems have been developed to assist the VI in detecting and avoiding obstacles. Visual sensor-based systems employ stereo [23]–[27] or monocular cameras [28]–[34] to acquire image data of the surrounding environment and analyze it to estimate obstacle positions. These solutions are generally cost-effective, accessible, require little or no infrastructure, are typically wearable, and can usually be installed or embedded into existing mobile computing devices. However, their performance deteriorates rapidly in uncontrolled real-world environments due to imaging factors such as motion blur, image resolution, video noise, etc., as well as changes in conditions such as illumination, orientation and scale. Other limitations include vulnerability to occlusion problems and high computational cost [7]. Also, stereo cameras are relatively expensive and require precise calibration [35]. Infrared-based obstacle detection systems for the VI typically employ infrared tags [36], infrared beacons [36]–[38] and general thermal input from the environment [39], [40]. These solutions offer the advantages of being affordable, discreet and unobtrusive (since infrared light is invisible to the naked eye), providing directional information (if beacons or tags are placed on obstacles) and being able to operate in the dark. However, such systems can detect objects only within a certain range and their performance may be negatively impacted by interference from other infrared sources, such as sunlight or fluorescent light [15]. Furthermore, most of these systems require retrofitting the environment with tags and beacons, which is costly, time consuming and limits the use of the system only to previously fitted areas. Nevertheless, some newly emerging sensor technologies (e.g., Microsoft's Kinect [41], Occipital's Structure Sensor [42], and, most recently, Google's Project Tango Tablet Development Kit [16]) are making it possible to exploit infrared light to extract 3D information about the environment without the need to install any equipment in the surroundings [15].

Recent development work on obstacle detection has specially focused on Kinect, either utilizing the data from its depth sensor alone ([43]–[46]) or from both its RGB and depth sensors ([47], [48]): Khan *et al.* [49] divide the depth image of the scene into 5x3 regions, calculate a depth metric for each region and instruct the user to go in the direction with the smallest probability of an obstacle. The Kinect sensor is mounted on the user's waist, the processing is done on a laptop computer and directions are generated via text-to-speech and conveyed to the user by Bluetooth headphones. Filipe *et al.* [44] extract six vertical line profiles at predefined locations from depth images acquired from a chest-mounted Kinect sensor; a feedforward neural network with backpropagation is then employed to classify each line profile and the user is informed about the location of any obstacles found in terms of right, left and center. Huang *et al.* [46] use the Least Squares method to approximate ground curves and to determine the ground height threshold from the depth image. Descending stairs are detected by finding possible stair edge points based on the threshold and transforming them into an edge line by applying the Hough Transform. The ground plane is then removed, a region growing approach is used to label different objects, the labelled objects are analyzed to determine if any of them are ascending stairs and the user is informed about how far he is from any obstacles in his path; if stairs are detected, the direction and distance to the stairs is conveyed. The Kinect sensor is mounted on the user's helmet, chest or waist, the processing is done on a laptop computer, and feedback to the user is provided via the Text-To-Speech (TTS) software on the laptop. Liu *et al.* [50] apply a multiscale voxel plane segmentation method on the 3D point cloud data for the current frame to extract planar structures. The ground plane is then removed and an area growing algorithm is exploited to segment all the non-ground regions into independent clouds which are regarded as generalized obstacles. The area in front of the user is then divided into three cuboids (left, center and right) and a multi-level voice feedback strategy is employed to alert the user to the presence of obstacles and to provide appropriate directions to avoid them. The Kinect sensor is mounted on the user's chest, the processing is done on a mini PC processor carried by the user in a backpack and voice feedback is provided via a Bluetooth headset. Brock and Kristensson [45] down-sample the depth data and split it into isolated structures, representing obstacles, at different depth levels using the marching cubes algorithm. The 3D location of the obstacle is sonified so that the horizontal position, vertical position and volume are encoded by the panning position, pitch and volume of the sound, respectively. The Kinect sensor is hung from the user's neck, the processing is done on a laptop also hung from the user's neck and the sonification is provided via headphones. Bernabei *et al.* [51] detect the floor based on the depth data acquired from a waist-mounted Kinect sensor and then simultaneously analyze the volume in front of the user to determine if there is sufficient room for him to move without colliding with an obstacle and the output from a wearable accelerometer to establish if the user is walking and if so, at what speed. The Kinect sensor is connected to a smartphone which does the audio processing and provides audio feedback to the user consisting of speech-based instructions for obstacle avoidance and sonification to convey the obstacle's location and distance from the user.

Some solutions have opted for tactile - instead of audio - feedback: Zöllner *et al.* [43] mount the Kinect sensor on the user's head and put vibe boards on the left, right and center of the user's waist to indicate the direction in which an obstacle was detected. A depth window is moved from left to right (respectively near to far) over the depth histogram of the current frame and stops, if the pixel area of that depth window exceeds a certain threshold area. The average depth value of the current depth window is then mapped to the pulse of the appropriate vibe board. Mann *et al.* [52] mount an array of six vibrating actuators inside a helmet. The depth sensing region of a Kinect sensor mounted on top of the helmet is divided into six zones; a distance map of the depth image is calculated and the vibration of each actuator is made inversely proportional to the distance of its corresponding zone to create the sensation of objects in the visual field pressing against the forehead before collision occurs – the sensation increases in strength as collision becomes more imminent.

All the above systems detect obstacles in general. However, a few other Kinect-based solutions focus on detecting specific obstacles such as staircases and traffic [53]–[55]. Data from Kinect has also been combined with other modalities such as ultrasonic and sonar for obstacle detection [56], [57].

Since the Kinect sensor module is not designed as a wearable or handheld device, these systems employ makeshift methods for affixing it to various locations on the user's body (e.g., head, chest or waist) resulting in awkward bulky contraptions that are unappealing from an aesthetic perspective and, thus, unlikely to be practically adopted by VI individuals for whom, as confirmed by several studies [58], [59], the cosmetic acceptability of an assistive device is even more important than its utility. Moreover, the processing of the data obtained from the Kinect sensor has to be done on an external server, which introduces the network connectivity issues and real-time performance challenges mentioned previously in the introduction.

The Project Tango tablet appears to have a distinct advantage over Kinect in that it is an aesthetically appealing, handheld, mobile device equipped with a powerful processor enabling it to execute computationally intensive code in real-time without the need to connect to a backend server. Moreover, it has several additional embedded sensors and in-built functionalities, which can be utilized for extending and improving the obstacle detection application in the future. Since the tablet has just recently been released in the market and has obvious potential for meeting real-time navigation requirements, there is a compelling need to initiate work to utilize its capabilities for developing navigational aids for the VI. These considerations have motivated us to use this platform for our development work. It should be noted that a few preliminary applications for the Tango tablet have already been proposed for this purpose [15]: The system presented by Anderson [60] collects depth information about the environment, saves it in a chunk-based voxel representation, and generates 3D audio for sonification which is relayed to the VI user via headphones to alert him to the presence of obstacles. Wang *et al.* [61] cluster depth readings of the immediate physical space around the users into different sectors and then analyze the relative and absolute depth of different sectors to establish thresholds to differentiate among obstacles, walls and corners, and ascending and descending staircases. Users are given navigation directions and information about objects using Android's TTS feature. Another indoor assistive navigation system developed for the Tango tablet by Li *et al.* [62] has an obstacle detection component that de-noises the point cloud data, de-skews it to align it with the horizontal floor plane, and projects it in both the horizontal and vertical directions. The vertical projection detects in-front and head-height obstacles locally while the horizontal projection is used to update obstacle information in a global 2D grid map maintained by the application. The Android TTS module is used to convey the obstacle detection results and navigation directions. A beeping alert sound is also issued, where the beep frequency signals the distance to the obstacle. However, none of these systems utilizes the Tango Unity SDK. Also, the first two applications need further development and are yet to be tested with the target users. Li *et al.* system [62] is reported to have been tested with blindfolded and blind subjects. However, details about the number of subjects and the data collection procedure and an in-depth analysis of the users' performance have not been provided.

It should be noted that one of the members of our research team is directing a related project for developing an obstacle detection and avoidance application for the VI on the Tango platform [15]. The approach for detecting obstacles differs significantly from the one presented in this paper since it relies on directly clustering and segmenting the point cloud data within a 2 m range of the depth sensor to isolate potential obstacles; feedback to the user consists of audio alerts for detected obstacles and speech and beep pattern-based directions for obstacle avoidance. A user-centered design methodology is being adopted for this project which is still in its early phases of development. We aim to eventually conduct a comparative evaluation of our system with this one to study any differences in terms of speed, accuracy and general usability.

III. SYSTEM DESIGN AND ARCHITECTURE

The application has been developed for the Google Project Tango Tablet Development Kit, an Android tablet equipped with a 2.3 GHz NVIDIA Tegra K1 processor with 192 CUDA cores running on the Android 4.4 KitKat operating system. It has 4 GB RAM and 128 GB flash memory (expandable via microSD). This device also has a depth-sensing array (an infrared projector, 4 MP 2μm rear-facing RGB/IR camera and 180° field of view fisheye rear-facing camera), several other sensors (a 120° front-facing camera, accelerometer, ambient light, barometer, compass, GPS, gyroscope), accurate sensor timestamping, and a software stack that enables application developers to use motion tracking, area learning and depth sensing [17], [63].

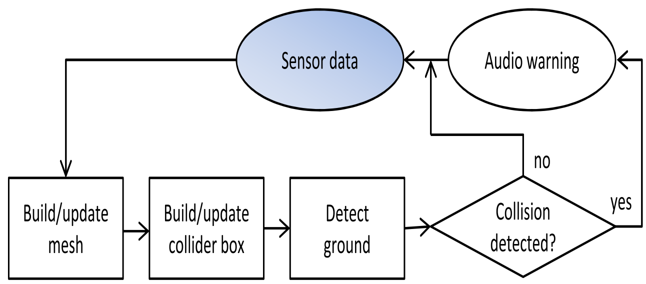

FIGURE 1. System overview.

An overview of the system is shown in Fig. 1. The application utilizes the Tango Unity SDK [64] to process the depth and motion tracking data obtained via the various sensors of the tablet to create and update a 3D reconstruction of the real-world environment in the form of a mesh as the user is walking. A rectangular box is created and updated around the user's position in this reconstruction. Some steps are taken to detect the ground to avoid false obstacle detection warnings being triggered by the box's coming in contact with the ground. If the box collides with any solid surface in the 3D reconstruction, an audio warning is relayed to the user via bone conduction headphones. The details of the system setup and data acquisition, the methodology used for detecting obstacles and the feedback given to the users are provided in the subsections below.

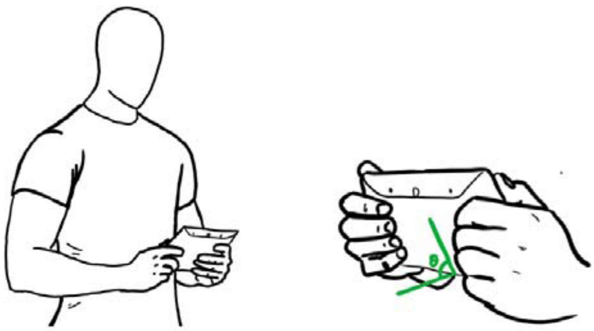

FIGURE 2. How the user should hold the tablet (in landscape orientation and slightly tilted).

A. SYSTEM SETUP AND DATA ACQUISITION

The user holds the tablet in his hands roughly at waist level with the screen facing towards him (Fig. 2). Since the Tango tablet has been so designed that the cameras on the rear top edge face directly in the front direction when the tablet is held in landscape orientation while slightly tilted (at an angle of about 60° to the horizontal plane), our system also requires the user to hold the tablet in this orientation and at this angle.

Under these constraints, the camera's field of view of 180° appears to be sufficient for covering the walking area in front of the user within which obstacles need to be detected. For a hands-free option, the tablet may also be mounted on the user's waist; however, it still needs to be tilted at a 60° angle. We are currently designing a 3D-printed wearable holder for the tablet which will lock it into the desired position.

Depth data is obtained via the tablet's integrated depth sensor using structured light in the form of point clouds [65] while location and orientation information is acquired based on the input from the various in-built cameras and sensors using visual-inertial odometry [66]. Tango offers APIs in C and Java and an SDK in Unity for accessing its depth perception, motion tracking and area learning services. For our project, the Tango Unity SDK [64] and the Android SDK (Android 4.2 'Jelly Bean' (APK level 17)) [67] are utilized for connecting to the Tango services for processing the depth and motion tracking data and for developing the application.

Since all the data is acquired via the tablet's built-in sensors and all the computations are carried out on the device itself, there are no other external hardware components required for the data input and processing.

Audio feedback about detected obstacles is provided to the user via Bluetooth bone conduction headphones connected wirelessly to the application.

B. OBSTACLE DETECTION APPROACH

The Tango Unity SDK [64] is used to acquire a 3D reconstruction of the surrounding environment in the form of a mesh which is created and updated in real-time [18]. Any empty regions in the resulting reconstruction represent empty spaces while the mesh represents solid surfaces and objects. So, to detect obstacles, the system simply needs to be able to determine whether the user is about to come in contact with the mesh as he moves through the environment.

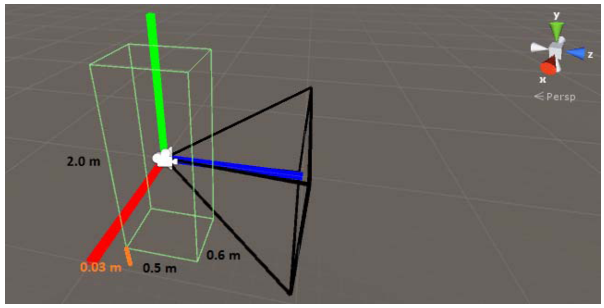

FIGURE 3. Collider box dimensions and distance from the ground.

Since Unity allows the creation of *collider* components [68] around an object and provides functions to detect when these components come in contact with any part of the generated mesh, we decided to utilize this useful feature to formulate a novel approach for detecting obstacles in the user's surroundings: A rectangular collider box is created around the device (which represents the user's position in the reconstructed environment). The dimensions of the box are chosen so that it would encompass an average user's body including allowances for the personal space required around him for unhindered head and limb movement during walking. Hence, the width of the box is set to 0.5 m [69], the height to 2 m [70], and the length to 0.6 m (Fig. 3). It should be noted that the width, height and length of the box are set along the x, y and z-axis, respectively, in Tango's device coordinate system. As the user walks, the mesh and the collider box are continuously updated. If at any time, the box comes in contact with the mesh, an audio-based obstacle warning is generated and relayed to the user.

Since the ground is a solid surface, a mesh will be created for it in the 3D reconstruction and thus, it, too, will be considered an obstacle by the system. To avoid triggering obstacle warnings for the ground, the following steps have been taken: When the application is started, Unity's raycast feature [71] is used to cast a ray straight down (along the y-axis in the device coordinate system). It is assumed that there is nothing between the tablet and the ground so that when the ray strikes the mesh, the distance between the ray's origin (the tablet) and the point it strikes the ground provides the distance between the tablet and the ground. To avoid having the collider box come in contact with the ground and thus, triggering an obstacle warning, the collider box is locked at a distance of 3 cm above the ground (Fig. 3). The raycasting to detect the ground is done every time the mesh is updated.

The rotation of the collider box is locked to avoid it being affected by slight changes in the user's posture (e.g., in the absence of rotation locking, if the user bends a bit forward,

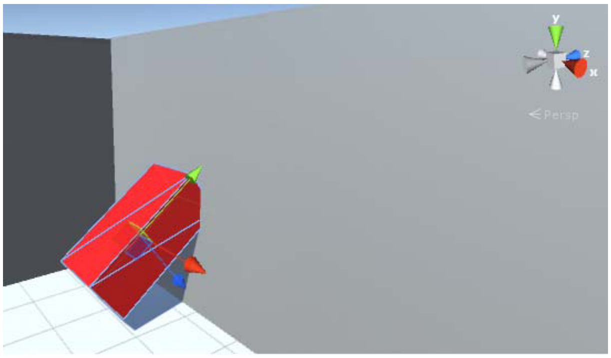

FIGURE 4. Effect of user's bending slightly forward if the collider box's rotation is not locked.

tilting the tablet down a little, the box will also tilt forward, which may cause it to collide with the floor, a wall or a nearby object triggering a false obstacle warning (see Fig. 4)).

Based on some initial testing wherein the resolution of the individual grid cells in the internal state for the mesh generation was varied from 0.01 m to 0.1 m and the resulting mesh was visually inspected, an optimal resolution of 0.05 m was chosen (decreasing the resolution further results in small and narrow objects not being properly meshed while increasing it considerably slows down the mesh generation process). The space clearing option is also selected to attempt to remove objects from the mesh that are no longer there when the mesh is updated. The Tango UX framework is utilized to handle general tasks related to user interaction with the application such as displaying the connection screen when the application is initializing and warning the user to hold the device steady as well as issuing notifications to the user when exceptions occur.

C. FEEDBACK TO USERS

Currently, only rudimentary audio feedback is being provided to the user: a voice message saying "Warning: obstacle detected" is relayed to him via wireless bone conduction headphones every time an obstacle is detected (bone conduction headphones were selected since the sound these produce is audible only to the user and not to those around him, thereby offering him a discreet means for receiving the system's output; moreover, these do not block his auditory channels). However, the feedback mechanism would be enhanced in the future to provide more details such as the approximate distance of the obstacle from the user (e.g., by using non-speech-based audio signals such as short beep sounds: the closer the obstacle, the higher the frequency of the beeps) and navigation directions to avoid the obstacle. Since vibrotactile feedback for navigation directions has been reported to offer several advantages over audio [72] such as discreetness, direct and intuitive matching of stimuli to body coordinates, the tactile channel being less overloaded than the auditory one and requiring less attention and cognitive effort, we have considered this option for output, too. In particular, we deliberated dividing the tablet screen into three areas – left, center, and right – and in the event of an obstacle being detected, have the area corresponding to the obstacle location vibrate (e.g., if the obstacle is on the right, the right area of the screen would vibrate). However, this idea was discarded upon discovering that the Tango tablet does not support such localized vibrations. We still intend to examine some alternative options utilizing such feedback, such as representing each warning/instruction by a unique vibration pattern or attaching a peripheral custom-designed wearable vibrotactile module to the user's body (e.g., on the wrist or chest).

Since the user interface design is so crucial to the eventual acceptance of the system by the target users, we plan to conduct semi-structured interviews with VI users at local institutions to gather their opinions about the type and frequency of feedback such an application should provide as well as their preferences for the output modality (tactile or audio or a combination of both).

IV. SYSTEM EVALUATION

A series of tests were conducted to evaluate if the system can detect obstacles correctly under different conditions and if it can avoid falsely detecting obstacles in some commonly encountered scenarios. For all tests, the settings of the system parameters, such as resolution, space clearing, etc., were as described in section III.B and the user held the tablet as described in section III.A. The details of the tests are provided below.

A. OBSTACLE DETECTION TESTS

The tests were conducted in an empty room containing a 3 m × 4 m × 2.4 m open space. The objective of the tests was to determine if the system could correctly detect obstacles if certain factors were varied. The specific factors studied and the variations applied to each, along with related experimental setup details, are provided below:

1) OBSTACLE SIZE

To test if the system can detect obstacles with sizes ranging from very small to very large, cardboard boxes of five different sizes were utilized (it should be noted that the simplest regular solid form objects (i.e., boxes) with no holes, curves, unusual textures, etc., were chosen. The same object was varied in size so as to avoid the results being affected by differences in other factors such as shape, texture and opacity). The sizes and dimensions of the boxes are shown in Table 1.

The boxes were placed on the ground so that the base (comprised of the width-length face) was flat against the ground (Fig. 5).

TABLE 1. Sizes and dimensions of boxes used as obstacles.

Size		Very large (s1)	Large (s2)	Medium (s3)	Small (s4)	Very small (s5)
Dimensions	Width	60 cm	51 cm	40 cm	18 cm	10 cm
	Length	50 cm	28 cm	23 cm	13 cm	10 cm
	Height	100 cm	82 cm	54 cm	25 cm	10 cm

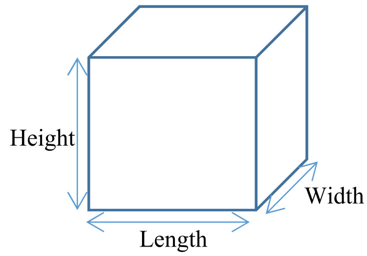

FIGURE 5. The labelling of the height, width and length dimensions of a cardboard box being used as an obstacle. The user faces the width-height face; the width-length face is flat against the ground.

2) DISTANCE FROM OBSTACLE

Since an obstacle warning is triggered whenever the collider box comes in contact with an obstacle, how distant an obstacle can be detected by the system is determined by the dimensions of the collider box. The height of the box (2 m) is more than the height of an average adult [70] and should, thus, be sufficient to ensure the detection of any overhanging obstacles. The width of the box (0.5 m) is about the width of an average adult [69]. We deliberately did not make the width significantly more than the width of an average adult since several studies have shown – and our own initial exploratory tests confirmed this – that people in general and VI individuals, in particular, tend not to walk in a straight line but are inclined to veer towards the right or left while walking [73] (Our initial exploratory tests on veering involved blindfolding three sighted users and asking them to walk in a straight line maintaining normal gait (i.e., not adjusting their gait to ensure a straight course – e.g., by placing one foot right in front of the other one) from one fixed point to another over a distance of about 10 meters in an open indoors space with a uniform floor. All the users walked slowly since they were concentrating on maintaining a straight course but still tended to wobble right and left.). Increasing the width of the collision box would, therefore, trigger obstacle warnings for obstacles which are not too close to the user resulting in the system trying to prevent the user from moving forward even though there may be a big enough gap for him to pass through. Thus, taking veering effects into account, the box width of 0.5 m should be sufficient for detecting obstacles on the right or left of the person while walking. The length of the box (0.6 m) is more than the side width of an average adult and should, thus, be sufficient to ensure the detection of any obstacles immediately in front of the user. However, since the user is walking forward, the distance to obstacles directly in front of the user keeps changing rapidly. The user may, therefore, want to be informed about obstacles which may be more than several cm in front of him.

We, thus, wanted to investigate if the system can correctly detect obstacles at various distances in front of the user. The maximum distance at which an obstacle can be detected is 4-5 m since this is the sensing range of the tablet's depth sensor. However, since the range of a traditional white cane is about 1 m and most assistive obstacle detection systems detect obstacles within 2 m while walking [74], the following distances within the range of 2 m were selected for testing the system: d1 = 0.5 m, d2=1 m and d3=2 m. The distance was varied by setting the length of the collider box to the desired distance.

3) OBSTACLE POSITION ALONG THE HORIZON

Since obstacles directly in front of the user, as well as those which are partially in front of him, should be detected by the system, we wanted to examine if the systems deals with these cases correctly by varying the position of the obstacles along the horizon. Five different positions were tested as described in Table 2.

TABLE 2. Descriptions of horizontal positions of an obstacle.

Horizontal positions	Descriptions
Far left (hp1)	Center of the obstacle aligned with left side of collider box/user
Left (hp2)	Left corner of the obstacle aligned with left side of collider box/user.
Center (hp3)	Center of the obstacle aligned with center of collider box/user.
Right (hp4)	Right corner of the obstacle aligned with right side of collider box/user.
Far right (hp5)	Center of the obstacle aligned with right side of collider box /user.

4) LIGHTING CONDITIONS

The data from the various visual sensors, used to calculate the user's orientation and position, is dependent on the amount of illumination. The performance of the system was, therefore, tested under two different lighting conditions: lc1=dimly lit, lc2=moderately bright. The amount of illumination was varied by turning on all the lights in the room (lc2) and then keeping only one fourth of the lights on while maintaining uniform luminance (lc1). The windows in the room were covered to prevent any sunlight from entering.

All possible combinations of the four factors described above were tested. A position in the room was marked as a destination point. For each test, the user positioned himself in front of the destination point at a distance of about 3 m with the center of his body aligned with the destination point and then started walking towards it. The user stopped either when an obstacle warning was issued or when he reached the destination point.

B. AVOIDING FALSE DETECTION OF OBSTACLES TESTS

The objective of these tests was to determine if the system incorrectly detects obstacles in some commonly occurring scenarios when, in fact, no obstacles exist. The details of the scenarios tested and the experimental setup for each are provided below.

1) GAP BETWEEN OBJECTS

Navigating in indoors environments frequently requires passing through gaps between objects (e.g., a cabinet and a chair,

TABLE 3. Results of obstacle detection tests.

Distance	d1										d2										d3									
Horizontal position	hp1		hp2		hp3		hp4		hp5		hp1		hp2		hp3		hp4		hp5		hp1		hp2		hp3		hp4		hp5	
Lighting condition	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2	lc1	lc2
Obstacle sizes for which the test failed	s4 s5	s4 s5	s5	s5	s5	s5	s5	s5	s4 s5	s4 s5	s4 s5	s4 s5	s5	s5	s5	s5	s5	s5	s4 s5	s4 s5	s4 s5	s4 s5	s5	s5	s5	s5	s5	s5	s4 s5	s4 s5

a table and a wall, a doorway, etc.). Hence, the system should not issue false obstacle warnings if the user is positioned in front of a gap between objects and the gap is large enough for the user to easily pass through. The system was, therefore, tested with the following gap sizes: g1=1.5 m, g2=1 m, g3=0.6 m, g4=0.5 m, g5=0.4 m.

For each gap size, two large objects (cardboard boxes were used) were placed side by side with a gap of the required size between them. The user started walking from a distance of about 2 m away from the gap such that the center of his body was aligned with the center of the gap. The user stopped either when an obstacle warning was issued due to the objects or when he passed through the gap. All tests were conducted under moderate lighting conditions (lc2).

2) WALKING DOWN A CORRIDOR

When walking down a corridor or narrow hallway – another frequently encountered scenario in indoor navigation – the system should not issue obstacle warnings for the walls. The system was, therefore, tested with the following corridor widths: c1=0.8 m, c2=1.2 m, c3=2 m.

The user started walking from a distance of about 2 m away from the corridor entrance such that the center of his body was aligned with the center of the corridor. The user stopped either when an obstacle warning was issued due to the corridor walls or when he reached the end of the corridor. All tests were conducted under moderate lighting conditions (lc2).

3) FLOOR TEXTURE AND LIGHTING CONDITIONS

Since the tablet's depth sensor utilizes IR light, the depth measurements can be adversely impacted in locations with high levels of ambient IR light, such as those lit by bright sunlight or incandescent light bulbs [75]. Also, the depth sensor cannot detect very dark, shiny and transparent materials [75]. Since indoor environments may have floors with varying textures and may be illuminated with ambient IR light, it is important to determine if the system produces false artifacts and consequently, issues false obstacle warnings under these conditions. Additional tests were, therefore, conducted under various ambient IR lighting conditions (indirect sunlight and direct sunlight) and floor textures (highly reflective floors (marble floors), semi reflective floors (tiled floors), non-reflective floors (matt surfaces, and carpeted floors)).

For each floor and lighting combination, an empty area free of any obstacles was selected. The user walked over a distance of about 4 meters across this area. Any problems with meshing and/or false obstacles and artifacts being produced were noted down.

V. RESULTS AND DISCUSSION

The tests described in section IV were conducted with a blindfolded sighted user. Three trials were conducted for each test and the results of the trials were averaged to get the final result. The results of the tests are reported and discussed in this section.

A. RESULTS OF OBSTACLE DETECTION TESTS

The results of the tests conducted for the various combinations of factors described in section IV.A are shown in Table 3. The results indicate that the system functioned correctly for very large, large and medium sized obstacles under all combinations.

However, for small obstacles, the system failed for the far left and far right positions while for very small obstacles, it failed for all combinations.

Since the width of the small and very small obstacles is just a few centimeters (13 cm and 10 cm, respectively), hence, due to veering effects, it is to be expected that the collider box will not come in contact with them when these are placed at the far left and far right positions. This is actually desirable since objects this small would not really be in the user's path if located on the far left and far right.

However, the system's failure to detect small obstacles directly in front of the user does present a problem. Since even a small bump on the ground or a small ledge on the floor may cause a person to trip and stumble and potentially fall down and injure himself, it is important to investigate how the system can be modified to enable it to accurately identify very small obstacles. As the collider box has been set 3 cm above the ground, it would be expected to collide with a 10 cm high object placed on the ground. However, due to the limitations of the hardware and the Unity software being employed, it is possible that the distance of the collider box from the ground may vary, becoming slightly more than 3 cm, as a result of the up and down motion and limb movement of the user as he is walking. Also, it may be possible that a higher resolution mesh is required to accurately capture objects this small on the ground. We, therefore, plan to experiment further with various mesh resolutions and collider box distances from the ground in order to determine how to improve the system's accuracy in detecting very small obstacles.

B. RESULTS OF AVOIDING FALSE DETECTION OF OBSTACLES TESTS

1) GAP BETWEEN OBJECTS

The results of the tests for various gap sizes are shown in Table 4.

TABLE 4. Results of tests for gaps between objects.

Gap size	1.5 m	1 m	0.6 m	0.5 m	0.4 m	0.5 m
Width of collider box	0.5 m	0.5 m	0.5 m	0.5 m	0.5 m	0.4 m
Collision detected?	No	No	No	Yes	Yes	No

The results indicate that the system functioned correctly and did not issue obstacle warnings as long as the gap size was at least 0.1 m more than the width of the collider box. Also, it functioned correctly and issued obstacle warnings when the gap size was less than the width of the collider box.

When the gap size was equal to the width of the collider box, the system incorrectly issued an obstacle warning. However, this was to be expected since, due to veering, the user would not pass exactly in the middle of the gap and therefore, the edges of the collider box would come in contact with the obstacles on either side of the gap. Reducing the width of the collider box by 0.1 m resulted in no collision being detected and appeared to solve this problem. However, as explained in section IV.A(2), the width of the collider box has been set to 0.5 m – which is on the lower end of the width of an average adult – in order to take the veering effects into account. Usability testing with VI users – which we eventually plan to conduct - will more definitively clarify if the collider box width needs to be decreased further to compensate for veering.

2) WALKING DOWN A CORRIDOR

The results of the tests for walking down a corridor or narrow hallway are shown in Table 5.

TABLE 5. Results of tests for walking down a corridor.

Corridor width	0.8 m	0.8 m	1.2 m	1.2 m	1.2 m	1.2 m	2 m	2 m
Collider box width	0.8 m	0. 5m	1.2 m	1.1 m	1 m	0.5 m	1.97 m	1.9 m
Collision detected?	Yes	No	Yes	No	No	No	Yes	No

The results here are similar to the gap test results: the system functioned correctly and did not issue obstacle warnings as long as the corridor width was at least 0.1 m more than the width of the collider box. Also, it functioned correctly and issued obstacle warnings when the corridor width was almost equal to the width of the collider box. In these tests, the width of the collider box was varied just to examine if obstacle warnings would be generated as the collider box width approached the corridor width. However, as can be seen in Table 5, the default collider box width of 0.5 m does not cause obstacle warnings even for quite narrow corridors (as exemplified by the results for the 0.8 m wide corridor).

3) FLOOR TEXTURE AND LIGHTING CONDITIONS

The test results indicate that direct sunlight caused severe problems in the mesh generation with either no mesh or an incorrect mesh being produced for all floor textures (including non-reflective matt surfaces).

Indirect sunlight resulted in the most accurate mesh being generated for all floor textures with the following exceptions: For highly reflective floors, bright spots of light caused a hole to appear in the mesh. Also, non-reflective floors with highly uneven surfaces (such as fuzzy carpets) caused distorted meshing.

The results imply that the system will not function correctly in areas with direct sunlight though it would operate well under indirect sunlight as long as the exceptions mentioned above do not occur.

The overall results of the various tests indicate that the system can correctly detect obstacles of medium to large sizes at various horizontal positions and at distances of up to 2 m under dim to moderate lighting conditions. Furthermore, the system functions correctly under indirect sunlight and with different floor textures except for bright light spots on highly reflective floors and highly uneven surfaces on non-reflective floors. Moreover, the system accurately deals with gaps between objects and corridors avoiding false detection warnings as long as the gap or corridor width is at least 0.1m more than the collider box width.

The system cannot operate under direct sunlight which is to be expected based the limitations of the depth sensor. Also, it is unable to detect very small obstacles – as mentioned in section V.A, we plan to conduct further experiments to investigate how to enable the system to detect such obstacles.

One issue that was observed when initializing the application was that it took a few seconds for the system to generate the mesh while the user had to stand in one place holding the tablet still. However, once the mesh was generated, it was updated in real-time while the user was walking at a slow pace. Since a new version of the Tango SDK was released recently, we updated the SDK on our tablet to this new version and re-conducted several of the above tests. Though the test results were the same in terms of the accuracy of the detection, the mesh generation and updating was much faster, thus, requiring less time for the initial mesh creation and allowing the user to walk at a faster pace.

VI. FUTURE WORK

The work presented in this paper focused on developing a Unity based application for obstacle detection on the Tango platform and evaluating the accuracy of the detection for this system. In the future, we intend to make the necessary modifications to the system to address the issues revealed by the empirical tests. A user-centered approach would be adopted in the remainder of this project with the next phases consisting of eliciting user preferences for the interface design via semi-structured interviews with VI individuals (as explained in section III.C), designing and developing the user interface based on the target users' feedback, and finally, conducting usability testing of the resulting system with VI users to identify any usability problems.

The current system is designed for obstacle detection in an unmapped environment and does not utilize the area learning capabilities of the tablet. In future iterations of the system, we plan to enable the area learning option provided in the Unity Tango SDK which would allow the system to remember where static obstacles in the environment are located. Since we are working in parallel on another project for developing a system for the Tango tablet which would assist VI users to macro-navigate from one indoor location to another, the ability to remember obstacles would be useful in path planning when the obstacle detection (micro-navigation) system is eventually integrated with the macro-navigation one.

The obstacles used in the current tests varied only in size. We plan to conduct further tests with obstacles which vary in shape, texture and opacity to evaluate the impact of these factors, too, on the system performance. Also, overhanging obstacles were not explicitly included in these tests. Since the current height of the box (2 m) may be much greater than the user's actual height, this may result in obstacle warnings being triggered by overhanging objects which the user can easily pass under without colliding. Therefore, we are considering reducing the height of the collider box and placing an additional collider box on top of the original box; the additional box can then be utilized to detect collisions with overhanging obstacles within a certain height range. Another option for dealing with this issue would be to customize the collider box dimensions for each user as discussed next.

Currently, the dimensions of the collider box are preset so that it can encompass an average adult. However, to make the system even more flexible enabling it to cater even to individuals whose dimensions deviate from the norm - i.e., they are unusually tall, short, stout or thin - we plan to give the user the option to customize the height and width of the box according to the dimensions of his own body and to set the length according to his preferences for at what distance from an obstacle he would wish to receive a warning. Moreover, since the user may wish to receive progressive warnings as his distance to an approaching obstacle changes (i.e., "obstacle at 1 m", "obstacle at 0.5 m", ...), we are considering using several collider boxes, instead of a single one, with different lengths.

More detailed information about the obstacles may also be provided to the user: Based on which regions of the box come in contact with the obstacle, the obstacle's position relative to the user can be relayed (i.e., left, right, top, bottom, etc.); Unity's built-in features can be utilized to extract information about the obstacle's size and shape and computer vision-based object recognition methods can be exploited to convey the obstacle's identity to the user. However, the kind and amount of information that would be output to the user would ultimately be decided based on the preferences expressed by the users in the semi-structured interviews.

VII. CONCLUSION

A novel visual and infrared sensor data-based application to assist VI users in detecting and avoiding obstacles in their path while independently navigating indoors has been presented in this paper. The application utilizes the functionalities of the Unity SDK of the Google Project Tango Tablet Development Kit to provide an aesthetically acceptable, cost-effective, portable, stand-alone solution for this purpose. A prototype version of the system has been developed and an extensive empirical evaluation of the obstacle detection component has been conducted, yielding favorable results and thus, confirming the potential of this application for future development work. We are currently modifying the system to address the issues revealed by the tests and adding some customization and detailed feedback options to enhance its functionality and usability.

A user-centered approach will be adopted for the remainder of this project with the next steps consisting of conducting semi-structured interviews to elicit user preferences for the interface design and then iteratively developing and testing the system with the target users to ensure that the final product is better adapted to their unique needs. We hope that providing VI users with a real-time mobile assistive stand-alone application on a cutting–edge device which allows them to detect obstacles independently in possibly unfamiliar indoor surroundings would significantly increase their autonomy. We also hope that our solution would inspire further research for assistive navigation solutions for VI individuals utilizing game engines and the capabilities of the new generation of mobile devices equipped with multiple sensors for gathering environmental data from various modalities as well as powerful processors capable of executing computationally intensive algorithms in real-time.

REFERENCES

[1] R. Manduchi and S. Kurniawan, "Mobility-related accidents experienced by people with visual impairment," *AER J., Res. Pract. Vis. Impairment Blindness*, vol. 4, no. 2, pp. 44–54, 2011.

[2] S. K. West, "How does visual impairment affect performance on tasks of everyday life? The SEE project," *Evidence-Based Eye Care*, vol. 3, no. 4, pp. 218–219, 2002.

[3] J. S. Karlsson, "Self-reports of psychological distress in connection with various degrees of visual impairment," *J. Vis. Impairment Blindness*, vol. 92, pp. 483–490, Jul. 1998.

[4] M. Wilson. (2015). *'Who Put That There!': The Barriers to Blind Partially Sighted People Getting Out About*. Accessed: Apr. 3, 2017. [Online]. Available: https://www.rnib.org.uk/sites/default/files/Who%20put%20that%20there!%20Report%20February%202015.pdf

[5] S. Shoval, I. Ulrich, and J. Borenstein, "NavBelt and the guide-cane [obstacle-avoidance systems for the blind and visually impaired]," *IEEE Robot. Autom. Mag.*, vol. 10, no. 1, pp. 9–20, Mar. 2003.

[6] D. M. Zuckerman, *Blind Adults in America: Their Lives and Challenges*. Washington, DC, USA: National Center for Policy Research for Women & Families, 2004.

[7] R. Jafri and S. A. Ali, "A multimodal tablet–based application for the visually impaired for detecting and recognizing objects in a home environment," in *Proc. 14th Int. Conf. Comput. Helping People Special Needs (ICCHP)*, vol. 8547. Paris, France, 2014, pp. 356–359.

[8] J. Coughlan and R. Manduchi, "Color targets: Fiducials to help visually impaired people find their way by camera phone," *EURASIP J. Image Video Process.*, no. 1, p. 096357, Dec. 2007.

[9] S. Mascetti, D. Ahmetovic, A. Gerino, and C. Bernareggi, "*ZebraRecognizer*: Pedestrian crossing recognition for people with visual impairment or blindness," *Pattern Recognit.*, vol. 60, pp. 405–419, Dec. 2016.

[10] R. Jafri, S. A. Ali, H. R. Arabnia, and S. Fatima, "Computer vision-based object recognition for the visually impaired in an indoors environment: A survey," *Vis. Comput.*, vol. 30, no. 11, pp. 1197–1222, 2014.

[11] H. Takizawa, S. Yamaguchi, M. Aoyagi, N. Ezaki, and S. Mizuno, "Kinect cane: An assistive system for the visually impaired based on the concept of object recognition aid," *Pers. Ubiquitous Comput.*, vol. 19, nos. 5–6, pp. 955–965, 2015.

[12] S. Wang and Y. Tian, "Detecting stairs and pedestrian crosswalks for the blind by RGBD camera," in *Proc. IEEE Int. Conf. Bioinform. Biomed. Workshops (BIBMW)*, Oct. 2012, pp. 732–739.

[13] WHO Media Center. (2014). *Visual Impairment and Blindness: Fact Sheet Number 282*. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/

[14] R. Jafri, "A GPU-accelerated real-time contextual awareness application for the visually impaired on Google's project tango device," *J. Supercomput.*, vol. 73, no. 2, pp. 887–889, 2017.

[15] R. Jafri and M. M. Khan, "Obstacle detection and avoidance for the visually impaired in indoors environments using Google's Project Tango device," in *Proc. 15th Int. Conf. Comput. Helping People Special Needs (ICCHP)*, vol. 9759. Linz, Austria, Jul. 2016, pp. 179–185.

[16] *Google Project Tango*. Accessed: Feb. 5, 2016. [Online]. Available: https://www.google.com/atap/project-tango/

[17] *Tango Developer Overview*. Accessed: Dec. 16, 2016. [Online]. Available: https://developers.google.com/tango/developer-overview

[18] R. Jafri, R. L. Campos, S. A. Ali, and H. R. Arabnia, "Utilizing the Google Project Tango Tablet Development Kit and the Unity engine for image and infrared data-based obstacle detection for the visually impaired," in *Proc. Int. Conf. Health Inform. Med. Syst. (HIMS)*, Las Vegas, NV, USA, 2016, pp. 163–164.

[19] Lenovo. *Phab 2 Pro*. Accessed: Dec. 17, 2016. [Online]. Available: http://shop.lenovo.com/us/en/tango/

[20] Asus. *ZenFone AR (ZS571KL)*. Accessed: Jan. 8, 2017. [Online]. Available: https://www.asus.com/Phone/ZenFone-AR-ZS571KL/

[21] *Unity*. Accessed: Jan. 2, 2017. [Online]. Available: https://unity3d.com/

[22] NVIDIA. *Tegra K1*. Accessed: Aug. 16, 2016. [Online]. Available: http://www.nvidia.com/object/tegra-k1-processor.html

[23] A. Rodríguez, J. J. Yebes, P. F. Alcantarilla, L. M. Bergasa, J. Almazán, and A. Cela, "Assisting the visually impaired: Obstacle detection and warning system by acoustic feedback," *Sensors*, vol. 12, no. 12, pp. 17476–17496, 2012.

[24] J. M. S. Martinez and F. E. Ruiz, "Stereo-based aerial obstacle detection for the visually impaired," in *Proc. Workshop Comput. Vis. Appl. Vis. Impaired*, 2008. [Online]. Available: https://hal.archives-ouvertes.fr/inria-00325455/

[25] H. Tang and Z. Zhu, "A segmentation-based stereovision approach for assisting visually impaired people," in *Proc. 13th Int. Conf. Comput. Helping People Special Needs*, Linz, Austria, 2012, pp. 581–587.

[26] S.-W. Lee, S. Kang, and S.-W. Lee, "A walking guidance system for the visually impaired," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 22, no. 6, pp. 1171–1186, 2008.

[27] T. Schwarze, M. Lauer, M. Schwaab, M. Romanovas, S. Böhm, and T. Jürgensohn, "A camera-based mobility aid for visually impaired people," *Künstliche Intell.*, vol. 30, no. 1, pp. 29–36, 2016.

[28] J. Hwang, Y. Ji, and E. Y. Kim, "Monocular vision-based collision avoidance system," in *Proc. 14th Int. Conf. Human-Comput. Interact. Mobile Devices Services Companion*, San Francisco, CA, USA, 2012, pp. 125–130.

[29] M.-C. Kang, S.-H. Chae, J.-Y. Sun, J.-W. Yoo, and S.-J. Ko, "A novel obstacle detection method based on deformable grid for the visually impaired," *IEEE Trans. Consum. Electron.*, vol. 61, no. 3, pp. 376–383, Aug. 2015.

[30] S. A. Bouhamed, J. F. Eleuch, I. K. Kallel, and D. S. Masmoudi, "New electronic cane for visually impaired people for obstacle detection and recognition," in *Proc. IEEE Int. Conf. Veh. Electron. Safety (ICVES)*, Jul. 2012, pp. 416–420.

[31] Q. Lin and Y. Han, "Safe path estimation for visual-impaired people using polar edge-blob histogram," in *Proc. World Congr. Eng. Comput. Sci.*, 2013, pp. 1–5.

[32] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia, "A smartphone-based obstacle detection and classification system for assisting visually impaired people," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 444–451.

[33] A. Caldini, M. Fanfani, and C. Colombo, "Smartphone-based obstacle detection for the visually impaired," in *Proc. Int. Conf. Image Anal. Process.*, 2015, pp. 480–488.

[34] P. Kaur and S. Kaur, "Proposed hybrid color histogram based obstacle detection technique," in *Proc. 3rd Int. Symp. Comput. Vis. Internet*, Jaipur, India, 2016, pp. 88–97.

[35] V.-N. Hoang, T.-H. Nguyen, T.-L. Le, T.-H. Tran, T.-P. Vuong, and N. Vuillerme, "Obstacle detection and warning system for visually impaired people based on electrode matrix and mobile Kinect," *Vietnam J. Comput. Sci.*, vol. 4, no. 2, pp. 71–83, 2017.

[36] V. Ivanchenko, J. Coughlan, W. Gerrey, and H. Shen, "Computer vision-based clear path guidance for blind wheelchair users," in *Proc. 10th Int. ACM SIGACCESS Conf. Comput. Accessibility*, Halifax, NS, Canada, 2008, pp. 291–292.

[37] K. Magatani, K. Sawa, and K. Yanashima, "Development of the navigation system for the visually impaired by using optical beacons," in *Proc. 23rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 2. Oct. 2001, pp. 1488–1490.

[38] T. Harada, Y. Kaneko, Y. Hirahara, K. Yanashima, and K. Magatani, "Development of the navigation system for visually impaired," in *Proc. 26th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (IEMBS)*, Sep. 2004, pp. 4900–4903.

[39] D. H. Do *et al.*, "Resolving subjects and measuring observer/subject distances with a thermal tactile imager," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBS)*, Aug. 2008, pp. 4302–4305.

[40] B. Mustapha, A. Zayegh, and R. K. Begg, "Wireless obstacle detection system for the elderly and visually impaired people," in *Proc. IEEE Int. Conf. Smart Instrum., Meas. Appl. (ICSIMA)*, Nov. 2013, pp. 1–5.

[41] *Microsoft Kinect*. Accessed: Dec. 23, 2015. [Online]. Available: https://dev.windows.com/en-us/kinect

[42] Occipital Inc. *Structure Sensor*. Accessed: Feb. 5, 2016. [Online]. Available: http://structure.io/

[43] M. Zöllner, S. Huber, H.-C. Jetter, and H. Reiterer, "NAVI—A proof-of-concept of a mobile navigational aid for visually impaired based on the microsoft Kinect," in *Human-Computer Interaction—INTERACT*, vol. 6949, P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, Eds. Berlin, Germany: Springer, 2011, pp. 584–587.

[44] V. Filipe, F. Fernandes, H. Fernandes, A. Sousa, H. Paredes, and J. Barroso, "Blind navigation support system based on Microsoft Kinect," *Proc. Comput. Sci.*, vol. 14, pp. 94–101, 2012.

[45] M. Brock and P. O. Kristensson, "Supporting blind navigation using depth sensing and sonification," in *Proc. ACM Conf. Pervas. Ubiquitous Comput.*, Zurich, Switzerland, 2013, pp. 255–258.

[46] H.-C. Huang, C.-T. Hsieh, and C.-H. Yeh, "An indoor obstacle detection system using depth information and region growth," *Sensors*, vol. 15, no. 10, pp. 27116–27141, 2015.

[47] Y. H. Lee and G. Medioni, "Wearable RGBD indoor navigation system for the blind," in *Computer Vision—ECCV*, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Cham, Switzerland: Springer, 2015, pp. 493–508.

[48] F. Ribeiro, D. Florêncio, P. A. Chou, and Z. Zhang, "Auditory augmented reality: Object sonification for the visually impaired," in *Proc. IEEE 14th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2012, pp. 319–324.

[49] A. Khan, F. Moideen, J. Lopez, W. L. Khoo, and Z. Zhu, "KinDectect: Kinect detecting objects," in *Computers Helping People With Special Needs*, K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds. Berlin, Germany: Springer, 2012, pp. 588–595.

[50] H. Liu, J. Wang, X. Wang, and Y. Qian, "iSee: Obstacle detection and feedback system for the blind," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput.*, Osaka, Japan, 2015, pp. 197–200.

[51] (2003). *H.R. 1902—108th Congress: Medicare Vision Rehabilitation Services Act of 2003*. Accessed: Sep. 13, 2016. [Online]. Available: https://www.govtrack.us/congress/bills/108/hr1902

[52] S. Mann *et al.*, "Blind navigation with a wearable range camera and vibrotactile helmet," in *Proc. 19th ACM Int. Conf. Multimedia*, Scottsdale, AZ, USA, 2011, pp. 1325–1328.

[53] S. Wang, H. Pan, C. Zhang, and Y. Tian, "RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs," *J. Vis. Commun. Image Represent.*, vol. 25, no. 2, pp. 263–272, 2014.

[54] T. J. J. Tang, W. L. D. Lui, and W. H. Li, "Plane-based detection of staircases using inverse depth," in *Proc. Austral. Conf. Robot. Autom. (ACRA)*, 2012, pp. 1–10.

[55] A. Pérez-Yus, G. López-Nicolás, and J. J. Guerrero, "Detection and modelling of staircases using a wearable depth sensor," in *Computer Vision—ECCV*, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Cham, Switzerland: Springer, 2014, pp. 449–463.

[56] R. Simpson *et al.*, "A prototype power assist wheelchair that provides for obstacle detection and avoidance for those with visual impairments," *J. NeuroEng. Rehabil.*, vol. 2, p. 30, Oct. 2005.

[57] K. Yelamarthi, D. Haas, D. Nielsen, and S. Mothersell, "RFID and GPS integrated navigation system for the visually impaired," in *Proc. 53rd IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2010, pp. 1149–1152.

[58] R. Golledge, R. Klatzky, J. Loomis, and J. Marston, "Stated preferences for components of a personal guidance system for nonvisual navigation," *J. Vis. Impairment Blindness*, vol. 98, pp. 135–147, Mar. 2004.

[59] R. Jafri and S. A. Ali, "Exploring the potential of eyewear-based wearable display devices for use by the visually impaired," in *Proc. 3rd Int. Conf. User Sci. Eng. (i-USEr)*, Shah Alam, Malaysia, 2014, pp. 119–124.

[60] D. Anderson. (2015). *Navigation for the Visually Impaired Using a Google Tango RGB-D Tablet*. [Online]. Available: http://www.dan.andersen.name/navigation-for-the-visually-impaired-using-a-google-tango-rgb-d-tablet/

[61] HackDuke. (2015). *Tiresias: An App to Help Visually Impaired People Navigate Easily Through Unfamiliar Buildings*. [Online]. Available: http://devpost.com/software/tiresias-ie0vum

[62] B. Li, J. P. Muñoz, X. Rong, J. Xiao, Y. Tian, and A. Arditi, "ISANA: Wearable context-aware indoor assistive navigation with obstacle avoidance for the blind," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 448–462.

[63] *Project Tango Tablet Teardown*. Accessed: Apr. 6, 2017. [Online]. Available: https://www.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148

[64] *Project Tango SDK*. [Online]. Available: https://developers.google.com/project-tango/downloads#project_tango_sdk_files

[65] *Tango Developers' Overview: Depth Perception*. [Online]. Available: https://developers.google.com/tango/overview/depth-perception

[66] *Tango Developers Overview: Motion Tracking*. [Online]. Available: https://developers.google.com/tango/overview/motion-tracking

[67] *Android Developers*. [Online]. Available: https://developer.android.com/index.html

[68] *Unity User Manual (V.5.5): Physics>Physics Overview>Colliders*. Accessed: Dec. 17, 2016. [Online]. Available: https://docs.unity3d.com/Manual/CollidersOverview.html

[69] D. Littlefield, *Metric Handbook*. Abingdon, U.K.: Routledge, 2008.

[70] Wikipedia: The Free Encyclopedia. Wikimedia Foundation Inc. *List of Average Human Height Worldwide*. Accessed: Apr. 3, 2017. [Online]. Available: https://en.wikipedia.org/w/index.php?title=List_of_average_human_height_worldwide&oldid=806422555

[71] *Unity User Manual (V.5.5): Physics.Raycast*. Accessed: Dec. 17, 2016. [Online]. Available: https://docs.unity3d.com/ScriptReference/Physics.Raycast.html

[72] C. Loconsole, M. B. Dehkordi, E. Sotgiu, M. Fontana, M. Bergamasco, and A. Frisoli, "An IMU and RFID-based navigation system providing vibrotactile feedback for visually impaired people," in *Proc. 10th Int. Conf. Haptics, Perception, Devices, Control, Appl. (EuroHaptics)*, London, U.K., Jul. 2016, pp. 360–370.

[73] D. A. Ross, "Implementing assistive technology on wearable computers," *IEEE Intell. Syst.*, vol. 16, no. 3, pp. 47–53, May 2001.

[74] S. Y. Kim and K. Cho. (2013, April 6, 2017)., "Usability and design guidelines of smart canes for users with visual impairments," *Int. J. Design*, vol. 7, no. 1, pp. 99–110, 2013. [Online]. Available: http://www.ijdesign.org/ojs/index.php/IJDesign/article/view/1209/559

[75] *Tango UX Best Practices*. Accessed: Oct. 29, 2016. [Online]. Available: https://developers.google.com/project-tango/ux/ux-best-practices

RABIA JAFRI received the B.S. degree in mathematics and physics from Islamia University, Pakistan, in 1995, and the B.S. and Ph.D. degrees in computer science from the University of Georgia, Athens, GA, USA, in 1999 and 2008, respectively.

She is currently an Assistant Professor of information technology with King Saud University, Saudi Arabia. Her research interests include image processing, computer vision, pattern recognition and related applications with a focus on assistive technologies, biometrics, and medical imaging. She is on the Editorial Board of *The Journal of Supercomputing* (Springer).

RODRIGO LOUZADA CAMPOS was born in Vitória, Espírito Santo, Brazil, in 1994. He is currently pursuing the B.S. degree in computer engineering with the Federal Center for Technological Education of Minas Gerais, Timóteo, Brazil.

From 2015 to 2016, he was an Exchange Student with the Brazilian Scientific Mobility Program by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior with the University of Georgia, Athens, GA, USA. His research interests include computer vision, neural networks, and assistive technologies.

SYED ABID ALI received the B.S. degree in computer science from the Georgia College and State University, Milledgeville, GA, USA, in 1992, and the M.S. degree in computer science from the University of Georgia, Athens, GA, USA, in 1998.

He served as the IT Manager for the Department of Biochemistry and Molecular Biology, University of Georgia, for over 12 years. He is currently a CIO/COO with the Umbrella Group Medical Technology Link Alliance (consisting of ISMTEC, Aracure, and Araware LLCs) out of the USA and a Partner with the Canadian Life Sciences Company, Connect Genomics. His latest research work is in the area of application development for the visually impaired and 3-D printing. His other research interests include parallel computer processing and architecture, stereo correlation, face recognition, gait recognition, and image processing in general.

HAMID R. ARABNIA received the Ph.D. degree in computer science from the University of Kent, Canterbury, in 1987.

He has been with the University of Georgia, Athens, GA, USA, since 1987, where he is currently a Professor of computer science. He is a Fellow and an Advisor of the Center of Excellence in Terrorism, Resilience, Intelligence and Organized Crime Research and an Academy Member of the University of Georgia Teaching Academy (program initiated by the Carnegie Foundation), inducted in 2015. His research interests include parallel and distributed processing techniques and algorithms, supercomputing, big data analytics (in the context of scalable HPC), imaging science (image processing, computer vision, and computer graphics), and other compute intensive problems. Methodologies that promote cross-disciplinary education are also of great interest. He has authored extensively in journals and refereed conference proceedings. He has authored or co-authored about 200 peer-reviewed research publications and also 250 edited research books in his areas of expertise. Applications of interest include health informatics, medical imaging, and security. Most recent activities include studying ways to promote legislation that would prevent cyberstalking, cyber harassment, and cyberbullying. He is an Editor-in-Chief of *The Journal of Supercomputing* (Springer).