
Blind's Eye: Employing Google Directions API for Outdoor Navigation of Visually Impaired Pedestrians

SABA FERAZ MEMON*, MOHSIN ALI MEMON*, SHEHNILA ZARDARI**, AND
SHAHZAD NIZAMINI*

RECEIVED ON 03.11.2016 ACCEPTED ON 21.02.2017

ABSTRACT

Vision plays a paramount role in our everyday life and assists human in almost every walk of life. The people lacking vision sense require assistance to move freely. The inability of unassisted navigation and orientation in outdoor environments is one of the most important constraints for people with visual impairment. Motivated by this problem, we developed a simplified and user friendly navigation system that allows visually impaired pedestrians to reach their desired outdoor location. We designed a Braille keyboard to allow the blind user to input their destination. The proposed system makes use of Google Directions API (Application Program Interface) to generate the right path to a destination. The visually impaired pedestrians have to wear a vibration belt to keep them on the track. The evaluation exposes shortcomings of Google Directions API when used for navigating the visually impaired pedestrians in an outdoor environment.

Key Words: Navigation, Visually Impaired, Braille, Android, Google Directions Application Program Interface, Arduino.

1. INTRODUCTION

Visual impairment is one of the most serious types of disabilities. Many people admit that visualizing ability plays a critical role in our daily life. According to WHO (World Health Organization) research of, there are over 285 million visually impaired people in this world, out of which 39 million people are totally blind and 246 million people have different kind of visual impairment [1]. The rate of blindness will roughly double worldwide by the year 2020 [2]. People having visual impairments, often face

numerous challenges when moving from one place to another in an outdoor environment. Although, it can be easier for visually impaired people to walk in a familiar environment like in their home streets because they are living there since birth. However, it is challenging for them to move in a new or unfamiliar outdoor environment and this incurs a high cognitive load on them. Many visually impaired pedestrians have even reported a fear of falling when they walk in an unfamiliar environment [2].

* Department of Software Engineering, Mehran University of Engineering & Technology, Jamshoro.

** Department of Computer Systems & Software Engineering, NED University of Engineering & Technology, Karachi.

Nandhini et. al. [3] highlights some traditional techniques that are available to guide visually impaired pedestrians. Some people prefer guide dogs, canes and some prefer humans to help them walk in unfamiliar surroundings. But, in this busy world, humans may not always be available to help a blind person. Also, guide dogs or guide cane provide limited aid to the pedestrians, are costly and burdening and require proper training [4].

Over the past few years, different researchers and developers tried their best to utilize mobile assistive technologies and developed such systems that could reduce the challenges faced by visually impaired pedestrians [5]. Lakde and Prasad's research [6] discusses that some of those systems have shown to be not very useful in real scenarios; they have a high installation cost, lack of usability or are limited to the indoor navigation only.

Owing to the difficulties faced by the visually impaired pedestrians, we developed an easy to use and cost effective blind navigation system that consists of a smart phone and vibration belt. The proposed system intuitively obtains the destination address via Braille keyboard and provides step by step navigations to visually impaired pedestrians with vibrotactile feedback. With our approach, we achieve several objectives; one of them is that visually impaired users will not have to continuously wear earphones so they are able to hear ambient sounds. Secondly, the system runs on android based smart phones, thus, providing a cheaper solution for navigation in outdoor environments. Last, the vibration belt is easily wearable and light weight solution to guide the user to reach the desired location.

2. RELATED WORK

A large number of systems have already been developed over the past few years to aid visually impaired people. We reviewed and analyzed existing research to find out

more about the current systems that help visually impaired people to navigate. In this section, we highlight the properties and limitations of such systems.

The study of Zöllner et. al. [7] was concerned with a system for visually impaired pedestrians that assists them in both micro-navigation (sensing of the immediate environment for obstacles and hazards) and macro-navigation (navigating to remote destinations beyond the immediate perceptible environment) inside buildings by providing vibration and voice feedback. The system consists of a vibrotactile waist belt that has been built with three pairs of ArduinoLilyPad vibe boards located at the left side, right side and centre of the person's waist for providing vibration feedback, a head-mounted Microsoft Kinect sensor and a simple backpack construction that carries the laptop and enables quick debugging. The system uses printed augmented reality markers to tag the points of interest on the desired path. The Kinect's RGB (Red, Green Blue) camera detects the markers and provides navigation instructions to the users depending on the distances of the user from the marker. However, this system has a few limitations. The waist belt, Microsoft Kinect sensor and backpack make the system burdening for the user. Furthermore, the system is limited to indoor navigation only.

A research was conducted whereby they developed a smart phone application "Crossing Guard" that provides sidewalk to sidewalk directions to the visually impaired pedestrians [8]. Additionally, it provides information about the shape and width of the intersections and about traffic control systems. Furthermore, it explicitly instructs the user when to cross intersections. The users must enter the destination location into the smart phone by using the keyboard or built in voice recognizers of the phone. The users can interact with the system via simple hand gestures with the phone. This indeed is a good system having two limitations. The system assumes that the user

enters the destination location by using the built-in keyboard or voice recognizers of the smart phone, which is a difficult task for a visually impaired person. Secondly, the system uses bone conducting head phones for providing feedback to the user while walking whereas the use of headphones while walking is illegal and have been prohibited in many countries.

Ren et. al. [9] is a system for visually impaired pedestrians that help them move independently and safely in both indoor as well as outdoor environments. The user can easily switch between the two environments by providing a simple voice command. It provides features such as 'adaptability to changes' and 'dynamic interaction' to the user by using a wireless connection, a wearable computer, a precise position measurement system and a vocal communication interface. It is designed in such a way that it delivers the information about the environment to the user in real time by using the auditory cues and provides information about the obstacles on request. However, the system is completely voice based and has a higher learning curve which makes it difficult for the visually impaired pedestrians to use and understand.

The study of Ben et. al. [10] deals with a cane-based system for visually impaired people that help them to detect and avoid various obstacles that appear in their way of moving towards a destination. It has been implemented on an Android-based Smartphone and uses three ultrasonic sensors (Upward Sensor, Downward Sensor and Ground Sensor) connected to an IOIO (pronounced yo-yo) circuit board to determine the height and proximity of various objects. The IOIO board along with the ultrasonic sensors could be mounted on the cane used by the visually impaired persons and could detect any objects within a range of 7 meters from the cane. The IOIO board translates the analog signals from the ultrasonic sensors and sends the information to the

Smartphone via a connected Bluetooth radio. The system has a few drawbacks such as the system requires that the visually impaired person use a cane to mount the IOIO board and the ultrasonic sensors. Also, the system uses different patterns of the built in vibrations and chimes of the Android phone to provide information about the height and proximity of the objects to the user, which can often confuse and mislead the user.

Lakde and Prasad's [11] research focused on making a shoe having depth, obstacle detection and RGB sensors. The proposed system consists of a number of infrared obstacle sensors that were connected on the left side, right side and front side of the shoes to detect the accurate position of the obstacle. RGB color sensor detects an object depending on the red, blue and green color level intensities of the object and is connected at the front of the shoes facing the ground. A processing unit consisting of an Arduino microcontroller, a power supply and an IC (Integrated Circuit) for converting voice to sound collects all the information from the sensors and sends commands to the response system for providing response to the user via voice and vibrations. The vibration alert is given to the user via a hand wearable device that produces vibrations to inform the user about an obstacle and intensity of the vibration depends upon the distance of the obstacle from the user. The voice alert is given to the user via an ear phone which is directly connected with an IC that records different alert messages at the time of development and provides alerts to the users depending upon the instructions from the processing unit. The system is helpful for the blinds but it doesn't provide any information about the current location of the user, has a high installation cost and is burdening for the user.

In the next section we will discuss about the prototype system designed to achieve our objectives.

3. PROTOTYPE APPLICATION

The prototype application has been developed on an Android-based smart phone “Samsung Galaxy S4 i9505” that works as a blind navigation system. The smart phone has the operating system Android 4.4 KitKat installed in it. However, the prototype system can run on any Android-based smart phone having USB Host mode and API level in between 13 and 22. Thus, the visually impaired people can easily access the system onto the Smart Phone owned by them.

While, many blind navigation systems [12-14] rely on Voice Recognition or Braille notetakers [15] for taking input from the blind users. The users of such systems

face many issues. The Braille notetakers are less adaptable and cumbersome to carry from one place to another. Using voice recognition invades the privacy of the user and such systems are easily affected by the factors such as individual accents, noisy environment and annotations. Therefore, we developed a touch screen Braille keyboard shown in **Fig. 1(a)** for taking input from the blind person via the smart phone. The design of this keyboard was inspired from Frey et. al. [16], and Southern et. al. [17]. Our keyboard consists of six buttons. In order to type, the user must hold the smart phone in the horizontal direction facing away from the user. The navigation buttons of the smart phone must be near the left hand of the user and the camera must be near the right hand of the user.



FIG. 1(a). TOUCH SCREEN BRAILLE KEYBOARD

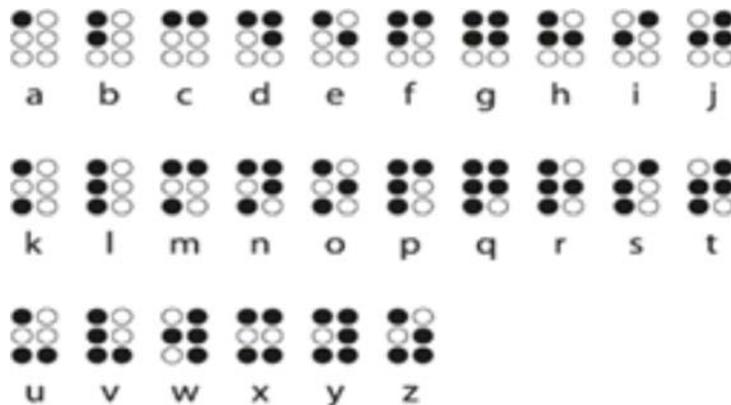


FIG. 1(b). BRAILLEKEYS [23]

By placing the first three fingers of each hand onto the three buttons on each side (Step-3 of Fig. 3) and by pressing different combinations of these buttons according to the Braille keys [18] shown in Fig. 1(b), the blind person can type different letters. The keyboard sends the audio feedback of the letters typed by the user. It also contains four swipe gestures for 'Space', 'Backspace', 'Enter' and 'Clear Screen' to make the interaction easier. Thus, the keyboard provides an intuitive and user friendly environment to the blind. Our system uses Google Fused Location Provider API [19] to find out the current location of the user. Google Fused Location Provider API uses a fusion of network, internet and GPS (Global Positioning System) providers to continuously monitor the location of the user.

This API provides various advantages over GPS based navigation systems [3,20]. For instance; it saves battery, provides an accuracy of approximately 10 meters when the priority is set to 'HIGH', uses the GPS receiver of the smart phone and eliminates the need of using costly and cumbersome GPS receivers. Since voice based feedback systems [10,12,13,21] have many drawbacks for instance the voice feedback is easily affected by noisy environment and can distract the user as mentioned in [22]. Therefore, many systems such as [22,23] make use of vibration feedback for providing navigational instructions to the visually impaired people. Some systems [10] even use a combination of voice and vibration feedback to guide visually impaired people. As we know that the vibrations of the smart phones have limited variations and different patterns of such vibrations can often be confusing and incur a high cognitive load on the user, a large number of wearable devices for providing vibration feedback have been made to assist visually impaired people [24].

Specifically, the use of vibration vests for providing navigational instructions to the visually impaired have been increased during the past few years. But the existing

vibrations vests are costly and burdening for the users as they contain a lot of vibration motors. Therefore, our research uses customized wearable vibration belt named as "Blind's Eye Vibration Belt" for providing step by step navigations to the users via vibrations. The cost of the belt is approximately \$50. The design of the belt was inspired from Zöllner's work [7]. The belt holds an 'Arduino Mega 2560Microcontroller [25] and three 'MT-VBR 1003 Vibration Motors [26]. We connected vibration motors with Arduino Board using jumper wires and soldered them. We connected Arduino Board with Android Phone via 'Usb OTG Cable'. The belt is used to provide continuous vibration feedback to the user (**Fig. 2**).

4. METHODOLOGY

We explain the methodology applied on the prototype application with the help of **Fig. 3**. Our system consists of five steps. In the first step, the blind user presses the power button of the smart phone twice to run Blind's eye application on the smart phone. In step 2, the system uses Google Fused Location Provider API to find out the current location (latitude, longitude) of the user, which is then converted into street address of the user.

In the next section, we explain the complete working as well implementation of our prototype system.

The system gives voice feedback to the user, informing him about his current location. Following, the use of our customized touch screen Braille Keyboard to enter the destination location. The system uses voice feedback only in the first three steps to interact with the system. In the next step, our system uses Google Directions API to find out the path from user's current location to the destination location entered by the user. Finally, in step 5, the application starts navigating the blind person via the three vibration motors attached with the vibration belt that is worn by the blind person.

Now we will discuss in detail how each step has been implemented in our system step by step.

4.1 Step-1: Using Power Button to Open Blind's Eye

As we all know that a blind person cannot directly navigate through the Smartphone. Therefore, we have provided a special feature to the user for running our system. The user can use the power button of the smart phone to run our application. For implementing this functionality, we used the concept of a foreground service

and receiver in android. Once the application has been installed on the Smartphone and the user runs it at least once, therefore ground service will start running. The service registers a receiver which listens for "ACTIONSCREEN ON" and "ACTION SCREEN OFF" events. Whenever both these events will be triggered together i.e. whenever the blind person will press the power button of the Smartphone twice, the OnReceive method of the receiver will run the activity of our application.

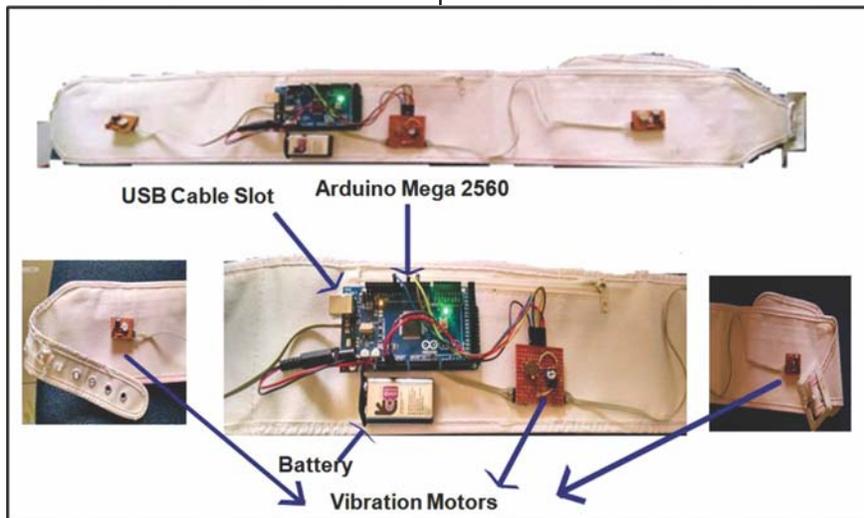


FIG. 2. BLIND'S EYE VIBRATION BELT

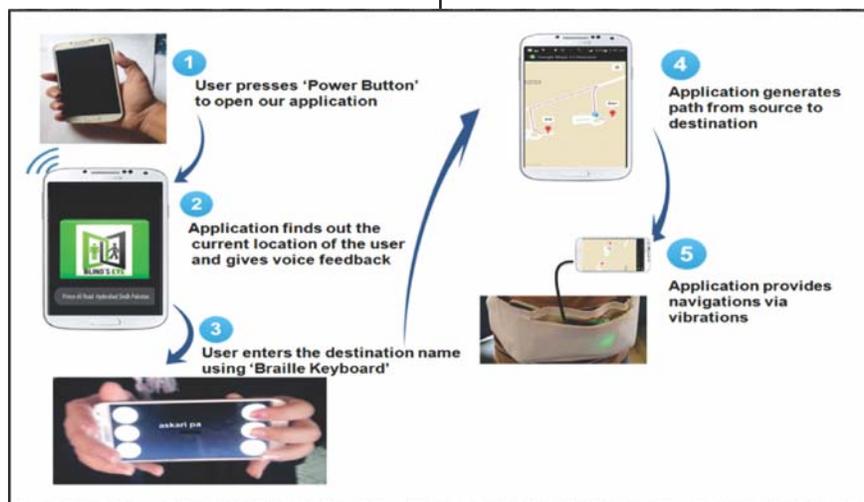


FIG. 3. WORKING STEPS OF BLIND'S EYE

4.2 Step-2: Finding Current Location of the User

Before entering the destination location, the blind user may require the information about his current location. For that purpose, we used Google Fused Location Provider API to find out the current location of the user. Google Fused Location Provider API returns the current location in terms of latitude and longitude. We used Reverse Geocoding in Android to convert the latitude longitude values into street address of the user. Finally, we used Android TTS (Text To Speech) [27] to convert the text of the street address into voice format and provide response to the user in the form of voice feedback.

4.3 Step-3: Using Braille Keyboard to Enter Destination Location

In this step, a touch screen Braille Keyboard is provided to the user to enter the destination location via the Smart Phone. The user must hold the Smart Phone as shown in Fig. 5 and type the destination location. The application uses Android OnTouch listener for handling multi-hand gestures in the keyboard and provides continuous voice feedback to the user about the letters typed. After typing the destination location, the user must swipe from left to right to enter the destination location. The system provides voice feedback to the user about the destination entered.

4.4 Step-4: Finding Path from Source to Destination

The current prototype system requires the destination locations to be already fed in the system. The system has its own SQLite database which contains latitude and longitude values of different locations of Hyderabad City (Table 1). When the user enters a destination location, the application traverses its database in order to find a match of that location in the database. If a match is not found, the application directs the user to step 3 and requests the user to enter the destination again by providing a voice feedback. If a match is found, the application fetches the corresponding latitude and longitude values from the database. The application uses Google Directions API to calculate a path from source to destination location. It provides the latitude longitude values of the source location (user's current location) and the destination location (entered by the user) along with mode of transport (walking) as parameters to the Google Direction's API via an Http Request. Directions API return the paths from source to destination as a series of way points. We developed a JSON Parser to parse the results obtained from Directions API and stored it into an array. We then imported Google Maps in our application by using Google Maps Android API and drew the path from source to destination via a polyline. The current prototype system navigates the user to a limited number of locations only because it sends the latitude and

TABLE 1. BLIND'S EYE DATABASE

Location Name	Location Latitude	Location Longitude
Department of Software Engineering	25.40486	68.26106333333334
Department of Biomedical Engineering	25.4045466666666	68.260115
Midway Restaurant	25.4074464	68.36667075
Gulab Pan House	25.40556242	68.36758393
Govt Girls Degree College	25.40502	68.345637
Kosar Masjid	25.403904	68.342191

longitude values of the destination location to the Directions API to find out the path to destination. These latitude/longitude values were manually collected by us for providing better accuracy to the user. However, the system can be easily modified to provide navigations to as many locations as wanted by the user by providing the name of the destination location directly to the Directions API. In such a case, the Directions API will use the latitude/longitude values of the destination location from Google's database (not complete enough for Hyderabad city) which may not be 100% accurate for some locations causing the user to reach away from the destination.

4.5 Step-5: Providing Step by Step Navigation to User

In the final step, our application uses Blind's Eye vibration belt for providing step by step directions to the user (Fig. 4). We used USBOTG cable for connecting the Smartphone with the Arduino board. The application communicates with the Arduino board by using Android USB Serial API.

When the path from source to destination has been ascertained, our application starts sending signals to the Arduino board, which in turn makes the motors attached to it vibrate. The three vibration motors are positioned on

the left side of the waist, right side of the waist and above the navel (Fig. 4).

If the motor at the left or the right vibrates, the user should slowly turn in the respective direction as long as he feels the vibration. If the motor at the front vibrates, the user should keep walking in the forward direction. The application performs periodic location updates to keep the user on the correct path. When the user reaches the destination location, all the three motors vibrate simultaneously to signal the user that he has reached his desired location.

Hence, the proposed system assists a blind person in reaching his desired destination with confidence and ease.

5. EVALUATION

In this section, we discuss the evaluation of our prototype system and present the results of our user study. Due to unavailability of blind participants, we employed non-blind participants to evaluate the system. Three Test Cases were setup to evaluate the functionality of the whole system. The purpose of the Test Case-I was to evaluate different versions of the Touch Screen Braille Keyboard and select the version which provides better accuracy where as Test Case-II and III were set up to evaluate our prototype's usefulness in real world navigation.



FIG. 4. APPLICATION PROVIDES DIRECTIONS FROM SOURCE TO DESTINATIONS VIA VIBRATIONS

5.1 Test Case-I: Evaluating the Braille Keyboard

We developed four versions of Braille keyboard (Fig. 5) with slight differences. In the first version, we used square buttons with spacing of 14 dpi (Density-independent pixels). In the second version, we used same square buttons but here we increased the spacing between these buttons to 30dpi. In the third version, we used circular buttons with spacing of 14 dpi and again in fourth version we increased the spacing between buttons.

We recruited five non-blind participants (3 female and 2 male) and briefed them about the purpose of the study and usage of the keyboard for fifteen minutes. The participants were aged between 20 and 22. We provided a Samsung Galaxy S4 smart phone to each participant in which all the four versions of the Braille keyboard were pre-installed. Due to unfamiliarity of the participants with the Braille Keys, they were requested to sit facing a Laptop displaying the Braille Keys. We asked them to type five words (Simla, Mehran, Karachi, Hyderabad and Jamshoro) using each version of the Braille Keyboard. A stop watch

was used to note the time (in seconds) taken by each participant to type each word using each version of the Braille Keyboard (Fig. 5).

5.2 Test Case-II: Evaluating Source to Destination Navigation (Short Route-I)

We recruited five non-blind participants (4 male, 1 female) and briefed them for fifteen minutes about purpose of the study and the usage of the system. The ages of the participants were between 23 and 35 and their educational qualification was between matriculation and graduation. We provided a Samsung Galaxy S4 Smart phone to each participant in which our application Blind's Eye was pre-installed. The participants were asked to wear Blind's Eye vibration belt and walk a short route by following the navigational instructions from the vibration belt. The short route (Fig. 6(a)) was 260 meters long. While walking, the participants were accompanied by one of the experimenters to avoid any hazardous situation. All the recruited participants reached the exact destination location. It was however noted that when the participants reached the exact destination, and location. The

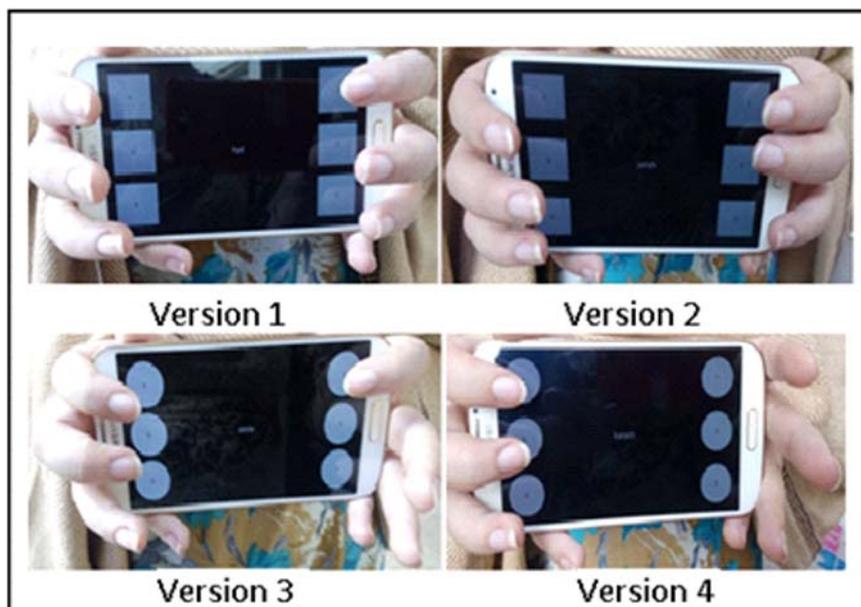


FIG. 5. VERSIONS OF BRAILLE KEYBOARD

application couldn't detect the stopping point and kept navigating the participants. As a result, the participants stopped at locations which were a little farther away (approximately 5-25 meters) from the exact destination. Each green dot in **Fig. 6(b)** shows the point where one of the participants stopped near the destination. We documented the time taken by the participants to reach the exact destination, time by the participants to reach further away from the destination and extra distance covered by the participants.

Similar to Test Case-II, we recruited five non-blind participants (4 male and 1 female) and briefed them for fifteen minutes about the purpose of the study and the usage of the system. The ages of the participants were between 16 and 23 and their educational qualification was

between matriculation and graduation. We provided a Samsung Galaxy S4 smart phone to each participant in which our application Blind's Eye was pre-installed. The participants were asked to wear Blind's Eye vibration belt and walk a short route by following the navigational instructions from the vibration belt. The short route (**Fig. 7(a)**) was 187 meters long. One of the five participants was requested to walk the route again with a blindfold. While walking, the participants were accompanied by one of the experimenters to avoid any hazardous situation. In this case, we noted the same error. I.e. the participants stopped at locations which were farther away from the destination as shown by green dots in **Fig. 7(b)**. However, here the error rate was very low as compared to Test Case-II.

6. RESULTS AND DISCUSSION

Initially, all the four versions of the Braille keyboard were evaluated. It was noted that the participants were easily able to use the keyboard after a short training and were quite satisfied with the design of the keyboard. It was also observed that version 3 of the Braille keyboard gives

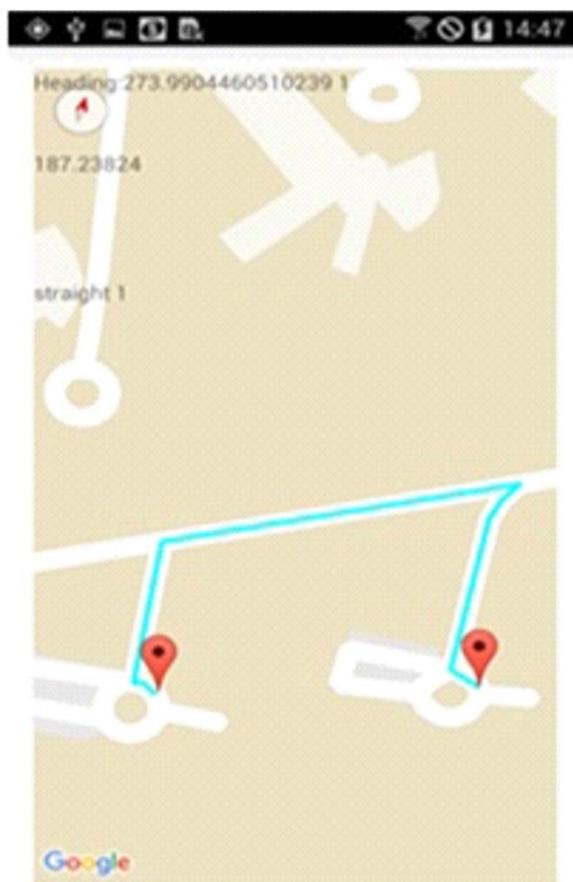


FIG. 6(a). SHORT ROUTE-I (LEFT)

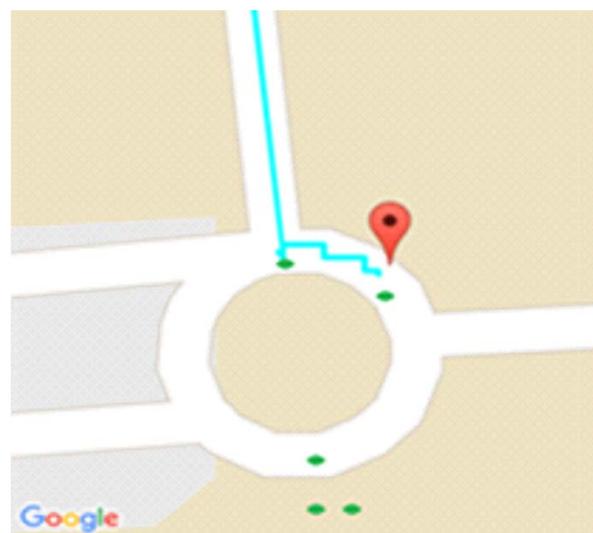


FIG. 6(b). POINTS WHERE THE PARTICIPANTS STOPPED NEAR THE DESTINATION (RIGHT) TEST CASE-III: EVALUATING SOURCE TO DESTINATION NAVIGATION (SHORT ROUTE-II)

better accuracy and was selected for inclusion in the final version of the prototype system. Then, in test case-II and test case-III we assessed the system's efficiency in providing real world navigation. It was noted that all the participants reached the exact destination location but stopped at locations which were a little farther away from exact destination. The average time taken by the participants to reach the exact destination location was quite close to the actual time (4 minutes) in both cases.

After detailed analysis of both short routes, we concluded that the errors occurred due to the presence of irregular paths near the destination locations. Fig.8(a-b). The irregular paths consisted of waypoints which were very close to each other. Some of the waypoints were even located at a distance of less than 1 meter from each other shown by red dots in Fig. 8(a-b). Whereas as stated above, Google Fused Location Provider API at its best can provide the current location on with an accuracy of

approximately 10 meters. The presence of closer waypoints makes it very difficult for the API to accurately detect when the user crosses a specific waypoint causing the application to show unexpected behavior in such situations. Furthermore, it was also noted that the average error rate in Test Case-II (1.37) was much higher than the error rate in test Case-III (0.11). The time taken by the participants in Test case-II and III and the error rate are shown in Fig. 9. The reason for such a huge variation was the presence of closer waypoints in short Route-I as compared to the waypoints in short Route-II. Thus, we can conclude that the error rate highly depends on the path near the destination location. If the path near the destination is highly irregular i.e. contains many closer waypoints as in short the problem of reaching away from destination due to Google Directions API was highlighted in the evaluation. This issue may be resolved by various methods such as, continuously calculating the distance

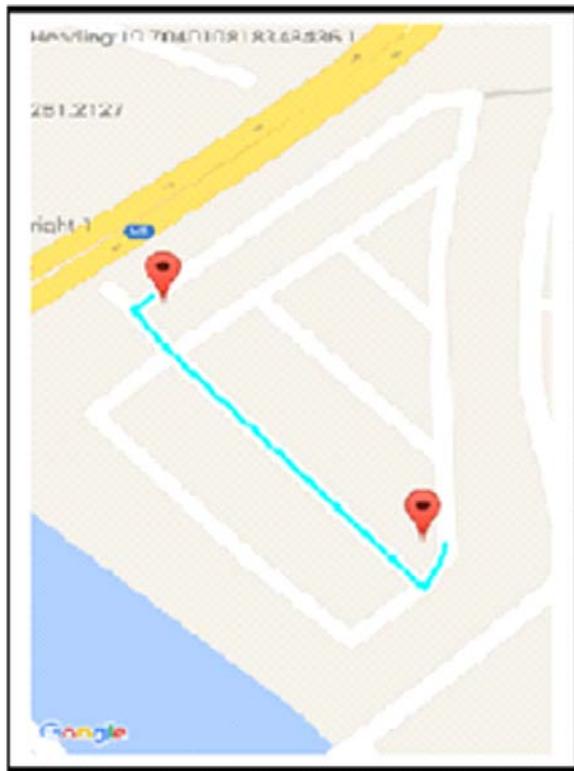


FIG. 7(a). SHORT ROUTE II (LEFT)

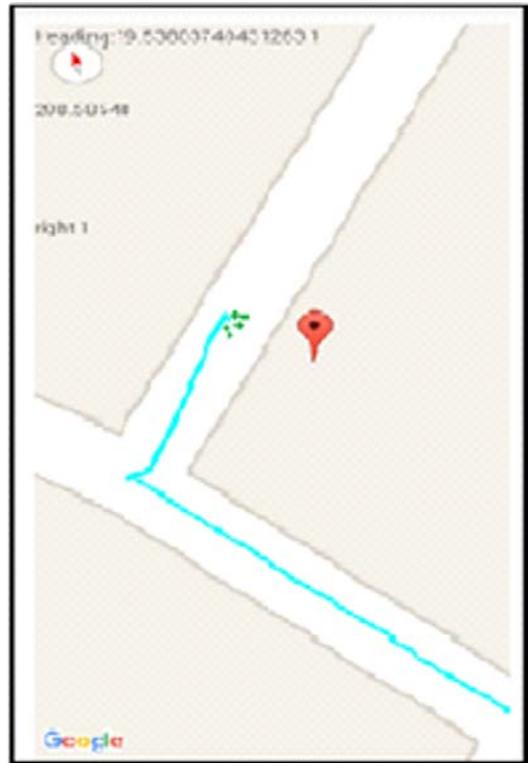


FIG. 7(b). POINTS WHERE THE PARTICIPANTS STOPPED NEAR THE DESTINATION (RIGHT)

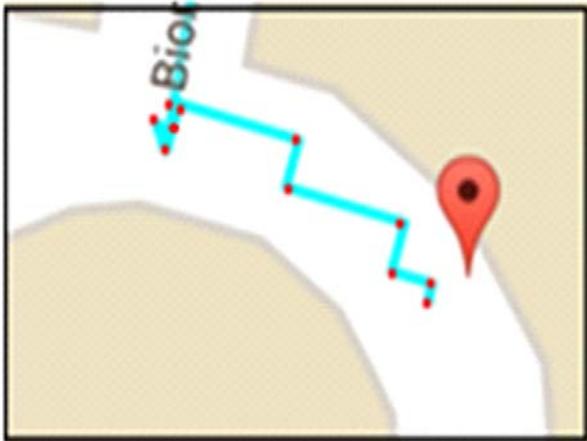


FIG.8(a). WAYPOINTS NEAR THE DESTINATION (SHORT ROUTE-I)

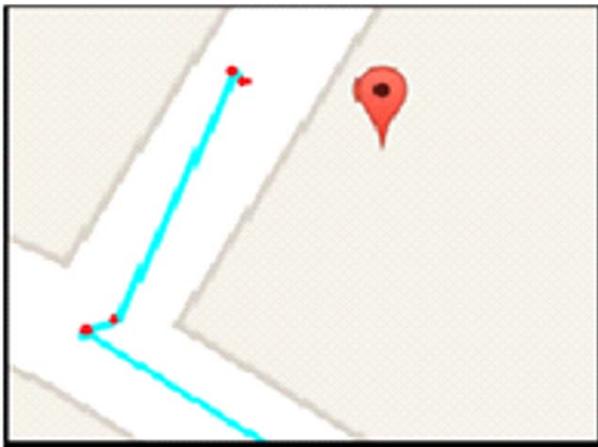


FIG.8(b). WAYPOINTS NEAR THE DESTINATION (SHORT ROUTE-II)

between user's current location and the destination location and stopping the user when the distance becomes approximately zero or adding RFID (Radio-Frequency Identification)tags at the locations through which the system can detect when the user reaches the exact destination.

However, in spite of the errors, the overall performance of the system was satisfactory. The users were easily able to use the system and gave positive feedback. The concept of using Touch Screen Braille Keyboard and wearable vibration belt was supported by the users, and with some amendments, the system can be used by visually impaired participants in the real world.

7. CONCLUSION

A simplified and user friendly navigation system with small learning curve for assistance of visually impaired pedestrians has been developed. Visually impaired can easily use this application after a short training. The initial assessment of the current system was promising and positive feedback was received from users about working of the system. However, Google Directions API used in

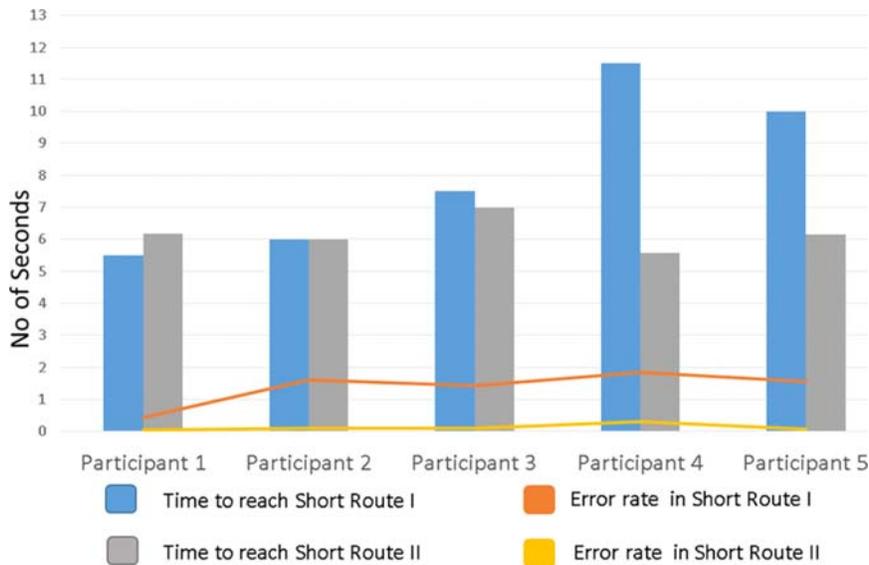


FIG. 9.TIME TAKEN BY PARTICIPANTS AND ERROR RATES IN TEST CASE-I AND II.

the prototype contains few limitations, therefore a proper stopping mechanism to stop the user at the exact destination location is required to prevent the user from walking farther away from the destination. Moreover, our application doesn't warn the users about the obstacles that come in their way of moving towards a destination which is also a limitation.

8. FUTURE WORK

Therefore, in future, we are thinking to do more work on the current prototype and add a way to properly detect the user reaching the exact destination so as to stop the user at the exact point. Also, we are working to attach an ultrasonic sensor (Ultrasonic Sensor) with the vibration belt and connect it with the Arduino board. The sensor will detect any obstacles and the Arduino board will provide the information about the obstacles to the application which in turn will warn the user about the obstacles by providing vibrations via the vibration belt.

ACKNOWLEDGMENTS

Authors are thankful to Mehran University of Engineering & Technology, Jamshoro, and NED University of Engineering & Technology, Karachi, Pakistan, for provision of facilities for this research. Special thanks to Internal and External Referees/Experts, for their valuable comments and suggestions regarding improvement of the paper.

REFERENCES

- [1] West, S., and Sommer, A., "Prevention of Blindness and Priorities for the Future", Bulletin of the World Health Organization, Volume 79, pp.244-248, 2001.
- [2] Norgate, S.H., "Accessibility of Urban Spaces for Visually Impaired Pedestrians", Proceedings of the Institution of Civil Engineers, Volume. 165, No.ME4, pp. 231-237, December, 2012.
- [3] Nandhini, N., Vinoth, C.G., and Deepa, G.P., "Talking Assistance about Location Finding Both Indoor and Outdoor for Blind People", International Journal of Innovative Research in Science, Engineering & Technology, Volume 3, No.2, pp. 9644-9651, February, 2014.
- [4] Ulrich, I., and Borenstein, J., "The GuideCane: Applying Mobile Robot Technologies to Assist the Visually Impaired", IEEE Transactions on Systems, Man, and Cybernetics, Part-A: Systems and Humans, Volume 31, No. 2, pp. 131-136, March, 2001.
- [5] Hakobyan, L., Lumsden, J., O'Sullivan, D., and Bartlett, H., "Mobile Assistive Technologies for the Visually Impaired", Survey of Ophthalmology, Volume 58, No. 6, pp.513-528, November, 2014.
- [6] Lakde, C.K., and Prasad, P.S., "Review Paper on Navigation System for Visually Impaired People", International Journal of Advanced Research in Computer & Communication Engineering, Volume 4, No. 1, January, 2015.
- [7] Zöllner, M., Huber, S., Jetter, H.C., and Reiterer, H., "NAVI: A Proof-of-Concept of a Mobile Navigational Aid for Visually Impaired Based on the Microsoft Kinect", Proceedings of 13th International Conference on Human-Computer Interaction, Part-IV, pp. 584-587, 2011.
- [8] Guy, R.T., and Truong, K.N., "CrossingGuard: Exploring Information Content in Navigation Aids for Visually Impaired Pedestrians", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, Texas, USA, 5-7 May, 2012.
- [9] Ran, L., Helal, A., and Moore, S.E., "Drishti: An Integrated Indoor/Outdoor Blind Navigation System and Service", Proceedings of IEEE 2nd Annual Conference on Pervasive Computing & Communications, pp. 23-30, 2004.
- [10] Ben, L.M., Halley, P., Shashank, B., and Patrick, C., "ioCane: A Smart-Phone and Sensor-Augmented Mobility Aid for the Blind", Computer Science Technical Reports, Paper 1031, 2010.

- [11] Lakde, C.K., and Prasad, P.S., "Microcontroller Based Navigation System for Visually Impaired People", *International Journal of Advanced Research in Computer and Communication Engineering*, Volume 4, No. 8, 2015.
- [12] Nilesh, J., Alai, P., Swapnil, C., and Bendre, M.R., "Voice Based System in Desktop and Mobile Devices for Blind People", *International Journal of Emerging Technology & Advanced Engineering*, Volume 4, No. 2, 2014.
- [13] Cha, J.S., Lim, D.K., and Shin, Y.N., "Design and Implementation of a Voice Based Navigation for Visually Impaired Persons", *International Journal of Bio-Science and Bio-Technology*, Volume 5, No. 3, 2013.
- [14] Kocieliński, D., and Pawłowska, J.B., "Improving the Accessibility of Touch Screen-Based Mobile Devices: Integrating Android-Based Devices and Braille Notetakers", *Proceedings of Federated Conference on Computer Science & Information Systems*, pp. 655-658, 2013.
- [15] Braille Notetakers (Online) Available: <http://www.afb.org/info/braille-technology/5> (Last Accessed: 5th March, 2016).
- [16] Frey, B., Southern, C., and Romero, M., "BrailleTouch: Mobile Texting for the Visually Impaired", *Proceedings of International Conference on Human-Computer Interaction*, 2011.
- [17] Southern, C., Clawson, J., Frey, B., Abowd, G., and Romero, M., "An Evaluation of BrailleTouch: Mobile Touchscreen Text Entry for the Visually Impaired", *Proceedings of 14th International Conference on MobileHCI*, 2012.
- [18] Google Fused Location Provider API (Online) Available: <https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi> (Last Accessed: 5th March, 2016).
- [19] Braille Keys (Online) Available: <http://www1.lvib.org/programs/braille/what-is-braille/> (Last Accessed: 5th March, 2016).
- [20] Rao, B.S., Deepa, K., Prasanth, H., Vivek, S., Kumar, N., Rajendhiran, A., and Saravana, J., "Indoor Navigation System for Visually Impaired Person using GPS", *International Journal of Advanced Engineering Technology*, Volume 3, No. 2, pp.40-43, 2012.
- [21] Renupriya, A., Kamalanathan, C., Kirubakaran, S., and Valarmathy, S., "Naviah: A Smart Electronic Aid for Blind People", *International Journal of Innovative Science, Engineering & Technology*, Volume 1, No. 8, 2014.
- [22] Pielot, M., Poppinga, B., Heuten, W., and Boll, S., "Pocket Navigator: Studying Tactile Navigation Systems", *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, 2012.
- [23] Kammoun, S., Jouffrais, C., Guerreiro, T., Nicolau, H., and Jorge, J., "Guiding Blind People with Haptic Feedback", *Pervasive Workshop on Frontiers in Accessibility for Pervasive Computing*, New Castle, UK, 2012.
- [24] Velázquez, R., "Wearable Assistive Devices for the Blind", *Wearable and Autonomous Biomedical Devices & Systems for Smart Environment: Issues and Characterization*, Chapter-17, pp. 331-349, 2010.
- [25] Arduino Mega 2560 (Online) Available: <https://www.arduino.cc/en/Main/arduinoBoardMega2560> (Last Accessed: 5th March, 2016).
- [26] Vibration Motors (Online) Available: <http://www.protostack.com/other/vibration-motor-10x3mm> (Last Accessed: 5th March, 2016).
- [27] Android Text To Speech (Online) Available: <http://developer.android.com/reference/android/speech/tts/TextToSpeech.html> (Last Accessed: 5th March, 2016).
- [28] Ultrasonic Sensor (Online) Available: http://www.pepperl-fuchs.us/usa/en/classid_182.htm?view=productgroupoverview (Last Accessed: 5th March, 2016).