18-500 ECE Design Capstone Electronic Seeing Eye Dog Final Report

Team: D8
Lam Wing Chan, Ning Guan, and Siying Jin
Electrical and Computer Engineering, Carnegie Mellon University
07 May 2018

I. Project Description

A. Motivations

Visually impaired people have difficulties navigating through places with obstacles. In the past, they rely on white canes to scan the surroundings for obstacles and to inform other participants in the traffic to pay special attentions when passing them. Recently, many visually impaired people start to use guide dogs. These dogs are trained to stop at curbs, steps, and navigate through traffics. Visually impaired people are guided by the dogs through the U-shaped handle attached to the dogs' necks.

Although these guide dogs are reliable for saving visually impaired people from dangers, many places today are not pet-friendly and still do not allow guide dogs to be present. This means visually impaired people lose their navigation ability. In addition to regulations, guide dogs are difficult and costly to train. Lack of professional trainers leads to the shortage of available guide dogs and inconsistency of the dog's navigation performance. Therefore, we intend to create a cheaper and more reliable solution to these issues, a wearable electronic device that performs the same duty as guide dogs. The electronic device will utilize a set of robust ultrasonic distance sensors and a thoroughly tested algorithm to carry out tasks accomplished by traditional tools, but with a higher-quality performance.

B. Goals

The goal of our project is to implement a wearable electronic device that provides both sound and vibrational alerts to visually-impaired users when obstacles are in their paths and within certain ranges. Our product can detect obstacles from five different directions and provide different feedbacks based on the direction of the object. To improve usability, we created a self-contained Android modular application, in which users can configure their heights and preferred alert thresholds, like how far from the object do they want to be alerted. Also, users have an option to choose their preferred form of feedback on Android, either sound, vibrations, or both.

II. Related Work

Current solutions to distance detection typically involves the use of video or image processing for image segmentation tasks. Other methods such as echolocation have been adopted to achieve the task of obstacle detection for visually impaired people. For instance, one research carried out by Lee, Ding & Taft [1] discussed the various performance trade-offs between using image segmentation techniques and audio processing. They attempted to create a depth map using the Tangos PointCloud Data API, and resolves sparsity issues by interpolating missing points to produce a fuller and more informative image for image segmentation process. However, such a complex work flow for video processing resulted in high latency, which caused inefficiency in providing feedbacks to users. In another approach, Lee et al. experimented dilation by ultilizing OpenCVs imgProc library. This method had low latency, but was not robust enough to the varying environments.

In addition to the algorithms and approaches in Lee et al.'s research project, which we found useful when designing our device, we considered the technical difficulties previous researchers encountered in their implementations. Most of the previous researchers abandoned Lucas Kanade's optical flow algorithm for computing object velocities, due to performance issues of Java environment, although the algorithm uses neighboring pixels for computation and dilation to produce results that are less sensitive to noise. We anticipate similar challenges in our design to Lee et al's experiment. In Lee's project, they considered the trade-offs between computation speed, depth of knowledge of the environment, and hardware costs when deciding which algorithm and hardware to use.

Lee's study also informed us potential issues between hardware capability and environmental conditions. In their ecolocation experiment, they found that environment is an important factor for success, because the angle at which the signal arrives the obstacle could lead to unexpected results that require complex calculations involving signal data received from multiple microphones. So in Lee's study, they imposed strict constraints on the environment to ensure that available hardware is sufficient to achieve their desired goals.

Other works we considered when designing our product include the use of detection and tracking methods of echo signals, as proposed by Hou & Wu [2]. In their research, Hou and Wu defined an initial scope of frequency point and an adaptive threshold detection scheme to determine if the data in the frame contains echo signals. The scope of frequency point for detection is continuously tracked and detected. On the other hand, Trump [4] presented a distance detection method that relies upon the pitch delay period in uplink and downlink of speed signals from a mobile device. He experimented modeling pitch estimation error with Laplacian and uniform distributions.

In Leopoldo Angrisani, Aldo Baccigalupi and Rosario Schiano Lo Moriello's [3] experiments, they introduced ultrasonic-based measurements. The distance between the object and the device is calculated using the time delay between the firing of the burst and the detection of the returned echo. Their implementation was built on the Discrete Kalman Filter for estimation of the state of linear stochastic processes. Similarly, Aziz, Mahamad, Mehat & Samiha [5] studied the use of echolocation principle and ultrasonic sensors for assisting the blind to avoid obstacles by detecting objects and triggering sound alerts.

III. Design Trade Studies

We believe our design has an edge because our goal is more specific and narrower, which is to alert visually impaired people if there is any obstacle in their way. Thus, we can improve our device efficiency by leaving out unnecessary features that cause latency. We abandoned the video and image processing approach because of its high latency. We do not think blind people need to know what is the object in front of him, other than the fact that there is an object. Rather, they would want a fast feedback whenever an object is present. Moreover, image processing is not the ideal approach for distance calculation, because finding the distances based on static images is complicated. Comparing to the echolocation approach using microphones, our design with ultrasonic sensors is more immune to environmental noises and more robust regarding signal direction problems.

IV. Design Requirements

Based on the goal of our project, we divide our problem to three parts: obstacle detection, obstacle distance calculation, and feedback. Our product consists of three separate devices wearing in the front of the body, on the left arm, and on the right arm. We will first provide an overview of the components of our device, and then talk about the hardware and software designs of our product.

A. System Design

Multi-direction obstacle detection is an important feature for enhancing user experience. However, multi-direction detection is not featured in any of the current solutions. But visually impaired users need to be wary of objects at different directions, on their side, on the ground, or in the air. Traditional white canes are only useful for detecting objects on the floor, but ineffective for detecting objects in the air, such as tree branches and etc. Therefore, our product is designed to provide more accurate object detection at a wider range.

Besides, most of the existing blind guiding tools which give sound feedback are not suitable for use at indoor areas. Our product seeks to solve this problem by providing different types of feedback for indoor and outdoor navigation. In this section, we will discuss the technical details of our three systems.

B. Obstacle Detection and Distance Calculation Fundamentals

Obstacle detection is achieved by ultrasonic sensors, which send out a sound wave and listen for that wave to bounce back. Figure 1 illustrates the functionality of ultrasonic sensors, in which the black circle is the obstacle reflecting sound waves. The reflected wave then triggers the Arduino echo pin to go high, and the time delay of the sound wave travel is recorded. Distance to the object is calculated from the time delay of the sound wave and the speed of sound with the formula given in Figure 2.

Fig. 1: Ultrasonic Distance Detection

$$distance = \frac{speed\ of\ sound \times time\ taken}{2}$$

Fig. 2: Distance Calculation

C. Product Components

1) Front Body Device Design: In the front system of our device, we used three ultrasonic sensors pointing at different angles, up, forward, and down. Objects at a lower level will trigger the buzzer at a lower frequency, while objects at a higher level will trigger the buzzer at a higher frequency. Objects picked up by the sensor pointing up trigger a sound alert at the frequency of 5000Hz; objects picked up by the sensor pointing forward trigger an alert at 3000Hz; objects picked up by the sensor pointing down trigger an alert at 500Hz. In this way, it is more intuitive for users to process the sound feedback and understand where the object is coming from. Since ultrasonic sensors are very directional, the positioning of our sensors ensures that our device does not only detect objects in one plane.

After some trials and errors, we set the default alert threshold for sensors in the front body device to be 75 centimeters. 75cm is a reasonable range for users to respond. Although a larger range might give users more time to take actions, the feedback system can get really noisy due to more objects' presence given a larger range.

To improve the accuracy of our distance calculation, we used Pythagorean theorem to calculate the actual distance from the user to the object given the input from the sensors pointing at angles. More specifically, as shown in Figure 4, the actual distance from the user to the schoolbag is smaller than the distance measured by the sensor pointing down. The same problem also applies to the sensor pointing up. In order to overcome this overestimation problem in distance detection, we used the Pythagorean theorem to calculate the actual distance with the formula in Figure 3, where h is the user input height and down ratio is default as 0.618, the proportion of the height below the waist of human body. We chose the default value of 0.618 because we suggest the users to wear the front body device at the level of their waist by adjusting the straps. With the step to adjust the distance calculation using Pythagorean theorem, we can ensure the alert consistency among the three sensors.

$$\boldsymbol{Actual\_Distance} = \sqrt{(\boldsymbol{Sensor\_Detected\_Distance})^2 - (\boldsymbol{h * down\_ratio})^2}$$

Fig. 3: Pythagorean Distance Calculation

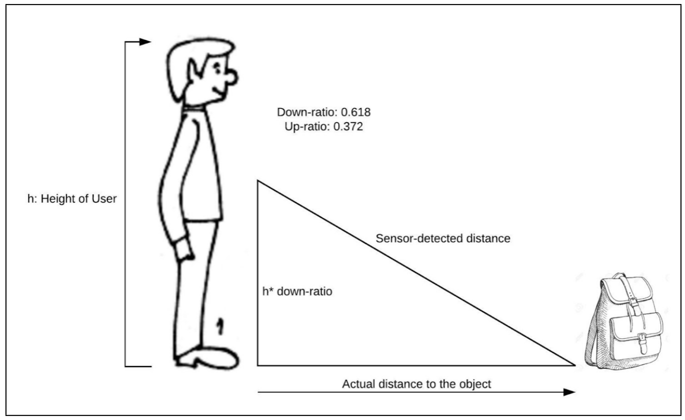

Fig. 4: System Set Up

To make our front body device wearable, we put the circuit into a customized bag and users will put the bag around their necks. We want to ensure the minimal effort required for users to put on our device, because the task of putting on complicated circuit is more challenging for our visually impaired users. Figure 5 is the look of our final front body device.

Fig. 5: Finished Front Body Device

2) Arm Device Design: The devices wearing on the left and right arms employ a similar design to the device wearing at the front. However, the device wearing on the arms has only one sensor pointing forward and one vibration motor which alerts users if any object is within 10cm on their side.

Figure 6 shows the circuit diagram of the arm devices.

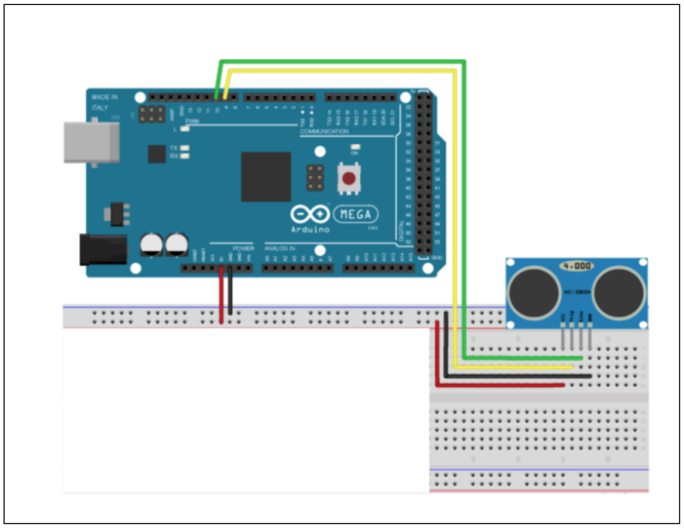

Fig. 6: Circuit Design

For the left and right arm devices, we used smaller breadboards for portability purposes. We customized sports armbands to hold the circuits. Figure 7 is the final look of our arm devices.

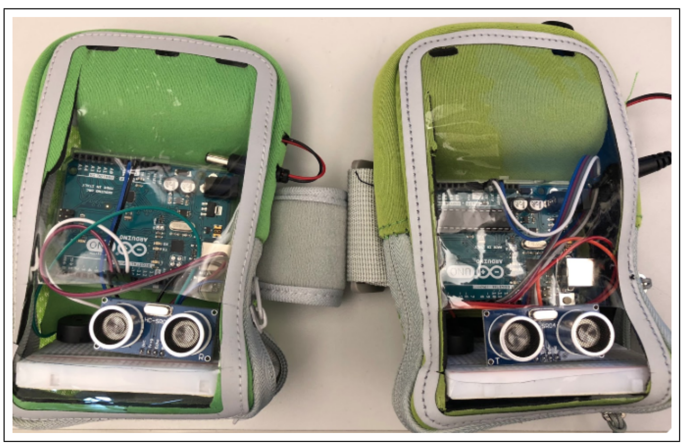

Fig. 7: Final Arms Devices

To make our devices portable, sturdy and stable, we used 9V batteries to power the Arduinos instead of laptops, and used ribbon wires to replace the traditional loose wires. We also customized the armbands and the bag to make plugging and unplugging battery power sources easy for users.

3) Android App Design: The buzzers are helpful for outdoor navigation. They are loud enough to be audible and also give pedestrians warnings. In indoor areas such as a quiet study area, however, the buzzers are too disturbing. Therefore, we designed an Android App to provide indoor navigation. The user is supposed to wear earphones to get the sound feedback from Android. Figure 8 shows a user interface of our Android App.

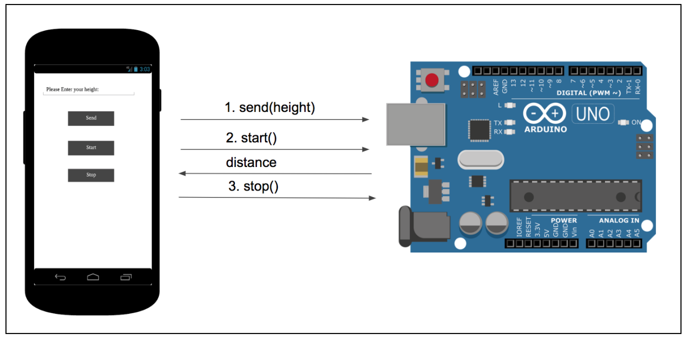

Fig. 8: System Overview

1. The SEND button is used to send user input height to the Arduino. 2. The START button is to start communication and receive distance data from the Arduino. If the received distance is within the threshold, beep sound and vibration will be played every 30ms using ToneGenerator and System Vibrator. 3. The STOP button is to stop the communication and clear everything in the buffer.

D. Feedback

Each of the three devices has its independent feedback system. To determine the default alert thresholds for each of our device, we tried various thresholds in different environments. For instance, we tested our front body device in a laboratory and a clear hallway with obstacles such as cardboards, walls, and boxes. We find that if the threshold is too large, such as 150cm, many objects will fall into the range, including those not in the user's path, and thus feedbacks are no longer reliable. On the other hand, if the threshold is too small, such as 40cm, users are not given enough time to respond to the obstacles. For the devices wearing on the arms, we set the alert threshold to be much smaller because, if in narrow hallways, the device will keep detecting the wall. Moreover, objects on the side should be less important than objects at front, the direction where the user is moving. After some trials and errors, we determine that 75cm and 10cm are reasonable thresholds for the front body system and arm systems.

We provide both sound and vibration feedbacks to our users. We decided to implement vibration feedbacks because visually impaired people are usually busy with other sounds in the environment as well. For example, they rely on their ears to listen to the traffic. Thus, we believe that an alternative form of feedback instead of sound would be helpful for our users' convenience. Vibration feedbacks also reduce the possibility of confusion of having too many different sounds.

In order to provide sound feedback, we will use a electronic buzzer, which has the alarm goes off if the distance between the user and the device is below a threshold as discussed. On the other hand, vibrational feedbacks will be produced by commercial vibration motors. Vibrational feedbacks would be produced when the distance between is within a particular threshold. Notice that this threshold is adjustable, which motivates us to integrate these functionalities on an Android device, granting users access to adjustable parameters for better user experience.

V. Architecture

A. Functional Architecture Overview

We will have 3 functional systems: Object Detection, Distance Computation and Feedback. A block diagram of architecture is provided in Fig. 9.

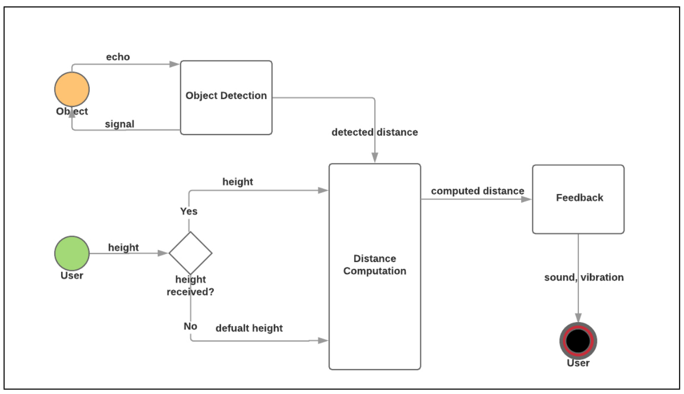

Fig. 9: Functional Architecture Overview

B. Workflow

1. User sends height for Distance Computation.
2. Distance Computation gets the height for Pythagorean distance calculation or use default height in there is not input.
3. Object detection utilizes echolocation;
4. Detected object distance is sent to Distance Computation system.
5. Distance Computation calculates distance based on height and detected distance.
6. Distance Computation sends result distance to Feedback System.
7. Feedback system provides sound and vibration if distance is within threshold.

VI. System description/depiction

A. System Overview

We utilize UDS and Arduino for Object Detection and Distance Computation. Feedback is generated using vibration motors, buzzers and Android App. Fig. 10 provides an overview of our system block diagram.

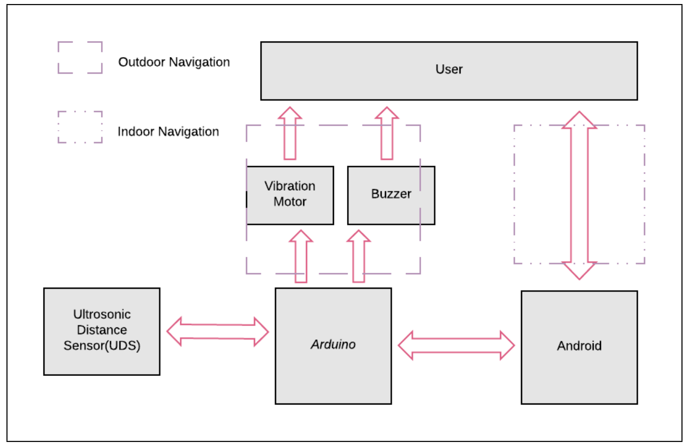

Fig. 10: System Overview

1. User enters height on Android App.
2. Android App sends height to Arduino.
3. UDS sends echoes to Arduino.
4. Arduino calculates distance of the object based on echoes and the input height.
5. Arduino drives indoor and outdoor navigation feedback systems.

B. Indoor Navigation Feedback

1. Arduino sends computed distance to Android device via serial port.
2. Android App provides beep sound and vibration to the user.

C. Outdoor Navigation Feedback

1. Arduino triggers Vibration Motors and Buzzers if distance is within threshold.
2. Vibration Motors and Buzzers provide vibration and sound to the user.

VII. Alpha Testing

A. Target Specification

Our goal is to keep the sensor output distance within 10% error from the actual hand-measured distance.

Since our device is based on ultrasonic sensor distance detection, we looked into the factors that impact ultrasonic sensor accuracy. According to a study by Nicolau et al. [12], ’the sound speed is affected by changes in the air properties, such as temperature, pressure, humidity, gas composition, and air turbulence. It has direct influence on the accuracy of distance measuring and the sound wavelength’ [p. 145]. So ultrasonic sensor performance is environment dependent.

We thus decided to test our product both indoors and outdoors. We tested the front body device and the arm device separately, since these devices have different alert thresholds. For each device, we did 15 trials indoors and 15 trials outdoors and recorded the error for each trial. The details of trial results are attached in Appendix A.

B. Testing Methodology

For each of our trials, we followed the steps below:

1) Hold our device in front of an object
2) Slowly move towards the object
3) Stop immediately when the alert goes off
4) Hand measure the distance between our device and the object
5) Record the difference (error) from the threshold value, and calculate error percentage by dividing the difference by the threshold
6) Repeat the same process

C. Testing Results

To visualize our testing results, we plot the errors in absolute values for the 30 trials of the front body device in Figure 11. We plot the errors in absolute values for the 30 trials of the arm device in Figure 12.

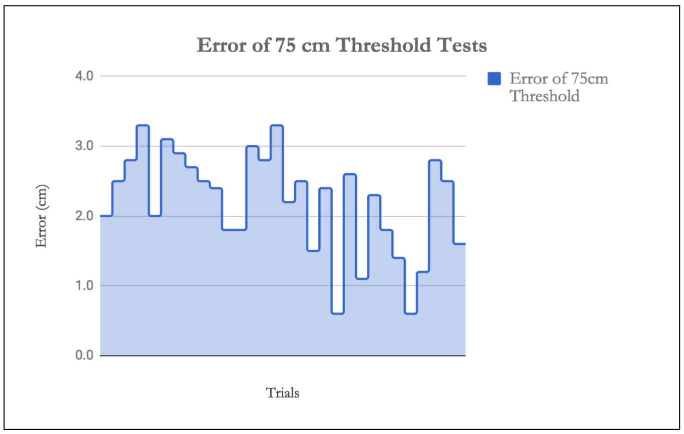

Fig. 11: Front Body Device Error in Absolute Value (75cm Threshold)

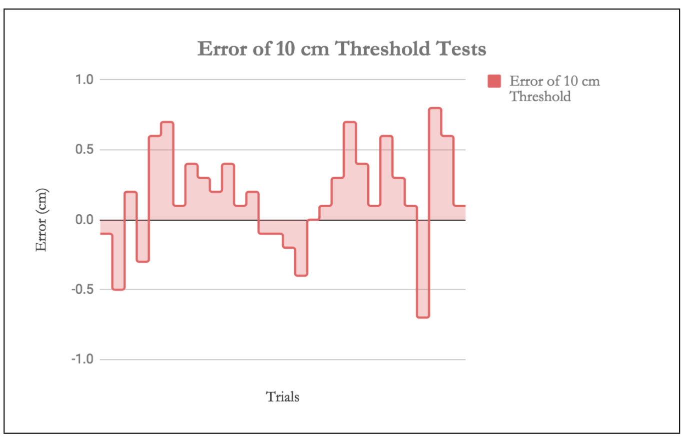

Fig. 12: Arm Device Error in Absolute Value (10cm Threshold)

To compare the error percentage of both devices to our target specification, we calculated the error percentage by subtracting the threshold from the hand-measured distance in each trial and dividing the difference by the threshold. The error percentage for each of the 30 trials of the front body device is plotted in Figure 13. The plot for the arm device is in Figure 14.

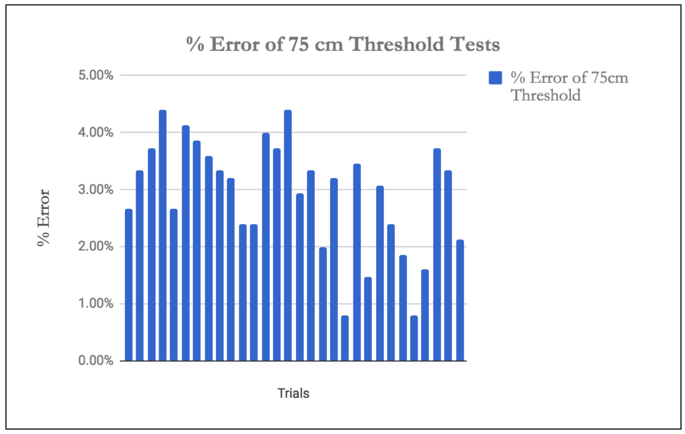

Fig. 13: Front Body Device Error Percentage

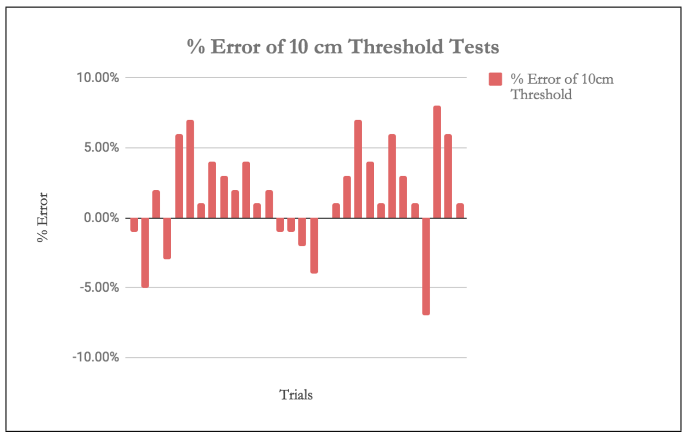

Fig. 14: Arm Device Error Percentage

In our front body device, the errors are all within 5% across 30 trials, as shown in Figure 13. In our arm device, the errors are all within 10% across the 30 trials, as in Figure 14. Thus, both devices have met our target specification of maximum 10% error.

When plotting indoor trial results and outdoor trial results on the same graph, from Figure 11 to Figure 14, we did not see significant differences between sensor performance indoors and outdoors. A potential reason might be that our indoor and outdoor testing environments did not differ a lot on temperature, humidity, air turbulence and etc. But on the other hand, this shows that our product would perform consistently given small variations of environment.

VIII. Beta Testing

A. Procedure

After alpha testing and seeing consistent and reliable performance of our product, we visited CLP - Library for the Blind & Physically Handicapped in Oakland, Pittsburgh for beta testing.

We met with Don, the library branch manager, and Ross, a visually impaired library staff member. Ross tried on our product to approach objects placed in different directions, such as chairs, walls, and door frames, and listen for alerts.

Fig. 15: Beta Testing Day

Fig. 16: Beta Testing with Ross

B. User Feedback

In our conversation with Ross, he told us that he once had an accident which he ran into a tree branch and scratched his face while he was using a cane to navigate on the street. He appreciated that we placed a sensor pointing up to the air to detect floating objects.

On another note, initially, our arm devices provided sound feedbacks instead of vibrations. However, Ross brought up that our alerts had too many sounds, because we used 5 different frequencies for objects detected at 5 different directions. He suggested that we change the sound alert on arm devices to vibrations, because he cares less about objects on his side than about objects in the front. With fewer sound alerts, he can focus more on listening to the traffic. Given Ross's advice, we changed the sound feedback on arms to vibrations. So in our final product, alerts on arms are vibrations for objects detected on user's side.

C. Overall Evaluation

Ross, our beta tester, and Don who witnessed our entire testing process both offered positive feedbacks on our product.

Ross: *'I like how the low frequency is down low and high frequency is up high. Its much more intuitive that way'*.

Don: *'I think your concept and the level to which youve developed the technology are much further along than some of the initial concepts weve seen before'*.

IX. Project management

A. Schedule

- Jan 18
  - ☑ Discussed each team members background and preferences

  - ☑ Narrowed down project area to Signals and Systems + Circuits + Software System

- Jan 25
  - ☑ Brainstormed a few possible projects and talked about how they can solve or improve current solutions
  - ☑ Decided that each member researches on three potential project topics

- Feb 1
  - ☑ Brought together nine different possible projects and presented to each other the unique / challenging parts of these project ideas
  - ☑ Discussed improvements on these projects we think we will be able to achieve

- Feb 8
  - ☑ Met up with different professors to discuss basic requirements of this class
  - ☑ Narrowed down to working on a project that seeks to help visually impaired people to navigate using ultra-sonic sensors

- Feb 15
  - ☑ Decided a few broad features i.e. distance sensing, software integration and feedback to implement
  - ☑ Studied a few related works and read through a few past project final reports to understand basic requirements
  - ☑ Listed out the parts we will need to purchase and how they will be connected together

- Feb 22
  - ☑ Researched methods of using Ultrasonic sensors and raspberry pi or Arduino
  - ☑ Evaluated the feasibility of implementing different features on an Android Device
  - ☑ Sent out purchase form and decided on each members main responsibility when parts come in

- Mar 1
  - ☑ Evaluated risks for using the device (i.e. robustness to noise in environment and accuracy of sensors)
  - ☑ Brainstormed other possibilities of feedback such as vibrations / lights
  - ☑ Decided on various ways to test our device and set up acceptable standards

- Mar 8
  - ☑ Software
    - Installed Android Studio and learned to build an Android App from tutorial
  - ☑ Hardware
    - Researched existing Android APK (Application Package Kit) online for ultrasonic object detection package

- Mar 15
  - ☑ Software
    - Made the user interface with four buttons
    - Researched existing Android/Arduino connection methods
  - ☑ Hardware
    - Modified code to take into account user input data, such as the user height (for determining sensor angle purpose) and the preferred alert threshold
    - Tested the functionality of our program by walking in a clear hallway with eyes covered, proposed and documented improvements

- Mar 29
  - ☑ Software
    - Did research on existing libraries for Android/Arduino connection communication;
  - ☑ Hardware
    - Modified the algorithm to calculate the distance to objects not on the same horizontal plane using Pythagorean theorem.
    - Added two more sensors for the horizontal plane detection and modified the algorithm to incorporate the changes

- April 5
  - ☑ Software
    - Enabled Android/Arduino connection with serial data transmission via USB cable
  - ☑ Hardware
    - Added different tones for alert from different directions
    - Improved Arduino power source by switching to portable batteries and ordered 9V rechargeable batteries

- April 12
  - ☑ Software
    - Buffered received data and enabled Android users to send their height to the Arduino
  - ☑ Hardware
    - Improved user experience by separating horizontal plane sensors to two separate boards, so that they can mount on users' left and right arms separately
    - Connected batteries to Arduinos to replace cable power source

- April 19
  - ☑ Software
    - Enabled sound and vibration feedback on Android

App

- ☑ Hardware
  - Brainstormed improvements on circuit layout to incorporate buzzers and vibration motors
  - Researched Android-Arduino bluetooth technology and studied its feasibility

- April 26
  - ☑ Software
    - Tested the app with different threshold;
    - Redesigned the user interface and type of sound based on usability testing
  - ☑ Hardware
    - Made the device wearable by customizing armbands and a purse to carry the hardware parts
    - Tested and determined effective thresholds for both side sensors and front sensors

- May 3
  - ☑ Software
    - Added comments and user manual for the Android App
  - ☑ Hardware
    - Visited Carnegie Library for the Blind to beta test our project with a visually-impaired staff member
    - Modified arduino code to drive vibration motors instead of buzzers for systems worn on left and right arms
    - Researched methods to reduce feedback noise by checking that the object has to be present for at least a certain amount of time before the system gives feedback. For example, if people wave hands in front of the sensor, it should not trigger the alert.

B. *Team Member Responsibilities*

The primary responsibility of Lam Wing Chan was on ultrasonic sensors and the vibrational feedback system. Her secondary responsibility was on code development and system integration.

The primary responsibility of Ning Guan was on hardware and testing. Ning was responsible for writing, modifying and debugging Arduino code, and designing alpha and beta testing methods. She also worked on designing and building circuits, developing the wearable product, and carrying out testings.

The primary responsibility of Siying Jin was on Android App design and implementations.

C. *Budget*

Ultrasonic Sensor Package (x2)	$19.98
Vibration Motors	$7.53
Reverse Backup Radar System	$32.99
Micro USB to USB cable	$6.99
Ribbon Wires	$7.86
Small Breadboards	$10.99
9V Batteries	$9.99
Battery Clips	$5.99
Rechargeable Batteries and Charger	$29.99
Sports Armbands (x2)	$18.98
Purse	$8.99
Transparent Coat	$18.99
TOTAL	**$179.27**

Fig. 17: Budget

D. *Risk Management*

Hardware Failure: To mitigate the risk of hardware failure, we always ordered more parts than the number we intend to use in our design. For example, we had 5 ultrasonic sensors in our design and final product, but we ordered 10 in case that some sensors will be broken. In fact, we broke 2 sensors in our soldering practice, so the backup sensors were extremely useful.

Connection: At first, we used bluetooth for Android/Arduino communication, but the connection was sometimes unstable and caused data loss. Therefore, we replaced bluetooth with OTG cable and read data from serial port.

APPENDIX A

...

	75cm Sensor	10cm Sensor	Error of 75cm Threshold	Error of 10 cm Threshold	% Error of 75cm Threshold	% Error of 10cm Threshold
1	77.0	9.9	2.0	-0.1	2.67%	-1.00%
2	77.5	9.5	2.5	-0.5	3.33%	-5.00%
3	77.8	10.2	2.8	0.2	3.73%	2.00%
4	78.3	9.7	3.3	-0.3	4.40%	-3.00%
5	77.0	10.6	2.0	0.6	2.67%	6.00%
6	78.1	10.7	3.1	0.7	4.13%	7.00%
7	77.9	10.1	2.9	0.1	3.87%	1.00%
8	77.7	10.4	2.7	0.4	3.60%	4.00%
9	77.5	10.3	2.5	0.3	3.33%	3.00%
10	77.4	10.2	2.4	0.2	3.20%	2.00%
11	77.2	10.4	1.8	0.4	2.40%	4.00%
12	76.8	10.1	1.8	0.1	2.40%	1.00%
13	78.0	10.2	3.0	0.2	4.00%	2.00%
14	77.8	9.9	2.8	-0.1	3.73%	-1.00%
15	78.3	9.9	3.3	-0.1	4.40%	-1.00%

Fig. 18: Outdoor Trials

	75cm Sensor	10cm Sensor	Error of 75cm Threshold	Error of 10 cm Threshold	% Error of 75cm Threshold	% Error of 10cm Threshold
16	77.2	9.8	2.2	-0.2	2.93%	-2.00%
17	77.5	9.6	2.5	-0.4	3.33%	-4.00%
18	76.5	10.0	1.5	0.0	2.00%	0.00%
19	77.4	10.1	2.4	0.1	3.20%	1.00%
20	75.6	10.3	0.6	0.3	0.80%	3.00%
21	77.6	10.7	2.6	0.7	3.47%	7.00%
22	76.1	10.4	1.1	0.4	1.47%	4.00%
23	77.3	10.1	2.3	0.1	3.07%	1.00%
24	76.8	10.6	1.8	0.6	2.40%	6.00%
25	76.4	10.3	1.4	0.3	1.87%	3.00%
26	75.6	10.1	0.6	0.1	0.80%	1.00%
27	76.2	9.3	1.2	-0.7	1.60%	-7.00%
28	77.8	10.8	2.8	0.8	3.73%	8.00%
29	77.5	10.6	2.5	0.6	3.33%	6.00%
30	76.6	10.1	1.6	0.1	2.13%	1.00%

Fig. 19: Indoor Trials

References

[1] Annabelle Lee, Wanqiao Ding, and Elliott Taft *Electronic Mobility Device for the Visually Impaired* Carnegie Mellon University 18-551 Spring 2017 Final Project

[2] S.L. Hou and S.L. Wu, *Detection and tracking method of echo signal for miss-distance measurement* 2009 IET International Radar Conference

[3] Leopoldo Angrisani, Aldo Baccigalupi and Rosario Schiano Lo Moriello (2009). *Ultrasonic-Based Distance Measurement Through Discrete Extended Kalman Filter, Kalman Filter Recent Advances and Applications,* Victor M. Moreno and Alberto Pigazo (Ed.), ISBN: 978-953-307-000-1, InTech

[4] Tonu Trump, *Detection of echo generated in mobile phones using pitch distance* 2008 16th European Signal Processing Conference

[5] Izzatdin Abdul Aziz and Sajee Mahamad andMazlina Mehat and Nazleeni Samiha *Blind echolocation using ultrasonic sensors* 2008 International Symposium on Information Technology

[6] C. J. Baker and G. E. Smith *Aspects of cognition and echolocation* Proceedings of the 2012 IEEE International Symposium on Antennas and Propagation

[7] Jascha Sohl-Dickstein ; Santani Teng ; Benjamin M. Gaub ; Chris C. Rodgers ; Crystal Li ; Michael R. DeWeese ; Nicol S. Harper *A Device for Human Ultrasonic Echolocation* IEEE Transactions on Biomedical Engineering

[8] Lucas Laursen *Echolocation by Smartphone Possible* IEEE Spectrum 2010

[9] D.Y.K. Sampath and G.D.S.P. Wimalarathne *Obstacle classification through acoustic echolocation* 2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF)

[10] Akhilesh Krishnan ; G Deepakraj ; N Nishanth and K. M. Anandkumar *Autonomous walking stick for the blind using echolocation and image processing* 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)

[11] Jong Hwan Lim ; J.J. Leonard *Mobile robot relocation from echolocation constraints* IEEE Transactions on Pattern Analysis and Machine Intelligence

[12] Viorel Nicolau, Constantin Miholca, and Mihaela Andrei *Fuzzy rules of sound speed influence on ultrasonic sensing in outdoor environments*, 2009, pp. 51-74.