The Bat Belt

Authors: Alex Lin, Kelton Zhang, Ning Cao
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—The Bat Belt is a smart sensing and feedback system aimed to help the visually impaired to move around obstacles. The design goal of this project is to achieve a better affordability and functionality balance between the cane and guide dog with a lightweight wearable that provides obstacle detection with intuitively actionable haptic feedback. A highlight feature of the system is that both ground-level and eye-level detection will be available through a depth camera and ultrasonic sensor array respectively. We expect our product to dramatically outperform the typical white cane in the overall perception of the user's environment, while being as smart as and much more low-maintenance than the guide dog.**

***Index Terms*—Computer Vision, Embedded Systems, Haptic Feedback, Ultrasonic Sensor, Wearable Device**

1 INTRODUCTION

This project aims to address a real pain point in the society at large through a technical solution that utilizes both hardware and software sides of our team's skill set. Following this thinking, we envisioned The Bat Belt, a smart sensing belt that gives real-time haptic feedback to the visually blind to avoid collision with obstacles. The two most adopted solutions, guide dog and cane, are either high maintenance or limited in functionality as guide dogs take extensive amount of training beforehand and care throughout its lifetime while the cane only offers ground level tactile feedback for contact points within a small range. Emerging technical products also each have its shortcomings, e.g. cumbersome and conspicuous neckwear, minesweeper-like cane bounded by the direction it is facing. Given the status quo, our project is designed to fill in the gap with a lightweight wearable that provides both ground-level and eye-level detection with intuitively actionable feedback. The implementation will be based on a off-the-shelf belt bundled with depth camera, ultrasonic sensor array, a set of vibration coin motors and Raspberry Pi 4 and Arduino Uno boards (Figure 1) to drive computation. To measure the engineering outcome of The Bat Belt, we will test the prototype indoor against a concrete set of metrics that includes detection accuracy, sensor range, feedback user testing etc as specified in the requirements section.

2 USE CASE REQUIREMENTS

From the rudimentary product definition of a wearable belt that alerts the visually impaired user with haptic feedback to avoid colliding with obstacles, we further develop the following qualitative use case requirements to guide the design and testing process :

1. Lightweight : should support a full day of use without fatigue
2. Reliable detection : should give confident product warnings within a reasonable range
3. Long Battery life : should support a full day's movement after one battery charge
4. Relatively low-cost : should be affordable compared to other options
5. Intuitively actionable real-time feedback : should give the user enough and as much as possible time to react with the right move

These qualitative use case requirements are specified with quantitative metrics in the design requirements section.

3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

Our system can be divided to four subsystems: sensing, computation, feedback, and power supply. For the sensing subsystem, we use a depth camera for ground-level sensing and an array of ultrasonic sensors for above-ground sensing; data collected from the sensors are either sent to an Arduino Uno, where they are processed into more comprehensive data, before being sent to the Raspberry Pi, or directly sent to the Raspberry Pi . The Raspberry Pi models the surrounding, classifying objects and rating their threat level, and send feedback commands to the Arduino Uno, which drives the vibration motors in the feedback subsystem. The power supply is connected to the Raspberry pi, which in turn powers all other subsystem; our power supply subsystem is versatile enough to provide the depth camera with direct power supply if necessary.

A block diagram of our system is shown in Fig. 1.

A physical image of our belt is shown in Fig. 2.

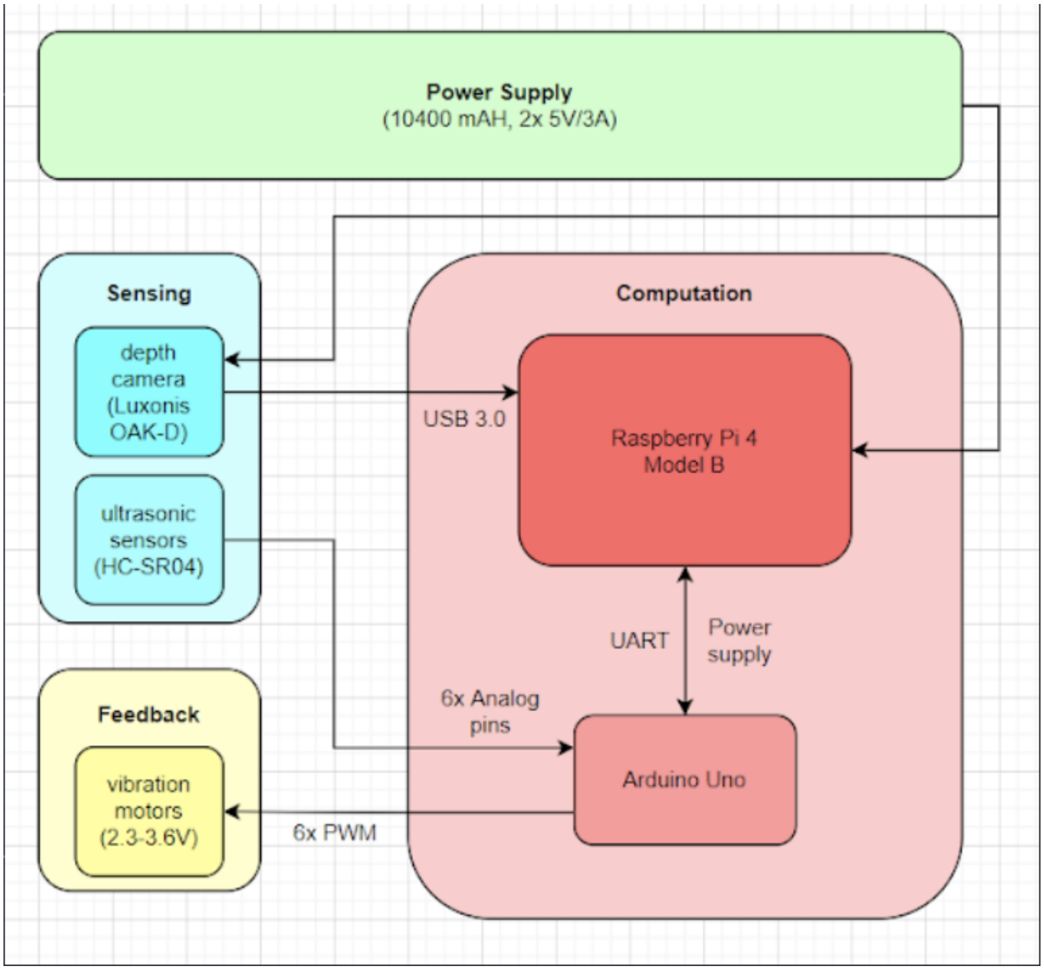

Figure 1: System Block Diagram.

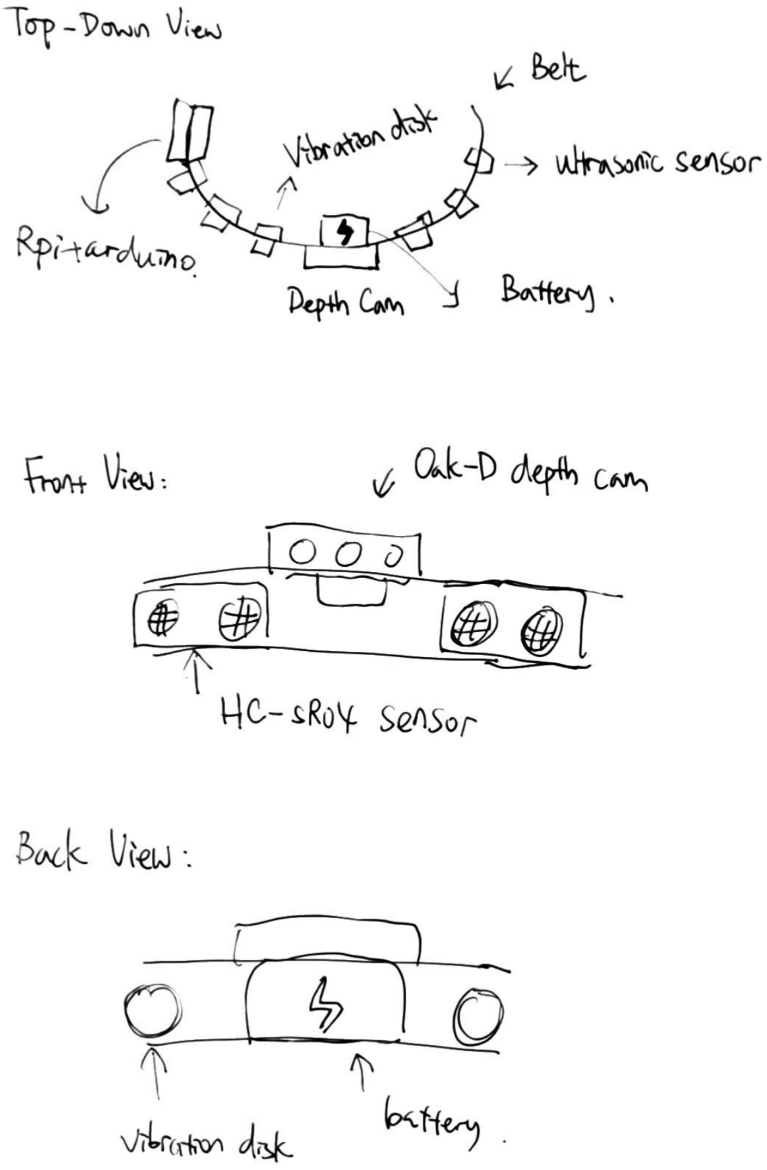

Figure 2: Prototype Sketch

4 DESIGN REQUIREMENTS

Based on the use case of alerting the visually impaired with real-time actionable feedback on a wearable form factor, our quantitative design requirements are

1. lightweight : < 1 kilogram
2. reliable detection : False positive < 10%, false negative < 5% within 5 meters
3. real-time actionable feedback : system response time from getting sensor signal to feedback within 0.1 second; sensible action within 1 second after feedback more than half of the time during blindfold user testing

Each of these requirements' quantitative constraint is placed by our hardware of choice and expected performance of our software.

In terms of weight, our hardware components besides the belt include Oak-D depth camera (115g), Arduino Uno (25g), Raspberry Pi (46g), 6 HC-SR04 ultrasonic sensors (6x8.5g), 6 vibration coin motors (6x1g) and 1-2 Charmest portable battery (2x187g) which add up to 617g. With the belt, it is reasonable to set the weight limit of the system to 1kg.

Regarding reliable detection, the scope is set to within 5 meters given our choices of ultrasonic sensor and depth camera. The ultrasonic sensor has been tested to satisfy a range of 5 meters. The depth camera, on the other hand, has a maximum perceivable distance of 35 meters and vertical POV of 49°, which when oriented downward at a certain angle, will cover the same 5 meter area. The accuracy of ground-level obstacle detection will depend on the computer vision algorithm or image classification model running on the depth camera and is yet to be tested. In the event that its accuracy does not meet our requirement, the depth camera module can be made optional. The field of view of the detection is not specified as it is different for ultrasonic sensing (150° horizontally with 6 sensors, subject to change with interference between echos of the sensors) and depth camera (72° horizontally).

To achieve real-time actionable feedback, we quantified the real-time part by requiring the latency from sensing to feedback to be less than 0.1 second and the actionable part by measuring whether users can make the right action within 1 second given feedback for more than half of the time during testing. **??**

5 DESIGN TRADEOFF STUDIES

To control the cost of our product while satisfying the design requirements in Section 4, it is crucial that we carefully examine each subsystem. In this section we will discuss the trade-offs when we design the different subsystems, as show in Fig. 1.

5.1 Sensing

Due to the different nature of above-ground sensing and ground-level sensing, we will discuss the trade-offs in two separate subsections.

5.1.1 Above-ground sensing

Unlike navigation of automatic vehicles, which typically have their sensors rotating on the top of the vehicle bodies, Our design adopts multiple sensors facing slightly different directions for better coverage. Therefore, the key specifications we care about are measuring angle, maximum detection distance, and precision. Table 2 shows the 3 different distance sensors we have taken into consideration and their specifications. Comparing these data against the design requirements in Section 4, we chose the HC-SR04 ultrasonic sensors as our distance sensors.

5.1.2 Ground-level sensing

Compared to above-ground sensing, ground-level sensing have the following properties:

- Increased complexity. A change in the terrain, like potholes in the road, ramps, or stairs upward/downward, requires higher resolution and better classification methods.
- Lower Relative speed. Typical human walk speed is 5km/h or 1.4m/s, and ground-level threats are assumed to be mostly stationary.
- Smaller field of view required. Users are expected to walk mostly forward, so a smaller area in front of the user is sufficient.

Based on these properties, we have chosen the Luxonis OAK-D depth camera, with a 70°horizontal and 50°field of view, 720P resolution, and individual Intel MyriadX chip for calculation of classification algorithms.

It should be noted that the inclusion of a depth camera with independent processing units places a great stress on power supply. We will discuss this in detail in Section 5.3.

5.2 Signal Processing & Computation

Limited by the form factor of wearable devices, we need a portable computational device with sufficient performance. We have adopted a Raspberry Pi 4 Model B with 2GB RAM for the computation, partly because it is readily available in the inventory. If in later stages of development we find this model We have also considered using a Raspberry Pi Pico, which is smaller and less power hungry, but we eventually gave up because it does not support the communication interface and power supplies we need for various devices, especially the depth camera (see Section 5.1.2). However, the Raspberry Pi 4 is much more power hungry than a Raspberry Pi Pico. We will discuss this in detail in Section 5.3.

Table 1: Distance sensors

Sensor type	Model #	Measuring angle	Max distance	Precision	Frequency	Power	Cost
ultrasonic	HC-SR04	30°	4.5m	3mm	50Hz	10mW	$3
infrared	HiLetGo	35°	30cm	-	-	-	$0.9
LiDAR	TF-Luna	2°	10m	1cm	250Hz	350mW	$26

In order to better interpret signals from the ultrasonic sensors and drive vibration motors, We have adopted an Arduino Uno as a motor driver and an interface for signal processing. There are sufficient GPIO pins available on the board, so purchasing a more powerful board like Arduino Mega would be unnecessary.

5.3 Power Supply

As briefly discussed in Section 5.1.2 and 5.2, the depth camera and Raspberry Pi are the main consumer of power. The typical working current of different parts of our system is shown in Table **??**. Given the battery life requirements in Section 4, we would need a battery of

$$((2 \times 6 + 900 + 50) \times 0.4 + 800) mA \times 10h = 11,848 mAh.$$

Combined with cost factor and the need of $5V/3A$ power supply for the Raspberry Pi, we decided to choose the Charmast Smallest USB-C Portable Charger with its 10,400 mAh battery and 2 5V/3A output.

5.4 Haptic Feedback

Weight and form factor are the most important aspects of vibration motors. After researching some widely used vibration motors, we decide to use the linear resonant actuators (LRAs) because of their small form factor, light weight, and short response time. More details about vibration motors will be discussed in Section 6.4.

6 SYSTEM IMPLEMENTATION

Again please use the guidelines on Canvas and the Word template for what to include in this section. This section should be complementary in content with the Architecture section 3 rather than redundant. You can refer back to earlier figures in section 3 using Fig. **??** and Fig. **??**.

There should be a subsection for each of the subsystems as shown below.

6.1 Sensing

6.1.1 Above-ground Sensing

We have discussed in detail why we choose to use ultrasonic sensors in Section 5.1.1. In this section we will discuss how we integrate these sensors into our system.

We plan to align an array of ultrasonic sensors in different directions to cover a total of 180°of area in front of the user. Directions of adjacent ultrasonic sensors will differ by approximately 30°apart to fully utilize the measuring angle of the sensors while minimizing the blind area. To avoid sensors interfering with each other, we choose to have only 2 sensors (e.g. 1/4, 2/5, 3/6) firing ultrasonic waves simultaneously, so we minimize sensor interference while keeping our detect frequency above 10Hz.

6.1.2 Ground-level Sensing

As briefly discussed in Section 5.1.2, we plan to use a Luxonis OAK-D depth camera for more sophisticated ground-level sensing. We will position the depth camera slightly downward (about 35°downward from the horizontal line), so it can capture more details closer to the user.

The depth camera can generate a mapping of distance information in front of the user and update at a frequency of 60 Hz. From the mapping we can easily detect obstacles by comparing the depth mapping to an "ideal" model in which we assume the ground in front of the user is flat.

If there is extra time after the implementation of MVP is complete, we might add classification algorithms to ground-level sensing to provide more accurate predictions.

6.2 Signal Processing & Computation

We use a Raspberry Pi 4 Model B with 2GB RAM as our source of computation. As shown in Fig. 1, the Raspberry Pi communicates with an Arduino Uno, receives readings from the ultrasonic sensors and the depth camera, and sends commands to the vibration motors.

The General Algorithm running on the Raspberry Pi is as follows: For each cycle,

- Collect data from ultrasonic sensors and the depth camera,
- Identify obstacles by comparing to "ideal" models,
- Rate "threat level" of each identified obstacle based on their distance and speed, and
- Send commands to the vibration motors.

The Arduino serves as an ADC for ultrasonic sensors and driver of the vibration motors. It is responsible for controlling the ultrasonic sensor array (briefly discussed in Section 6.1.1), translating the pulse signal of ultrasonic sensors into distance data, and controlling the voltage output to the vibration motors through its PWM pins. It communicates with the Raspberry Pi through USB ports, using UART protocol with a Baud rate of 9,600.

Table 2: Distance sensors

Parts	Working Current (typ. value, mA)	Expected Uptime ratio
ultrasonic sensors	$2 \times 5 = 10$	40%
depth camera	900	40%
Arduino	50	40%
Raspberry Pi	800	100%

6.3 Power Supply

We have adopted a (power bank name) with 10,400 mAh capacity and 2x 5V/3A output. Our current implementation plan is to power all other subsystems through the Raspberry Pi; however, we do notice that the OAK-D depth camera is also quite power hungry, so if the current design proved to be beyond the power supply capacity of the Raspberry Pi, we will directly power the depth camera with the power bank.

In the worst-case scenario where one battery is unable to supply the entire system, we would need to purchase another battery, which would guarantee a sufficient power supply but add significantly to the weight of the system.

6.4 Haptic Feedback

As briefly discussed in Section 6.2, we use 6 LRA coin motors as the feedback system. By controlling the output voltage through PWM pins, we can control the intensity of vibration from weakest (2.0V) to strongest (3.6V). Currently, we plan to implement 3 levels of intensity to signal the user of the threat of certain obstacles.

7 TEST & VALIDATION

For testing, we will mainly be conducting two classes of tests. The first class is verifying the design requirements. This includes verifying that our sensors are able to react to objects within a 4 meter radius with acceptable accuracy, verifying our product has the specified battery life, and measuring weight of our product etc. For this class of tests, the main idea is to provide the system with controlled input and check if the output is as desired.

The second class is verifying our product's performance in the desired use-case, making sure that it will indeed help blind user's avoid obstacles while walking. For this test, we plan to create a controlled environment that simulates the minimal viable product scenario, and evaluate our belt's performance based on pre-determined metrics. A detailed explanation will be provided in the sections below.

7.1 Tests for Battery Life

While we have calculated the expected battery life, we want to be able to test how well our product performs with the entire system intact. For this test, we will test the battery life in two modes. For mode 1, we will internally set the system to turn on depth camera, all six sensors, and operate all six vibration unit at all times. This is the maximum power consumption for the system, and while in real life scenario this will likely not occur, we still want to get an estimate of the worst case scenario. In this test, we will simply lie down our belt and monitor the state of our battery(which comes with an estimate of how much power is left), until the system stops working. For mode 2, we will operate the belt in a close to real life scenario, having a belt marathon where each member will wear the belt for about three hours, and walk around in daily life until the system runs out of battery. This is rather achievable as the expected lifetime is at most 10 hours, and we would be able to get an estimate of real life operating time. If the real life battery life satisfy the benchmark(8h) we would conclude that our system is appropriate in terms of battery life. The worst case battery life will be used as a reference, to make sure our system is still somewhat reliable in extreme cases.

7.2 Tests for Sensor Detection Range and Reliability

For sensor detection range, we want to make sure that the sensor components we use have desired accuracy and range when it comes to detecting obstacles. To conduct this test, we will place the belt on a flat surface with sensors off. Then we will conduct the test on 3 different obstacles, a small obstacle that is around $10cm \times 10cm$, a medium obstacle that is around $20cm \times 20cm$, and a large obstacle that is around $40cm \times 40cm$. We will place the obstacle from 6 different directions to the belt in an 180 degree radius, and at 3 different distances, 1 meter, 2 meter, and 4 meters. Each time after placing the obstacle, we would read the system readings from the belt to know where the belt think the obstacle is, or if the belt ignores the obstacle. Using this method, we would be able to produce 3 detection maps for obstacles of three different sizes, where each dot on the map indicates the location of the obstacle, and the error of the belt for detecting the obstacle. For example, if an obstacle is placed at a location with measured distance 4 meters, and the belt detects the obstacle at distance 3.5 meters, the error would be marked as $\frac{4-3.5}{4}$. In total we want the error rate of the obstacles to be restricted to within 10 percent across all points and for all obstacles.

7.3 Tests for Depth Camera Obstacle detection

Since the depth camera is mainly used to detect ground level threats, we would simply conduct a similar test for the sensors, but with ground level obstacles. To do this, we would have one of the team members wear the belt, and walk past the three obstacles used in the sensor test, which are placed on the ground. Then we would record the depth camera footage and feedback, and view the footage playback. For this test, we want to measure the time in which the obstacle appeared in the field of view of the camera, and compare it to the time the system is able to mark the obstacle as a threat. Comparing these two times, we can determine the accuracy of the depth camera for the three obstacles of different sizes. In this case, we want to achieve 90 percent accuracy for all three obstacles.

7.4 Tests for Sensing-Vibration response time

For this test, we want to make sure that when an obstacle is present, our system is able to trigger the vibration unit response quickly. In this test, we would be using the $20cm \times 20cm$ obstacle, and inserting it into the detection range. A team member would be wearing the active belt, with his hands stretched forward horizontally, holding the obstacle in his hands. The obstacle would be tied to his wrist with a line, such that when he lets go of the obstacle, the obstacle would fall and dangle at approximately the same height as the belt. (This simulates a sudden inserted obstacle into the detection range). The team member would be holding a timer, and around the same time as he drops the obstacle, he would press and start the timer, he would then press again and stop the timer as soon as he feels vibration from the belt. This recorded time would give us a rough estimate of the system response time. And accounting for human reflex in pressing a button(averaging around 0.25s), we would conduct that the system is reliable if the recorded time is less then 0.5s. (In real life, the user would feel the response almost immediately when the obstacle enters detection range). We would conduct this experiment multiple times and compute the average.

7.5 Tests for Use Case

For this test, we want to get an estimate of how well our product actually performs. We would first create a controlled environment in an enclosed room, with a specified starting location and a finish line. This would create a track where the volunteer would have to navigate through. We would then place obstacles in the form of soft paper boxes along the track. The paper boxes would be of different height and sizes to simulate all kinds of ground level and above ground level threats. Then we would invite volunteers(likely friends), to be blind folded and lead into the room. The volunteers would be instructed on the usage of the belt, and informed that their mission is to walk to the finish line as fast as possible and hitting as little obstacles as possible. To guide the volunteer in the correct direction, a music track would be placed at the finish line. In the first run, the blindfolded volunteer would be provided with no equipment, and asked to walk to the finish line. In the second run, we would alter the track(to avoid the influence of memorizing tracks), keep the number of obstacles constant and make the difficulties similar(this can be done my placing the obstacles in a symmetric position), and ask the volunteer to attempt the task again, this time with the help of the belt. In the end, we would record the volunteer's finish time and number of unintended collisions(touching does not count). We would conduct this experiment with up to 5 participants, and compare their performance between the two rounds. We would hope to see a significant decrease in their finish time as well as number of collisions.

8 PROJECT MANAGEMENT

8.1 Schedule

The detailed schedule is shown in Fig. 4. towards the end of the report.

8.2 Team Member Responsibilities

Xiaoran Lin:

- Arduino interface code
- software and hardware integration for sensor and vibration unit.
- Python code for raspberry pi and arduino serial communication
- Battery and circuit connection

Ning Cao:

- Depth Camera Integration with Raspberry Pi
- CV code for depth camera obstacle detection
- Potential obstacle classification for depth camera

Zhuoran Zhang:

- Raspberry Pi integration
- Raspberry Pi python code for threat level processing based on arduino and depth camera input
- Managing schedule and materials
- Belt physical form management

8.3 Bill of Materials and Budget

Our table is shown below in table 3 3.

Table 3: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Depth Camera	Oak-D	Luxonis	1	$199	$199
Raspberry Pi	Pi 4 2GB	Canakit	1	$139.99	$139.99
Battery	Smallest 10000mAh USB-C Portable Charger	Charmast	1	$23.39	$23.39
Arduino Uno	REV3	Arduino	1	$22.77	$22.77
Ultrasonic Sensor	HC-SR04	Smraza	6	$3.198	$19.188
Belt	Tactical Belt	FAIRWIN	1	$14.99	$14.99
Raspberry Pi SD Card	Pi 32GB Preloaded (NOOBS)	Raspberry Pi	1	$9.9	$9.9
Vibration coin motor	MIniVibration Motor For Mobile Phone Bluetooth	Tegg	6	$1.17	$6.99
					$458.988

8.4 Risk Mitigation Plans

Our main risks include the following

- **Operating Multiple Sensors at desired rate:** Since we are operating multiple supersonic sensors, one of our main risk is interference between the sensors. To handle this risk, our proposed plan is to operate sensors sequentially in a cycle, so that only one sensor is operating at a given time. The potential setback of this plan is that we might have a limit on how many data we can receive in a given second. The backup plan to increase data rate is to operate two sensors at a same time in a cycle, for example, if the sensors are numbered 1,2,3,4,5,6 from left to right on the belt, we can operate (1,4),(2,5),(3,6) by pairs. This way we are less likely to have interference since the sensors in each pair are relatively far, and we can also achieve higher data rate.

- **Arduino Pin management:** Since we are utilizing so many sensors and vibration unites, we might run into the case where we do not have enough operating pins on the arduino uno model. The current plan to solve this is to operate the sensor input with a one-directional diode, so that multiple sensor input can share a pin. At the same time, we could also operate sensor trigger with a decoder so that we can operate multiple sensors with only 2 pins. If this fails to work, the backup plan is to switch from arduino uno to arduino mega, which is a slightly larger board with significantly more pins.

9 RELATED WORK

During ideation, our team has extensively researched technical products currently available in the market with the same value proposition of helping the visually impaired navigate. A capstone at SUTD, prototype built by MIT CSAIL, and startup product are linked below for reference.

- MIT CSAIL Lab – https://techcrunch.com/2017/06/02/mit-develops-a-vibrating-wearable-to-help-people-with-visual-impairments-navigate/
- Strap Tech startup – https://strap.tech
- SUTD Capstone – https://capstone2021.sutd.edu.sg/projects/n-able-a-navigational-wearable-for-the-blind-and-visually-impaired

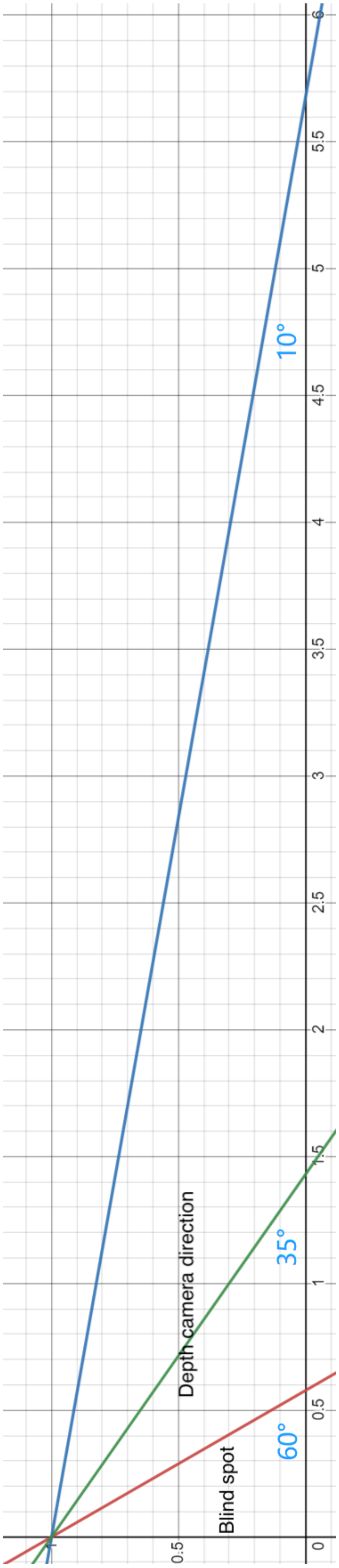

Figure 3: Depth camera cover area

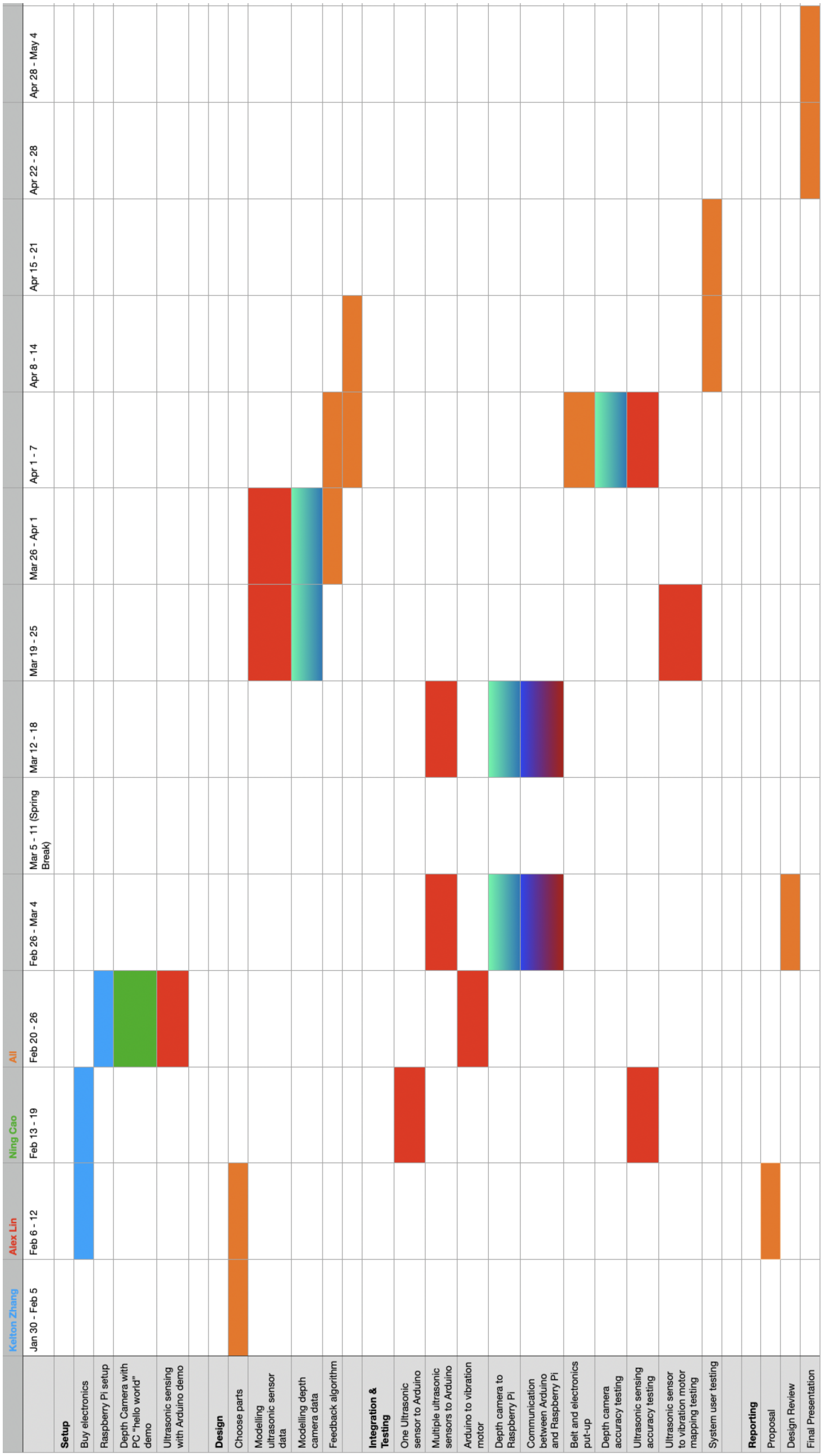

Figure 4: Gantt Chart