ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Smart Assistive Walking Stick for the Visually Impaired

Team #16

YUCHENG ZHANG
(yz90@illinois.edu)
SANHE FU
(sanhefu2@illinois.edu)
YIHAN HUANG
(yihanh4@illinois.edu)
HAOYANG ZHOU
(hz74@illinois.edu)

Professor: Yushi Cheng
TA: Hongtai Lv

April, 6 2025

Contents

1 Introduction **1**
1.1 Problem and Solution Overview . . . 1
1.2 Visual Aid . . . 3
1.3 High Level Requirements List . . . 3

2 Design **5**
2.1 Diagrams . . . 5
2.1.1 Block Diagram . . . 5
2.1.2 Physical Diagram . . . 6
2.1.3 Circuit Diagram . . . 6
2.2 Descriptions . . . 7
2.2.1 Control Module: . . . 7
2.2.2 Information Collection Module: . . . 9
2.2.3 User Response module: . . . 14
2.2.4 Power Module: . . . 17
2.3 Tolerance Analysis . . . 17
2.3.1 Speed Tolerance . . . 19
2.3.2 Positional Tolerance . . . 20

3 Cost and Schedule **21**
3.1 Cost . . . 21
3.2 Schedule . . . 21

4 Discussion of Ethics and Safety **23**

References **26**

1 Introduction

1.1 Problem and Solution Overview

More than 250 million people worldwide suffer from varying degrees of visual impairment, which has a profound impact on their physical health, mental well-being, and overall quality of life. Impaired vision can influence people's mobility, and thus impact their life quality [1] [2]. Individuals with impaired vision face three key challenges when navigating their surroundings: obstacle avoidance, indoor path planning, and key object localization. Therefore, it is important to have a tool to help blind people locate obstacles and plan routes. Especially in the intersections on the road, the road conditions in this area are complicated, and the blind cannot directly identify the traffic lights and traffic signals. In China, most city intersections do not have voice prompts like those in the United States, so blind people do not have the important information to cross an intersection in China. At the same time, the traffic flow at Chinese urban road intersections is very large with fast speed, so it is more dangerous to pass this section.

The most commonly used assistive tool for visually impaired individuals is the white cane, which provides users with tactile feedback. However, the standard white cane has a limited detection range, only sensing obstacles within its physical length, and cannot identify distant or elevated obstacles. But these situations are very common in the intersections. Moreover, the white cane provides only basic physical feedback and lacks the capability to convey detailed environmental information, such as road intersections and navigation directions. As a result, in unfamiliar or complex environments, relying solely on a white cane makes precise navigation difficult, forcing users to depend on external assistance or their memory of previously traveled routes.

Our solution to this problem is the development of an intelligent smart cane. [3] The smart cane can improve walking speed and safety both outdoors and indoors, and we have designed it with a focus on blind people crossing intersections. The sensors can be used to detect environmental information and help users address navigation problems [4]. Our smart cane will be equipped with LIDAR sensors to measure the distance to obstacles and estimate the user's position [5] [6]. A GPS system can be used for precise outdoor positioning [7]. And computer vision technology can be leveraged to capture detailed environmental information, such as traffic signs and other critical landmarks [8]. Additionally, the smart cane features motor-controlled omnidirectional wheels for directional guidance and provides real-time voice feedback to assist users in navigating their surroundings with greater ease, speed, and confidence. In outdoor environments, aid from GPS can not only help the user to walk in strange environments that are not similar but also help them to be more confident in their familiar environments. It will also show a great ability when navigating the users to walk in indoor environments, where the obstacles are usually many and unexpectable. When the user passes through the intersection area, GPS will help give the alert, the camera takes information about the surrounding environment, such as traffic lights and their duration, traffic signs, and whether there are vehicles around. This information is identified by computer vision algorithms and then prompted by voice to the user. And the strong detecting ability provided by the laser sensor of our smart cane can help to avoid crashing into obstacles, especially to avoid crashing into people and objects that are moving fast speed.

The smart assistive walking stick we designed is different from other existing products mainly in that it has mature obstacle location, route planning, and information acquisition functions. The guide sticks on the market are only equipped with voice prompts at most, and the blind need to obtain environmental information by themselves through touch. Most of the smart poles mentioned in the paper are equipped with sensors to detect obstacles or plan routes[9]. Their design works well to help blind people navigate roads, but they don't account for particularly dangerous sections. Our guide poles focus on the safety and efficiency of blind people when crossing dangerous sections such as intersections. To achieve this goal, our design not only integrates obstacle detection, path planning, and information acquisition, but also specifically enhances the application of computer vision algorithms to recognize the signal status of intersections, the direction of vehicle travel, and pedestrian priority rules. Through advanced image processing and deep learning algorithms, the guide stick can analyze the status of traffic lights and traffic flow density in real time to intelligently evaluate the best time to cross the road, thus improving the efficiency of walking for the blind under the premise of ensuring safety.

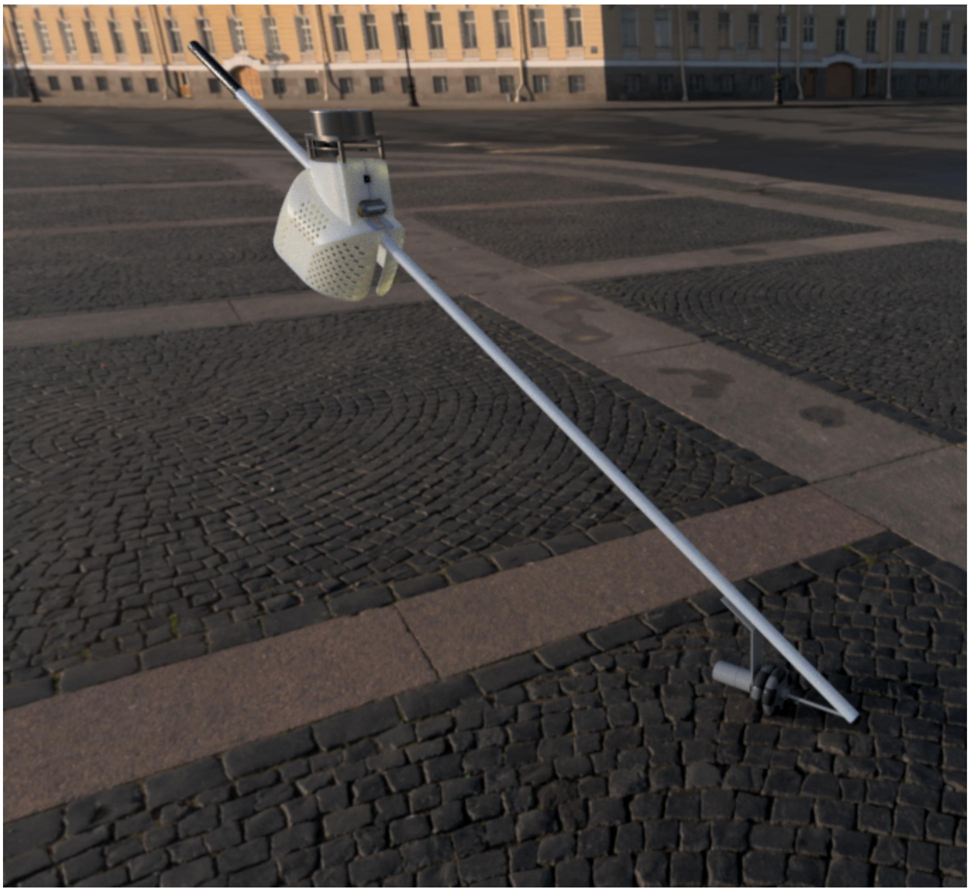

Figure 1: Smart Cane Overview

1.2 Visual Aid

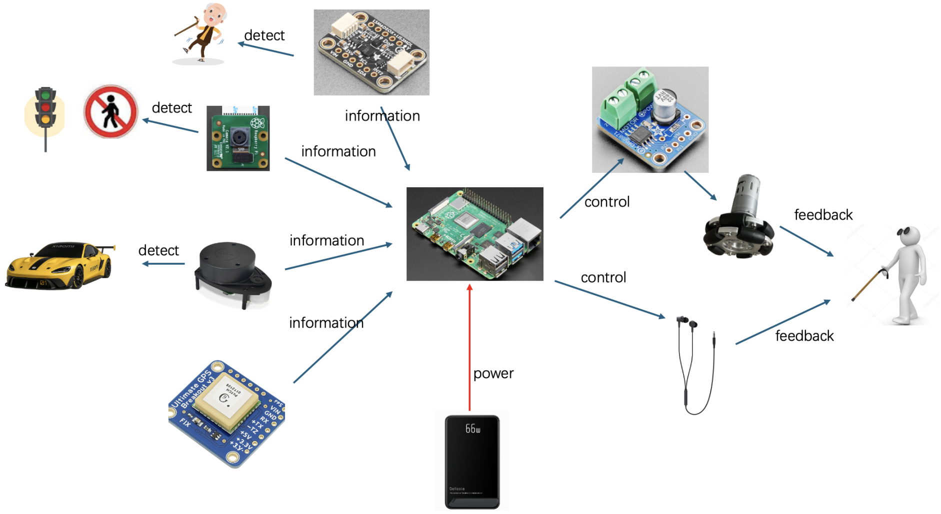

Figure 2: Visual Aid

1.3 High Level Requirements List

High accuracy of computer vision algorithms

The Raspberry Pi camera must capture traffic lights and traffic signs within a 20-meter range and process images at a resolution of 3280 × 2464 pixels. The YOLOv7-based computer vision algorithm should achieve an accuracy of at least 90% in detecting and classifying these elements. The total processing time, from image capture to Raspberry PiRaspberry Pi feedback, must be within 1.5 seconds to ensure timely decision-making for the user.

The accuracy of the sensors to identify obstacles

The walking stick must detect both stationary and moving obstacles within a 12-meter range. The system should differentiate between moving and static objects with a detection accuracy of at least 90%. For immediate hazards such as the obstacles detected within 2 meters, the Raspberry Pi should be able to trigger haptic feedback through a vibration motor with an intensity proportional to the obstacle's proximity.

The accuracy and real-time of GPS

The GPS Breakout module must achieve a positioning accuracy of 1.8 meters and update location data at a minimum rate of 10 Hz. The GPS data must be processed and transmitted to the Raspberry Pi, which will compare the user's real-time location with the predefined route. Additionally, the GPS module must be able to detect when the user approaches major intersections, triggering an alert to indicate when it is safe to cross based on traffic signal recognition from the Raspberry Pi.

The energy efficiency of the entire system

The smart walking stick must operate on a battery-powered system with a minimum runtime of 6 hours per full charge under normal usage conditions. The Raspberry Pi must regulate peripheral power dynamically, ensuring that non-essential modules enter low-power mode when not actively needed, reducing overall energy consumption by at least 30% compared to continuous operation. A battery level monitoring system must provide real-time alerts when battery life drops below 20%, ensuring the user is notified well in advance of power depletion.

2 Design

2.1 Diagrams

2.1.1 Block Diagram

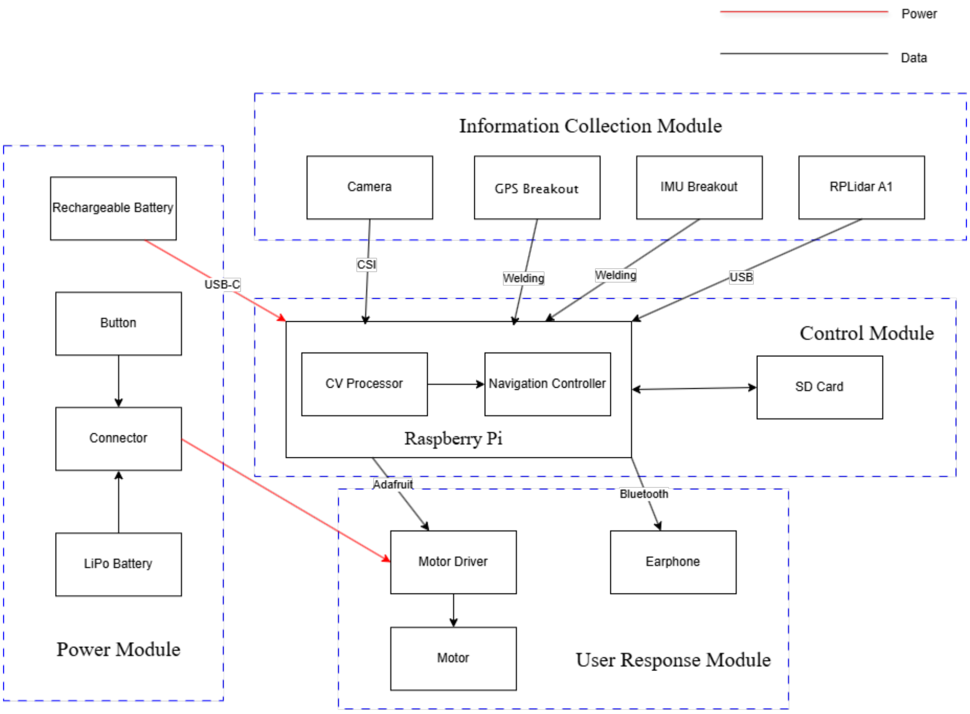

Figure 3: Block Diagrams

2.1.2 Physical Diagram

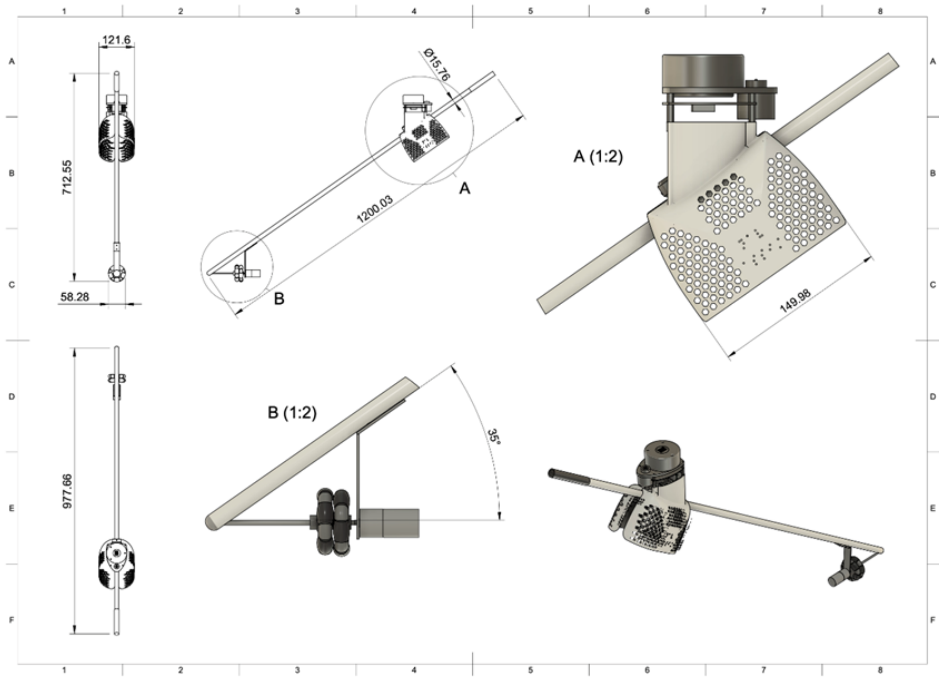

Figure 4: Physical Design Diagrams

2.1.3 Circuit Diagram

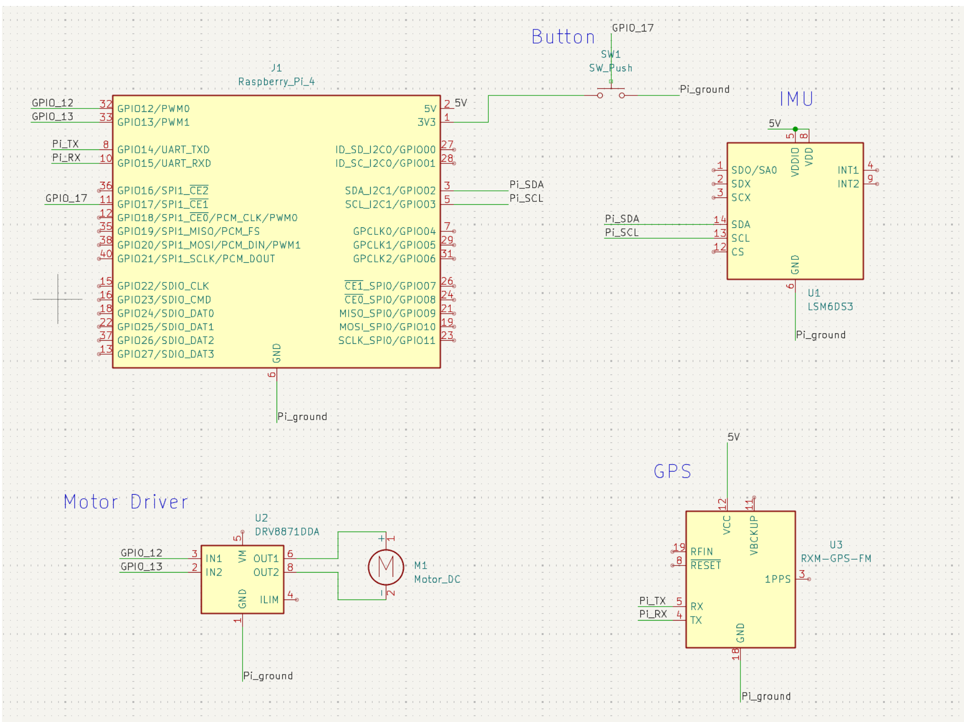

Figure 5: Circuit Diagrams

2.2 Descriptions

2.2.1 Control Module:

The control module gets in and handles the environment information from laser radar, camera, IMU, and GPS through SPI and UART communication, it will control the motor to avoid obstacles and follow the planned routes. Also, the information will be informed to the users through Bluetooth earphones. The Raspberry Pi will handle the data from laser radar, IMU, and GPS directly, and the image captured by the camera will be first processed by computer vision processing unit in Raspberry Pi and then used in the Raspberry Pi central processing unit.

1) Central Processing Unit:

   We chose the Raspberry Pi 4 as the central processing unit for our smart assistive guide stick due to its high computational performance, rich interface support, and ability to handle complex multi-sensor processing tasks. Equipped with a 1.5GHz quad-core Cortex-A72 processor and 4GB of RAM, the Raspberry Pi 4 is capable of efficiently processing data from the RPLIDAR A1 LiDAR, Adafruit Ultimate GPS, IMU, and camera input in real-time. It supports UART, I2C, SPI, GPIO, and MIPI CSI interfaces for reliable peripheral communication.

   In addition, the Raspberry Pi 4 includes built-in Bluetooth 5.0 and dual-band Wi-Fi, allowing seamless wireless communication with Bluetooth earphones or mobile devices. Unlike traditional microcontrollers, it can run a full Linux OS, enabling advanced features such as real-time image processing with OpenCV and deep learning inference using TensorFlow Lite or YOLOv7. This eliminates the need for a separate microcontroller, reducing system complexity and improving integration.

   Taking into account computing power, peripheral compatibility, wireless communication, and software ecosystem, the Raspberry Pi 4 is the ideal all-in-one control platform for this project.

Requirement Description	Verification Method
• The Raspberry Pi 4 must communicate with external sensors (RPLIDAR A1, GPS, IMU) over UART, SPI, and I2C. • It must ensure stable data acquisition with an update rate of at least 10Hz. • It must control the motor and transmit Bluetooth audio with latency no more than 100ms. • The Raspberry Pi must optimize power usage via peripheral management and software-level sleep routines.	• Connect sensors to Raspberry Pi using appropriate interfaces. Use Python or C libraries to validate data reception and check real-time decoding accuracy. • Log timestamps of received sensor data and confirm ≥ 10 updates per second per sensor using system logs. • Use GPIO-based PWM control to drive the motor; log feedback timing. Use Bluetooth tools to measure audio feedback latency during event triggers. • Monitor power consumption using USB analyzer while dynamically enabling/disabling modules; evaluate power profiles with and without active tasks.

2) Computer Vision Processing Unit:

The Raspberry PI 4 also has a sub-module to be used as a computer vision processing unit to capture real-time images captured by Raspberry PI cameras and perform traffic light recognition, traffic sign detection, and road condition analysis. The Raspberry PI 4 is powered by a quad-core Cortex-A72 64-bit processor at 1.5GHz and supports 4GB of RAM to efficiently run OpenCV, YOLOv4-Tiny, or TensorFlow Lite for real-time target detection. Its USB 3.0 interface provides data transfer rates of up to 5Gbps, greatly improving the processing speed of camera data, while dual-band Wi-Fi and Gigabit Ethernet ensure stable communication capabilities in high-data applications. The results of the computer vision processing, such as traffic light status, recognized road signs and traffic flow information, are transmitted to the central processing unit for further control of the motor, voice feedback or other auxiliary functions.

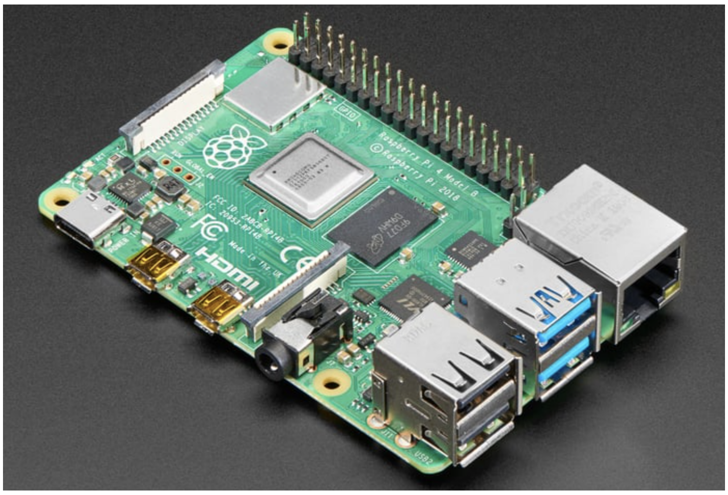

Figure 6: Raspberry PI 4

Requirement Description	Verification Method
• The Raspberry Pi 4 must capture images at 3280 × 2464 resolution using the Pi Camera. • It must process the images using computer vision algorithms for traffic light recognition, traffic sign detection, and road condition analysis with high accuracy. • The system must maintain stable data transmission via dual-band Wi-Fi or Gigabit Ethernet, ensuring uninterrupted operation in real-world environments.	• Use raspistill or Python OpenCV to capture full-resolution images; check EXIF metadata or image dimensions to confirm resolution. • Run detection model (YOLOv7-tiny or TFLite) on labeled images and evaluate performance metrics (precision, recall) to confirm high recognition accuracy. • Place time log points before image capture, before/after CV processing, and after UART send; confirm total duration is ≤ 1.5 seconds. • Use Wi-Fi and Ethernet file transfer or socket streaming over 10 minutes of simulated runtime; log transmission continuity and packet integrity.

2.2.2 Information Collection Module:

The Information Collection Module utilizes the devices including RPi Camera, GPS breakout, IMU, lidar and so on to collect the information from the outer environment of the cane and send the information back to the control unit through SPI, URAT communication and the ports on Raspberry Pi for an optimal nest step decision. And then we will introduce the functionality of the information collection devices one by one.

1) RPi Camera:

The RPi Camera Module we used in our design is the Raspberry Pi Camera Module v2, which can capture high-resolution images and videos. It interfaces with the Raspberry Pi via the MIPI CSI-2, which allows for high-speed data transmission. The camera supports resolutions up to 8 megapixels. The Raspberry Pi camera interface consumes low power, making it suitable for embedded and battery-powered projects.

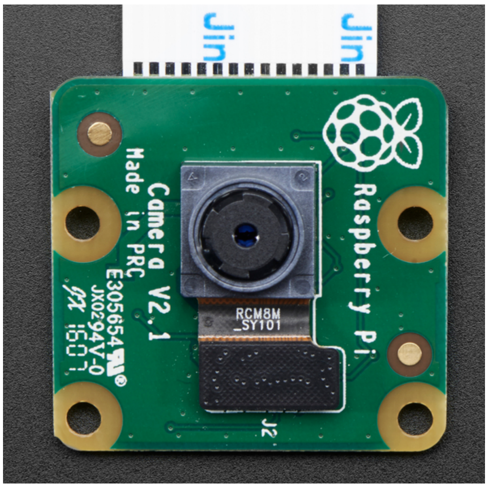

Figure 7: RPI Camera

Requirement Description	**Verification Method**
• The RPi Camera Module v2 must capture images at a resolution of 3280 × 2464 pixels. • It must record 1080p video at 30fps. • The camera module must operate with low power consumption (<250mW) for battery efficiency.	• Capture still images using raspistill and verify the image resolution using metadata or image viewer tools (e.g., ImageMagick identify command). • Use raspivid to record 1080p@30fps video and verify frame rate and resolution through media info tools like VLC or ffprobe. • Connect the camera to the Pi and record power usage with a USB inline current/power meter while taking images/videos; confirm consumption <250mW.

2) IMU Breakout:

We used the Adafruit LSM6DSOX + LIS3MDL IMU as the inertial measurement unit of the smart guide stick to monitor the attitude, direction and motion state of the guide stick in real-time. The module integrates the LSM6DSOX 6-axis inertial sensor and LIS3MDL 3-axis magnetometer to provide 9-DOF motion tracking. The LSM6DSOX has an acceleration measurement range of ±2g to ±16g and an angular velocity detection capability of ±125 to ±2000 dps (degrees per second), ensuring that the guide stick can accurately detect tilt Angle, vibration and rotation direction as it travels. The LIS3MDL magnetometer supports a magnetic field measurement range of ±4 to ±16 gauss and can be used in an electronic compass function to provide an absolute directional reference to help blind people stay oriented in complex environments.

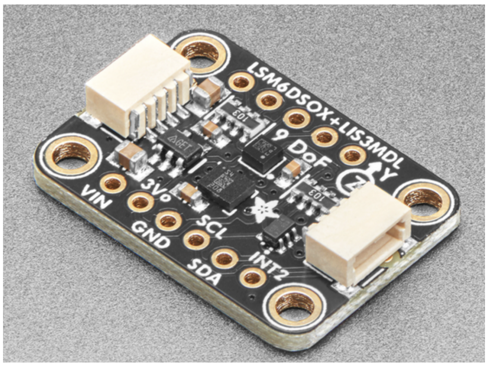

Figure 8: IMU Breakout

Requirement Description	**Verification Method**
• The IMU module must measure acceleration in the range of ±2g to ±16g. • It must measure angular velocity between ±125 dps to ±2000 dps. • The update frequency of the IMU must be no less than 100Hz to ensure accurate motion tracking.	• Mount the IMU on a test rig and use controlled motion to generate known linear acceleration; verify readings in all three axes fall within the ±2g to ±16g range as configured. • Rotate the sensor using a turntable or controlled rotation device to observe angular velocity; verify values correspond to actual rotational speed and lie within the configured ± 125–2000 dps range. • Use timestamped logging of continuous IMU data and compute sampling intervals; verify update rate is $\geq$ 100 samples per second.

3) RPLIDAR A1 Lidar:

RPLIDAR A1 Lidar is used as the core perception module of the guide stick, which is responsible for measuring the distance of obstacles and transmitting the data to the Raspberry Pi 4 for real-time processing. It can work excellently in all kinds of indoor environment and outdoor environments without sunlight. The RPLIDAR A1 has 360° environmental scanning capability and can complete up to 8,000 distance measurements per second, ensuring safe navigation for blind people in complex environments. It has a maximum detection range of 12 meters, allowing the wand to sense obstacles in advance and make travel decisions. Lidar uses an adjustable scan frequency of 2-10Hz, allowing it to optimize data acquisition at different walking speeds, while its low-power design (about 5W) is suitable for portable battery-powered devices. The ranging data is transmitted through the SPI interface to the Raspberry Pi, which directly parses the data and controls the motor to help the user avoid obstacles and follow a safe path.

Figure 9: RLIDAR A1 Lidar

Requirement Description	Verification Method
• The RPLIDAR A1 must perform at least 8,000 distance measurements per second. • It must cover a full 360° scanning range. • It must have a minimum detection distance of 12 meters. • It must maintain stable communication with the Raspberry Pi 4 via SPI.	• Use the RPLIDAR SDK to collect live scan data and calculate the measurement rate in samples/second; verify it meets or exceeds 8,000. • Run visualization tools or custom scripts to check scan coverage across the full 360° field. • Place reflective objects at varying distances up to 12 meters and log the detection distances; confirm all are correctly recognized. • Connect via SPI to Pa and monitor signal stability using logic analyzer; check for consistent packet reception without errors.

4) GPS:

The Adafruit Ultimate GPS Breakout is used as a navigation module for the smart guide cane, providing highly accurate geolocation information. The GPS module supports 66-channel MTK3339 chip and has a 10Hz update rate to ensure real-time navigation data. Its positioning accuracy and velocity accuracy can reach 1.8 meters and 0.1meter/s separately, and it has an ultra-high sensitivity of -165 dBm in open environments, even in urban tall buildings or under the shade of trees, and can receive satellite signals stably. The Adafruit Ultimate GPS has a built-in antenna and an external antenna interface to optimize signal reception and provide more stable navigation in different environments. Power usage is incredibly low, only 20 mA during navigation.

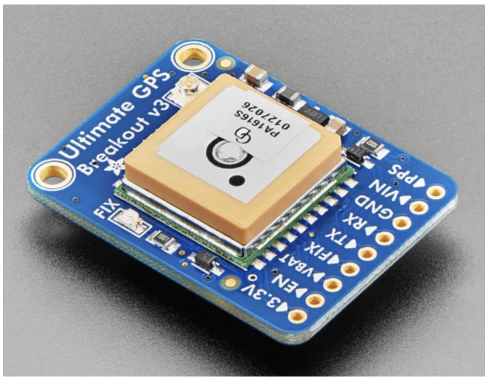

Figure 10: GPS Breakout

Requirement Description	Verification Method
• The GPS module must support a minimum update rate of 10Hz. • It must provide positioning accuracy within 1.8 meters. • It must function reliably in urban environments with a sensitivity of -165 dBm. • It must operate at a power consumption of $\leq$ 20mA to support extended battery life.	• Log continuous GPS NMEA sentences and timestamp updates; confirm $\geq$ 10 position updates per second. • Compare logged GPS coordinates to surveyed ground-truth points; calculate average positioning error. • Test GPS performance in areas surrounded by tall buildings or under tree cover and confirm signal acquisition and tracking. • Use multimeter or USB power logger to monitor current draw during operation and ensure $\leq$ 20mA.

2.2.3 User Response module:

The User Response Module primarily consists of a motor, motor driver, and earphone, serving as the key components for delivering the Raspberry Pi's processed decision outputs to the user. It connects to the Raspberry Pi via GPIO and UART, ensuring efficient communication. By utilizing the motor for haptic feedback and the earphone for auditory feedback, the system delivers information to the user quickly and accurately, enhancing responsiveness and interaction.

1) Motor:

   We use motors to provide stable, precise motion control for the cane. We chose a 12V DC brush motor combined with a 4.4:1 metal gear reduction box, which can reach 1800 RPM under no load, ensuring smooth running of the guide stick in different ground environments. In addition, the motor provides 0.71 kg·cm (10 oz·in) of torque at maximum load and has a 1.8A lock-in current, which is suitable for the efficient power output of the smart guide stick.

Requirement Description	Verification Method
• The motor must be capable of operating at 12V DC. • It must achieve at least 1800 RPM under no load. • It must deliver a torque of 0.71 kg·cm (10 oz·in) at maximum load. • It must maintain a power efficiency of at least 85%. • It must support precise motion control via PWM signals.	• Apply 12V from a bench power supply and observe motor start-up and stable operation. • Use a tachometer to measure rotational speed under no load; verify $\geq$ 1800 RPM. • Mount to a torque-measuring rig or dynamometer to verify load torque at maximum rated output. • Measure input and output power, compute efficiency $\eta = \frac{P_{out}}{P_{in}}$; verify $\geq$ 85%. • Send PWM signals with varying duty cycles from Raspberry Pi and observe proportional changes in motor speed and direction.

2) Motor Driver:

We use the Adafruit DRV8871 DC Motor Driver Breakout Board as the motor drive module of the smart guide stick, which is responsible for controlling the motor. The DRV8871 driver supports 6.5V-45V input voltage, is fully compatible with 12V DC motors, and provides up to 3.6A peak current output, ensuring stable operation of the guide stick under different terrain and load changes. The driver is controlled by PWM signals from the Raspberry Pi 4, which enables precise speed adjustment and direction control. This allows the guide stick to dynamically respond to different travel speeds and obstacle avoidance strategies without the need for an external microcontroller.

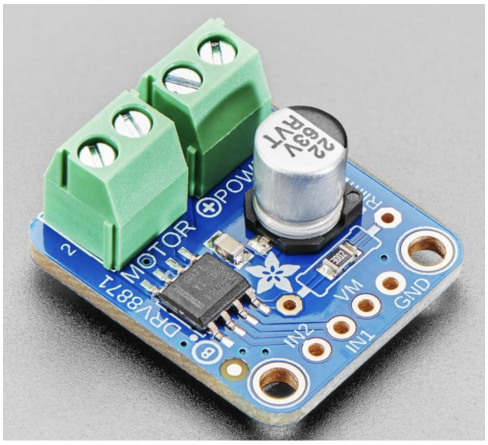

Figure 11: Motor Driver

Requirement Description	Verification Method
• The motor driver must support an input voltage range of 6.5V–45V. • It must provide a peak current output of 3.6A to ensure stable operation. • It must support PWM-based speed and direction control via Raspberry Pi 4. • It must include built-in thermal protection, current limiting, and under-voltage protection.	• Use a variable power supply to apply different input voltages from 6.5V to 45V; verify stable operation. • Connect a current logger and load motor to maximum load; verify driver can deliver $\geq$ 3.6A peak current without shutdown. • Use Raspberry Pi to generate varying PWM signals through GPIO; observe motor behavior and verify correct direction/speed response. • Intentionally overheat or short the output momentarily under supervision; confirm driver auto-shutdown or fault indication kicks in.

3) Earphone:

The earphones used in our design provide wireless audio transmission and hand-free communication. They connect to Raspberry Pi 4 via Bluetooth, sending audio information to users with low latency, low power consumption and high accuracy, making them a good choice for conveying the decision make by the smart cane.

Requirement Description	Verification Method
• The earphones must support Bluetooth 4.2/BLE. • They must have an audio transmission latency of no more than 100ms to provide real-time feedback. • They must operate at a power consumption of less than 50mW to ensure prolonged usage without frequent recharging.	• Pair with Raspberry Pi 4 using its built-in Bluetooth module and confirm Bluetooth version compliance via protocol logs. • Play audio samples with embedded timestamps; record playback using high-speed camera or waveform comparison to confirm latency $\leq$ 100ms. • Measure average power draw using USB power analyzer during continuous playback; verify usage $<$ 50mW.

2.2.4 Power Module:

Our power supply unit utilizes high-performance lithium batteries, equipped with premium lithium-ion cells. Despite its lightweight of only 40 grams, it delivers a rated energy of up to 4.995Wh, providing long-lasting and stable power support for devices. It also boasts robust power output capabilities, with a continuous discharge current of up to 33.75A, easily meeting the power demands of high-power devices such as motor drives, ensuring stable operation and preventing performance degradation due to insufficient power supply. Additionally, it features a wide voltage range, with a rated output voltage of 11.1V and a full charge voltage of 12.6V, making it compatible with a variety of devices and providing stable voltage output to effectively protect device circuits. For interface connectivity, it uses a standard XT30 plug for a convenient connection.

Requirement Description	**Verification Method**
• The power module must provide a minimum rated energy of 4.995Wh. • It must support a continuous discharge current of at least 33.75A. • The battery must operate within a voltage range of 11.1V to 12.6V. • It must include a standard XT30 plug for secure connectivity.	• Fully charge the battery and discharge it using a controlled electronic load; calculate total energy delivered (Wh) to verify $\geq$ 4.995Wh. • Set the load to 33.75A continuous draw and monitor voltage stability and heat dissipation; verify no shutdown. • Use multimeter to verify voltage remains within 11.1–12.6V throughout normal usage. • Connect and disconnect XT30 plug 50+ times, checking for secure contact and no signal loss; verify mechanical integrity.

2.3 Tolerance Analysis

In our project, the most critical part is the accuracy and latency of computer vision processing, which directly affects the user's perception of traffic lights, road signs and obstacles, as well as the final walking decision. If the computer vision recognition accuracy is too low or the processing delay is too high, blind people may not be able to access critical information in a timely manner, leading to potential security risks.

Design challenges include how to improve recognition accuracy and reduce processing latency. Computer vision needs to be at least 85% accurate to provide reliable information to the user, and is subject to changes in lighting conditions, occlusion, and weather. The Raspberry PI 4 takes less than 1.5 seconds to complete the process from camera capture to analysis and transmission to the central processing unit.

The computer vision processing time is determined by the computational power of the Raspberry PI and the computational complexity of the model. High-resolution images (3280 × 2464) may result in increased processing time, so it may be necessary to adjust the frame rate or reduce the image resolution to ensure real-time performance.

Mathematic Analysis:

- Let N_{total} be the total number of targets and $N_{correct}$ be the number of correctly identified targets. The recognition accuracy is given by:

$$Accuracy = \frac{N_{correct}}{N_{total}} \times 100\%$$

- The goal is to maintain recognition accuracy $\geq 85\%$. If the accuracy decreases in low-light or occluded environments, algorithm adjustments are required, such as Data Augmentation or using a more robust model.

Processing Delay Calculation:

- The total image processing time $T_{\text{processing}}$ is mainly composed of:

$$T_{\text{processing}} = T_{\text{capture}} + T_{\text{preprocess}} + T_{\text{inference}} + T_{\text{communication}}$$

  where:

  - T_{capture}: Image acquisition time (approximately **30ms**)
  - $T_{\text{preprocess}}$: Image resizing and normalization (approximately **100ms**)
  - $T_{\text{inference}}$: Model inference time (approximately **1.2s**, YOLOv7-Tiny on Raspberry Pi 4)
  - $T_{\text{communication}}$: Transmission via to Raspberry Pi (approximately **50ms**)

- **Total processing time target:** $\leq 1.5s$

Solution: Use YOLOv7-Tiny or quantized TensorFlow Lite for faster reasoning.Appropriately reduce the camera resolution, reduce the amount of data, and improve the frame rate. Optimize data transmission, using UART or Wi-Fi to ensure stable data transmission and avoid additional latency. If the processing time is longer than 1.5s, the computational complexity can be reduced, for example, by using region cropping to analyze only a specific region, such as the location of a traffic light.

LIDAR Sensor Tolerance:
The LIDAR sensor is crucial for obstacle detection along the cane's 12-meter effective range. It must account for potential misalignments and sensor noise to avoid missing small or fast-approaching obstacles.

$$\text{Error Margin} = d_{\max} \cdot \tan(\theta_{\text{error}}) + \delta \quad (1)$$

Where:

- $d_{\max}$ is the maximum detection distance (12 m).
- θ_{error} is the angular misalignment error in radians.

- δ represents the sensor's noise deviation in standard deviation (meters).

Description:
This formula models how deviations in sensor alignment and inherent noise affect the precision of obstacle area. The term $d_{\text{max}} \cdot \tan(\theta_{\text{error}})$ illustrates the geometric expansion of error with increasing misalignment, while δ fixes a fixed noise component. Our design mandates that the error margin remain below a safety threshold (e.g. $\leq 0.5\,\text{m}$) to guarantee that obstacles are detected within roughly one meter of their actual location. Consequently, both angular misalignment and sensor alignment routines are implemented continuously to monitor and adjust these parameters.
GPS Module Tolerance:
Accurate positioning is essential to trigger timely route corrections and intersection alerts. The GPS module's tolerance affects the system's ability to distinguish between safe and hazardous zones.

Key Formula:

$$\text{GPS Error}_{\text{total}} = \sqrt{\sigma_{\text{lat}}^2 + \sigma_{\text{long}}^2} \tag{4}$$

Where:

- σ_{lat} is the standard deviation of the latitude measurement.
- σ_{long} is the standard deviation of the longitude measurement.

Description:
This equation computes the overall positional uncertainty by combining the independent errors in the latitude and longitude dimensions. An error of up to 1.8 m is acceptable under typical operating conditions. In challenging environments, such as urban canyons where multipath effects are common, our design anticipates larger deviations. In such cases, supplementary strategies—like sensor fusion with IMU data—are employed to compensate for the GPS error, ensuring that the positional data remains reliable for navigation decisions.

Motor Control Tolerance:
The motor control system drives haptic feedback and directional guidance, necessitating precise speed and position control to deliver real-time assistance.

Key Formulas:

2.3.1 Speed Tolerance

$$\text{Motor Speed Error (\%)} = \frac{|\text{Measured\_RPM} - \text{Target\_RPM}|}{\text{Target\_RPM}} \times 100\% \tag{5}$$

Description:
This metric measures the deviation of the motor's actual speed from the desired value. A maximum deviation of 5% is targeted, ensuring that the response time and haptic feedback remain consistent. Any significant fluctuations beyond this range would compromise the user's ability to receive accurate and discrete directional cues.

2.3.2 Positional Tolerance

$$\text{Positional Error} = |\text{Actual\_Position} - \text{Desired\_Position}| \tag{6}$$

Description:
This formula quantifies the error in the physical positioning of the cane. With a maximum allowable deviation of 2 mm, the system can maintain a precise path, reducing the risk of sudden directional pivots. Continuous monitoring and calibration algorithms are implemented to correct deviations in real time.

Worst-Case Scenario and Design Adjustments:

For each subsystem, we simulate worst-case tolerance deviations and observe their cumulative impact on overall performance. For example, if the computer vision algorithm's accuracy falls below 85% due to challenging lighting or heavy occlusion, the system is designed to automatically adjust detection parameters or switch to alternative detection methods. Similarly, if the LIDAR's error margin exceeds 0.5 m in a given environment, recalibration protocols or more aggressive scanning algorithms are initiated. For the GPS module, if the combined positional error nears the 1.8 m threshold, sensor fusion with the IMU is used to refine the estimated position.

Design Adjustments:

- **Software:** Adaptive algorithms, region-of-interest processing, and dynamic resolution scaling are employed to mitigate increased sensor delays or recognition errors.
- **Hardware:** Regular calibration, enhanced sensor mounting precision, and the potential use of high-precision sensors (with tighter tolerances) are considered to ensure robustness.
- **Testing:** Comprehensive lab and field tests under extreme conditions are planned to verify that each component maintains its performance within the defined tolerance limits.

3 Cost and Schedule

3.1 Cost

Part	Item	Cost (RMB)
Control module	Raspberry Pi 4B 4 GB	714.00
Control module	Micro SD card	64.99
Information collection module	RPLIDAR A1M8	498.00
Information collection module	GPS and antenna	588.00
Information collection module	IMU	35.00
Information collection module	Raspberry Pi Camera v2	258.99
User response module	Motor	10.00
User response module	Motor driver	25.29
User response module	Wired earphone	39.00
Power module	Rechargeable battery	198.00
Power module	LiPo battery	99.98
Power module	LiPo battery charger	74.10
Structure	White cane	25.80
Structure	Raspberry Pi case	24.57
Structure	Omni wheel	60.00
Structure	Pi Proto Hat and wire	47.00
Total Cost		**2762.72**

Table 1: Cost

3.2 Schedule

time	Yihan Huang	Sanhe Fu	Yucheng Zhang	Haoyang Zhou
2.17–3.2	Project planning and design, sensors selection	Project planning and design, sensors selection	Project planning and design, computer vision algorithm learning	Project planning and design, computer vision algorithm learning
3.3–3.9	Writing RFA, contract and Proposal, material purchasing	Writing RFA, contract and Proposal, material purchasing	Writing RFA, contract and Proposal	Writing RFA, contract and Proposal
3.10–3.16	Test Raspberry Pi, GPS, IMU, Motor driver	Test RPLidar, Pi camera and sound feedback from raspberry Pi	Test and evaluate different version of Yolo and other vision detection models	Test and evaluate different version of Yolo and other vision detection models
3.17–3.23	Learn the basic use of RPLidar, develop the logic to achieve the obstacle detection	Learn the basic use of RPLidar, develop the logic to achieve the obstacle detection	Locally deploy YOLO on personal computer and test on the general version of weights and generate the executable file	Investigate the priority and advantage of different yolo version and optimization methods
3.24–3.30	Build the code structure for obstacle avoidance by controlling Motor driver	Build the code structure for obstacle detection with RPLidar	Deploy the YOLOv7-tiny model on Raspberry and do evaluation on pictures	Prepare the audio files and assign them in different function
3.31–4.6	Combine the code of obstacle detection and avoidance, add sound feedback	Combine the code of obstacle detection and avoidance, add sound feedback	Build environment for Pycamera and combine them together to work	Design the function of human posture detection and fall prevention with IMU module

4.7–4.13	Learn and design the use of GPS, develop the logic of navigation function with GPS	Learn and design the use of GPS, develop the logic of navigation function with GPS	Finish the code for the whole process of sign detection and audio feedback and combine it with the main code	Finalize and optimize the algorithm for traffic light detection and integrate it with the navigation system
4.14–4.20	Construct the smart cane, test and debug the function of obstacle detection and avoidance	Construct the smart cane, test and debug the function of obstacle detection and avoidance	Locally train a better version of weight for the detection of traffic signals	Conduct system-level integration tests for computer vision modules and refine algorithm parameters
4.21–4.27	Test and debug the GPS navigation function	Test and debug the GPS navigation function	Locally train a better version of weight for the detection of traffic signals	Perform system integration and cross-module optimization for computer vision and navigation modules
4.28–5.19	Overall function test and evaluate. Prepare for the demo.	Overall function test and evaluate. Prepare for the demo.	Overall function test and evaluate. Prepare for the demo.	Overall function test and evaluate. Prepare for the demo.

Table 2: Schedule

4 Discussion of Ethics and Safety

Our project's overarching ethical priority is to improve the welfare and autonomy of individuals with visual impairments while ensuring no additional hazards are introduced. We follow the IEEE Code of Ethics [10], which obligates engineers to uphold safety, transparency, privacy, and public welfare, as well as the OSHA guidelines for electrical and workplace safety. Below, we outline the primary ethical and safety concerns, our mitigation strategies, and the rationale behind each design choice.

Identifying Ethical and Safety Risks

- Device Reliability & Versatility:

  If undetected faults or limited environmental adaptability compromise performance, users may face unsafe conditions—especially when navigating intersections or encountering obstacles.

- Data Privacy:

  Our device collects environment and positional data, so unauthorized access to these records could endanger user privacy.

- User Awareness:

  Miscommunication about any limitations—like reduced sensor accuracy in low light or adverse weather—could expose users to unexpected risks.

Danger Mitigation Procedures

- Comprehensive Testing:

  We perform repeated lab and field tests under various lighting and weather conditions to uncover edge cases. Any defect (e.g., delayed detection of traffic lights or obstacles) is corrected before production.

- Robust Power and Thermal Design:

  We include over-voltage, over-current, and temperature safeguards to prevent battery malfunctions. Components are selected and assembled in line with UL/CSA or equivalent safety standards.

- User Training & Transparency:

  Clear documentation and user instructions highlight both the system's capabilities and known operational constraints, allowing users to decide how best to use the walking stick.

- Data Protection:

  We secure sensor data with encrypted storage/transmission (where feasible) and limit data collection to essential functionality in compliance with recognized privacy guidelines.

Relevant Ethics and Regulatory Standards

- IEEE Code of Ethics:

  Emphasizes honesty about system capabilities, transparency with stakeholders, and the obligation to avoid harm [10] [11]

- OSHA Requirements:

  Guide our lab safety practices, including proper PPE usage, safe soldering procedures, and hazard labeling.

- FCC & Industry Certifications:

Wireless modules (e.g., Bluetooth, Wi-Fi) comply with FCC Part 15 to reduce interference and ensure safe, legal operation.

Justifying Design Decisions for Safety & Ethics

- Sensor Selection:

  We chose sensors with high detection accuracy to minimize false positives and negatives, thereby upholding user safety.

- Battery Management:

  By implementing real-time battery monitoring and robust power regulation, we avoid unexpected shutdowns in critical moments.

- Enclosure and Mechanical Design:

  Smooth surfaces, protected wiring, and stable mounting for internal components reduce the risk of physical harm.

- Feedback Mechanisms:

  Haptic and audio alerts are simple, immediate, and tested with diverse users to ensure accessibility, aligning with fair and equitable treatment principles [10].

References

[1] J. M. Crewe, N. Morlet, W. H. Morgan, *et al.*, “Quality of life of the most severely vision-impaired,” *Clinical & experimental ophthalmology*, vol. 39, no. 4, pp. 336–343, 2011.

[2] H. Hörder, I. Skoog, and K. Frändin, “Health-related quality of life in relation to walking habits and fitness: A population-based study of 75-year-olds,” *Quality of life research*, vol. 22, pp. 1213–1223, 2013.

[3] A. D. P. dos Santos, F. O. Medola, M. J. Cinelli, A. R. Garcia Ramirez, and F. E. Sandnes, “Are electronic white canes better than traditional canes? a comparative study with blind and blindfolded participants,” *Universal Access in the Information Society*, vol. 20, no. 1, pp. 93–103, 2021.

[4] S. Real and A. Araujo, “Navigation systems for the blind and visually impaired: Past work, challenges, and open problems,” *Sensors*, vol. 19, no. 15, p. 3404, 2019.

[5] S. Bajracharya, “Breezyslam: A simple, efficient, cross-platform python package for simultaneous localization and mapping,” *Washington Lee university*, 2014.

[6] G. Fusco, S. A. Cheraghi, L. Neat, and J. M. Coughlan, “An indoor navigation app using computer vision and sign recognition,” in *Computers Helping People with Special Needs: 17th International Conference, ICCHP 2020, Lecco, Italy, September 9–11, 2020, Proceedings, Part I 17*, Springer, 2020, pp. 485–494.

[7] E. Cardillo and A. Caddemi, “Insight on electronic travel aids for visually impaired people: A review on the electromagnetic technology,” *Electronics*, vol. 8, no. 11, p. 1281, 2019.

[8] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[9] P. Slade, A. Tambe, and M. J. Kochenderfer, “Multimodal sensing and intuitive steering assistance improve navigation and mobility for people with impaired vision,” *Science Robotics*, vol. 6, no. 59, eabg6594, 2021. DOI: 10.1126/scirobotics.abg6594.

[10] IEEE, *IEEE Code of Ethics*, [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html, 2016.

[11] The Grainger College of Engineering, University of Illinois at Urbana–Champaign, *Safety – ECE 445: Senior Design Laboratory*, [Online]. Available: https://courses.grainger.illinois.edu/ece445zjui/guidelines/safety.asp, Accessed: 2024.