See For Me : A prototype for blind and visually impaired people

By Group 1
Vincent BERARDI
Flavien MATHIEU
Joris JACQUET
Antoine BEAULIEU
Tutor : Antonio FAONIO

Engineering Degree Semester 5 Project

Contents

1 Introduction **3**
1.1 Project context . . . 3
1.2 Main goal . . . 3

2 Project Organization **4**
2.1 Task Distribution . . . 4
2.2 Main Work Phases . . . 5
2.3 Support and Guidance . . . 5

3 Technical Approach and Prototype **6**
3.1 Hardware . . . 6
3.2 Image Processing . . . 7
3.3 Integration with Raspberry Pi . . . 8
3.4 Results and Performance . . . 9

4 Carbon Footprint and Cost **10**
4.1 Carbon Footprint Analysis . . . 10
4.2 Sustainable and Modular Design . . . 10
4.3 Privacy Protection . . . 10
4.4 Cost Breakdown . . . 11

5 Conclusion **12**
5.1 Summary of Achievements . . . 12
5.2 Challenges and Lessons Learned . . . 12
5.3 Potential Improvements . . . 12

6 User Manual **13**

7 Bibliography **15**

8 Annex **16**

List of Figures

1 Example of a path witch we need to detect 3
2 Diagram of the algorithm 8
3 Final result of the algorithm 9
4 Prototype 14

List of Tables

1 Detection rates for different obstacles 9
2 Cost of all components 11
3 Technical specifications of the prototype components 16

1 Introduction

1.1 Project context

Today Visually Impaired People (VIP) have a lot of issues in daily life, like walking outside alone safely. Today the tools used by VIP are the white cane or a dog. But these tools are limited, for example the white cane has a limited scope or an impossibility to detect far or high obstacles. It may also take several minutes to identify an obstacle on the street. Also, with the quick growing of new technologies, the range of possibilities to help VIP is also growing. Because this disability must be considered in our society, we implemented an algorithm using modern technologies to help VIP walking in the street.

1.2 Main goal

Our main goal in this project is to develop a system based on a raspberry Pi to help VIP go forward straight, detect an obstacle and identify it at the same time. To do that we can alert the user with haptic and audio signal.

Figure 1: Example of a path witch we need to detect

2 Project Organization

2.1 Task Distribution

Tasks were divided into the project's core pillars:

- **Image Processing:**
  - Flavien worked on red line detection, shape recognition (initially circles, later adapted to squares), and obstacle identification using OpenCV. He refined detection methods, integrated the PiCamera, and implemented Hough line detection with interpolation for better path tracking.
  - Antoine contributed by adjusting parameters for circle detection and linking the program to the Raspberry Pi camera.
  - Vincent worked on developing and fine-tuning shape detection algorithms, testing recognition accuracy on real-world markers.
- **Audio and Haptic Feedback:**
  - Antoine and Joris researched sound generation in Python and worked on the vibrator system for distance feedback.
  - Flavien contributed by linking detected objects to corresponding sounds and haptic feedback.
- **System Integration:**
  - Flavien ensured seamless communication between the Raspberry Pi and the detection software, enabling SSH access for remote monitoring.
  - Joris and Vincent worked on ways to attach the Raspberry Pi to the user's body for ergonomic integration.
- **Environmental Considerations:**
  - Vincent and Joris explored ways to improve the usability of the system for visually impaired users, considering practical constraints.
  - Antoine worked on the poster layout, organizing information about algorithmic and technical aspects.

2.2 Main Work Phases

The project was organized into weekly sessions (2-3 per week) and three main phases:

Phase 1: Initial Study During this phase, we focused on familiarizing ourselves with the hardware and tools, including understanding how the components work, such as the Raspberry Pi, sensors, and camera. We also defined both technical and user requirements, setting the foundation for the project's development.

Phase 2: Software and Hardware Development In this phase, we worked on the development of image processing, audio, and haptic feedback systems. Specifically, we explored different methods to detect crosses and circles (eventually adapting to squares instead of circles). We also discussed how to utilize audio and haptic feedback: which signal to use for a cross/square/line, whether to use sound or voice, and if stereo audio could indicate the right/left position of an object. This phase also involved integrating hardware and software components to ensure seamless communication and functionality.

Phase 3: Testing and Validation This phase involved testing the prototype in real-world scenarios to assess its reliability and usability. Based on feedback, the prototype was validated and adjusted to improve performance and user experience.

2.3 Support and Guidance

Tutor and experts provided feedback and specialized knowledge on:

- Raspberry Pi, sensors, and modules. (We consulted Prof. Aurelien Francillon regarding a connection issue between the TF Luna and the Raspberry Pi.)
- Regular coaching sessions and meetings were held to address ongoing challenges, provide feedback, and ensure continuous progress.

3 Technical Approach and Prototype

3.1 Hardware

Main components:
- **Raspberry Pi 4**: Equipped with a quad-core processor, sufficient RAM, and multiple GPIO ports, it serves as the central platform for data acquisition and processing. It provides connectivity via Wi-Fi and Bluetooth.[2]
- **PiCamera 2**: A high-resolution camera module with a wide field of view, fully compatible with the Raspberry Pi.
Role: Captures images for real-time path analysis.
- **TF Luna LiDAR Sensor**: A compact and energy-efficient distance sensor capable of short-range obstacle detection.[5]
Role: Provides precise depth measurement for obstacle avoidance.
- **Haptic Feedback System**: Two vibration motors placed on the left and right sides. The intensity and duration of vibrations vary based on detected obstacles.
Role: Assists visually impaired users by giving physical feedback.
- **Audio Feedback Module**: Uses pre-recorded sound alerts and directional cues.
Role: Provides auditory guidance based on object detection.

Advantages and disadvantages:
Advantages: Low cost, energy-efficient, and adaptable for different scenarios.
Disadvantages: Limited sensor range, susceptible to environmental conditions such as low light and background noise.

3.2 Image Processing

The image processing module utilizes OpenCV [1] to detect four key elements in the captured frames [3]: - Crosses - Squares - Cardboards - Red Lines

A loop continuously processes each frame, detecting and categorizing elements using specialized functions.

Cross Detection:
A Gaussian filter and morphological operations smooth the image, enhancing contour detection. The contour properties are analyzed, ensuring sufficient area and a specific number of vertices (12). The detected crosses are then added to a tracking list.

Square Detection:
Squares are detected by isolating white areas using a bitwise filter. Image moments confirm their shape and rule out elongated objects. Image moments also help get the coordinates of the object's centroid:

$$X_{\text{coordinate}} = \frac{M_{10}}{M_{00}}$$
$$Y_{\text{coordinate}} = \frac{M_{01}}{M_{00}}$$

-M['m00'] is the total sum of pixel intensities inside the contour. As we apply it to a binary image, M['m00'] represents the area of the contour.
-M['m10'] is the sum of x-coordinates weighted by pixel intensities (it measures the distribution of pixel intensities along the x-axis).
-M['m01'] is the sum of y-coordinates weighted by pixel intensities (it measures the distribution of pixel intensities along the y-axis).

The centroid can locate an object in an image, and is very easy to implement using moments. That is why we chose to calculate the centroid rather than the center of the square. Then, the system ensures the detected shape has an aspect ratio close to 1:1.

Cardboard Detection:
This function filters brown hues in the image using predefined color thresholds. The system validates detected contours by ensuring a minimum area requirement is met.

Red Line Detection:
To ensure the user stays within the safe path, red lines are detected and

their intersection point is calculated: - The function checks if valid contours exist. - The image is converted to grayscale if necessary. - `cv2.HoughLinesP` detects line segments.[4] - The slope m and y-intercept b of each line are computed. - The intersection point is used to determine deviation from the center

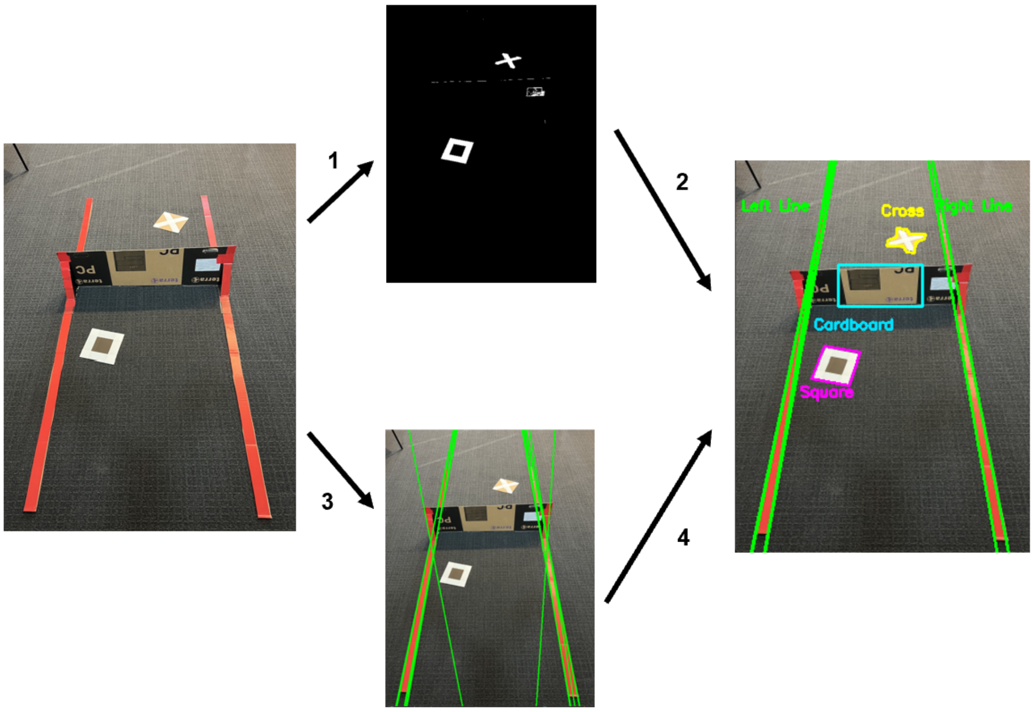

Figure 2: Diagram of the algorithm

3.3 Integration with Raspberry Pi

The system is designed to run efficiently on the Raspberry Pi, utilizing its processing capabilities for real-time image and sensor analysis. The integration includes:

- Configuring GPIO pins for sensor input and haptic feedback.
- Optimizing the image processing pipeline to maintain a smooth framerate.
- Managing power consumption to ensure continuous operation over long periods.

For hardware connections, we have wired the TF Luna LiDAR sensor, the haptic feedback system, and the audio module to the Raspberry Pi's GPIO pins. The LiDAR sensor communicates using the UART interface, requiring only four connections: power (VCC), ground (GND), and the serial communication pins (TXD/RXD).

3.4 Results and Performance

The system was tested in various environments to evaluate its detection accuracy. We did 35 tests. For each one, we wrote the objects that were detected. The table below presents the success rate for each type of object.

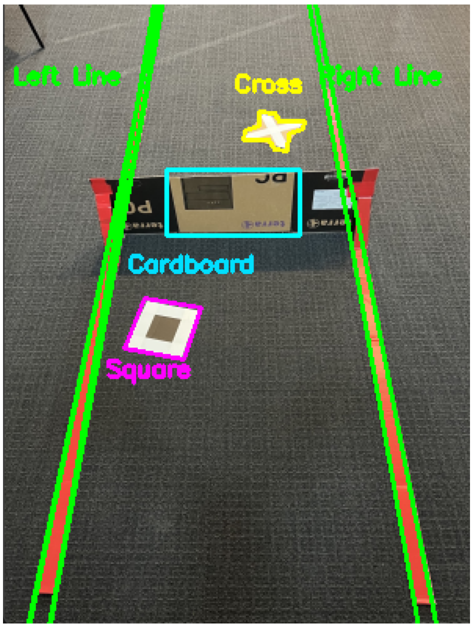

Figure 3: Final result of the algorithm

Obstacle	Detection Rate
Cross	94%
Square	91%
Cardboard	89%
Red Line	97%

Table 1: Detection rates for different obstacles

Limitations Encountered:
- Challenges in detecting obstacles under low-light conditions.
- Sensitivity of LiDAR sensor affected by reflective surfaces.
- Potential false positives in object classification due to background noise.

Despite these limitations, the system provides reliable navigation assistance, offering real-time haptic and auditory feedback for visually impaired users.

4 Carbon Footprint and Cost

4.1 Carbon Footprint Analysis

Lifecycle Analysis:

- **Manufacturing of components:** Re-use of a second-hand Raspberry Pi, camera, haptic sensors, and other elements.
- **Transportation:** Importation of components and material movement for assembly.
- **Utilization:** Energy consumption during prototype operation.
- **End of Life:** Assessing recycling/disposal impact, including emissions, energy use, and non-recyclable materials.

The major impact relates to electronic component manufacturing. Optimization includes reducing energy use via algorithm efficiency and exploring recycled materials. We chose a cardboard box for the device to be eco-friendly, light, cheap, resistant, and avoid energy-consuming custom manufacturing like 3D printing. However, it is less effective under bad weather.

4.2 Sustainable and Modular Design

To ensure sustainability, we focused on modularity and reusability:

- **Modular Components:** Raspberry Pi, haptic sensors, camera, and LiDAR can be easily removed, replaced, and reused.
- **Eco-Friendly Housing:** Housed in the Raspberry Pi's original cardboard box, reducing extra materials.
- **Low Environmental Impact:** Fully dismantlable design minimizes electronic waste and supports a circular economy.

4.3 Privacy Protection

- **Camera Privacy and Cybersecurity:** Local processing on the Raspberry Pi eliminates remote/cloud risks.
- **User Privacy:** No image or personal data storage, ensuring complete anonymity.

4.4 Cost Breakdown

To remain sustainable and competitive, we used second-hand components. Table 2 summarizes the costs.

Components	**Cost**
Raspberry Pi	30 €
Camera	15 €
TF Luna	10 €
Audio module	10 €
Haptic sensor	10 € each
Total	**85 €**

Table 2: Cost of all components

Additional costs:

- **Audio module:** 10 € (basic earbuds) or 100 € (bone conduction headphones so that your ears are free).
- **Software:** Python, OpenCV (Open source, free).
- **Extras:** Cardboard box, Velcro, adhesive tape (5 €).

Total estimated cost: 100 €.

5 Conclusion

5.1 Summary of Achievements

The development of a functional prototype integrating modern technologies, including Raspberry Pi, image processing, and haptic sensors, has been successfully achieved. A system capable of effectively guiding visually impaired persons (VIPs) along a marked path was also implemented, ensuring real-time detection of obstacles and providing both haptic and audio feedback.

5.2 Challenges and Lessons Learned

Lessons learned: During initial testing, our system struggled to detect obstacles accurately and sometimes identified non-obstacles as obstacles. We realized that modifications to the code and parameter adjustments were necessary, highlighting the importance of rigorous testing to understand system limitations and improve performance.

Additionally, we underestimated the time required for marketing aspects, such as poster design and presentation preparation. As deadlines approached, we encountered delays and increased pressure. To address this, we introduced weekly progress reviews and improved task distribution, teaching us the importance of maintaining productivity even when the core code is operational.

5.3 Potential Improvements

Several enhancements could improve the system:

- **Implementation of adaptive algorithms:** Enhancing performance in varied environments (e.g., low light, reflective surfaces) to ensure reliability in all weather conditions (sunny, cloudy) and at any moment of the day (evening or even nighttime). This could be achieved by broadening the range of detectable colors while preventing false detections.
- **Integration of renewable energy sources:** Exploring the feasibility of portable solar panels to power the device. This requires energy efficiency calculations, cost analysis, and discussions on wearability to determine practicality.
- **Customizable usage modes:** Introducing a configuration menu allowing users to personalize settings such as vibration intensity and sound volume. This feature could be a key selling point, enabling users to create and save preferred configurations by adjusting program parameters.

6 User Manual

Introduction

This manual explains how to use the wearable navigation system for visually impaired users. The device provides real-time haptic and auditory feedback for obstacle detection and navigation.

Setup and Wearing the Device

- Attach the Raspberry Pi-based module to a belt around the waist.
- Ensure it is securely fastened to prevent movement.
- The camera and LiDAR sensor must face forward for accurate detection.

Operation

- The system starts automatically when powered on.
- The camera detects objects (squares, crosses, red lines).
- The LiDAR sensor scans for obstacles.
- Haptic feedback vibrates left or right based on proximity to obstacles or red lines.
- Auditory feedback provides directional alerts.

Understanding the Feedback

Haptic Feedback:

- Right motor vibrates if near a red line on the right; left motor if on the left.
- Both vibrate if an obstacle is directly ahead.

  Audio Feedback:

- **"Cross/Square on right/left/center"**: Indicates object position.
- **"Obstacle ahead"**: A cardboard obstacle is detected.

Safety and Maintenance

- Ensure sufficient battery or power connection.
- Keep the camera and LiDAR sensor clean.
- Store in a dry place when not in use.

Figure 4: Prototype

7 Bibliography

[1] OpenCV Foundation. *OpenCV Documentation*. 2024. URL: https://docs.opencv.org/.

[2] Raspberry Pi Foundation. *Raspberry Pi Documentation*. 2024. URL: https://www.raspberrypi.com/documentation/.

[3] GeeksforGeeks. "How to Detect Shapes in Images in Python using OpenCV". In: (2023). URL: https://www.geeksforgeeks.org/how-to-detect-shapes-in-images-in-python-using-opencv/.

[4] OpenCV. "Canny Edge Detection and Hough Transform". In: (2024). URL: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html.

[5] Makers Portal. "Distance Detection with the TF Luna LiDAR and Raspberry Pi". In: (2022). URL: https://makersportal.com/blog/distance-detection-with-the-tf-luna-lidar-and-raspberry-pi.

8 Annex

Prototype Specifications

General Specifications

- **Dimensions**: 150 mm × 80 mm × 50 mm
- **Weight**: Approx. 450 g (including battery)
- **Power Supply**: 5V DC via USB-C (minimum 3A)
- **Battery**: 1820 mAh / 6.7 Wh (PiJuice HAT)
- **Autonomy**: Approx. 3-4 hours of continuous use

Component Specifications

Component	**Specification**
Raspberry Pi 4	Quad-core Cortex-A72 @ 1.5 GHz 2 GB LPDDR4-3200 SDRAM Wi-Fi 802.11ac, Bluetooth 5.0 2× USB 3.0, 2× USB 2.0
TF-Luna LiDAR	Measurement Range: 0.2 m – 8 m Accuracy: ±6 cm (0.2 m - 3 m), ±2% (3 m - 8 m) Refresh Rate: 1 – 250 Hz (adjustable)
PiCamera	8 MP Resolution Video: 1080p30, 720p60, 640×480p60/90 Sensor Size: 3.68 mm × 2.76 mm
Haptic Sensors	Operating Voltage: 5V DC Rotation Speed: 9000 RPM Rated Current: 60 mA

Table 3: Technical specifications of the prototype components

Method to Use

- **Obstacle Detection**: The LiDAR and camera analyze the surroundings in real time, detecting obstacles and providing feedback.
- **User Guidance**: Haptic sensors vibrate to indicate direction and obstacles, allowing intuitive navigation.