Cultivating Navigational Autonomy in the Visually Impaired: A Novel Approach with *VirtualEYE*

Aishwarya Singh
School of Computer Science and Engineering
Nanyang Technological University
Singapore
SI0001YA@e.ntu.edu.sg

Smitha K G
School of Computer Science and Engineering
Nanyang Technological University
Singapore
smitha@ntu.edu.sg

***Abstract*—This research project presents *VirtualEYE*, an advanced indoor navigation system for visually impaired individuals, integrating Bluetooth Low Energy (BLE) technology and innovative application design. The system, initially implemented at the university premises, uses strategically placed BLE beacons and a purpose-built Android mobile application to enhance indoor navigation. The integration of Radio Frequency Identification (RFID) beacon localization techniques with computer vision-based object detection provides real-time, voice-enabled assistance. Experimental results showed a 20.48% decrease in navigation time for visual navigation and a 37.34% decrease for non-visual navigation across all participants, along with increased navigational confidence. This validates *VirtualEYE*'s potential to improve the quality of life for visually impaired individuals and contributes to the broader discourse on the role of AI in healthcare.**

***Index Terms*—Indoor navigation systems, Bluetooth Low Energy (BLE) technology, Spatial orientation, Assistive technology, Computer vision-based object detection, Artificial Intelligence**

I. Introduction

The past decade has seen a significant rise in navigation tools, particularly for outdoor localization. However, the complex nature of indoor navigation, with signal blockages and multi-path interferences, necessitates innovative solutions. Existing indoor navigation systems [10] [11], despite advancements, have limitations. These systems are often not diverse and are applicable only to specific scenarios. They often rely on hardware-dependent and environment-specific solutions. However, these solutions may lack effectiveness in diverse real-world scenarios. With approximately 2.2 billion visually impaired individuals globally [14], there is a pressing need for a robust system that can be used for a wide range of indoor navigation scenarios. Such a system should not require expensive and bulky hardware or installation costs, and should be readily accessible.

A. Objective and Scope

The primary research objective is to develop *VirtualEYE*, an indoor navigation system for the visually impaired. This Android application interfaces with FUJITSU FWM8BLZ02 BLE Beacons for enhanced auditory and kinaesthetic navigation. The project integrates *VirtualEYE*, a Flask-based Server, and BLE beacons, with the server handling computationally intensive tasks for real-time responsiveness. *VirtualEYE* offers features like obstacle detection, tactile and audio feedback, beacon localization, and visual navigation. Further sections discuss the system architecture, implementation, testing, and evaluation of *VirtualEYE*'s effectiveness.

II. Background Information

While outdoor navigation has improved with tools like **Global Positioning System (GPS)** and **Global Navigation Satellite System (GLONASS)**, indoor environments pose challenges for these satellite-dependent systems due to signal obstructions and multi-path interference [7], [13]. This has led to the development of **Indoor Positioning Systems (IPS)** for indoor navigation.

A. Indoor Positioning System (IPS)

Indoor Positioning Systems (IPS) are vital for real-time localization in complex indoor environments. They use designs like self-positioning and infrastructure positioning, and technologies like **Radio Frequency (RF)**, vision-based methods, and inertial sensors [6]. Understanding these technologies is key for system selection, considering factors such as accuracy response time, availability, and scalability.

1) Radio Frequency-based IPS: Indoor positioning in diverse environments uses technologies like **Internet of Things (IoT)** devices, RFID devices, and Wi-Fi networks. RF-based IPS uses range-based and range-free methods.

- Range-based Radio Frequency IPS - Uses methods like signal propagation time, **Angle of Arrival (AoA)**, and RSSI to extract geometric information from nodes. RSSI-based algorithms calculate user-node distances using signal intensity and a log-distance path loss model (assuming signal attenuation aligns with proportional distance travel) [15]. Calibration involves collecting RSSI values at known positions, typically using regression methods, to estimate distances via the Maximum Likelihood Estimator [17].
- Range-free Radio Frequency IPS - Uses connection data to estimate positions without measuring the reach at a node. Proximity algorithms determine the user's position based on nearby nodes [18]. The fingerprinting location system divides the region into small cells and collects data to establish a database, guiding the approximation of

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00081

positions by comparing collected data to stored records. The downside is the effort required to build the database.

2) Inertial Sensors-based IPS: Independent of physical infrastructure, these systems use the geographic coordinate system for absolute positioning. Real-time inertial location is enabled by sensors in **Inertial Measurement Units (IMUs)**, which include accelerometers, gyroscopes, and magnetometers. Despite the scalability advantage and cost-effectiveness, especially with smartphone integration, inertial drift introduces accuracy considerations [12].

3) Computer Vision-based IPS: These systems use cameras and image processing algorithms for indoor navigation, with users capturing and processing images or videos on smartphones [8]. While single-sensory IPS face limitations, hybrid systems enhance accuracy and reliability. Integrating artificial intelligence and object detection techniques in IPS contributes to enhanced spatial awareness. Algorithms such as convolutional neural networks analyze visual data to identify obstacles or landmarks, refining the IPS's overall efficacy.

B. Bluetooth Low Energy (BLE) Beacons

These devices use Bluetooth 4.0 standard technology to transmit radio signals with alphanumeric data at regular intervals [1]. They operate at 2.4GHz with Gaussian Frequency Shift Keying modulation, achieving a power-efficient throughput of around 300kbps. This allows them to run on coin-cell batteries for long periods. BLE beacons broadcast configurable 'IDs' and initiate a discovery process with advertising, scanning, and connection. In passive scanning, devices identify each other without feedback to the advertiser. The scanner selects an advertiser based on advertising data like device name and RSSI. The BLE Link Layer connection involves an Initiator and an Advertiser, leading to data packet transmission [9].

This section underscores the significance and challenges of IPS in complex environments. Drawing from these insights, VirtualEYE aims to leverage the strengths of these technologies while mitigating their limitations. By integrating RF-based methods with Inertial Sensors and Computer Vision techniques, VirtualEYE seeks to enhance accuracy, reliability, and scalability in indoor positioning. Furthermore, the use of BLE Beacons is expected to provide efficient and effective real-time localization, thereby improving the overall user experience in indoor navigation. This endeavor aligns with the broader goal of advancing indoor navigation systems to match, if not surpass, the effectiveness of their outdoor counterparts.

III. System Architecture

This section delves into the architecture of the proposed indoor navigation system, progressing from the initial planning and considerations to a detailed description of the final system architecture of *VirtualEYE.*

A. Initial Considerations

The Fig. 1 architecture shows components interacting via the Android app *VirtualEYE*. User instructions initiate WebSocket communication. The smartphone's Bluetooth manager scans for BLE beacons, verified by the server. ImageAI API processes image frames for obstacle detection, with identified obstacles relayed back via the **TTS** engine. The system uses BLE beacons for indoor navigation and real-time path calculation. Challenges like unreliable audio input, insufficient environmental data, and feedback delays necessitated architecture re-evaluation.

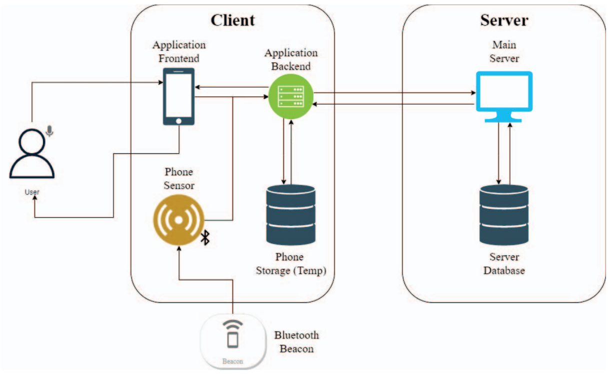

Fig. 1: VirtualEYE's Initial System Architecture - This diagram illustrates the user's interaction with the application and the server, emphasizing the use of BLE beacons for indoor navigation and the ImageAI API and YoloV4 model for obstacle detection. The architecture also highlights the challenges encountered during implementation, which necessitated further refinement of this design.

B. Final Architecture

The refined architecture introduces new input and feedback sources to address initial challenges. User interaction is facilitated through UI, audio, and shake inputs, while feedback is conveyed visually, audibly, and tactilely. Additional magnetometer and accelerometer sensors complement the Bluetooth scanner and camera for environment interaction, providing bearing, direction, speed, and movement data for localization during navigation. The Flask server replaces WebSocket to enhance communication efficiency, and hosting on a Linux Virtual Machine on Azure Cloud improves overall responsiveness. The final architecture is detailed in Fig. 2.

The landing screen initiates the Text-to-Speech (TTS) controller and sensor manager, delivering a welcome message and accepting kinetic or UI input (shaking or button press). This input redirects the user to either Free Roam or Regular Navigation mode. Regular Navigation Mode supports vision-based navigation, integrating sensors for direction, bearing, and distance calculations. BLE beacons enhance proximity ranging, validating data with sensors. The beacon is considered in range based on the distance calculation from RSSI values (Methodology), allowing for seamless real-time navigation updates. Free Roam Mode adds a camera stream for obstacle detection, awaiting kinetic input for Assisted Navigation. This mode mimics the Regular Navigation logic but prioritizes comprehensive feedback for visually impaired users, including distance updates, tactile feedback, and continuous environmental information. To enhance obstacle detection, images of common obstacles, including chairs, humans, and other objects, were placed in the testing environment for the app.

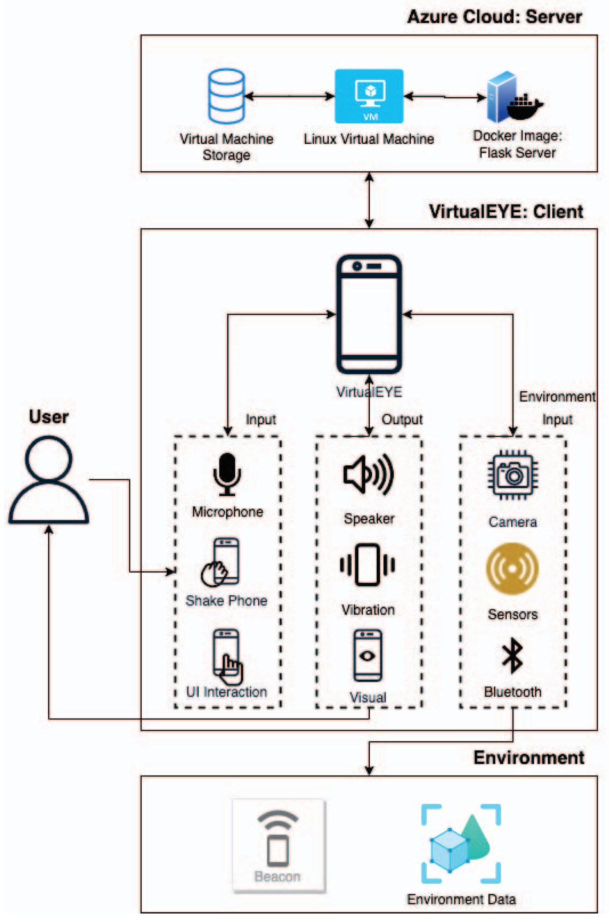

Fig. 2: Enhanced VirtualEYE System Architecture - This diagram illustrates the updated architecture with diversified user inputs and outputs, additional sensors for environment interaction, and an efficient Flask server hosted on Azure Cloud.

Pictures of these obstacles were taken and used to train a pre-trained convolutional neural network model (MobileNetV2) using transfer learning. While effective, performance may vary with objects not seen during training. To address this, diversification of the training dataset is planned. Further strategies, such as data augmentation, are being considered to enhance the model's generalization capabilities. With the server hosted on the cloud, computation became much faster, improving the responsiveness of the app and the overall user experience.

IV. Methodology

A. Indoor Mapping

University premises being our chosen testing environment, were mapped using SketchFab with a reference from the university's official map [3]. Integration into the Android app utilized the Subsampling Scale Image View library [5], offering efficient display of large maps and custom markers for improved indoor mapping and navigation in *VirtualEYE*.

B. Beacon Detection and Distance Estimation

The BLE beacons used in this project require custom code for tailored use due to the lack of an SDK. They only advertise MAC Address and RSSI. The hardware details were obtained from the manufacturer's document [2], and configuration and testing were done to optimize indoor navigation and localization. The Android BluetoothAdapter communicates with the beacons. The code starts a BluetoothLeScanner if the adapter is not null, which finds BLE devices and calls the callback method (mScanCallback). Scan settings are set for real-time scanning with filters based on MAC address [4].

Multiple experiments were conducted to establish a distance estimation method for indoor localization using RSSI values. The first experiment maintained a constant distance while noting RSSI values at intervals, resulting in averaged values. Various equations (Linear, Quadratic, Root) were derived from these values, but accuracy was limited to distances under 200cm. The second experiment introduced an obstacle (wall) for a more realistic scenario, yet accuracy remained insufficient for the project's purpose. The final experiment applied the literature method, using measured RSSI values to calculate the distance d using the equation -

$$d = \frac{10^{Tx-\text{RSSI}}}{10n} \tag{1}$$

where Tx is the transmitted signal power, and n is the path loss exponent. The optimized values for Tx and n were determined through trials, demonstrating improved accuracy compared to the previous experiments. Fig. 3 and Fig. 4 demonstrate the beacon experiment setup and the result for the same.

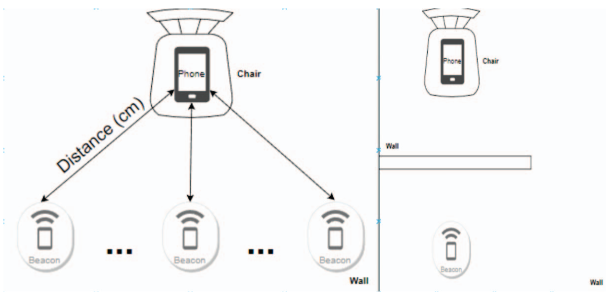

Fig. 3: Experimental Setup for Proximity Ranging in Indoor Localisation - This setup, designed to derive a custom relationship between RSSI values and beacon distance, involved multiple experiments with constant distances and varied beacon angles. An additional obstacle was introduced to simulate a realistic indoor scenario and account for potential obstacles.

V. System Implementation

A. Server

The server, built in Python 3.8.5 using Flask, handles client-server communication and shortest path calculation. **Breadth-First Search (BFS)** was the best algorithm among the tested ones for its efficiency in navigating the server's graph structure. Cloud migration was done through Docker, allowing deployment on Azure Linux Virtual Machine to improve performance and internet connectivity, removing the need for client and server to be on the same network.

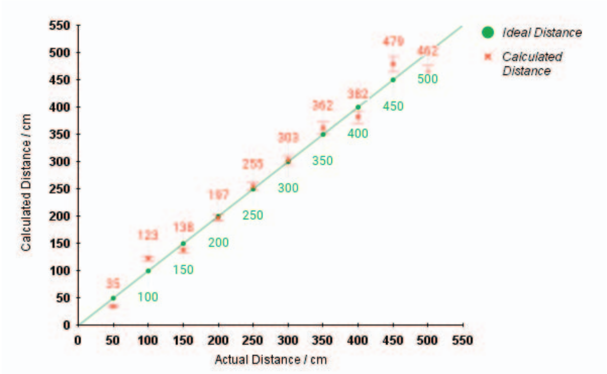

Fig. 4: Scatter plot illustrating the comparison between the Calculated Distance and the Ideal Distance against the Actual Distance. The Ideal Distance is depicted as a linear correlation with the Actual Distance, serving as the benchmark. The Calculated Distances, represented by red crosses, exhibit variability around the Ideal Distance, with the deviation captured by the error bars (to account for random errors during measurement).

B. Client

The client-side application, *VirtualEYE*, created in Kotlin using Android Studio, includes TTS control, shake input, HTTP requests, voice input, compass functionality, and object detection. TTS and sensor initialization provide voice output upon accelerometer changes. HTTP requests, enabled by Kotlin coroutines, allow asynchronous communication with the server for path information retrieval. Voice input functionality uses the SpeechRecognizer class, handling user commands through speech recognition. Compass functionality determines device orientation in relation to cardinal directions. Object detection uses camera preview, utilizing the aforementioned trained model to identify and announce the recognized objects as demonstrated in Fig. 5

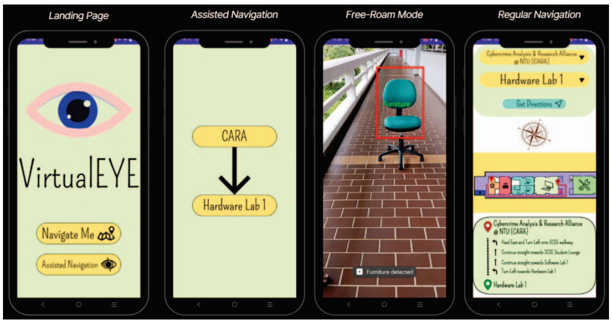

Fig. 5: An illustration of the different functionalities of *VirtualEYE*

VI. Experimentation

The controlled variables for the experiments conducted involved visually-abled participants (combination of university and non-university students), obstacles (chairs, people), a predefined navigation path, and the use of an Android device. An evaluation metric was set up to measure the effectiveness of the indoor navigation system. This metric included the following variables:

- **Metric 1** - Time: Measured using a timer - time taken for the participants to navigate from Landmark A to Landmark B across all test cases. This variable evaluates the efficiency of the algorithms used in the system.
- Participants' Feedback: The ratings (Ratings on a scale of 0 (Impossible) to 5 (Excellent)) and comments given by the participants on various metrics at the end of each test case. These metrics were:
  - **Metric 2** - Ease of navigation
  - **Metric 3** - User Interface / Non-visual Feedback / Visual Cues (Depending on the test case)
  - **Metric 4** - User Experience

The experimentation involves a series of structured test cases to rigorously evaluate the performance of the proposed system. Each test case was meticulously designed to assess specific functionalities and aspects of *VirtualEYE*. The following test cases were implemented:

- **Test Case 1: Visual Navigation without VirtualEYE**: Participants navigated through a predefined route relying solely on visual cues. This served as a control case for subsequent visual navigation tests.
- **Test Case 2: Visual Navigation with VirtualEYE**: Similar to Test Case 1, participants underwent visual navigation, but with the assistance of VirtualEYE.
- **Test Case 3: Non-Visual Free Roaming without VirtualEYE**: Participants navigated freely without visual assistance, emphasizing the reliance on non-visual cues. This scenario was set as a control for the non-visual navigation tests.
- **Test Case 4: Non-Visual Free Roaming with VirtualEYE**: Building upon Test Case 3, participants engaged in non-visual free-roaming activities with the support of VirtualEYE.
- **Test Case 5: Non-Visual Navigation without VirtualEYE**: Participants faced the challenge of non-visual navigation, blindfolded and without the aid of VirtualEYE. This scenario assessed the inherent difficulties in navigating unfamiliar surroundings without visual input.
- **Test Case 6: Non-Visual Navigation with VirtualEYE**: Extending the previous test, participants navigated non-visually with the assistance of VirtualEYE.

Table I and Table II illustrate the results of these test cases across 6 participants - 3 University and 3 Non-University students.

VII. Evaluation of Results

Across the six test cases, *VirtualEYE* demonstrated superior performance, notably reducing navigation time in both visual and non-visual scenarios. Fig. 6 illustrates the time metric results.

Test Case 1 and Test Case 5 were the control cases for visual and non-visual navigations. For both scenarios, *VirtualEYE* reduces the time taken for the participants. The percentage reduction in time for the University students from Test Case 1 to Test Case 2 is 12.50%. This may not be due to *VirtualEYE*

TABLE I: Summary of Quantitative Results - This table summarizes the quantitative results for all the test cases, presenting average time measurements and ratings for both categories of university and non-university participants.

Test Case	Participants	Metric-1	Metric-2	Metric-3	Metric-4
Test Case 1	University	01:12	4.0/5.0	4.0/5.0	4.0/5.0
Test Case 1	Non-University	02:03	3.0/5.0	3.5/5.0	3.45/5.0
Test Case 2	University	01:03	4.0/5.0	4.25/5.0	4.5/5.0
Test Case 2	Non-University	01:28	4.0/5.0	4.5/5.0	4.5/5.0
Test Case 3	University	-	0.0/5.0	0.0/5.0	0.0/5.0
Test Case 3	Non-University	-	0.5/5.0	0.2/5.0	0.0/5.0
Test Case 4	University	-	3.0/5.0	3.0/5.0	3.0/5.0
Test Case 4	Non-University	-	3.2/5.0	3.9/5.0	3.5/5.0
Test Case 5	University	05:32	0.6/5.0	0.5/5.0	0.13/5.0
Test Case 5	Non-University	05:02	0.3/5.0	0.2/5.0	0.37/5.0
Test Case 6	University	03:14	3.5/5.0	4.3/5.0	4.1/5.0
Test Case 6	Non-University	03:22	4.18/5.0	4.58/5.0	4.38/5.0

TABLE II: Summary of Qualitative Feedback - This table provides a concise summary of qualitative feedback across various test cases, capturing insights from both university and non-university participants.

Test Case	Feedback - University	Feedback - Non-University
Test Case 1	Following visual cues was simple; familiar surroundings.	Took a while to understand directions; lack of familiarity made it confusing.
Test Case 2	Application helped keep track of direction and location.	Easier navigation with regular feedback about the current location.
Test Case 3	Very uncertain of surroundings; difficult to roam without fear of collision.	Afraid to take steps without knowing what or whom walking into.
Test Case 4	Much more confident with the application; better understanding of surroundings.	Process was easier, better awareness of obstacles.
Test Case 5	Most difficult exercise; no idea where heading; uncertain.	Experience similar to Test Case 3; hesitant and uncertain of direction.
Test Case 6	Non-visual cues aided the most in navigation; vibration and audio feedback increased confidence.	Surprised by the effectiveness; continuous feedback made navigation easy.

alone, as the participants could have been familiar with the chosen university premise. The non-university students, who were new to the premise, had a higher percentage reduction in time for visual navigation (28.46%), hence indicating the effectiveness of *VirtualEYE*. For Test Case 5 and Test Case 6 (non-visual navigation), prior area knowledge did not have a profounding impact as the participants were blindfolded. The percentage reduction in times for both participants from both categories were 41.57% and 33.11%, again demonstrating the power of *VirtualEYE*.

Additionally, *VirtualEYE* is also successful in non-visual free-roaming scenarios (Test Case 3 and Test Case 4), as showcased Table I and Table II. It aids in significantly boosting participants' confidence using computer vision and Artificial Intelligence, highlighting its versatility and comprehensive utility. The qualitative feedback in Table II revealed the strengths and weaknesses of *VirtualEYE*. The participants found *VirtualEYE* very effective in non-visual navigation and

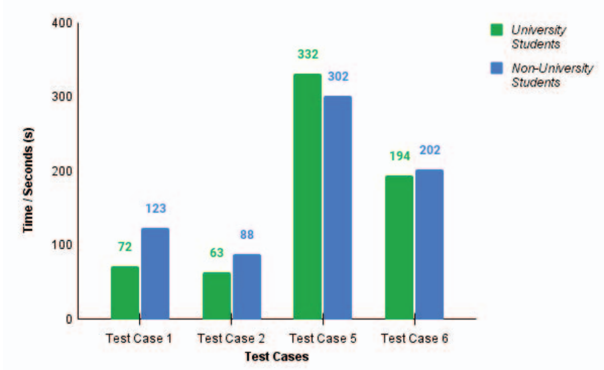

Fig. 6: Comparative Analysis of Evaluation Metric: Time (in seconds) for Test Cases 1, 2, 5, and 6 between University and Non-University Students. The bar graph illustrates the distinct time durations taken by both groups to complete the respective test cases.

free-roam modes. The constant feedback and interaction with the user made it a capable navigation tool for the blindfolded participants. Visual navigation was most effective for the non-university students. The quantitative feedback/scores in Table I give a statistical view of the system's performance. For all test cases, *VirtualEYE* improves user ratings across all metrics. Fig. 7 depicts the average user ratings for both the participant groups across all the test cases. There are improvements in ratings in all the test cases, significantly in vision-less scenarios. The metrics most affected by the application are user experience and visual cues/non-visual feedback. These ratings are good indicators for *VirtualEYE*'s performance. Table III presents a comparative analysis of the change in ratings between the controls and their respective test cases. Standard Deviations of average participant ratings are calculated to provide a comprehensive insight into the variability and consistency in the responses among the two distinct participant groups. These values range from 0.00 to 2.00, indicating the spread of the data from the mean for each metric.

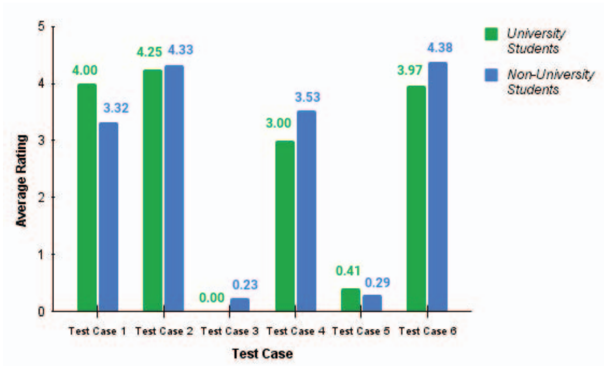

Fig. 7: Comparative Analysis of Average User Ratings across all Test Cases for University and Non-University Students. The bar graph represents the six test cases on the x-axis and their corresponding average ratings on the y-axis, ranging from 0 to 5.

Following the qualitative feedback, a survey based on a

TABLE III: Comparative Analysis of User Experience Metrics for University and Non-University Students. This table presents the standard deviations (SD) for various user experience metrics, including ease of navigation, visual cues, non-visual feedback, and overall user experience, segregated between university and non-university students. Each metric is evaluated across three distinct test cases (1 to 2, 3 to 4, and 5 to 6)

Test Cases	Metrics	SD: University Students	SD: Non-University Students
	Ease of Navigation	0.00	0.50
1 to 2	Visual Cues	0.13	0.50
	User Experience	0.25	0.53
	Ease of Navigation	1.50	1.35
3 to 4	Non-visual feedback	1.50	1.85
	User Experience	1.50	1.75
	Ease of Navigation	1.45	1.94
5 to 6	Non-visual feedback	1.90	2.19
	User Experience	1.99	2.00

recorded video demonstration [16] was conducted to further assess the system's performance. The survey results, illustrated in Fig. 8, showcase the effectiveness of *VirtualEYE*, providing an additional layer of validation beyond participant feedback. The following questions were asked, with the same rating scale from 0 to 5 -

- **Q1**: How likely are you to use *VirtualEYE* for indoor navigation in settings with high footfall such as Airports and Malls (For Visual Navigation)?
- **Q2**: How effective do you think is *VirtualEYE*'s interface in easing indoor navigation for its users?
- **Q3**: How effective do you think is *VirtualEYE* in navigating the visually impaired in unfamiliar indoor settings?
- **Q4**: How effective do you think the vibration and audio based feedback is for the visually impaired users when navigating indoors using *VirtualEYE*?

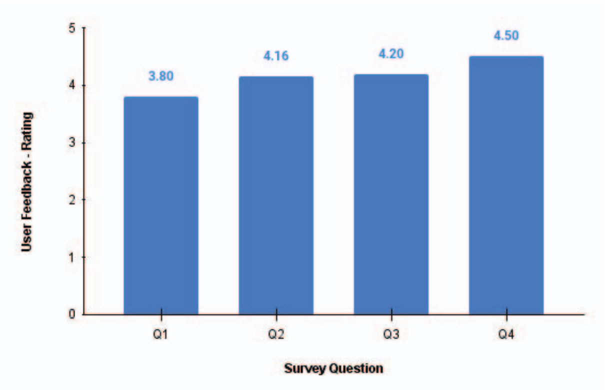

Fig. 8: Bar Graph Representing User Ratings - Survey Feedback. This figure illustrates the average user feedback ratings for four different survey questions (Q1 to Q4). The x-axis represents the survey questions, while the y-axis indicates the user feedback rating, ranging from 0 to 5.

VIII. Conclusion

VirtualEYE, a novel solution for indoor navigation, leverages BLE beacons and computer vision for landmark recognition and obstacle detection. Unlike existing systems, which often rely on specific environmental features or specialized hardware, VirtualEYE is versatile and robust, designed for a wide range of indoor navigation scenarios. Future developments include integrating wearable technology, enabling multi-floor navigation, leveraging augmented reality, and enhancing obstacle detection. A crucial part of this future work involves testing with real visually-impaired individuals. This ensures that VirtualEYE is not only technically robust but also practically useful and user-friendly. Furthermore, VirtualEYE, aligning with the concept of AI in healthcare, aims to improve the well-being and independence of the visually impaired through techniques such as computer vision and speech recognition.This positions *VirtualEYE* as a significant contribution to the field, addressing the limitations of existing systems.

References

[1] Bluetooth® low energy discovery process. https://microchipdeveloper.com/wireless:ble-link-layer-discovery.

[2] Fujitsu beacon documentation. https://www.mouser.com/ProductDetail/Fujitsu/FWM8BLZ02-109042?qs=JUmsgfbaopSMsHNjonqsbw%3D%3D.

[3] Official university map. https://maps.ntu.edu.sg/#/ntu/d386ffa80e4e46f286d17f08/poi/search.

[4] Scancallback : android developers. https://developer.android.com/reference/android/bluetooth/le/ScanCallback.

[5] Subsampling image library for android. http://davemorrissey.github.io/subsampling-scale-image-view/javadoc/.

[6] A. Alarifi, A. Al-Salman, M. Alsaleh, et al. Ultra-wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, page 707, 2016.

[7] D. Dardari, P. Closas, and P. M. Djuric. Indoor tracking: Theory, methods, and technologies. *IEEE Trans. Veh. Technol.*, 64:1263–1278, 2015.

[8] F. Fraundorfer and D. Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics Autom. Mag.*, 19:78–90, 2012.

[9] M. Ji, J. Kim, J. Jeon, and Y. Cho. Analysis of positioning accuracy corresponding to the number of ble beacons in indoor positioning system. In *2015 17th International Conference on Advanced Communication Technology (ICACT)*, 2015.

[10] Masaki Kuribayashi, Seita Kayukawa, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima. Linechaser: A smartphone-based navigation system for blind people to stand in lines. In *Proceedings of the Conference*, 2021.

[11] Deren Li Yujin Chen Guangyi Guo Zhipeng Cao Mengyun Liu, Ruizhi Chen and Yuanjin Pan. Scene recognition for indoor localization using a multi-sensor fusion approach. *Sensors*, 17(12), 2017.

[12] E. Munoz Diaz. Inertial pocket navigation system: Unaided 3d positioning. *Sensors*, 15:9156–9178, 2015.

[13] A. Navarro, J. Sospedra, R. Montoliu, et al. Geographical and fingerprinting data to create systems for indoor positioning and indoor/outdoor navigation. challenges, experiences and technology roadmap intelligent. *Data Centric Syst.*, pages 1–20, 2019.

[14] World Health Organization. Blindness and visual impairment, 2023.

[15] F. Palumbo, P. Barsocchi, S. Chessa, and J.C. Augusto. A stigmergic approach to indoor localization using bluetooth low energy beacons. In *Proceedings of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Karlsruhe, Germany, August 2019.

[16] Aishwarya Singh. Virtualeye: An indoor navigation tool for the visually impaired. https://www.youtube.com/watch?v=6d57YgNWh-w, 2023. 01/12/23.

[17] A. Zanella. Best practice in rss measurements and ranging. *IEEE Commun. Surv. Tutor.*, 18:2662–2686, 2016.

[18] Y. Zhao, C. Fritsche, F. Yin, et al. Sequential monte carlo methods and theoretical bounds for proximity report based indoor positioning. *IEEE Trans. Veh. Technol.*, 67:5372–5386, 2018.