2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS) | 979-8-3503-8311-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/COMSNETS59351.2024.10427277

SonicGlass: An Obstacle Detection and Navigation System Using Smartglass-Based Ultrasonic Sensors

Hrishikesh Govindrao Kusneniwar, Surjya Ghosh, Sougata Sen
Dept. of Computer Science and Information Systems
BITS Pilani, Goa campus, India

***Abstract*—Enabling continuous obstacle detection and providing real-time navigational assistance for people with visual impairment allows them to navigate in an indoor space without dependence on others. In this paper, we present the design, development, and evaluation of SonicGlass, a wearable obstacle detection and indoor navigation system. SonicGlass consists of a custom 3D printed wearable smartglass that can replace the individual's existing smartglasses. The smartglass is embedded with multiple sensors and microcontrollers. We performed a user study with 10 participants to determine the performance of SonicGlass and observed that SonicGlass could easily guide individuals in an indoor space while detecting obstacles with an F1-score of 84.7% (at 100% precision), and yielded a localization error of 1.35 meters when using a very simplistic regression model. Overall, SonicGlass can be a helpful tool for people with visual impairment to navigate in indoor spaces without assistance.**

***Index Terms*—smartglass, Accessibility, Indoor Navigation, Wearable Sensing, Obstacle Detection.**

I. INTRODUCTION

According to a 2015 report, nearly 253 million people worldwide suffer from visual impairment, of which 36 million suffer from complete blindness [1]. Recent advances in mobile and wearable technologies have enabled researchers to develop various systems for the Blind and Visually Impaired (BVI) individuals. One such category of systems assists the BVI individuals in navigating through indoor spaces while avoiding obstacles [2]. These systems usually rely on users donning bulky custom-made devices [2], or a wearable camera [3], or they use infrastructure sensors [4]. Although these systems enable individuals to navigate through indoor spaces while avoiding obstacles, however, they pose some major challenges. Firstly, BVI individuals are expected to don devices in addition to their regular attire. Additional devices often introduce unnecessary burdens on individuals. Secondly, although it is easy to capture images from cameras and apply image processing to detect objects in the image, however, cameras often raise serious privacy concerns for both users as well as bystanders [5]. Thirdly, purely infrastructure sensors (non-camera-based systems that do not expect any identifiers on individuals) often fail to identify individuals, thus failing to provide targeted personalized notifications. These challenges impact the usability and acceptability aspects of such devices.

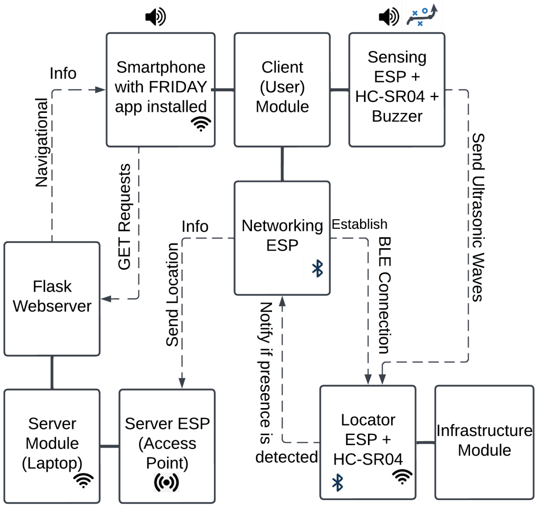
We developed SonicGlass, a smartglass-based system that overcomes these aforementioned challenges without overburdening the individual or raising privacy concerns for bystanders. SonicGlass allows individuals to navigate easily in an indoor space while avoiding obstacles. In this paper, we describe the SonicGlass system, and approaches that we undertook to validate the system. SonicGlass consists of (a) a smartglass that the BVI individuals wears, (b) an infrastructure-based module that enables locating the user, and (c) a central server that compute movement-related details of the individual. SonicGlass's **key contributions** are:

- We designed an obstacle detection and indoor localization system using ultrasonic sensors and BLE radios. SonicGlass uses a custom 3D printed vision-blocking smartglass and a slightly modified form of several off-the-shelf components.
- We designed an obstacle path in our lab, and conducted a user study with 10 participants. Overall, our system was able to locate an individual with a Root Mean Squared Error (RMSE) of 1.35 meters, and obstacle detection with an F1-score of 84.7%.

II. RELATED WORK

Navigational assistance technologies for people with visual impairment have been explored by various researchers. Several works develop techniques for finding environmental obstacles and ensuring that the individual does not collide with them [6]. Drishti is one of the earliest works to explore the ultrasound sensor for enabling navigation in indoor spaces [2]. The authors developed a custom wearable device to locate a person in both indoor and outdoor spaces. However, the Drishti system is bulky and expensive, advocating for a cheaper and more usable solution. More recently, work such as ALVU has presented the possibility of real-time navigational assistance for BVI individuals [6]. In this work, the authors developed a wearable belt-like device and used the Time of Flight (ToF) sensor to detect obstacles. The use of multiple sensors and feedback mechanisms makes this solution energy-hungry, however. Another wearable location that researchers have explored is the bracelet – Abusukhon et al. developed a bracelet with an ultrasonic sensor and a microcontroller to help BVI individuals navigate in indoor spaces [7]. A recent work, similar to ours, relies on multiple ultrasonic and other sensors for assisting BVI individuals in indoor spaces [8]. However, similar to the ALVU system, this system also relies on more expensive sensors and actuators, reducing the overall battery life. The common alternative approach taken by several researchers focused on using a camera. Work such as that by Mustafa et al. used a 5-megapixel camera along with a Raspberry Pi to

Fig. 1: The overall system architecture of SonicGlass.



detect obstacles, and faces, and for navigational assistance [9]. Alternatively, Chaudary et al. transferred the video feed from a wearable camera to a caretaker who was sitting remotely [10].

III. SONICGLASS: DESIGN AND IMPLEMENTATION

The goal of this work is to develop a system that can help BVI individuals detect obstacles and navigate in an indoor space. To attain this, we have developed SonicGlass, a system that consists of a Server Module (located remotely, described in more detail in Section III-A), an Infrastructure Module (located in the space of interest, described in more detail in Section III-B), and a Smartglass Module (on the user to support mobility, described in more detail in Section III-C). Fig. 1 presents the overall system design.

A. Server Module (Server):

The server module consists of (a) a *microcontroller-based access point* attached to the server via its serial port, (b) *a server backend* that *models* and *controls* the server logic, (c) *a dynamic mapper* the provides the *view* (Graphical User Interface (GUI)) for an end user, (d) a Flask webserver that allows multiple Smartglass and Infrastructure Modules to connect to the Server Module simultaneously. Fig. 2a presents the working of the software on the server.

(a) The microcontroller-based access point acts as the interface between the server and other devices in the system. In our current implementation, we use a laptop running Ubuntu 20.04 as our server. The laptop is connected to an access point via its Serial port. We use an ESP32-Wroom microcontroller as our access point. The ESP32-Wroom microcontroller already has an inbuilt RF module with Wi-Fi capabilities, and thus using it as an access point is straightforward. The microcontroller routes the captured wireless data to the laptop.

(b) The server backend has complete system-related information about the location – the coordinates of infrastructure modules, the coordinates of all static obstacles, and the dimensions of the location all these information are provided to the server backend in form of static config files. The server backend is capable of handling multiple locations. When a new client connects to the server backend, the server module creates a new thread to handle this client. An incoming connection indicates that a smartglass module is within range of an area of interest (it can listen to the infrastructure sensors). The server module uses the incoming data to (a) determine the user's approximate location (based on the infrastructure module it is closest to, (b) determine the user's probabilistic trajectory by augmenting the current location and trajectory data with previous location data, (c) determine the subsequent responses that must be sent to the client module. The server module logs the received information and then passes the location information to the dynamic mapper module.

(c) The dynamic mapper module is a GUI module present on the server that continuously updates all user locations on a map present on the server side. This module reads a file that is constantly updated by the server backend module. If multiple users are present in the location of interest, a unique identifier is presented on the map to uniquely identify the user.

(d) The Flask webserver is a webserver that has been implemented using Flask, a micro web application framework. The webserver handles connections with the Smartglass Module as well as the Infrastructure Module over a HTTP connection. We utilize the Flask library for multi-client interaction. The webserver avoids any unnecessary communication with the Infrastructure and Server Modules if there are no updates to information; those modules use any cached data in case they do not hear response from the server. Such a strategy reduces communication and energy overheads.

B. Infrastructure Module (Locator):

The infrastructure modules (shown in Fig. 2b from the actual in-lab deployment) are mounted at pre-defined positions in the location of interest. Each of these modules consist of an ultrasonic sensor and a microcontroller.

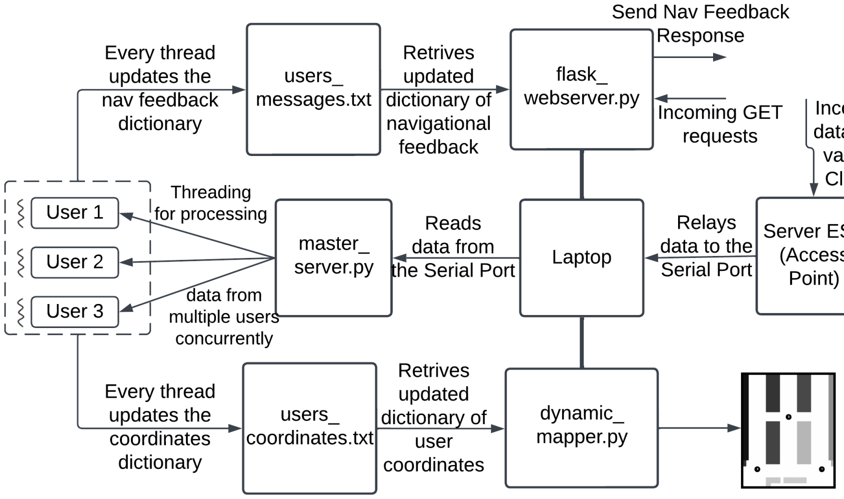
We modified the ultrasonic sensor (HC-SR04) in the Infrastructure Module physically; the ultrasonic sensor in the Infrastructure Module does not have its transmitter unit. We achieved this by de-soldering the transmitter and shorting the exposed pins. This change is observable in Fig. 2b This modification allowed the sensor's receiver to listen to any transmitted ultrasonic sound with a specific pattern.

The Infrastructure Module has a dedicated microcontroller. We used the ESP32-Wroom microcontroller in the Infrastructure Module. The microcontroller is powered by a 3.7V LiPo battery. However, the ultrasonic sensor requires 5V input. We used a boost converter to step up the voltage.

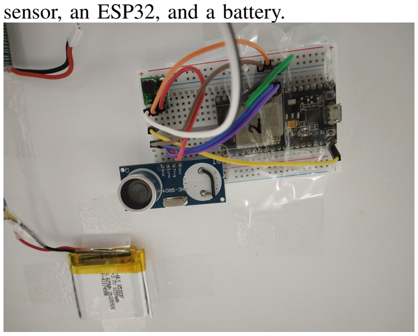
The ESP32 is capable of connecting to a mobile hotspot via Wi-Fi. We utilize this RF capability to perform Over The Air (OTA) programming. This enables remote re-programming of the microcontroller. Currently, the ESP32 continuously advertises a beacon over BLE until it hears a response from a client. The advertising interval is currently set to 1 second.

Fig. 2: The working details and components of various modules in the SonicGlass system.

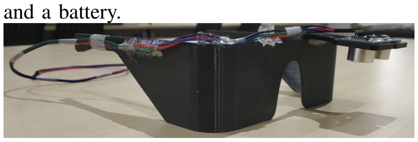
(a) Details of the functioning of the server module. It consists of an ESP32 connected to the laptop.



(b) The infrastructure module is ceiling-mounted and consists of a modified ultrasonic sensor, an ESP32, and a battery.



(c) The smartglass is a custom 3D-printed glass with two ESP32s, an ultrasonic sensor, a buzzer, and a battery.



Once a client establishes connection with the Infrastructure Module, the Infrastructure Module continuously polls the ultrasonic sensor to determine if the client module is directly below itself. Additionally, whenever the client is detected, the Infrastructure Module notifies the client that it is directly below that Infrastructure Module.

C. Client (User) Module (Smartglass):

The client module (shown in Fig. 2c) is a 3D-printed wearable smartglass. We designed a completely opaque smartglass that blocks the participant's vision. The smartglass currently is lightweight, comfortable to wear, and is capable of mounting a sensors and actuators onto itself. The smartglass constitutes one ultrasonic sensor, one buzzer, two ESP32 microcontrollers. Note that the ultrasonic sensor is placed in front of the left eye in Fig. 2c, while the buzzer is on the left handle (not in view in the figure). Of the two microcontrollers, one microcontroller is responsible for handling the sensing data, while the other is responsible for the networking. In future, we will combine these two functionalities onto a single microcontroller. The ESP32 is capable of running TinyML, and we will use that to advantage in future [11]. These components are powered by a 3.7V LiPo battery. In the current implementation, the battery is carried in a backpack.

Sensor data handling microcontroller: The ultrasonic sensor used on the smartglass is an unmodified sensor that is oriented downward (See Fig. 2c). The ultrasonic sensor detects obstacles based on the ToF of the ultrasonic wave. The same signal is also used by the receiver of the infrastructure sensors to determine the user's presence. The sensor data handling microcontroller triggers the ultrasonic sensor continuously, and computes the eye-to-floor distance. An ultrasonic sensor value lower than a pre-configured threshold indicates the presence of an obstacle. If an obstacle is detected, the microcontroller controls the buzzer to provide adaptive audio feedback to the user by adjusting the frequency of the buzzer. The frequency of the buzzing increases if the user is closer to the obstacle and reduces as the user moves away from the obstacle.

Smartphone app: The microcontroller is connected to a smartphone carried by users. We developed a custom Android app, the 'FRIDAY' app. This app enables the smartphone to connect with the webserver module in the server backend. The webserver continuously provides text feedback. The *FRIDAY* app converts this text to speech on a need to know basis.

Networking microcontroller: The microcontroller that is tasked with the networking duties is already deployed with the Server access point's credentials, and a bluetooth handle that is known to the smartphone and server. It connects to the server's ESP-based access point to send location data for further processing. The microcontroller already has a default path embedded onto it. However, when it sends its current location to the server, it anticipates a response from the server – the response must include the location of the *n* future access points that the user must visit. In case the client does not receive a response from the server, it continues with its pre-defined path. The microcontroller also performs continuous BLE scanning, connects with a close-range infrastructure module, and sends location information to the server.

IV. USER STUDY

To understand the feasibility of SonicGlass, we conducted a user study with 10 participants (5 males, 5 females) aged between 19 and 22 years in a laboratory setting. The location where the user study was conducted was a research lab. Participants were asked to visit the lab for a duration of 30 minutes. Prior to the user study, we installed 4 Infrastructure Modules in the lab. The participants were instructed to wear the Smartglass module before they entered the laboratory. For the study, participants were instructed to walk through an

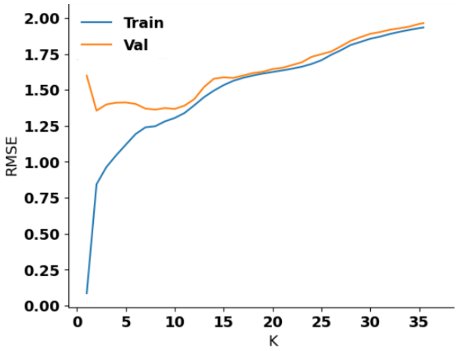


Fig. 3: Variation of RMSE for various values of K. We plotted the average value of K from the cross-validation.

indoor obstacle course. A research staff was always within some distance of the participant while they were performing the user study. Experiments were repeated twice for each participant. Participants were given a short break between the two data collection rounds. We collected a video while the participant performed the task to obtain the ground truth.

V. EVALUATION

In this section, we evaluate SonicGlass. First, we discuss the evaluation criteria (Section V-A) and then show system performance for each of these criteria (Section V-B, V-C).

A. *Evaluation Criteria*

Overall, using data from the user study, we aimed to answer the following three questions:

Q1. How accurately could SonicGlass identify the position of individuals in a closed space?
Q2. How precisely could SonicGlass detect obstacles?

B. *Performance of location detection*

To determine the location detection performance, we performed a leave one person out k-nearest neighbor-based (KNN-based) regression. We used the received RSSI from each of the Infrastructure Module as the feature. Thus, at every instance of time, we obtained 4 RSSI values. We used the video data to add a location label to each such row. We used 9 participant's data as the train set, and the left out participant's data as the validation set. We repeated this process 10 times so that each user was part of the validation set exactly once. We varied K (Number of RSSI-location vectors to consider) to determine its best value. For each value of K, we computed the distance difference between the actual location and the predicted location. We then used the distance for all locations to compute the RMSE value. Overall, we observed that the training error was the least for $K = 2$, however, we attribute this to overfitting. The performance for both the train and validation set start converging at approx. $K = 10$. Fig. 3 presents the variation of RMSE for different values of K. From the figure, we observe that at $K = 10$, the RMSE is 1.35 m. This indicates that if we have approximately 10 readings from a particular location, we can approximately determine the users' location.

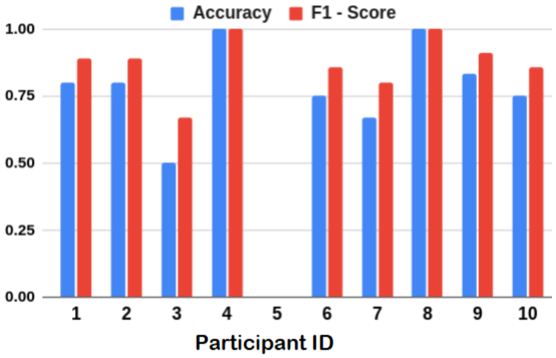


Fig. 4: User-wise obstacle detection accuracy & F1-score.

C. *Effectiveness in detecting obstacles*

To determine the effectiveness of SonicGlass's obstacle detection algorithm, we noted the total number of obstacles in the participant's path. There were three obstacles during each experiment round, resulting in 60 obstacle observations in the 20 rounds (two rounds per participant). However, in some cases, the participants did not encounter the obstacle as the obstacle was not in their navigation path. Overall, the participants encountered a total of 50 out of the total 60 obstacles. Among these 50 obstacles, SonicGlass was able to detect the obstacles with an accuracy of 73.5%. To understand if SonicGlass missed detecting obstacles or issued false positive alarms, we also computed the precision and recall of detecting obstacles and observed that the precision was 100%, while the recall was 73.5% – the overall F1-score was 84.7%. This result indicates that SonicGlass could precisely locate obstacles (i.e., there we no false positives). Fig. 4 presents the participant-wise accuracy and F1-score for SonicGlass. From the figure, we observe that for one participant (participant id: 5), the F1-score and accuracy are both zeros. This participant was taller than the others. Even though there were obstacles, the distance was above the fixed threshold, resulting in missing the obstacles.

VI. CONCLUSION

In this paper, we described the design of SonicGlass, a system to enable assistance-free navigational support to BVI individuals. The design of SonicGlass ensures that it is low cost and draws little power. The design also provides the capability of adding more components to the system in the future. To test the performance of SonicGlass, we conducted a user study and observed that SonicGlass could detect location with an RMSE of 1.35 m. The results indicate the possibility of building a practical obstacle detection & navigational system.

ACKNOWLEDGMENT

This work is supported partly by Science & Engineering Research Board's SERB-SURE project number SUR/2022/002735 and partly by BITS Pilani's grants GOA/ACG/2021-2022/Nov/05. All findings and recommendations are those of the authors and do not necessarily reflect the views of the funding institutes.

REFERENCES

[1] P. Ackland, S. Resnikoff, and R. Bourne, “World blindness and visual impairment: despite many successes, the problem is growing,” *Community eye health*, vol. 30, no. 100, p. 71, 2017.

[2] L. Ran, S. Helal, and S. Moore, “Drishti: an integrated indoor/outdoor blind navigation system and service,” in *IEEE Annual Conference on Pervasive Computing and Communications (PerCom).*, 2004.

[3] Y. H. Lee and G. Medioni, “Rgb-d camera based wearable navigation system for the visually impaired,” *Computer vision and Image understanding*, vol. 149, pp. 3–20, 2016.

[4] T. Grosse-Puppendahl, X. Dellangnol, C. Hatzfeld, B. Fu, A. Kuijper, M. R. Hastall, J. Scott, and M. Gruteser, “Platypus: Indoor localization and identification through sensing of electric potential changes in human bodies,” in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2016.

[5] R. Alharbi, S. Sen, A. Ng, N. Alshurafa, and J. Hester, “Actisight: wearer foreground extraction using a practical rgb-thermal wearable,” in *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2022, pp. 237–246.

[6] R. K. Katzschmann, B. Araki, and D. Rus, “Safe local navigation for visually impaired users with time-of-flight and haptic feedback device,” *IEEE Trans. on Neural Systems & Rehab. Engineering*, vol. 26, 2018.

[7] A. Abusukhon, “Iot bracelets for guiding blind people in an indoor environment,” *Journal of Communications Software and Systems*, vol. 19, no. 2, pp. 114–125, 2023.

[8] Y. Bouteraa, “Smart real time wearable navigation support system for bvip,” *Alexandria Engineering Journal*, vol. 62, 2023.

[9] A. Mustafa, A. Omer, and O. Mohammed, “Intelligent glasses for visually impaired people,” in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2022.

[10] B. Chaudary, S. Pohjolainen, S. Aziz, L. Arhippainen, and P. Pulli, “Teleguidance-based remote navigation assistance for visually impaired and blind people—usability and user experience,” *Virtual Reality*, vol. 27, no. 1, pp. 141–158, 2023.

[11] H. M. Singh, S. Agashe, S. Jain, S. Ghosh, A. Challa, S. Danda, and S. Sen, “(poster) insights from executing tinyml models on smartphones and microcontrollers,” in *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 2023, pp. 89–92.