University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Volume 14 No. 1, June, 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com

University of Ibadan
Journal of Science and Logics in ICT Research

Augmented Reality Navigation Assistance System for Visually Impaired Individuals

[1] **Okparaeke C. J.,** [2] **Sakpere A. B.**

[1]*Department of Computer Science, University of Ibadan – Nigeria*
[2]*Department of Statistics, University of Ibadan – Nigeria*
chukwuemekaokparaeke@gmail.com, ab.sakpere@ui.edu.ng

Abstract
Visually impaired people face a lot of issues navigating and moving around their environment, mostly depending on their mobility cane, memory, and familiarity with the environment to navigate. Technologies designed for them are usually expensive, hard to get, or uncomfortable. This research presents the design, implementation and evaluation of an Augmented Reality (AR)-based indoor navigation system for visually impaired individuals, compatible with Android and iOS devices. Using an Agile methodology, we conducted interviews, gave out questionnaires, and made observations with a number of visually impaired individuals in order to establish user requirements and refine the scope of the project. The system was implemented using the AR library, Niantic Lightship ARDK with real time meshing and semantic segmentation features, the Unity 3D engine, and the C# programming language. The AR navigation tool digitally maps the environment, detecting obstacles and filtering out the ground. When an obstacle is detected within one meter, the system provides haptic and auditory feedback, alerting the user until they move away. Usability testing was conducted with 18 visually impaired participants through questionnaires, interviews, and observations. The system's usability was assessed using the John Brooke System Usability Scale (SUS), achieving a score of 81.39, classified as "Best Imaginable." This research contributes to the field of AR-based assistive technologies.

Keywords: *Visually Impaired Navigation, Augmented Reality (AR), AR-Based Assistive Technology for the Blind, Human-Computer Interaction (HCI), Usability Testing, System Usability Scale (SUS)*

1. Introduction

The visually impaired people in our society face many challenges in their daily lives, such as finding their way around unfamiliar places, avoiding obstacles and hazards, accessing relevant information about their surroundings, and also lacking confidence due to restricted mobility. Due to the increase in the number of people becoming visually impaired, a lot of research has gone into the development of different technologies to help visually impaired individuals navigate both their indoor and outdoor environments [1, 2, 3].

In recent years, augmented reality has been used to create solutions to help visually impaired people navigate their environment because it has the ability to virtualize physical objects or an individual's movement, mixing virtual and real elements in an immersive scenario [4]. It also boosts the implementation of functionalities needed for navigation systems, such as obstacle detection and recognition [4]. Augmented reality can be defined as a live direct or indirect view of a physical, real-world environment whose elements are augmented by computer-generated sensory input, such as sound, graphics, or GPS data [5]. It can also be defined as a system that enhances the real world by superimposing computer-generated information on top of it [6].

A couple of AR-based solutions have been developed to aid visually impaired individuals. For instance, NAVIG, an AR guidance system aimed at empowering visually impaired individuals with navigation and object recognition capabilities [7]. NAVIG integrates geographic information systems with object data to recommend routes and provide guidance. It employs real-time object detection using embedded vision algorithms and offers

Okparaeke C. J., and Sakpere A. B. (2025). Augmented Reality Navigation Assistance System for Visually Impaired Individuals. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 14 No. 1, pp. 221 - 231

3D spatialized audio instructions that adjust based on the user's head position, facilitating intuitive guidance. The development of NAVIG incorporated feedback from visually impaired users and educators to ensure the design meets their specific needs. Another example is ARIANNA+ designed to assist users in both indoor and outdoor navigation by creating virtual paths without relying on physical landmarks like QR codes or colored tapes [8]. Leveraging a smartphone's AR capabilities, ARIANNA+ enables users to load previously recorded virtual paths and provides automatic guidance through haptic, speech, and sound feedback. It also recognizes surrounding objects and buildings, offering relevant information to enhance accessibility.

Despite the effectiveness of these solutions, certain limitations persist. Many systems require loading previously recorded virtual paths, limiting their real-time adaptability in new environments. Consequently, when a visually impaired user enters an unfamiliar setting, the system may struggle to provide adequate assistance due to the absence of pre-recorded data. Additionally, some systems necessitate extra devices and hardware components, potentially making them expensive and cumbersome, leading to reliance on non-governmental organizations (NGOs) for access [9]. There is therefore a pressing need for systems that are affordable, easily accessible, user-friendly, compact, and comfortable to use.

To address these challenges, we propose an AR system compatible with both Android and iOS devices, utilizing the Niantic Lightship ARDK, which offers features such as real-time meshing and semantic segmentation. Real-time meshing involves creating a 3D digital representation of the environment by capturing its geometry and texture using the device's camera. This process allows the system to interact with any environment instantaneously, even without prior exposure. Semantic segmentation assigns labels to each pixel in an image based on categories like sky, ground, foliage, water, or person. This feature enables the system to identify the ground and instruct it to disregard the ground as an obstacle. By combining these features, our system can generate a 3D mesh of any environment in real-time while ignoring the ground. To assist visually impaired users, the system calculates the distance between the user and obstacles using the 3D mesh and greedy algorithm. If an obstacle is detected within one meter, the system triggers both haptic and auditory feedback to alert the user. This feedback ceases once the user moves away from the obstacle.

This study aims to design, implement, and evaluate an augmented reality navigation assistance system for visually impaired individuals using Niantic Lightship ARDK's meshing and semantic segmentation. The research objectives are:

- Design an Augmented Reality navigation assistance system for Visually impaired people
- Develop a prototype of the proposed system using the Niantic Lightship ARDK SDK
- Conduct user testing and VIPs to assess the prototype's usability, effectiveness, satisfaction, and acceptance.
- Analyze the results and provide recommendations for future work.

2. Related Works

Over the years, significant advancements have been made in assistive technologies to enhance navigation for visually impaired individuals. In 2012, Katz *et al.* [7] introduced NAVIG, an augmented reality guidance system integrating an adapted geographic information system (GIS) with real-time vision algorithms and spatialized audio cues. This system enabled users to receive customizable route planning and real-time information about points of interest, thereby increasing personal autonomy. However, the study highlighted the need for further refinement in system parameters and enhancements to GIS databases to improve accessibility.

Sato [10] introduced NavCog3, a smartphone-based indoor navigation assistant designed for visually impaired individuals (PVI). Unlike traditional navigation systems, NavCog3 emphasizes semantic features, such as information about nearby points of interest and obstacles, in addition to accurate turn-by-turn navigation. The inclusion of semantic features was highly valued, as it enhanced users' spatial understanding and confidence in navigating complex spaces. However, the system has limitations, including its dependence on predefined routes and inputted points of interest, restricting its use to environments that have been pre-configured. Real-time data processing was not supported, limiting its adaptability to dynamic environments.

In 2021, Lo Valvo *et. al.* [8] presented ARIANNA+, an augmented reality navigation system leveraging ARKit and machine learning to create virtual paths for indoor and outdoor localization. By eliminating the need for physical markers, ARIANNA+ provided real-time feedback through haptic, speech, and sound cues, enhancing accessibility for visually impaired users. Despite its effectiveness, the system's reliance on pre-recorded virtual paths limited adaptability to changing environments, and its dependence on ARKit restricted usage to iPhone users.

Sanchez-Garcia *et al.* [11] introduced an augmented reality navigation system designed to assist users of visual prostheses in navigating unknown environments. This system incorporates reactive navigation and path planning software to guide users along obstacle-free routes. Key components include locating the subject on a map, planning their trajectory, visually displaying the path, and dynamically re-planning to avoid obstacles. Results demonstrated that the AR system significantly enhanced navigation performance by reducing the time and distance needed to reach the goals and minimizing collisions with obstacles compared to baseline methods. Despite its potential, the study has limitations. The sample size was small, and the results may not generalize to a broader population. Furthermore, participants with varying degrees of visual impairment were not included, which limits the system's applicability across diverse user groups.

In 2024, Raythatha *et al.* [12] developed an augmented reality (AR)-based indoor navigation system designed to facilitate efficient wayfinding. Implemented using Vuforia and Unity, the system incorporates voice assistance to enhance the navigation experience, particularly for visually impaired users. By employing the A* algorithm for optimal pathfinding, it provides real-time directions, enabling users to navigate complex environments with ease. The application is versatile, suitable for various settings such as universities and other public spaces.

These developments underscore the ongoing efforts to leverage technology in enhancing navigation and independence for visually impaired individuals, with each system building upon previous innovations to address existing limitations and improve user experience.

3. Methodology

In this section we present an overview of the system, requirement gathering, program design and algorithm, system component and the user interface.

3.1 System Overview

The augmented reality navigation assistance system is designed to aid visually impaired users by creating a real-time 3D representation of their environment, detecting objects and obstacles, and alerting them when they are within 1 meter of an obstacle. This system is compatible with both Android and iOS devices, leveraging the widespread availability and familiarity of smartphones among visually impaired individuals, thereby ensuring accessibility and cost-effectiveness. Its design emphasizes comfort and ease of use, as it operates seamlessly on everyday mobile phones without the need for additional cumbersome equipment.

The system's architecture involves key components: the mobile phone camera, the Niantic Lightship ARDK, a distance calculation algorithm, and feedback mechanisms. The camera captures visual and sensor data from the environment. The Niantic Lightship ARDK processes this data to create a real-time 3D mesh of the surroundings and performs semantic segmentation to identify and ignore the ground, preventing it from being mistakenly recognized as an obstacle. The distance calculation algorithm measures the space between the user and detected obstacles using this 3D mesh. When an obstacle is within 1 meter, the system triggers haptic (vibration) and auditory (sound) feedback to alert the user, enhancing their spatial awareness and navigation safety.

Figure 1. shows the context diagram of the system. In the context diagram, we have two entities, which are the visually impaired user and the environment. And we have the Augmented Reality Navigation Assistance System as the process 0. The visually impaired user sends in the data flow named “Initial Movement” into the system. The visually impaired user receives two data flows from the system which are “haptic feedback” and “audio feedback.”. The environment sends the “Raw visual/ Sensor data” data flow to the system.

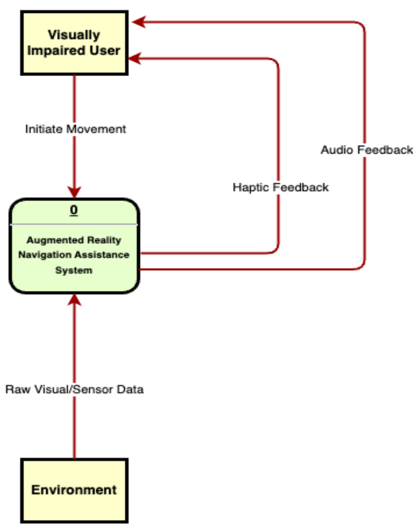

Figure 1. Context Diagram

3.2 Requirement Gathering

To ensure the system meets the specific needs of visually impaired users, a comprehensive requirement analysis was conducted. Data was gathered through questionnaires, interviews, and direct observations at various locations, including the Department of Special Education at the University of Ibadan, a Center for the Blind in South West of Nigeria, and a Commercial Grammar School in Ibadan, Oyo State. The data gathering was done to find out the challenges visually impaired people face, the existing ways they overcome them, if they own a mobile phone and how efficient they are at using it, and finally, if they know what Augmented Reality is.This approach provided valuable insights into the challenges faced by visually impaired individuals, their current navigation methods, smartphone usage proficiency, and familiarity with augmented reality technology.

A total of 18 participants particpated in the requirement gathering phase. Result from the survey among visually impaired individuals revealed that 77.8% (14 particpants) experienced vision loss during childhood, while smaller percentages reported onset during adolescence (11.1% - 2 participants), adulthood (5.6% - 1 participant), or since birth (5.6% - 1 participant). Regarding support from family and friends, 83.3% described it as moderate, 11.1% as strong, and 5.6% as limited. In daily navigation, 66.7% rely on mobility aids like canes or guide dogs, whereas 33.3% navigate independently. When locating objects at home, 88.9% depend on memory and familiarization, with only 11.1% seeking assistance. Notably, 88.9% own a mobile phone or tablet, predominantly Android devices (55.6%), while 33.3% use basic phones, and none reported using iOS devices. Comfort with technology is high, with 88.9% rating their comfort level between 8 and 10 on a 10-point scale. However, none of the participants were familiar with augmented reality (AR), indicating a need for this research.

3.3 Program Design And Algorithm

Greedy algorithm was used in the calculation of the distance of the user to an obstacle. The system only focuses on a local optimal solution. What that means is that, even though there are many obstacles in front of the system camera, the system camera only focuses on the particular obstacle it can see that has the shortest distance to the user.

Algorithm for Distance Calculation

1. Get the current frame from the camera.
2. Calculate the screen center position.
3. Cast a ray from the camera towards the screen center.
4. Check if ray hit something within a set distance.
5. If hit:
    - Check if hit object name contains "MeshCollider" or "Interior_".
    - If name matches and is not ground mesh:
        - Calculate the distance

from the camera or ray origin to the hit point.
  - Log hit distances
  - Check if the hit distance is below the threshold of 1 meter.
  - If conditions are met:
    - Start vibration and play sound effects.

3.4. System Component

This section describes the structure and components that make up the Augmented Reality Navigation Assistance System. Since this system is built for visually impaired users, we made sure to make the system as easy as possible to use and not cumbersome for the visually impaired user. To accomplish this, we kept the number of screens to one (1). The system has the following components that make up its implementation:

1. The mobile phone camera
2. The Niantic Lightship ARDK
3. The Distance Calculation Algorithm
4. The Haptic Feedback Engine
5. The Audio Playback Engine

3.4.1 The Mobile Phone Camera

For this system to work properly, it needs access to the camera system of the mobile phone it is running on. The camera provides the system with the visual data and sensor data it needs to carry out activities like real-time meshing, semantic segmentation, and many more.

3.4.2 The Niantic Lightship ARDK

The Niantic Lightship ARDK is an augmented reality library or SDK similar to ARCore, ARKit, and Vuforia that allows the mobile phone to access different augmented reality features like plane detection, image tracking, real-time meshing, semantic segmentation, and more. For this system, Niantic Lightship ARDK uses two important features, which are the real-time meshing feature and the semantic segmentation feature.

3.4.2.1 Real-Time Meshing feature

Niantic Lightship ARDK takes the visual and sensor data from the camera system of the phone, and using the real-time meshing feature, it creates a 3D representation of the physical world in digital form by overlaying a 3D mesh on the visual feed of the real world it receives from the camera.

This 3D mesh of the physical work is very important because it allows us to carry out different activities, such as measuring the distance of the system to physical obstacles in the environment. This is possible because the physical environment has a digital representation within the system, so we can do more with the data we have about the physical world.

3.4.2.2 Semantic Segmentation Feature

The semantic segmentation feature of the Niantic Lightship ARDK labels the video feed that is sent by the camera into categories such as Ground, Sky, Person, Tree, Forage, and more. With this feature, we focused on the label Ground. The reason for this is that we want to tell the system to ignore the ground when it is taking the distance between it and obstacles in the environment. In summary, we are telling the system not to see the ground as an obstacle.

3.4.3 The Distance Calculation Algorithm

Once the Niantic Lightship ARDK has created the real-time mesh of the environment and has successfully created a digital representation of the world, the system employs an algorithm to measure the distance from the system to the detected obstacles in the environment while ignoring the ground as not being an obstacle. This distance is calculated every frame, and the system is constantly checking if the distance is less than 1 meter. If the distance is less than 1 meter, then it triggers both the haptic feedback and audio playback.

Figure 2 is the image of the actual source code of the distance calculation algorithm implemented in C# programming language.

3.4.4 The Haptic Feedback Engine

The haptic feedback engine starts the vibration of the mobile phones once the system distance calculation has a distance of less than 1 meter.

```
public void RaycastAndDistance()
{

    var currentFrame = _session.CurrentFrame;
    if (currentFrame == null) return;

    if (Camera == null) return;

    Vector2 centerOfScreen = new Vector2(Screen.width / 2f, Screen.height / 2f);
    centerOfScreenImage.rectTransform.position = centerOfScreen;

    var worldRay = Camera.ScreenPointToRay(centerOfScreen);
    RaycastHit hit;

    if (Physics.Raycast(worldRay, out hit, 1000f))
    {

        if (hit.transform.gameObject.name.Contains("MeshCollider") || hit.transform.gameObject.name.Contains("Interior_"))
        {
            Vector3 hitPosition = hit.point;

            float hitDistanceFromSource = hit.distance;
            float distanceFromCamera = Vector3.Distance(Camera.transform.position, hit.point);

            string hitDisMsg = $"{hitDistanceFromSource}";
            string cameraDisMsg = $"{distanceFromCamera}";

            hitDisText.text = hitDisMsg;
            //cameraDisText.text = cameraDisMsg;


            Debug.Log($"Hit.distance result: {hitDistanceFromSource}");
            Debug.Log($"Distance from Camera calculation: {distanceFromCamera}");

            // check if the distance of the user to the mesh is less than 0.6 and
            // the mesh is not the ground. Then Vibrate and play sound effect

            if (hitDistanceFromSource < 0.6f && !isGroundChannel)
            {
                // Start vibration and play sound effect

                StartHapticVib();
                audioSource.PlayOneShot(audioClip);

            }
```

Figure 2 Sample Source code

3.4.5 The Audio Playback Engine

The audio playback engine plays the alert sound once the system distance calculation has a distance of less than 1 meter.

3.5 User Interface

The main screen is very simple-looking, displaying the Video Feed of the Augmented Reality Camera, with a simple group of debug text at the top-right corner showing the distance of the system to the obstacle directly in front of the system and another group of text showing a boolean value (true or false) if the system is looking at the ground or not. The image of the screen is shown in Figure 3. To use the app, the visually impaired user holds the mobile phone with the AR camera turned on and moves around in the environment. Our prototype uses augmented reality, real-time meshing, and semantic segmentation to get the distance between the mobile phone and obstacles in the environment and also recognize objects in the environment. If the distance between the user and the obstacle is below a certain value, the prototype alerts the user by triggering sounds and vibrations to notify the user that they are in very close proximity to an obstacle. Once the users change direction and move away from the obstacle, the alert stops, signifying to the user that there is no obstacle on the way. Figure 4 was designed to keep track of the history of objects recognized and locations.

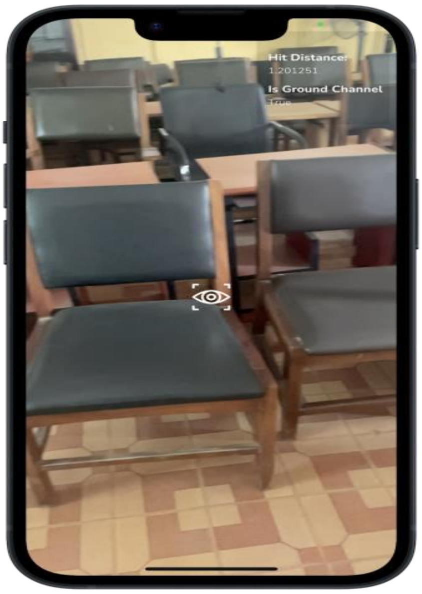

Figure 3. The Main Screen

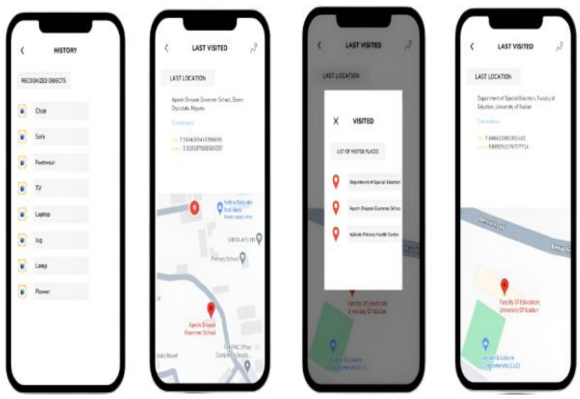

Figure 4: The Screens Showing History

4. Results and Evaluation

For the evaluation phase, we made use of the John Brooke System Usability Scale (SUS) [13] to evaluate the system. In addition to that, we also gave the users some tasks to carry out and we observed how they carried it out and the success and error rate they had during the task. Our criteria for recruitment involved recruiting individuals with varying levels of visual impairment. The research was conducted in three (3) places over two different days - Department of Special Education of University of Ibadan, a School for the Blind and a Commercial Grammar School. A total of 18 people were involved in the usability testing with 27.8% as females and 71.2% as males. 66.7% of the participants were in the age group of 11 - 20, 16.7% in the group of 21 -30 and another 16.7% in the age group of 31 - 40.

4.1 Usability Experiment

The participants in the usability study had the chance to test our prototype, and they were given tasks to carry out. During the usability experiment, we gave the participants different tasks to carry out, which included glance testing, task testing and accessibility testing.

For the glance testing, participants were asked to observe the app's home screen for a few seconds and provide initial thoughts on the overall layout and visual appeal. They were also asked to express their initial thoughts on how easy or challenging it appears to navigate within the app. For the task testing, participants were given the following tasks:

- Navigate to a designated spot in a location
- Utilize the haptic feedback feature to sense proximity to obstacles.
- Utilize the app's auditory cues to assist in navigation.
- Explore your environment and identify a specific item

For the accessibility testing, the following were considered:

- Measure the proximity to the obstacle when the feedback is triggered during the navigation
- Rate the ease of locating objects of varying sizes within the environment.

4.2 Evaluation

The System Usability Scale (SUS), developed by John Brooke is widely used and well-respected tool for measuring the subjective usability of a product or system [13]. It offers a simple, efficient, and reliable way to gather user feedback and assess the overall user experience.

4.1 **Usability Study Findings**

A 5-point Likert scale was used for users to indicate their level of agreement. The scale was interpreted as follows: (1) Strongly disagree; (2) Disagree; (3) Neither agree nor disagree; (4) Agree; (5) Strongly agree. The SUS focuses on 10 factors: Frequency of Use, System Complexity, Ease of Use, Technical Support, Proper Integartion of Functions, System Inconsistency, Gaining Familarity, Cumbersome, Confident to use and the need to learn before usage.

The system usability scale (SUS) score was computed according to the guidelines by John Brooke (1996). Using the guideline, the final SUS score was obtained by multiplying the individual SUS score by 2.5. An average score of 81.39, which falls within the range of 80.8 - 84.0 was obtained. This range has the interpretation of Excellent. According to the guideline, our prototype with a SUS score of 81.39 can be said to be Best Imaginable. Table 1 visualizes the average response of users during the usability testing.

4.2.1 Frequency of Use

This item measures how frequently users will prefer to use the app. As seen in table 1, this factor has a mean score of 4.67. This is a strong indication that the users would like to use the app frequently. Our prototype solves a very important problem for visually impaired people which is navigation and this is something they do frequently every day, so that explains why the participants are willing to use the software frequently.

4.2.2 System Complexity

This item measures how complex the users found the app. An average response of 1.94 showed that the app is not unnecessarily complex. Reduction of complexity was top of mind for us when designing this prototype. We know that our target users are virtually impaired, which means that they will have difficulty seeing our system and using the app. So we made sure that we had only one screen, which is the main screen that uses the Augmented Reality camera to help them avoid obstacles. By doing this, the user does not need to worry about moving from one part of the app to another. Once the app launches, they can start using it immediately.

Table 1: John Brooke System Usability Scale

User ID	Frequency of Use	System Complexity	Ease of Use	Technical Support	Proper Integration of Functions	System Inconsistency	Gaining Familarity	Cumbersome	Confident to Use	Need to Learn	SUS Score	Final SUS score
1	5	2	5	2	4	1	4	2	4	3	32	80
2	5	1	5	1	4	2	5	2	5	2	36	90
3	5	1	5	1	5	1	5	1	5	1	40	100
4	5	1	5	1	5	1	4	2	4	2	36	90
5	4	1	4	2	5	1	5	2	5	2	35	87.5
6	4	1	5	1	5	1	5	1	5	1	39	97.5
7	5	1	5	3	5	4	5	1	5	1	35	87.5
8	4	1	5	1	5	1	5	1	5	1	39	97.5
9	5	4	5	2	4	2	2	2	2	2	26	65
10	5	1	5	2	4	1	5	2	4	4	33	82.5
11	5	2	4	2	4	2	4	2	4	2	31	77.5
12	4	2	5	1	4	2	5	2	5	5	31	77.5
13	4	2	5	1	3	3	3	5	2	4	22	55
14	5	2	5	1	4	3	5	2	5	1	35	87.5
15	5	2	5	2	5	2	4	1	5	1	36	90
16	5	2	5	1	5	1	5	2	5	1	38	95
17	5	5	5	3	5	4	2	5	5	5	20	50
18	4	4	5	4	5	4	5	5	4	4	22	55
Mean	4.67	1.94	4.89	1.72	4.5	2	4.33	2.22	4.39	2.33		81.39

4.2.3 Ease of Use

This item measures how easily the users found the app. Users rated the app as relatively easy to use, with a mean value of 4.89. We made sure that the prototype was very simple and easy to use. There were no complex UIs or screens. All we have is a single screen. The users do not have to do any extra things to use the app. All they need to do is launch the app and point it in the direction they are moving.

4.2.4 Technical Support

This item questions whether users will need the support of a technical person to be able to use the app. An average value of 1.72 indicates that most of the participants do not need any technical assistance to be able to use the app. It was nice to see that all our participants had a mobile phone, and they had high proficiency in using it. Our prototype, as stated initially, was developed to be simple, which means that as long as the user knows how to use a smartphone, he or she will easily use our prototype without the help of any technical support.

4.2.5. Proper Integration of Functions

This item measures whether the app functions were well integrated. 4.5 was obtained as the mean value of users' responses. This shows the various functions of the app are integrated satisfactorily. During the design of the prototype, we ensured that all functions were properly integrated. All the user has to do is launch the app and point the phone in the direction they are walking, and when they approach any obstacle, the prototype triggers a sound and haptic feedback to the user to alert them of the obstacle in front of them.

4.2.6 System Inconsistency

This item measures whether the users found the app inconsistent. There is little or no inconsistency in the app. This was proven by an average score of 2. We made sure that every part of the prototype was consistent and that all the functionality worked as expected all the time.

4.2.7. Gaining Familarity

This item measures how quickly users will learn to use the app. The majority of users will learn to use the app very quickly, as indicated by a mean value of 4.33. From our observation, once the participants used the prototype for the first time, it became clear to them immediately what the prototype was all about. The good thing is that the prototype works similarly to how their mobility cane works and the only difference is that this is a mobile phone. So they can transfer that familiarity with the mobility cane into the prototype.

4.2.8. Cumbersome

This item measures how cumbersome the users found the app. 2.22 was the mean value derived from users' responses. This implies that the app is not exactly cumbersome to use. During the design of the prototype, we made sure not to add a lot of screens and functionalities, so that the prototype is not too cumbersome for the users. We kept the prototype simple so that the users did not have to do much to use it.

4.2.9 Confident to use

This item measures how confident users felt while using the app. A good number of users felt confident while using the app and an average value of 4.39 was reported. During the user testing, we observed that when the users used the prototype for the first time, they appeared confident enough to continue using the prototype. Most of them were very free to use the prototype even without our presence.

4.2.10 The Need to Learn Before Usage

This item questions if the users needed to learn a lot of things before they could get going with the app. Users' responses reported a mean value of 2.33, indicating that, to a considerable extent, users will not need to learn a lot of things before they use the app. The prototype was built so that the users do not need to learn something else before they use it. Their knowledge of the use of mobile phones and their knowledge of the use of their mobility cane is enough for them to use the prototype effectively.

4.3 Inferential Analysis

In order to validate if the level of education has an influence on the potential to use immersive technology/augmented reality for navigation amongst the visually impaired, we calculated the correlation between SUS score and education level using a Python script as shown in Figure 5. The correlation coefficient is 0.3522, which indicates a very weak positive correlation between SUS score and education level. There is no statistically significant relationship between the two variables. This means that education level does not appear to have a strong influence on the usability of the prototype, as measured by the SUS score.

5. Conclusion

Navigating unfamiliar environments poses significant challenges for visually impaired individuals, often leading to reliance on assistance or specialized tools. To address this, we developed an Augmented Reality (AR) application designed to facilitate indoor navigation by detecting obstacles in real-time. Leveraging real-time meshing and semantic segmentation, the app interprets the user's surroundings, identifies potential obstacles, and provides alerts when they are within one meter of an obstruction.

A key advantage of our solution is its integration with everyday mobile devices. Given that a substantial majority of visually impaired individuals own smartphones, our app offers an accessible and cost-effective alternative to specialized hardware, which can be expensive and less readily available. Existing literature indicates that many AR navigation systems for the visually impaired rely on pre-mapped environments, limiting their adaptability to new settings.

```
import pandas as pd

# Create a pandas dataframe from the provided new data
data = {
    "Age": ["31 - 40", "31 - 40", "11 - 20", "21 - 30", "11 - 20", "21 - 30", "31 - 40", "21
- 30", "11 - 20", "11 - 20", "11 - 20", "11 - 20", "11 - 20", "11 - 20", "11 - 20", "11 -
20", "11 - 20", "11 - 20"],
    "Education Status": ["Undergraduate", "Graduate", "Post Secondary",
"Undergraduate", "Undergraduate", "Undergraduate", "Postgraduate",
"Undergraduate", "Post Secondary", "Post Secondary", "Post Secondary", "Post
Secondary", "Post Secondary", "Post Secondary", "Post Secondary", "Post
Secondary", "Post Secondary", "Post Secondary"],
    "SUS Score": [80, 90, 100, 90, 87.5, 97.5, 87.5, 97.5, 65, 82.5, 77.5, 77.5, 55,
87.5, 90, 95, 50, 55]
}

df = pd.DataFrame(data)

# Encode the categorical data to numerical values
df['Age'] = df['Age'].astype('category').cat.codes
df['Education Status'] = df['Education Status'].astype('category').cat.codes

# Calculate the correlation between SUS score and Age
correlation_age = df['SUS Score'].corr(df['Age'])

# Calculate the correlation between SUS score and Education Status
correlation_education = df['SUS Score'].corr(df['Education Status'])

# Print the correlation coefficients
print("Correlation between SUS Score and Age:", correlation_age)
print("Correlation between SUS Score and Education Status:",
correlation_education)
```

Figure 5: Source code to calculate the correlation coefficient:

In contrast, our application functions in real-time, allowing users to navigate any indoor space without prior mapping. This adaptability enhances user independence and mobility across various environments.

We conducted usability testing with eighteen visually impaired participants. The results demonstrated that the app effectively alerted users to nearby obstacles, enabling them to navigate their indoor environments more safely and independently. However, the current version of the app has limitations, notably its reduced functionality in low-light conditions. Future developments will focus on improving performance in such environments to ensure consistent reliability.

In summary, our AR application represents a significant step toward enhancing indoor navigation for visually impaired individuals, offering a practical, real-time solution that leverages the capabilities of devices many users already possess.

References

[1] Bamdad, M., Scaramuzza, D., & Darvishy, A. (2024). Slam for visually impaired people: A survey. *IEEE Access*.

[2] Messaoudi, M. D., Menelas, B. A. J., & Mcheick, H. (2022). Review of navigation assistive tools and technologies for the visually impaired. *Sensors*, *22*(20), 7888.

[3] Kuriakose, B., Shrestha, R., & Sandnes, F. E. (2020). Tools and Technologies for Blind

and Visually Impaired Navigation Support: A Review. *IETE Technical Review*, *39*(1), 3–18. https://doi.org/10.1080/02564602.2020.1819893

[4] Real, S., & Araujo, A. (2019). Navigation systems for the blind and visually impaired: Past work, challenges, and open problems. *Sensors*, *19*(15), 3404.

[5] Grier, R. A., Thiruvengada, H., Ellis, S. R., Havig, P., Hale, K. S., & Hollands, J. G. (2012, September). Augmented reality–implications toward virtual reality, human perception and performance. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 56, No. 1, pp. 1351-1355). Sage CA: Los Angeles, CA: SAGE Publications.

[6] Nikolaidis, A. (2022). What is significant in modern augmented reality: a systematic analysis of existing reviews. *Journal of imaging*, *8*(5), 145.

[7] Katz, B. F., Kammoun, S., Parseihian, G., Gutierrez, O., Brilhault, A., Auvray, M., ... & Jouffrais, C. (2012). NAVIG: Augmented reality guidance system for the visually impaired: Combining object localization, GNSS, and spatial audio. *Virtual Reality*, *16*(4), 253-269.

[8] Lo Valvo, A., Croce, D., Garlisi, D., Giuliano, F., Giarré, L., & Tinnirello, I. (2021). A navigation and augmented reality system for visually impaired people. *Sensors*, *21*(9), 3061.

[9] Yu, K. N. (2019, March 19). *Ways NGOs can Help the Blind. AsianNGO. Retrieved July 2, 2023,* from https://asianngo.org/magazine/post-magazine/article/article-detail/159/ways-ngos-can-help-the-blind

[10] Sato, D., Oh, U., Naito, K., Takagi, H., Kitani, K., & Asakawa, C. (2017, October). Navcog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 270-279).

[11] Sanchez-Garcia, M., Perez-Yus, A., Martinez-Cantin, R., & Guerrero, J. J. (2021). Augmented reality navigation system for visual prosthesis. *arXiv preprint arXiv:2109.14957*.

[12] Raythatha, D., Syed, K., Makadia, R., & Rathod, D. (2024, December). An AR-Based Indoor Navigation System with Improved Accessibility. In *2024 13th International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 576-584). IEEE.

[13] Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4-7.