Indoor Positioning for Visually Impaired people based on Wi-Fi RSS Fingerprinting

Marco Bologna, Caterina Bruchi, Federico Casu, Giacomo Moro
m.bologna2@studenti.unipi.it, c.bruchi@studenti.unipi.it, f.casu1@studenti.unipi.it, g.moro4@studenti.unipi.it

https://github.com/caterinabruchi9/MSSS-Project.git

ABSTRACT

With the rapid development of wireless communication technology, various indoor location-based services (ILBSs) have gradually penetrated daily life. Although many other methods have been proposed to be applied to ILBS in the past decade, Wi-Fi-based positioning techniques have attracted attention in the field of wireless transmission. The aim of this paper is to provide an indoor positioning and guiding system thought for visually impaired people using the sensors of a smartphone by developing an Android application access friendly. A map containing the RSS of the Wi-Fi access point was realized and will be used to perform a 1-NN to find the sample nearest to the user and locate them. After the localization using the azimuth computed from the accelerometer and magnetometer of the smartphone checked against some reference value included in the map also the direction towards which the user is walking will be computed. Given this information some guidelines will be provided via audio interface. The application was tested with various maps, focusing on the response time and difference in performance depending on the trajectory.

1 Introduction

There are many technologies for indoor positioning, such as visual technology, infrared technology, Wi-Fi technology, ultra-wideband technology, Bluetooth technology, inertial navigation technology, and magnetic technology. In recent years, various methods have been compared, and their advantages and disadvantages have been analyzed.

Indoor positioning systems can be divided into three classes based on their architecture:

1. *self-positioning*;
2. *infrastructure positioning*;
3. *assisted by self-directed infrastructure*.

In self-positioning architecture, devices determine their locations by themselves. In infrastructure positioning architecture, device positions are estimated using the locations of devices positioned in the environment. In architecture assisted by self-directed infrastructure, an external system calculates the position and sends it to the tracked user in response to a request.

Additionally, indoor positioning systems are categorized according to the primary technology they employ to determine their position within indoor environments. The indoor positioning systems which are the subject of this work were initially classified into two groups: based or not based on radio frequency signals (Figure 1).

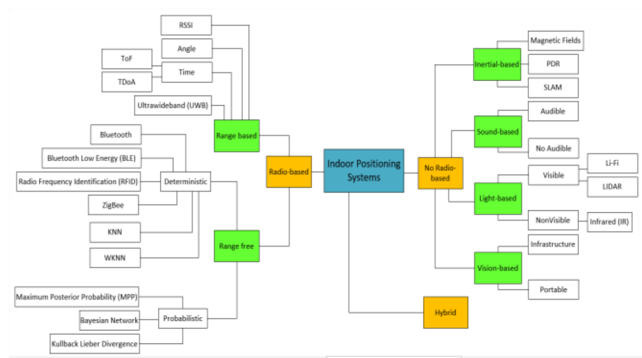

Figure 1: Types of Indoor-Positioning systems

Positioning technique	Advantage	Disadvantage	Positioning accuracy
Ultrasonic indoor positioning [1]	High positioning accuracy	Multipath effect, thermal drift effect, high cost, severe decay	Centimeter scale
RFID indoor positioning [2]	Low cost, high speed	Short distance, low communication capacity	Centimeter scale
Ultra-wideband (UWB) indoor positioning [3]	Strong resistance to interference, strong penetration, high positioning accuracy	High cost	Sub-meter scale
Bluetooth indoor positioning [4]	Low power consumption, easily deployable	Short distance	Meter scale
Infrared indoor positioning [5]	Low cost, mature technology	Poor resistance to interference	Meter scale
Zigbee indoor positioning [6]	Low cost, low power consumption	Poor stability	Meter scale
WiFi indoor positioning [7]	No additional hardware required for deployment, low cost, wide application range	Tedious fingerprint collection, degeneration of WiFi signal	Meter scale

Figure 2: Advantage and Disadvantage for different techniques.

Systems Based on Radio Frequency

Many different wireless networks can be deployed in an indoor environment, from traditional Wi-Fi networks, which are used in millions of buildings around the world to provide Internet access, to models with more specific purposes. Among the various models available are WSN models, which are designed to connect the Internet of Things (IoT) devices, popular RFID devices, and Bluetooth beacons, among other alternatives. A more simplified

way to classify network-based indoor positioning systems is to consider how information is obtained. In this classification model, there are two groups: (1) methods based on range and (2) methods without range.
Range-based methods extract geometric information (distance or angle) from signals from different wireless nodes and then combine the geometric constraints of each anchor to obtain the user's position. Free-range methods are based on the connectivity information between nodes or the identification of signal resource patterns that depend on location.
Inside the range-free methods we have the category of fingerprint based.

Fingerprinting location systems use the characteristics of information perceived in certain locations, which must be previously identified. Thus, these systems require two phases to be observed: an offline phase and an online phase. In the first phase, the area is divided into small subregions to specialize in the identification of each place of interest, allowing the collection of data to form a database. The central idea of the digital printing method is to store a signal pattern in a similar way to a relational database. This pattern is used as a reference in the search for a value by equivalence or approximation. In the second phase, the position is estimated by comparing the captured data with the records stored in the database.
When the fingerprinting method is used to map locations through radio signals, the main concern is to perceive the singularity of the signals in different positions due to propagation problems caused by the complexity of indoor environments. There are two types of fingerprint methods: those that adopt deterministic strategies and those that adopt probabilistic strategies.
Deterministic models indicate high levels of accuracy, provided by high data consumption, the strong density of sensory coverage per square meter, or the combination of techniques to reduce spurious data, as perceived in SNR applications for RSSI signals on a Wi-Fi network. In addition to the k-nearest neighbor (KNN) and the weighted k-nearest neighbor (WKNN), other systems are also used to determine positions, such as support factors (SVMs) or linear discriminant analysis. In the case of KNN or WKNN, the degree of matching between the RSS vector and the fingerprint is characterized by the Euclidean distance. The Euclidean distances between the target location and all the fingerprints in the radio map are ranked in ascending order. Subsequently, the average of the coordinates of the K (K>2) reference points with the smallest distance is taken as the position of the target location. The fingerprint positioning algorithm based on KNN is widely used because of its simple hard-ware facilities, less complexity, and less computation. [5]

Systems Based on Inertial Sensors
Network-based systems estimate user positions by measuring the resources of signals received from a wireless network. This model requires an intervention in the environment with the implementation of physical infrastructure. Systems based on inertial information calculate their positions without having to apply external resources.
Estimating a future position given an initial position associated with speed and direction is one of the oldest methods of navigation, generally called "dead reckoning". An obvious problem with this model of dead reckoning is the increase in the progressive accumulation of errors since a small error of direction can mean a huge error over the distance covered.
Information about the position of a target must be accurate. Therefore, these models adopt the geographic coordinate model, composed of latitude, longitude, and altitude, to identify the absolute (inertial) position. The sensors allow the absolute (inertial) position to be immediately perceived in the roll, pitch, yaw (RPY) system using the x-, y-, and z-axes, based on the RPY. In such a system, the x-axis indicates the nominal direction (front), y (slope) is orthogonal to the x-axis and points to the left side, and the z-axis (yaw) points up.

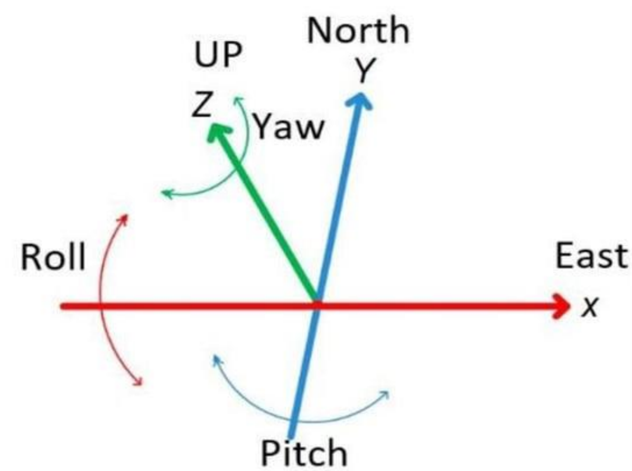

Figure 3: IMU Schema

Generally, inertial sensors are assembled, forming units of inertial measurement (IMUs), which have an accelerometer, a gyroscope, and a magnetometer, each with three axes, the magnetometer is not an inertial sensor.
From this approach we used the accelerometer and magnetometer to measure the azimuth and learnt the direction towards which the user is heading. We use only azimuth since we assume the user to keep the phone horizontally facing up which make the use of pitch and roll unnecessary.

1.1 Wi-Fi Based Approaches

The pervasive nature of Wi-Fi technology has led to its widespread adoption in various sectors such as public safety, industry, and healthcare.
Within the current landscape of ILBS applications, Wi-Fi technology offers three distinct advantages:

1. Broad distribution of hotspots – Wi-Fi hotspots are prevalent in diverse indoor settings including residences,

hotels, and shopping centers, facilitating the applicability of Wi-Fi positioning across various indoor environments.

2. Low access conditions – Exploiting the existing infrastructure of Wi-Fi networks eliminates the need for extensive network reconstruction or expansion, thereby reducing implementation costs for Wi-Fi-based positioning systems.
3. High flexibility – Wi-Fi signals are not severely affected by obstacles in the complex indoor environment. [1]

1.2 Implications for Visually Impaired People

These strategies we just talked about can be applied to the field of Visually Impaired people considering some aspects. The broad choice of using the smartphone was because of the possibility of using its sensors and its high availability among the visually impaired public.

Main strategies can be grouped into two broad classes: the systems that use maps and the systems of free mapping. In models that use maps, the systems depend on the data obtained by the sensors to establish addresses that allow representing faithfully or approximately the environments, such as buildings, schools, and hospitals.

On the other hand, others stated that indoor environments have characteristics that make it difficult to read addresses through sensors and suggest as a solution free mapping strategies which perceives the scenario itself, within a radius of sensory vision.

Of course, this latter choice will achieve higher performances in noisier environments with the presence of metallic components that affect the operation of the magnetometer and the Wi-Fi sensor. Some of them also use a camera to indicate the position of colored lines on the floor or similar kind of signs. If this is to be done through the creation of a specific map it will be extremely time-consuming.

Also, there are some criteria that must be stringently met in future approaches, such as the ability to adapt to numerous positions of the user, the heterogeneity of transmitters and receivers, and the consumption of energy. Since we are dealing with users which could be highly disadvantaged by malfunctioning and delays, the application must be portable and easily usable.

- *Adaptability to indoor positioning*: The most adherent algorithms and technologies are those that allow their application with a higher rate of adaptation. For example, portable systems can be offered for applications on smartphones and other devices that can be used in various positions on a person, such as in a pocket, purse, or the user's hand, fixed at the height of the head, etc.
- *Heterogeneity*: In an indoor location and an indoor navigation application for the visually impaired that is intended to be made available to large masses of users, the hardware and algorithms must be able to deal with the various characteristics of closed environments. Multiple wearable devices, such as smartwatches, smartphones, and the countless gadgets presented in recent years, can be used to provide different sensitivities and data patterns.
- *Power consumption*: It is known that battery life on many commercial smartphones is not as good as desired. The same is true for other wearable devices. Therefore, it is crucial to develop new algorithms that are energy-efficient, with reduced computational complexity.

This paper's main aim is to try to provide a simple but useful tool to orientate VI people inside buildings, without the use of external infrastructure and with the provision of all the accessibility standards used in literature.

2 Architecture

The application's functioning can be divided into two phases: an initial phase involves locating and downloading the map locally, and a subsequent phase which utilizes this map to determine the user's location within the building and provides basic directions via text-to-speech functionality.

2.1 Map Selection

To accommodate visually impaired users who may find navigating complex menus challenging, we have streamlined the process by integrating GPS functionality. Specifically, all maps now include coordinates stored in a dedicated table within the database. During this initial stage, only this table will be downloaded locally and used to identify the nearest maps to the user, with a 200 meters radius. Then the top four options are displayed from the closer to the more distant.

As for the VI-friendly menu, it will display the options in a scrollable format, allowing users to navigate using the volume up button on their phone. Each option will be announced using text-to-speech technology.

Then the user will select the desired option with a click of the volume down button. Only after this step, the Wi-Fi fingerprint map will be downloaded in JSON format locally.

Here is a simple scheme of the functioning of this phase:

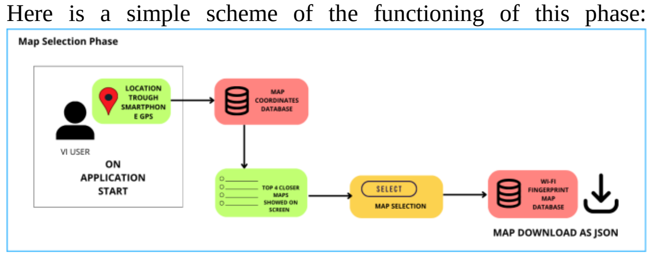

Figure 4: Map selection procedure.

A brief guide on how to use the volume buttons to navigate the menus is spoken through text-to-speech when the user starts the application.

Admin Options

In the initial panel there is a button for the admin options, which allows you to create a new map and add fingerprints to existing maps.

In this menu there are also many options used during the preliminary development phase to test the various tools used in the application.

2.2 Positioning and Guidance Phase

Now that the map has been selected, the user can start the navigation.

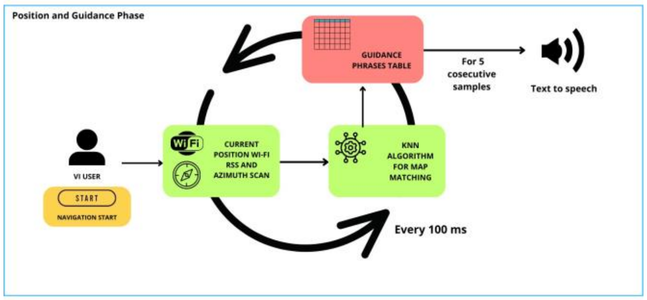

Figure 5: Navigation phase schema.

1-NN algorithm

The 1-NN algorithm we have developed follows the traditional approach: it calculates Euclidean distances based on the RSS values, component by component. Consequently, when comparing different locations, it's possible that some locations may not have the same number of active Access Points (APs) as others. This might create problems in the minimum distance computation. To address this issue, we've opted to assign a value of -100dB to APs that are not present during the matching process. This value is intentionally set to be very high, effectively penalizing such fingerprints in the nearest neighbor computation. The rationale behind this is that if an AP doesn't match, it's highly likely that the sample to be localized is not situated in the same area as that fingerprint.

```
fun findNearestSample(samples: List<Sample>, allBSSIDs: Set<String>): Pair<Pair<Int, Int>, Double> {
    var minDistance = Double.MAX_VALUE
    var nearestSample: Pair<Int, Int>? = null

    for (sample in samples) {
        val distance = this.euclideanDistance(sample, allBSSIDs)
        if (distance < minDistance) {
            minDistance = distance
            nearestSample = sample.zone to sample.sample
        }
    }

    return (nearestSample ?: (0 to 0)) to minDistance    }

}
```

```
fun euclideanDistance(other: Sample, allBssids: Set<String>): Double {
    val rssVector1 = allBssids.map { bssid ->
        fingerprints.find { it.bssid == bssid }?.rss
            ?: -100 // Uses allBss to ensure the same dimensionality and fill missing values with -100
    }
    val rssVector2 = allBssids.map { bssid ->
        other.fingerprints.find { it.bssid == bssid }?.rss
            ?: -100 // Uses allBss to ensure the same dimensionality and fill missing values with -100
    }

    // euclidean distance
    return sqrt(rssVector1.zip(rssVector2).sumOf { (rss1, rss2) ->
        (rss1 - rss2).toDouble().pow(2)
    })
}
```

Figure 6: 1-NN code

Guidance Phrases and Database

A guidance phrase will be issued when the zone detected is the same for 5 consecutive samplings. The database holds for each zone 2 distinct phrases, one for each walking direction, since most of the references were “on your left”, “on your right” etc.

The azimuth allows us to understand in which direction of the corridor the user is walking. For the 2 different cases we have different explanations of the road ahead, so the user can be aware of what they are going to face.

The phrases are triggered when both the zone and the direction changes, to better handle special cases like corners, where a change in direction might maintain the same reference point as before, thus making the message misleading.
Here are some examples:

zone	sample	azimut	threshold	info
1	1	290.0	90	Behind you is the entrance to the department, to your left are the stairs to the first floor.
1	1	110.0	90	In front of you is the entrance to the department, to your right are the stairs to the first floor.
2	1	290.0	90	To your right is the upper entrance of classroom ADII1, continuing in this direction you will find A28.
2	1	110.0	90	To your left is the upper entrance of classroom ADII1, continuing in this direction you will find the department.
3	1	290.0	90	To your right is the lower entrance of classroom ADII1, continuing in this direction you will find A28.
3	1	110.0	90	To your left is the lower entrance of classroom ADII1, continuing in this direction you will find the department.
4	1	290.0	45	You are at the entrance of classroom A28, turn left for the bathrooms, ExA27, and the stairs to the first floor.
4	1	20.0	45	At your left there is the entrance of classroom A28, turn right for classroom ADII1 and the entrance of the department.

Table 1: Location info table

The table includes reference azimuth values, which allow us to determine the direction in which the user is walking.
All the azimuth values have been sampled following the corridors' direction; in order to give information coherently when the user make a turn, we insert the Threshold value.
The "threshold" value is used to differentiate how changes in direction are detected between corners and straight corridors.
In corners, the change in direction occurs within a stricter range of angles. Thus, a direction change must be detected when the angle changes by at least 45°, unlike in straight corridors where a at least 90° change is expected, since in corridors there are only two possible directions, while in corners there are virtually four.

2.3 Android Structure

We will now give a brief overview of where the main functionalities of the application are located in the codebase.

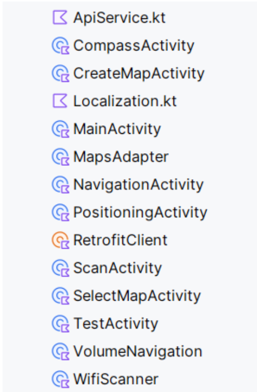

Figure 7: Kotlin classes and Activities

The **`MainActivity`** is an interface that sends the user into the admin options or the actual navigation application, through the buttons "Select Map" or "Admin Options".

`TestActivity` provides a central interface including all the Admin Options from managing maps to changing server addresses, and initiating different activities related to fingerprinting and positioning within the app.

The **`CreateMapActivity`** is one of the Admin options. In this activity the admin can create a new map by providing a building name and the number of rooms. Then it generates a Unique map ID using "`UUID.randomUUID().hashCode()`" and finally sends the `MapInfo` object, that contains the aforementioned information, to the database on the server.

If you press in the main activity the Select Map button, you will be redirected to the **`SelectMapActivity`**. This activity provides a user-friendly and location-aware way to select a map based on proximity to the user's current location. First the activity checks for location permissions and fetches the last known location. If the location is available, it proceeds to fetch and display maps based on the user's current location. It returns a list of maps that are within 200 meters of the user's location, that is handled by a custom adapter that takes a list of maps and sets up click listeners on each map item. When a map is selected, it starts the NavigationActivity and passes the selected map's ID.

The **`NavigationActivity`** uses a combination of Wi-Fi scanning and sensor data to determine the user's position in the indoor environment. It fetches the pre-recorded fingerprints and position information from the server, continuously scans for Wi-Fi networks, compares the results to find the nearest match, and provides audio feedback using TTS. The use of sensor data helps in determining the user's orientation, which enhances the accuracy of the navigation system. The activity provides the user information about his current position automatically whenever the user changes the zone he's in, or it repeats the information if manually prompted touching the screen.

2.4 VI accessible interfaces

To make the application more friendly we used high contrast coloring, and the buttons are in fact realized by splitting the screen into big sections like in this image.

At the start of the application,

In the menus the user can scroll the options using the volume up button and when selected they will be read aloud by the text to speech to make sure the user properly knows what they are selecting. Then the option can be confirmed by using the volume down button.

The selected option is the one highlighted in black (which might appear counterintuitive, but is more VI friendly since it's easier for them to read high contrast texts)

After pressing select map a list of maps will appear looking like this

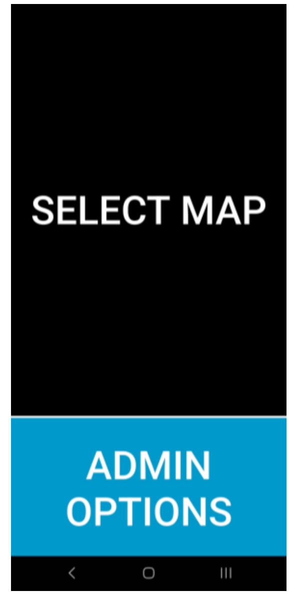

Figure 8: Home Screen

Figure 9: Map Selection

The user can select the desired one with the same mode as explained above (in this example only 2 maps are present but the 4 closer will be returned if present)

During the navigation the guidance phrases will be shown on screen and read aloud from the text to speech

If the user touches the screen anywhere the current guidance message will be repeated, else they will only be said when the user reaches the reference zone.

Inside the admin options there is the possibility to create your own map and add it to the server for it to be used later.

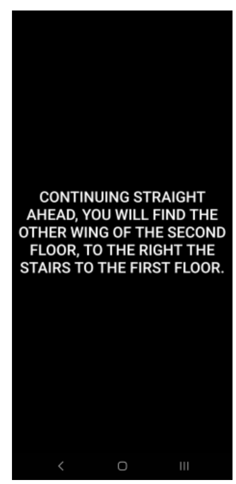

Figure 10: Navigation Phase

3 Experimental results

3.1 Map Structure

For the experimentation, we focused on mapping only the east side of the second floor of Polo A at the School of Engineering, University of Pisa. We needed to conduct several experiments to determine the optimal placement of the fingerprints on the map.

Initially, considering that V.I. individuals typically use a stick to walk while following walls, we placed fingerprints on both sides of the corridor (40 cm from the wall).
This was the initial disposition we tested:

Figure 11: First map created

However, the Euclidean distances based on Wi-Fi RSS values between the two sides of the corridor was insufficient, and the samples within each zone were too close together. This caused the algorithm to frequently misplace the user and made the computations intensive, as each run required calculating many distances (with each sample averaging 50-55 Wi-Fi RSS values).
We could have tried an approach where the comparison was performed with the adjacent zones but the changes in RSS values were still too low for this to work effectively as its shown in the heatmaps below.

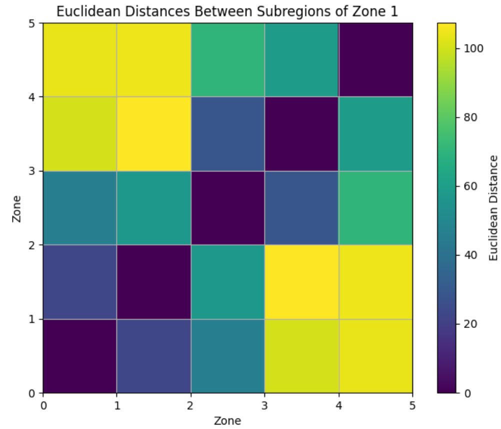

Figure 12: Distances between Subregions of zone 1

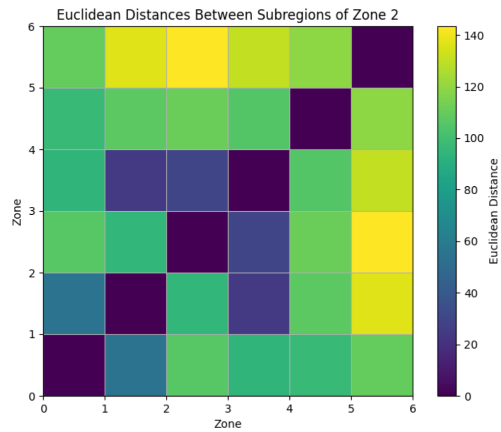

Figure 13: Distances between Subregions of zone 2

As an example there are shown only zone 1 and 2 but the case was the same for all the other zones, all the heatmaps can be found in the GitHub folder.

Therefore, we revised our strategy and mapped only the significant locations in the corridor.

This was the second map we obtained; we passed from 36 samples to just 8 making the computation a lot faster (30ms).

Figure 14: Final map with less fingerprints

During the map sampling mobile routers were detected so we removed them from the dataset along with the private Wi-Fi from the houses around.

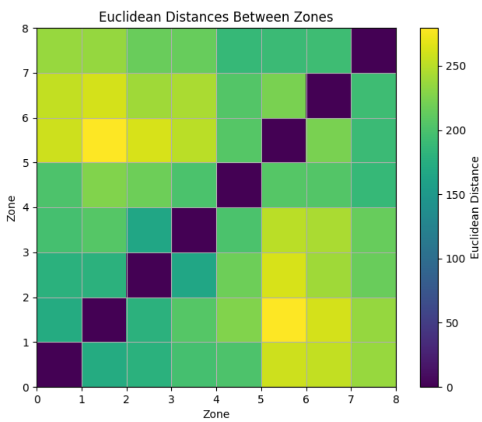

Figure 15: Distances between each zone of the map

3.2 Performances

For performance testing, we aimed to examine all scenarios that could be relevant for a visually impaired person. This included assessing whether performance degraded when walking along the edges of the corridor, as visually impaired individuals often use the wall for guidance, and whether the response time was fast enough to be practical.

The tests were performed on a Samsung A53 With Android 14 (2023) and a Realme X2 with Android 11 (2019), this latter one was also the one used to realize the maps.

3.2.1 General performance

First, we aimed to determine the accuracy of the application's zone predictions. To do this, we conducted 7 walks at approximately 0.6 m/s with two phones in the corridor, walking in both directions while staying in the center.

We considered a detection to be correct if it occurred inside the zone boundaries that we deemed optimal for a visually impaired individual to receive information about the surrounding Area of Interests (AoI), the delay present in the detection will be analyzed in the next section.

The correctness is expressed in percentage:

Correctness %	Samsung A53	Realme X2
ZONE 1	100%	100%
ZONE 2	100%	100%
ZONE 3	66%	100%
ZONE 4	33%	100%
ZONE 5	100%	100%
ZONE 6	100%	100%
ZONE 7	0%	100%
ZONE 8	100%	100%

Table 2: Table of testing going from the stairs to the department

Correctness %	Samsung A53	Realme X2
ZONE 1	100%	100%
ZONE 2	100%	0%
ZONE 3	66%	100%
ZONE 4	66%	100%
ZONE 5	100%	100%
ZONE 6	100%	66%
ZONE 7	0%	66%
ZONE 8	100%	100%

Table 3: Table of testing going from the department to the stairs

3.2.1 Walking sidelines or in the center

We tested the app also along the sides, to evaluate its compatibility with the walking habits of visually impaired individuals. We were particularly concerned about potential signal interference caused by the walls. Throughout the experiments, we found no significant changes in the results we obtained from the ones obtained walking in the center.

Correctness %	Samsung A53	Realme X2
ZONE 1	100%	100%
ZONE 2	100%	100%
ZONE 3	100%	100%
ZONE 4	100%	100%
ZONE 5	100%	100%
ZONE 6	100%	100%
ZONE 7	0%	100%
ZONE 8	100%	100%

Table 4: Table of sideline testing going from the department to the stairs

Correctness %	Samsung A53	Realme X2
ZONE 1	100%	100%
ZONE 2	100%	0%
ZONE 3	100%	100%
ZONE 4	100%	100%
ZONE 5	100%	100%
ZONE 6	100%	66%
ZONE 7	0%	66%
ZONE 8	100%	100%

Table 5: Table of sideline testing going from the stairs to the department

3.2.1 Delay Analysis

We also wanted to check the difference between reality and expectations about where the transition zones are located.

These are the differences obtained proceeding from the main stairs to the department.

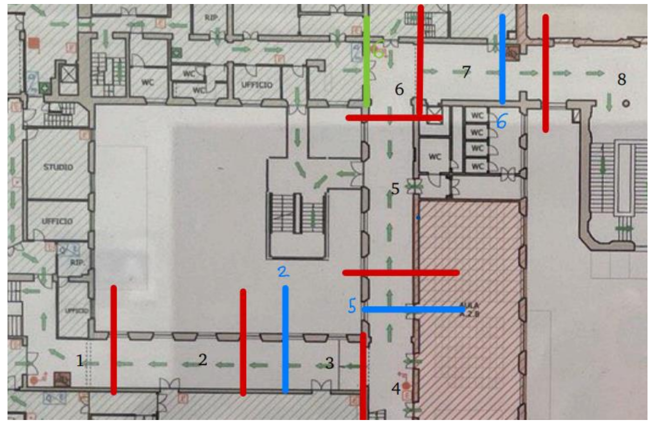

Figure 16: Delay map, path from stairs to Department

In red the expected transitions, in blue the ones obtained with the Samsung A53s and in green the ones obtained with the Realme X2.

Here are the ones in the opposite direction:

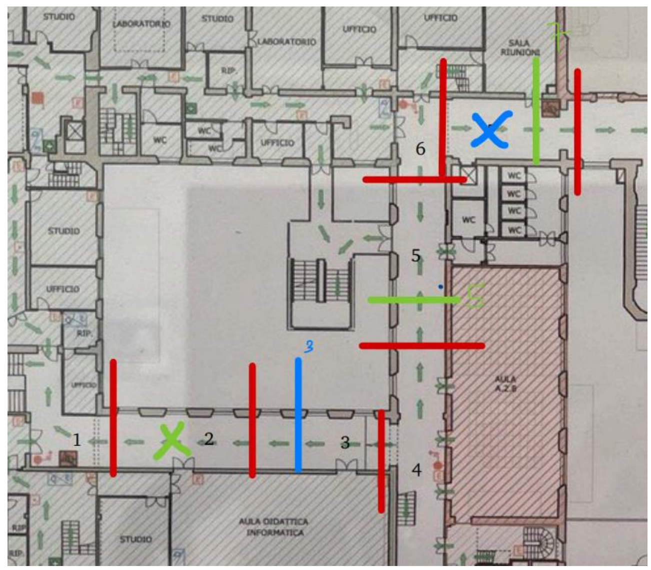

Figure 17: Delay map, path from Department to stairs

The X are the not detected zone (so the ones that were skipped and the device detected directly to the following one) as also explained in the tables.

3.2.3 Energy Consumption

We manually checked the energy consumption levels of the application on our smartphones and calculated the average consumption to determine the energy usage during app use. We obtained the following values:

- Realme X2: 237 mAh
- Samsung A53: 255.7 mAh

For comparison, we also checked the energy consumption values for Instagram on the same devices:

- Realme X2: 345 mAh
- Samsung A53: 337 mAh

This is of course dependent also form the very different use of the screen, since the nature of the application for visually impaired induces the use of high contrast text (so white text on dark screen). However, despite the very high frequency sampling rate the energy usage is not that high.

4 Conclusion

We could consider the results obtained as decent. We adopted one of the lighter algorithms and used only fingerprints and Wi-Fi signals making the application fast and portable since it doesn't need any kind of infrastructure.

Of course, map creation is a fundamental step which determines the performances of the algorithm, as it was explained above a "bad" map could lead the application to not working at all. So, the map creation step is the more time consuming one and takes the most effort, even though it makes the online phase very light.

The audio information provided is still of use to orientate inside the corridor despite what could be interferences and delays which we talked about in the dedicated section.

The map structure appears to be suitable; the main problems are the variation due to the device used for the application; this could be improved by performing multiple samples for each location at a different moment of the day and different devices and then averaging them to create a more robust value.

Of course, also the algorithm could be improved by combining it with the many listed in the literature above.

REFERENCES

[1] Liu, F., Liu, J., Yin, Y., Wang, W., Hu, D., Chen, P. and Niu, Q. (2020), Survey on Wi-Fi-based indoor positioning techniques. IET Communications, 14: 1372-1383. https://doi.org/10.1049/iet-com.2019.1059

[2] Simões, W.C.S.S.; Machado, G.S.; Sales, A.M.A.; de Lucena, M.M.; Jazdi, N.; de Lucena, V.F., Jr. A Review of Technologies and Techniques for Indoor Navigation Systems for the Visually Impaired. *Sensors* 2020, *20*, 3935. https://doi.org/10.3390/s20143935

[3] Mahida, P.; Shahrestani, S.; Cheung, H. Deep Learning-Based Positioning of Visually Impaired People in Indoor Environments. *Sensors* **2020**, *20*, 6238. https://doi.org/10.3390/s20216238

[4] T. L. N. Nguyen, T. D. Vy, K. -S. Kim, C. Lin and Y. Shin, "Smartphone-Based Indoor Tracking in Multiple-Floor Scenarios," in IEEE Access, vol. 9, pp. 141048-141063, 2021, doi: 10.1109/ACCESS.2021.3119577.

[5] Shang, S., Wang, L. Overview of Wi-Fi fingerprinting-based indoor positioning. *IET Commun.* 16, 725–733 (2022). https://doi.org/10.1049/cmu2.12386