Multi-technology Indoor Localization with Data Fusion

Daniel Zheng Dong
Master's Computer Science
Departamento de Ciência De Computadores
2020/2021

Supervisor
Rui Pedro De Magalhães Claro Prior, Assistant Professor,
Faculdade de Ciências da Universidade do Porto

Co-Supervisor
Sérgio Armindo Lopes Crisóstomo, Assistant Professor,
Faculdade de Ciências da Universidade do Porto

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/_________

Abstract

Indoor Location-Based Services (ILBS) are more and more important nowadays, whether it is to track our positions, guide us to a specific place or to issue context-aware commands, such as turn on the lights of this place. And for that purpose, Global Positioning System (GPS) is not reliable or precise enough for indoor environments.

With each passing year, there are more and more technologies and techniques becoming available to implement an indoor localization system. In this work, we want to address the possibilities of such a system based on WiFi Round Trip Time (RTT). This technology improves previous techniques used for WiFi, mainly Received Signal Strength (RSS). However, it still suffers from multipath propagation and obstacles. We will enhance the reliability of the position estimated using more information from other methods that use motion sensors, QR codes and data fusion with Kalman Filter. In the end, we evaluate the performance of each module and compare it between themselves and in different environments. In more indoor localization oriented placements of Access Points (APs) we achieve an error rate below 2m at 70% of time.

Resumo

Hoje em dia, serviços baseados em localizações interior são cada vez mais importantes, quer para localizar as nossas posições, para nos guiar até um lugar específico ou permitir comandos contextuais, tais como, liga as luzes deste lugar. E para tal, Global Positioning System (GPS) não é fiável ou preciso suficiente para ambientes interiores.

Cada ano que passa, temos cada vez mais tecnologias e técnicas disponíveis para implementar um sistema de localização de espaços interiores.Neste trabalho, queremos abordar as possibilidades de um sistema baseado em WiFi Round Trip Time (RTT). EEsta tecnologia melhora as antigas técnicas de WiFi existentes, principalmente Received Signal Strength (RSS). Contudo, este também sofre problemas de multi-caminho e obstáculos. Queremos melhorar a confiança das posições estimadas pelo sistema utilizando mais informações de outros métodos que utilizam sensores de movimento, códigos QR e combinação de dados com filtro de Kalman. Ao final, avaliamos o desempenho de cada módulo, comparando-os em diferentes ambientes. Em um ambiente onde a colocação dos pontos de acesso levou em consideração a localização interna, conseguimos obter posições com erro inferior a 2m, 70% das vezes.

Acknowledgments

I would like to thank professor Rui Prior and Sérgio Crisóstomo for the opportunity and guidance along with the development of this work. I also want to express my gratitude to the Instituto de Telecomunicações (UIDB/50008/2020) for hosting this work. And a special thank you to SafeCities (POCI-01-0247-FEDER-041435) for providing a scholarship and necessary equipment for the development of this work.

I would like to thank my family for being there and supporting me. Thank you to my friends and DinoTeam for looking after me.

Contents

List of Tables

List of Figures

Acronyms

AOA Angle Of Arrival

API Application Programming Interface

AP Access Point

CDF Cumulative Distribution Function

CNN Convolutional Neural Network

CRF Conditional Random Fields

EKF Extended Kalman Filter

FTM Fine Time Measurement

GNSS Global Navigation Satellite System

GPS Global Positioning System

GSM Global System for Mobile Communications

HMM Hidden Markov Model

HW Hybrid Wireless

IEEE Institute of Electrical and Electronics Engineers

ILBS Indoor Location-Based Services

KNN K-Nearest Neighbor

LOS Line of Sight

NFC Near Field Communication

NLOS Non-Line of Sight

PAN Personal Area Networks

PDR Pedestrian Dead Reckoning

RSS Received Signal Strength

RTT Round Trip Time

SVM Support Vector Machines

TDOA Time Difference Of Arrival

TOA Time Of Arrival

TSML Two Stage Maximum Likelihood

ToF Time Of Flight

UKF Unscented Kalman Filter

UWB Ultra Wide Band

WLAN Wireless Local Area Network

dBm Decibel-Miliwatts

Chapter 1

Introduction

The world is becoming more and more connected with each passing year. Not only information is travelling at a faster rate than ever. Even physical distance is not an obstacle for us anymore. In a blink of an eye, we are on the other side of the world.

With the increase of movements, visiting unknown places is becoming such a frequent occurrence in our life that there is also an increased need for indoor guidance. Indoor Location-Based Services (ILBS) provides localization in environments where the Global Positioning System (GPS) could not reach and higher precision required for indoor environments.

ILBS has been studied for a long time, using a variety of technologies such as WiFi, Bluetooth and the uses have become more and more widespread with a market value of 7.0 billion USD and is expected to be worth 19.6 billion USD[17].

Besides the obvious applications of guiding a user through a mall to a specific shop or restaurant, ILBS can also help us keeping track of people, such as in a hospital, we can keep track of patients, given them a bigger sense of freedom but in case of emergency we can still easily find them and give treatment. It can cooperate with other system that require spatial context, such as in a museum, give the user more information about an art piece if the user is in front of said art or for a smart house to understand commands like, light up this room.

This comes with a lot of challenges as it is often indoor environments where the signal from satellites can not easily reach the device. It also requires much higher precision that GPS based localization provides as inside a typical house a difference of couple meters can result in being in a completely different room.

1.1 Problem statement

With that being said, this is still an area with a lot of improvements to do, with new technologies and techniques becoming available each passing year.

We want to address the possibilities of a hybrid indoor localization system that uses WiFi Round Trip Time (RTT) as the core piece.

Objectives

With this question in mind, we want to implement a Multi-Technology Indoor Localization system for Android devices.

To complete this system, we defined 5 objectives to achieve our goal:

1. Design a system that can utilize WiFi RTT
2. Identify the strong points of the technology and its limitations
3. Implement modules that can compensate the downsides of WiFi RTT
4. Evaluate the performance of each module of the localization system
5. Evaluate the improvements made by the hybrid localization module

In the end, we want to achieve a localization system that can both provide real-time positioning as well as high accuracy.

1.2 Structure

In chapter 2 we discuss the state of the art of indoor localization technologies, techniques and methods. In chapter 3 we introduce some background knowledge about the technologies used and Android implementations of those technologies. In chapter 4 we present the system design and discuss some challenges that we had to solve. In chapter 5 we explain the experimental setups and evaluate the performance of each module and the improvements made by data fusion. Lastly, in chapter 6 we present the main conclusions of this work and point a few directions for future work.

Chapter 2

State of the Art

In this chapter, we discuss the state of the art of indoor localization, the different technologies available, the techniques advantages and disadvantages and the methods that use said techniques. We also talk about localization improvements with little to no extra infrastructure cost as the hardware used is already available for other purposes.

2.1 Technologies

There are many technologies available[28] that provides indoor localization services. In this section, we discuss radio wave-based technologies, such as WiFi, Bluetooth and Ultra Wide Band (UWB).

2.1.1 WiFi

WiFi is a network protocol based on IEEE 802.11 standard, primarily used to connect devices to the Internet wirelessly. In indoor environments, it has a reception range of 20m, but some modern Access Point (AP) claims to have a range up to 150m in outdoor environments. This drastic difference in reception range is due to the frequency that WiFi operates, 2.4GHz or 5GHz, and because it is easily disturbed by obstacles. The most commonly used techniques for this technology is Received Signal Strength (RSS) and Round Trip Time (RTT) discussed with more detail in section 2.2.1 and 2.2.4.

Most places now have WiFi coverage, and most mobile devices are WiFi-enabled, making WiFi an easy choice for indoor localization and also one of the most studied localization technologies[5, 14, 21, 22]. More recently, the IEEE 802.11mc amendment introduced WiFi RTT allowing more precise distance measurement than previously possible.

2.1.2 Bluetooth

Bluetooth is wireless technology primarily used to exchange data between devices in the vicinity and building Personal Area Networks (PAN). The range of Bluetooth devices depends on the class, with class 2 being the most common in smartphones, ranging from 5 to 10m. The actual range can be affected by the conditions of the environment.

Bluetooth and WiFi shares some similar applications, but the lower power consumption and lower infrastructure cost can make Bluetooth more desirable in some cases. This technology is also well studied[21, 28] using RSS but Bluetooth 5.1 introduces major improvements in Angle Of Arrival (AOA) allowing more techniques and more accurate localization[6]. These techniques are talked in detail in section 2.2.1 and 2.2.5.

2.1.3 UWB

UWB is a radio technology that can use reduced energy level, short-range, high bandwidth communications, typically used for radar imaging. Unlike WiFi and Bluetooth that are supported, by most smartphones, only recently the high-end smartphones support UWB.

This technology has an emphasis on using Time Of Flight (ToF) based techniques that we discuss with more details in sections 2.2.2, 2.2.3 and 2.2.4. In contrast to traditional radio transmissions that use varying power levels, frequencies or phases to transmit information, UWB generates wide bandwidth radio energy at specific intervals. Having pulse lengths in the order of nanoseconds helps solve the multipath propagation problem, and the large bandwidth improves the penetration of obstacles, improving the accuracy of ToF[18, 24].

2.2 Techniques

The available technology allows us to use numerous techniques, each having its strengths and weaknesses. In this section, we discuss these techniques.

2.2.1 RSS

RSS is the signal strength, typically measured in Decibel-Miliwatts (dBm) and the closer the receiver is to the transmitter, the higher is the measured strength. Based on the measured value, it is possible to infer the distance between the devices. However, RSS is not a reliable technique due to the deterioration of the measurements caused by the multipath propagation and Non-Line of Sight (NLOS)[26, 28]. On the flip side, it is the simple and most available technique as it doesn't require any special hardware. In section 2.3.2 we talk about the fingerprinting that resolves the issue but, it is still a relatively coarse localization technique.

2.2.2 TOA

Time Of Arrival (TOA) is one of the techniques that use ToF to derive the distance between 2 devices by multiplying the time the packet uses to travel by the speed of light. Because of this, even the smallest errors in the measurement of the time can cause a huge error in the calculated distance. Although still affected by the problems RSS, it is more accurate. The downside is the requirement of specialized hardware and strict time synchronizations between the transmitters and receivers.

2.2.3 TDOA

Due to the strict time synchronizations, it may be impractical to use TOA. A variant, Time Difference Of Arrival (TDOA), can be used, which lessens the constraints on the synchronizations of the receiver, only requiring strict synchronization between the transmitters. This technique is used in multilateration methods with prior knowledge of the positions of the transmitters, TDOA measures the difference between the times of arrival to infer the location of the receiver[28].

2.2.4 RTT

This technique works similarly to TOA and TDOA. All three uses ToF, but, unlike the last two, RTT doesn't require strict time synchronization because the distance calculated from the two-way propagation delay (round trip time) is measured by exchanging timestamped data packets.

In particular, Wifi RTT[7], using the 802.11mc protocol capable devices, the smartphone(receiver) sends a Fine Time Measurement (FTM) request and the AP (transmitter) responds with a FTM packet. When the smartphone receives this packet, it responds with an acknowledgement packet. By the end of this exchange, the smartphone has every timestamp it needs to calculate the distance to the transmitter. In figure 2.1 we can see how the distance is calculated using WiFi RTT.

2.2.5 AOA

AOA can be measured with an array of antennas by calculating the TDOA to the different elements of the array. The main advantages are that it requires only needs 2 transmitters when the receiver is the mobile beacon and no strict synchronization. However, it requires specialized and expensive receivers, and a small error in the measurement can result in a big error in the position estimation.

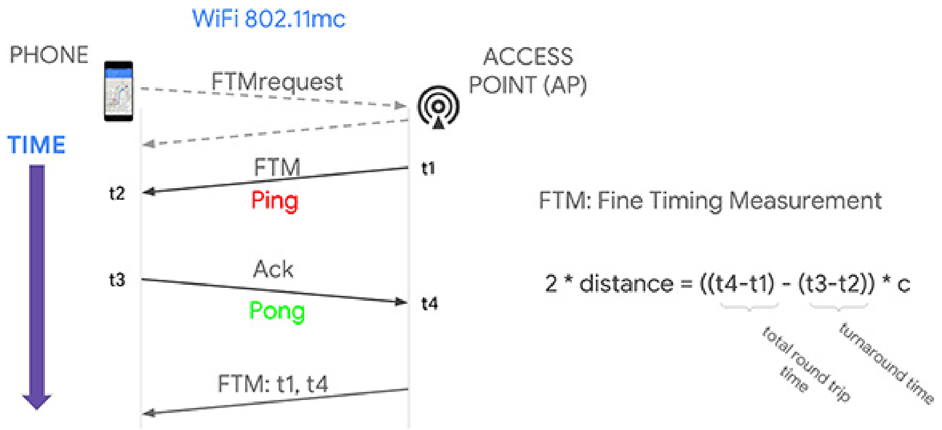


Figure 2.1: WiFi RTT principles[7]

2.2.6 Proximity

This technique utilizes various readers such as Near Field Communication (NFC) or to some extent Bluetooth APs, identification card terminals and other devices that require close proximity to be able to read. These devices are placed in known key areas, and, when a user is in range, their tag, card or any other identification device is read or vice versa, thus knowing their position.

This method trades the ability to get real-time positioning for high accuracy in a certain spot. It is best used when the accuracy is more important than the frequency, such as detecting if the user has passed through a gate[28] or to calibrate other relative methods, for example, Dead Reckoning.

2.3 Localization Methods

We will now discuss the state-of-the-art methods using the above techniques. In particular, we will discuss multilateration using ToF, fingerprinting with RSSs, and Dead Reckoning using motion sensors. We will also discuss hybrid approaches using a combination of methods.

2.3.1 Multilateration

Multilateration algorithms compute the position of the user from estimated distances to reference points with known position (anchors) by using trigonometry. Distances are estimated using signals such as WiFi, Bluetooth, radio waves, UWB measuring the RSS, TOA, TDOA or AOA. Different techniques will have different strengths and weaknesses. And more adequately applied to certain methods than others.

2.3.1.1 Least Squares

One of the most popular methods is least squares, an optimization technique whose objective is to minimize the sum of the square of the residuals of a set of equations. In the case of localization, the equations are the distance equations to each AP:

$$d_i^2 = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 \tag{2.1}$$

where the user is located at (x, y, z), the coordinates of i-th AP is (x_i, y_i, z_i) with distance d_i.

Note that the equations are quadratic, and trying to solve least squares in a non-linear setting is computationally expensive. However it is possible to linearize theses equations[8] with a reference AP r. Then, all other equations are subtracted from the reference AP.

$$d_i^2 - d_r^2 = [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2] - [(x - x_r)^2 + (y - y_r)^2 + (z - z_r)^2] \tag{2.2}$$

Then, all equations are expanded and simplified

$$d_i^2 - d_r^2 = (x^2 - 2xx_i + x_i^2 + y^2 - 2yy_i + y_i^2 + z^2 - 2zz_i + z_i^2) - (x^2 - 2xx_r + x_i^2 + y^2 - 2yy_r + y_i^2 + z^2 - 2zz_i + z_r^2) \tag{2.3}$$

$$d_i^2 - d_r^2 = -2xx_i + x_i^2 - 2yy_i + y_i^2 - 2zz_i + z_i^2 + 2xx_r - x_i^2 + 2yy_r - y_i^2 + 2zz_i - z_r^2 \tag{2.4}$$

After this operation, the unknown quadratic variables are gone, and we may rearrange the equations to be solved by a least squares solver.

$$2x(x_r - x_i) + 2y(y_r - y_i) + 2z(z_r - z_i) = d_i^2 - d_r^2 - x_i^2 - y_i^2 - z_i^2 + x_r^2 + y_r^2 + z_r^2 \tag{2.5}$$

After this step, it is simply a matter of using a solver to solve a linear least squares problem[19].

2.3.1.2 Bayesian Grid

Another way is to calculate the probability of the user is at certain coordinates. In the work of Horn [14], a Bayesian grid is used where each cell has a probability of the user being there, based on the measured distances.

In figure 2.2, Horn collected data and observed that the ratio between the reported distance and actual distance were mostly independent of the distance.

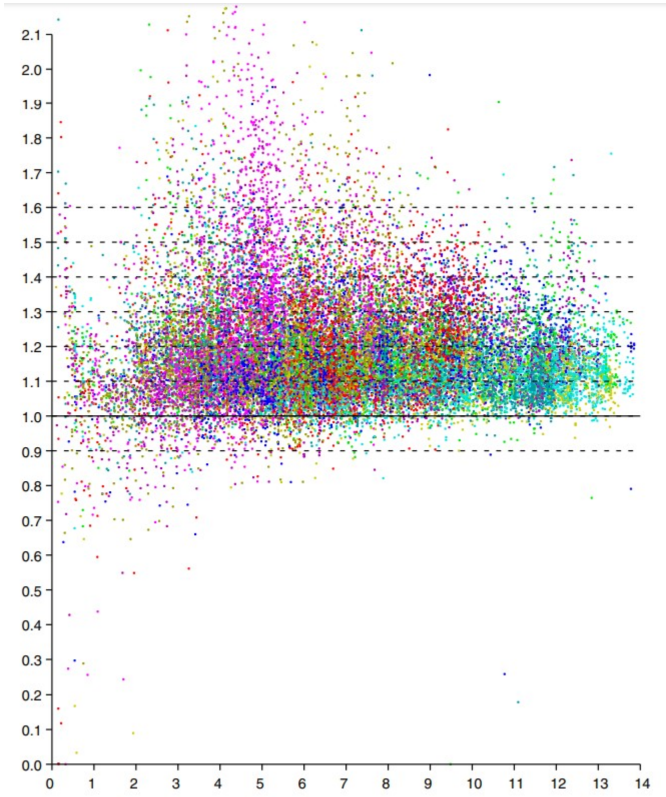


Figure 2.2: Scattergram of ratio of reported to actual distance (vertical axis) versus actual distance (horizontal axis—meters). [14]

So based on the collected data, Horn presents a double exponential with flat top model:

$$h(r) = \frac{1}{H} \begin{cases} e^{-(r_l - r)/s_l} & r < r_l \\ 1 & r_l \leq r \leq r_r \\ e^{-(r_r - r)/s_r} & r_r < r \end{cases} \tag{2.6}$$

In figure 2.3, we can see the heuristic(blue line) matches almost perfectly with the data collected. The red curve is smoothed version of the heuristic[14]. Where r is the ratio of the measured distance to actual distance, $H = s_l + (r_r - r_l) + s_r$ normalizes the distribution so the integral with respect to r is 1, r_l and r_r are the ratio where the flat top begins and ends. Finally, s_l and s_r are parameters estimated from the scattergram that can depend on the structure and material of the building, but Horn's experiments show that there is no need for great precision for reasonable results.

Unlike the fingerprinting method discussed in the next section, this method is not designed specifically for one location, as the values used can be generalized and deployed anywhere.

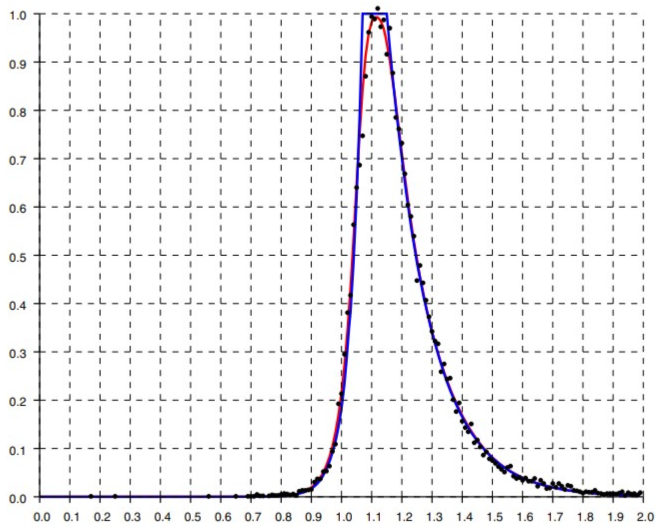


Figure 2.3: Histogram of the ratio of reported to actual distance based on about 20,000 measurements (scaled so that the peak equals one).[14]

2.3.2 Fingerprinting

This method is based on the use of machine learning techniques to locate a user by training a model. Similarly to a human with unique fingerprints, each place has a unique combination of signals, and the more signals each position can receive the more unique each place becomes.

There are two stages for this methods[28]. The first one is an offline phase, where data are collected for an area. These datasets are then used to train a model, using, for example, an artificial neural network such as Convolutional Neural Network (CNN)[16] or K-Nearest Neighbor (KNN)[19].

The main advantage of using fingerprinting is using values from which the distances cannot easily be directly inferred, for example, RSS. However, the data set used for fingerprinting is custom for each location and non-transferable, so the amount of work needed can be tedious before getting a functioning model.

2.3.3 Dead Reckoning

Another popular method is Dead Reckoning, a relative localization method that uses motion sensors such as an accelerometer to get the distance travelled and a gyroscope or compass to get the direction to compute the relative position at set intervals.

The work of Kang and Han[15] introduces SmartPDR, a smartphone-based Pedestrian Dead Reckoning (PDR) taking advantage of the inertial sensors present in most modern smartphones. This method has its challenges. For example, in a straight line, the way a person holds the smartphone may vary, changing the angle measured, unlike a foot-mounted inertial sensors where the measurement stays relatively consistent.

In figure 2.4 is the architecture of SmartPDR can be divided into 3 key processes: step detection, heading estimation, and step length estimation. The step detection and step length estimation are based on the reading from the accelerometer, and the heading estimation is done using a combination of the magnetometer and gyroscope.

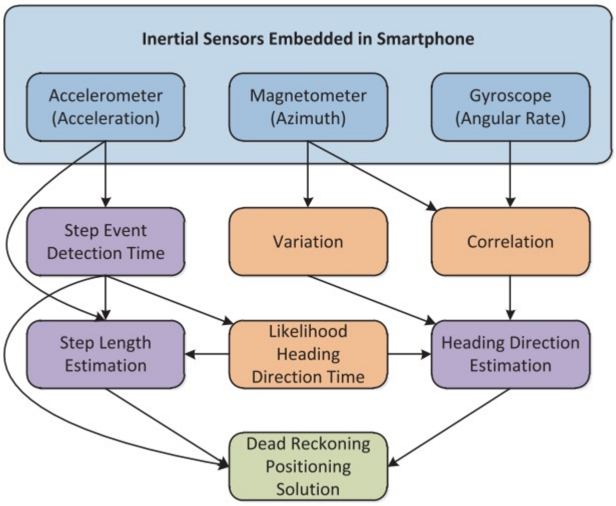


Figure 2.4: SmartPDR system architecture[15]

Step detection is done with the reading of the z-axis from the accelerometer, and the set of points where the following three conditions are met are registered step events:

$$t_{peak} = \{t | a_t^{step} > a_{t+i}^{step}, a_t^{step} > a_\tau^{peak}, |i| \leq \frac{N}{2}, i \neq 0\} \tag{2.7}$$

$$t_{pp} = \{t | max(a_t^{step} - a_{t-i}^{step}) > a_\tau^{pp}, max(a_t^{step} - a_{t+i}^{step}) > a_\tau^{pp}, 1 \leq i \leq \frac{N}{2}\} \tag{2.8}$$

$$t_{slope} = \{\frac{2}{N} \sum_{i=t-\frac{N}{2}}^{t-1} (a_{i+1}^{step} - a_i^{step}) > 0, \frac{2}{N} \sum_{i=t+1}^{t+\frac{N}{2}} (a_i^{step} - a_{i-1}^{step}) < 0\} \tag{2.9}$$

where:

- t_{peak} is the set of peak points that exceed a threshold a a_τ^{peak} (0.5m/s2)
- t_{pp} is the set of points where the the difference of the peak to the previous and next valley exceeds the threshold (1m/s2)
- t_{slope} is the set of points where the slope previous to the point is positive and the slope after the point is negative

The intersections of these three sets are the points where step events are registered.

Heading estimation is a combination of the reading of the gyroscope, compass and previous heading and are chosen based on the following rules.

$$\begin{cases} \omega^{pmg}(\omega^{prev}h_{t-1} + \omega^{mag}h_t^{mag} + \omega^{gyro}h_t^{gyro}) & h_\Delta^{cor} \leq h_\tau^{cor}, h_\Delta^{mag} \leq h_\tau^{mag} \\ \omega^{mg}(\omega^{mag}h_t^{mag} + \omega^{gyro}h_t^{gyro}) & h_\Delta^{cor} \leq h_\tau^{cor}, h_\Delta^{mag} > h_\tau^{mag} \\ \omega^{pg}(\omega^{prev}h_{t-1} + \omega^{gyro}h_t^{gyro}) & h_\Delta^{cor} > h_\tau^{cor}, h_\Delta^{mag} > h_\tau^{mag} \\ h_{t-1} & h_\Delta^{cor} > h_\tau^{cor}, h_\Delta^{mag} \leq h_\tau^{mag} \end{cases} \tag{2.10}$$

where

- h_t^i is the heading at timestamp t for the sensor i
- ω^i is the weight parameter form the sensors i

And finally, the step length estimation is calculated using the difference between the current step peak and the previous valley of the acceleration.

$$l_{step} = \beta \sqrt[4]{a_{peak} - a_{valley}} + \gamma \tag{2.11}$$

$$l_{step} = \beta \log(a_{peak} - a_{valley}) + \gamma \tag{2.12}$$

Kang and Han claim that the logarithmic equation performs slightly better when the steps are longer, in other words when the difference of the acceleration is larger[15].

There are several advantages to this method. For example, it does not require any transmitters to locate a user, so zero additional cost to the infrastructure. Also, most modern smartphones already come equipped with motion sensors, so there is also no additional cost to the end-user.

There are also a few disadvantages, and the biggest issue is the drift caused by each iteration small errors that accumulate over time. After a few iterations, the precision suffers greatly. And with this method alone, there is no way to get the absolute position, only the relative position to the initial given position.

2.4 Hybrid Localization Methods

This section discusses Hybrid Methods, which take advantage of the strengths of individual methods and compensate for each other's weaknesses. With this approach, it is possible to achieve more accurate localization than with individual methods.

2.4.1 Motion Sensors and WiFi

The coverage of WiFi signals is usually the whole building, with many APs spread around the area, but they can suffer from NLOS and multi-path propagation. Previous works used RSS which had problems with real-time positioning and was not always reliable due to the complex nature of indoor environments. Meanwhile, Dead Reckoning uses the motion sensors from a smartphone so the measurement errors will not depend on the location. This reading will be used in small intervals of time or using a step detection algorithm and the measurements will be reasonably accurate but this method only computes the relative position and, used alone, would suffer from drift.

The work of Yu et al[27] uses Multi Pattern-based Dead Reckoning, which can detect lateral and backward walking, something that traditional dead reckoning algorithms could not detect since the held smartphone's heading doesn't change for these types of movement. The results were 2% error for forwarding step detection, 5% for backward step and around 8% for lateral steps, and the step length module has error rates no more than 5%.

Yu et al also combine motion sensors with WiFi to achieve better results. The accelerometer is used to detect steps that are combined with step length. The gyroscope and magnetometer are fed to a Extended Kalman Filter (EKF) calculate the heading. The IEEE 802.11mc protocol is used to query the distances to each APs, and the position is computed using the least squares algorithm. Finally, all this information is fed to the Unscented Kalman Filter (UKF) compute the final position.

The final position computed is within 2m from the real position. The help of the WiFi multilateration method corrects the natural drift of the dead reckoning method.

2.4.2 WiFi and UWB

The work of Monica and Bergenti[18] proposes a hybrid localization algorithm using Wifi and UWB. The reason for using these technologies is the availability of WiFi infrastructure in most indoor environments, consequently little to no extra cost of infrastructure, but WiFi RSS doesn't provide accurate information to pinpoint the position. To solve this problem, UWB is used. UWB devices transmit short-duration pulses and signals that have a wider range and lower frequency. These characteristics help penetrate through obstacles and, thus, obtain more accurate ToF.

Two Stage Maximum Likelihood (TSML) algorithm uses two stages to solve two least squares problems to get the position with maximum likelihood. In the first stage, Monica and Bergenti[18] solve the system as if it were linear, obtaining the solution vector v composed of 4 components. The second stage of the algorithm addresses the fourth component of the vector, which is based on the first three components, obtaining the final estimate of the position.

They did experiments in a 4 by 4m room with 3m height, and the transmitters were placed in the middle of the walls at different heights, in four different configurations, where they varied the number of UWB nodes.

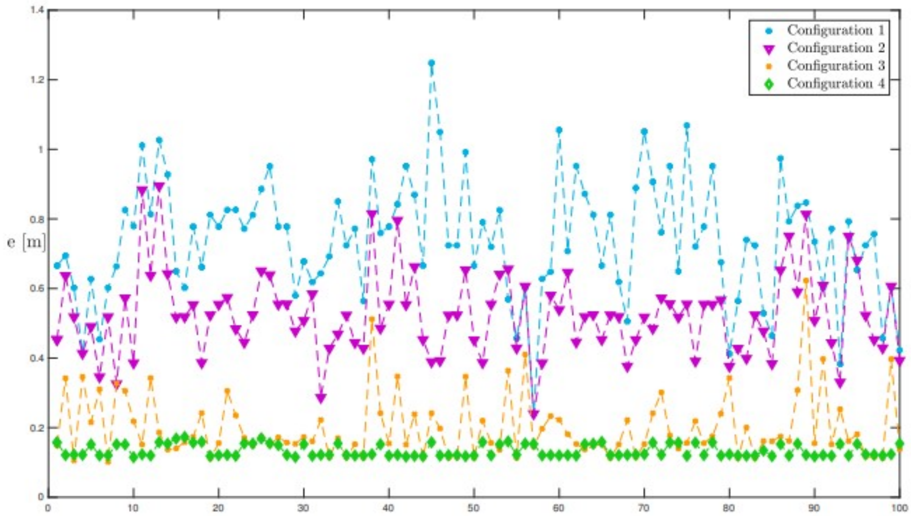


Figure 2.5: Localization errors for different configurations [18]

With 4 WiFi transmitters, errors were around 50~70cm. When 2 of them were replaced by UWB transmitters, errors were reduced to around 20cm. When all of the nodes were UWB, the results were improved, but not significantly as shown in figure 2.5.

2.4.3 RSSI HYBRID

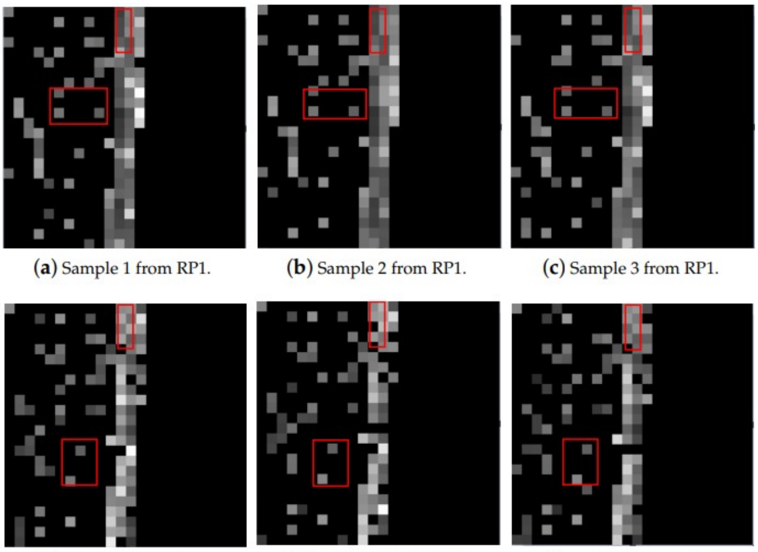
Liu et al[16] proposed a localization method using Hybrid Wireless (HW) fingerprinting based on a CNN improving the accuracy compared to other methods.

RSS are often unreliable. Short term disturbances, such as moving objects or even people walking around, or long term disturbances, like humidity caused by the weather, can affect the accuracy of a localization module. To solve this problem Liu et al propose HW fingerprint because they observed that different the RSS value can vary because of the weather, the ratio between AP stay relatively stable when every APs is under the same conditions.

The HW fingerprint is a sequence of RSS followed by the ratio fingerprint, which is calculated as follows:

$$ratio_p = \begin{cases} \frac{RSS_k}{RSS_i} & RSS_k \neq 0, RSS_i \neq 0, 1 \leq k < i \leq v \\ 0 & RSS_k = 0 \vee RSS_i = 0, \end{cases} \tag{2.13}$$

CNN has strong feature extraction capabilities in image classification. Liu et al transformed indoor localization into an image classification problem. HW fingerprint is transformed into an image where RSS fingerprint is compressed into a distribution of 0-1 bound by RSS max and RSS min, and ratio fingerprint is normalized, then they reshape the vector into a matrix, obtaining an image in grayscale as in figure 2.6.



(a) Sample 1 from RP1. (b) Sample 2 from RP1. (c) Sample 3 from RP1.
(d) Sample 4 from RP2. (e) Sample 5 from RP2. (f) Sample 6 from RP2.

Figure 2.6: Grayscale images constructed by HW-fingerprint[16]

Liu et al did experiments with different models, KNN, Support Vector Machines (SVM) and CNNi, and the accuracy of each model is 67.79%, 79.97% and 84.17% and the HW fingerprint on average improved the result by 3.39%, 8.03% and 9.03% respectively.

2.5 Localization Improvement

Besides improvement in accuracy based on hybrid methods, it can also be improved using spatial context which has no added cost in terms of hardware and is usually already available. Prior knowledge of a floor plan can be used to constrain the estimated motion of a user, for example, if there is a wall between the previous and next position. With this information, the system can safely assume that the user did not go through the wall, but got a lot closer. There are mainly two methods that are based on a floor plan, map matching and spatial models, and another method that uses landmarks to correct the position.

2.5.1 Map matching

Gu et al[13] compiled map matching methods and divided them into three main categories: point to point, trajectory and probabilistic graphical model, each with its advantages and disadvantages.

Point to point method is the simplest of the three, by matching the location to a point according to the floor plan. It is also computationally efficient but may not offer the best performance.

Trajectory matching is based on the topology information from the floor plan to match trajectory paths. This method can be a complement to dead reckoning algorithms, adjusting the movements detected with the most likely path. Compared to point to point is more robust and accurate, but it is also more complex and may have real-time capability issues.

The probabilistic graphical model computes the position based on the probability of the position, then the probabilities are updated according to the model and spatial constraints. The most popular models are Conditional Random Fields (CRF) and Bayesian techniques like particle filters and Hidden Markov Model (HMM). These models can provide even better accuracy than the previous methods, but at a cost of high computational requirements. Xiao et al [25] developed a lightweight model based on CRF that can be done on the smartphone in under 10ms.

2.5.2 Spatial Model-based Methods

Unlike map matching, a spatial model has more information compared to a simple floor plan. It can contain additional information about each room such as the location of the furniture, the position of the doors or windows. At the cost of being more complex, this method can be more accurate than the former. Gu et al[13] separated these methods into two main categories, grid model and graph model.

Grid models divide the space into cells of the same size and shape, each cell has a probability of the user being in it, and unreachable cells have 0%, each cell will update its probability at each iteration based on the previous information of the position and information of the spatial model.

Graph models, on the other hand, represent indoor structures with a graph composed of nodes and edges. Each node can represent a room, door or other relevant location, and the edges connect nodes containing the information of distance.

2.5.3 Landmarks

Another way to improve localization is using landmarks, which are points of interest with unique characteristics. The landmarks can be visual or more abstract, but the more distinct the characteristics are the better the landmark is.

Gu et al[13] presented several landmarks, detecting indoor or outdoor using Global Navigation Satellite System (GNSS) or Global System for Mobile Communications (GSM), proximity landmarks using WiFi or Bluetooth, turn landmarks with gyroscope or compass, movement-based using accelerometer, and other miscellaneous landmarks.

2.5.4 Indoor or outdoor

The readings of GNSS and GNSS can be used as an indoor or outdoor landmark due to the inability of the satellites signals to penetrate walls or other obstacles, So the reduction of numbers of satellites or GSM towers that can communicate with a device when positioned at an entrance or windows the numbers and that change can be used as a landmark to situate or limit the locations.

2.5.5 Proximity

Although not the primary use of Bluetooth and WiFi, the RSS can be a bit unstable due to the environment but at close proximity the RSS will peak and be stable enough to be used as a landmark, thus helping locate a user. For this purpose Bluetooth beacons are preferred due to consuming less energy and being relatively cheap, consequently, it is more economically efficient to spread more transmitters throughout the building at key areas to obtain more useful information.

2.5.6 Corners

Gyroscope and compass can measure the angular velocity and directions, with this information we can use it to detect when the user is taking a left or right turn. With knowledge about the floor plan, it is possible to locate or at least limit the positions of the user. Gyroscope, unlike the compass, is not affected by the nearby magnetic field but it may miss subtle turns, while the compass can detect these subtle turns, it is influenced by nearby magnetic fields.

Acceleration

Acceleration landmarks are the change of state when the user is opening a door, going up or down a flight of stairs or riding an elevator. The measurements can be captured by an accelerometer and these different actions have their own different patterns of measurements. These patterns can limit where a user can be, thus improving or locating the said user.

Acoustic

Acoustic landmarks use the microphone to record the sound, and some locations have some unique sounds such as vending machines, washing machines or any other sources of distinct sounds and based on this information, it can limit the positions available.

Visual

Visual landmarks use the camera to capture the surrounding area, and by using some kind of image recognition method to recognize objects that are limited to certain places, such as partings or posters, fire extinguishers, door plates, signs, QR codes or other distinct objects.

Barometer

Barometer landmarks are based on detecting changes in air pressure to derive changes in altitude and can be obtained from a barometer. This change can be used to know when a user goes up or down the floor and limits the position to the entrance of the floor such as the entrance of an elevator or stairs. These readings can be influenced by the temperature or other factors, but the disturbances can be negligible compared to the change of altitude.

Magnetic

Magnetic landmarks are the Earth's own magnetic field and magnetic anomalies. These anomalies can be caused by magnetic objects in the surrounding area. The anomalies are present in the most indoor environment such as household appliances, elevators.

Light

Light landmarks use light sensors to detect the intensity of light reaching the device. The light sources can be artificial lights, such as ceiling lights or lamps, or natural light near windows or entrances and at certain places the intensity is at its peak and used as a landmark to pinpoint a user.

Hybrid

Some landmarks can be classified into multiple categories. For example, washing machines can be acoustic, visual and magnetic landmarks. This increases the uniqueness of said landmark and can locate better than the simple landmark.

Chapter 3

Background

In this chapter, we present some basic knowledge about the technology used in our implementation, alongside software and Application Programming Interface (API) available for Android.

3.1 Google Mesh Router

In this work, the Access Points (APs) used are Google Mesh Routers. Like normal routers, they distribute WiFi services. They can be managed using the Google Home app[10]. These routers connect wirelessly with each other as long they are in range. Each router can cover a range up to 85m^2. The most important feature of these routers is the support for the 802.11mc protocol, allowing us to query the distance that a smartphone is compared to the router.

3.2 WifiRttScan App

WifiRttScan[11] is the official app that allows us to test the WiFi Round Trip Time (RTT) functionality of the Google Mesh Routers RTT capabilities. In figure 3.1, the app shows information about the distance to the router, its mac address, the Received Signal Strength (RSS), and other relevant information useful for debugging. In figure 3.1 is a screenshot of an example of the information reported by the app.

3.3 Android

Android is a free and open-source Operating System(OS) based on Linux, designed primarily for touchscreen devices such as smartphones and tablets. It was founded by Andy Rubin, Rich Miner, Nick Sears, and Chris White, in October 2003, and later bought by Google in July 2005. It is mostly developed by Google and the Open Handset Alliance. The first public version, Android beta, was released in November 2007, and the first commercial version, Android 1.0,

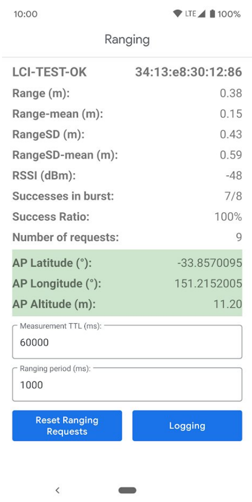


Figure 3.1: Information about a query to a Google Mesh Router [11]

was released in September 2008. Currently, there are 11 major versions, with the latest Android 11.0 having been released in September 2020. Google provides an official integrated development environment(IDE) to develop Android apps, the Android Studio.

According to Statcounter Global Stats[9], in the year 2020 Android is the most popular mobile OS, with just a bit over 71%. More than 60% of its users have Android 9.0 or a more recent version. Starting with version 9.0, it includes a feature to support the measurement of WiFi round trip time(RTT)[1].

3.3.1 WiFi RTT

WiFi RTT is a feature introduced in 802.11mc, the third maintenance/revision group for the Institute of Electrical and Electronics Engineers (IEEE) 802.11 Wireless Local Area Network (WLAN) standards that allow devices for example a smartphone to measure the distance to a WiFi AP.

Android 9.0 comes the WiFi-RTT API, to be able to take advantage of this feature, but there are some requirements[1]:

- The device making ranging requests and the AP must implement 802.11mc
- The device making the request must be running Android 9.0 or a more recent version
- The device making the ranging request must have location services enabled and Wi-Fi scanning turned on
- The app making the ranging request must have the ACCESS_FINE_LOCATION permission and be in the foreground

3.3.2 Sensors

Android provides an API for sensors, which is divided into 3 categories: motion sensors, positions sensors, and environmental sensors. In this section, we talk about some of them, which are later used in Section 4.3 to develop a Pedestrian Dead Reckoning (PDR) method.

Accelerometer

The accelerometer, as the name suggests, measures the acceleration applied to itself on its reference coordinate system. Using this reading, it is possible to infer movement to estimate the distance travelled.

The sensor manager provides a vector containing 3 values corresponding to the acceleration on the x, y and z axis in meters per second squared. The accelerometer is constantly subject to gravitational acceleration, so a resting phone on a flat surface always has a reading of 9.81m/s^2.

Gyroscope

The gyroscope reads angular velocity. Similarly to the magnetometer, it can provide the direction the device is facing but does not suffer from interference from nearby magnetic fields. Instead, it suffers drift due to not providing absolute direction.

The gyroscope API provides the angular velocity on the three-axis in a vector, but it is important to note that the measurements are extremely noisy, and it may not provide accurate information by itself.

Magnetometer

The magnetometer is a device that measures the magnetic field. It may be used to find the magnetic north of Earth, which can then serve as a reference for estimating the direction of motion. A downside is that the readings can be affected by local magnetic fields, such as those caused by nearby metallic objects.

The magnetometer readings also provide a vector with 3 values, the x, y and z axis, measured in micro-Tesla.

Barometer

The barometer provides measurements of the atmospheric pressure, which we can use to compute the altitude with the help of a reference with known altitude and pressure[20].

The pressure sensor on Android provides the atmospheric pressure in millibar.

Chapter 4

System Design

In this section, we discuss the design and implementation of the hybrid localization application for Android. We present the multilateration methods, Pedestrian Dead Reckoning (PDR) and the data fusion method, along with the complementary methods such as altitude detection and proximity methods using QR codes.

4.1 Architecture

We decided to to a modular hybrid localization method, where each module should be able to locate the user based on its inputs. These positions are then passed to a data fusion module that combines the information received from the different modules to obtain a more accurate localization.

Figure 4.1 illustrates the general architecture of the system. It includes several localization modules and a central hybrid module that combines the information received from the others modules.

The primary localization module is the multilateration module based on WiFi Round Trip Time (RTT), and it is this module that will provide the position most of the time.

The secondary module is the PDR method, which will complement the multilateration module or take over when there are not enough Access Point (AP) to provide an accurate position.

There are also two auxiliary modules. The QR code-based proximity method will calibrate the ground truth of the other modules and provide position whenever available. The barometer-based altitude calculation with the atmospheric pressure will mostly be used to lower the computational cost of the multilateration module by reducing one dimension of search space.

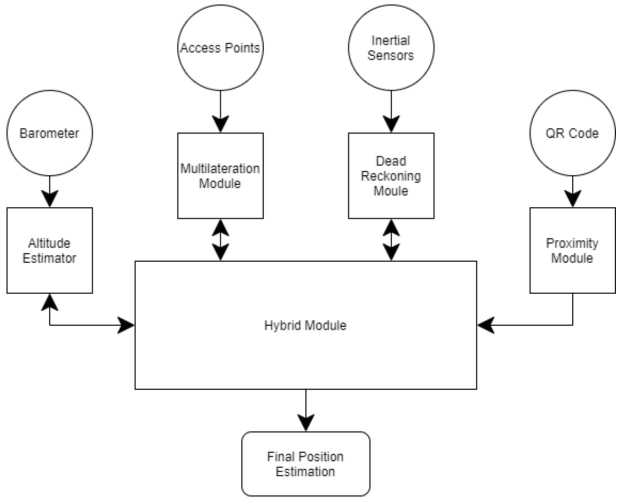


Figure 4.1: Architecture of Localization System

System Development

The app will be implemented for Android devices that support WiFi RTT and motion sensors. It will be developed using Kotlin and compiled via Android Studio. The communication between the software and hardware is through the Application Programming Interface (API) and libraries that Android provides[1–3].

It will be a modular app where we implement each module based on state of art. In particular, the multilateration module is based on the work of Horn[14], the PDR module is a combination of the work of Kang and Han[15] and the composite sensors provided by the Android sensor API[2]. The auxiliary method, altitude estimator is done by measuring the atmospheric pressure and temperature and calculated based on the work of Stull[20]. The proximity-based method is a combination of images captured by the camera using the CameraX library[3] and using the machine learning kit[12] to read the QR codes captured in the images.

4.2 Multilateration

4.2.1 Bayesian Grid

We observed that the measurements of RTT from the AP had a systematic error, but were non-constant and varied with the real distance in addition to the random noise. To combat that we decided to implement a Bayesian Grid-based approach. Because this method allows us to easily correct this error by slightly adjusting the values of the parameters. In the distribution based approach, we could adjust the average of each distance with the measurements in the collection phase. In the case of the heuristic-based, we could skew the values to better fit the data collected.

With this method, the search space is divided into a grid. Due to the random noise present in the measurements of about 2m, it does not make sense to make each cell small, as the accuracy would not allow such precision. And making each cell too big would cause a loss of precision of the final position estimation, so we decided that 10cm on each side of the cell would be a good balance. Depending on the resolution of the estimated location, we can also increase or decrease the size of the cell if the measurements are more or less accurate and if the measurements allow more precise localization. Using this grid, it is possible to compute the probability of the user being in a specific cell-based of measurements from the APs with 2 different ways to calculate the probability.

4.2.1.1 Distribution Function

In the first approach, In the first phase, we collect the measurements given by the AP at known distances. The reason for this collection phase is due to the systematic errors present in the measurements. In the second phase, we use the distribution of the values to compute the probability of the position. For the in-between values, the expected value μ and variance σ^2 are obtained using a weighted average.

The probability of each cell is calculated using the following formula where the distance measured by the AP is denoted as o and the distance of that cell to the position of the corresponding AP is denoted as d:

$$P(o|d) = \frac{1}{d}\left(\mathcal{N}(\mu, \sigma^2)\right) \tag{4.1}$$

This method does not work well in practice due to the obstacles in a real-life scenario where the distribution does not hold strong and the effect of a single measurement going to zero causes the probability of the cell going to zero. As we can see in figure 4.2, when the smartphone is at 10m of the AP, in Line of Sight (LOS) environments the measurement will have a high probability of reporting 11m. However this would not hold when there are obstacles between the smartphone and AP where the measurement of the same distance will be much higher. As the data is collected in LOS, the distribution function will be similar to the first graph and the probability of the user being at 10m of the AP is 0 if the reported distance is 13, the case of a Non-Line of Sight (NLOS) measurement.

4.2.1.2 Heuristic

A second way to compute the probability is using a heuristic, Horn[14] proposes a double exponential with a flat top heuristic:

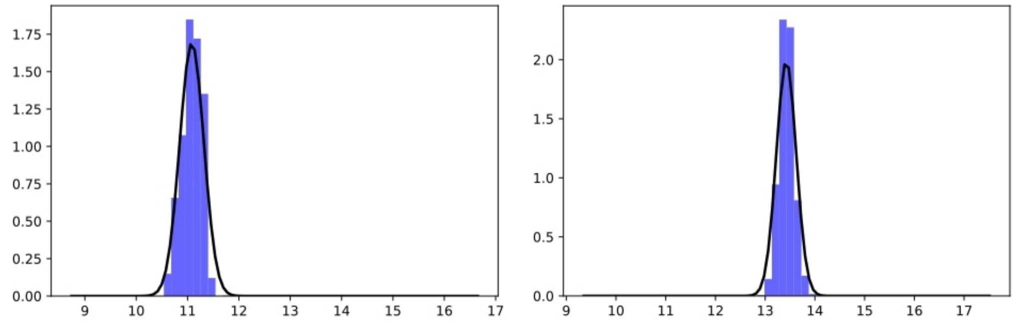


Figure 4.2: Distribution of measurements with LOS(on the left) and NLOS(on the right) at 10 meters from the AP

$$h(r) = \frac{1}{H}\begin{cases} e^{-(r_l - r)/s_l}, & r < r_l \\ 1, & r_l \leq r \leq r_r \\ e^{-r_r - r)/s_r}, & r_r < r \end{cases} \tag{4.2}$$

Then the probability for the distance measured o knowing that the distance to cell d is:

$$P(o|d) = \frac{1}{d}\left(h\left(\frac{o}{d}\right)\right) \tag{4.3}$$

We modified the multilateration module to give two positions with different confidence levels: the first one, with higher confidence, where we only consider distance measured with RTT smaller than d_τ, and the other one, with lower confidence, where we consider every measurement, that is later used in section 4.6.

4.2.2 Limitations

There were a few problems regarding the discovery of new APs that come within the reception range due to the implementation of WiFi scan throttling, introduced in Android Pie, that limits foreground applications to 4 scans in 2 minutes and background applications to once every 30 minutes. Such changes can cripple an application whose main focus is indoor localization based on WiFi RTT. Fortunately, Google also implemented a way to disable WiFi scan throttling using developer options.

In practice, our experience showed that, in the best-case scenarios, it still takes a few seconds to detect APs that should be in range and, in some extreme cases, the AP was not detected until restarting the app.

4.3 Pedestrian Dead Reckoning

There are many approaches to Dead Reckoning, but they generally boil down to two major steps: distance estimation and heading direction estimation. In this work, we use the accelerometer to estimate step length and simulate a pedometer to detect step events[23]. The gyroscope and magnetometer measurements are used to estimate the heading direction.

4.3.1 Heading Estimation

The heading estimation is computed using a combination of motion sensors. The Android Open Source Project provides a rotation vector sensor[4]. This composite sensor will use, if available, the accelerometer, magnetometer and gyroscope, and computes the device's orientation with respect to Earth's coordinate system as a unit quaternion.

Using this information, we get the azimuth of the phone on the global coordinate system. This sensor also provides information about the rotation matrix that is used in the next section to convert the local coordinate system to a global coordinate system.

4.3.2 Distance Estimation

The distance estimation for humans is a combination of estimation of the step length and step detection. Usually for steps are detected using a pedometer but most smartphones do not come equipped with such a sensor. It can, however, be simulated using the accelerometer[23].

SmartPDR[15] uses the acceleration on the vertical axis relative to the ground in other words the z axis on the global coordinate system. Since the accelerometer gives the acceleration in the phone relative coordinate system, in order to convert it to the global coordinate system, we rotate the values using the rotation matrix obtained from the rotation vector sensor discussed above.

A valid step event is registered when the acceleration on the z axis satisfies the following 3 conditions described in equations 2.7, 2.8 and 2.9.

The step length estimation is also done using the measurements from the accelerometer on the global z axis, using the difference of acceleration at step detection a_{step} and the valley between current($t = step$) and previous($t = step - 1$) step, a_{valley}

$$a_{valley} = min(a_t), \qquad (step - 1) < t < step \tag{4.4}$$

$$a = a_{step} - a_{valley} \tag{4.5}$$

To calculate the step length, we use the following formula:

$$l = \begin{cases} \beta \sqrt[4]{a} + \gamma, & a < a_\tau \\ \beta \log_{10} a + \gamma, & a \geq a_\tau \end{cases} \tag{4.6}$$

Kang and Han[15] noticed that these equations performance were almost equal but at bigger accelerations, the logarithmic functions are slightly better than the fourth root equation.

4.4 Altitude Estimator

The altitude of the user can be derived from the atmospheric pressure measured by the environmental sensors, which is provided by the Android API.

By measuring the pressure at a known altitude, it is possible to calculate the altitude using only the measured pressure and temperature[20]:

$$z_2 - z_1 \approx a \times T_v \times \ln \frac{P_1}{P_2} \tag{4.7}$$

where T_v is the average virtual temperature between heights z_1 and z_2, the constant a is $\Re_g/|g| = 29.3mK^{-1}$, and $P_1 and P_2$ are the pressure values at z_1 and z_2, respectively.

The altitude measurements can be used as a landmark to detect stairs or elevators. But the purpose of this module will be to reduce the complexity of the search space of the multilateration method. It doesn't play a bigger role due to the barometer is not as common as other sensors in smartphones yet.

4.5 Proximity

The proximity module is based on QR codes. Depending on the situation, the QR code can contain information directly or indirectly about a location such as the coordinates and direction.

We settled with QR codes instead of other options such as Near Field Communication (NFC) because QR codes are already present in some areas such as museums where the art at display is identified by a QR code. And even in places that there were no QR codes present, it would not require any other hardware necessary to deploy this method.

Google Vision[12] from Machine Learning kit is used for many other applications, but its QR code scanning abilities are convenient. We provide the images captured by the camera it detects the QR code and reads it.

To capture the images, we used CameraX[3] library, because it allows us to include an image analyzer that directly feeds the images to the Google Vision kit, streamlining the process.

Although this method can locate the user with high precision, it is not convenient to be scanning these codes, and only used as a calibration method and not for general use.

4.6 Hybrid Localization

Although each module is capable of localization on its own, we can combine the modules and cover each other's weaknesses and provide an accuracy that is higher than the sum of the modules. In this section, we present the details of the data fusion and where each module will be used to provide more accurate information.

4.6.1 Module Roles

We do not want the information to flow only from the localization modules to the hybrid module, but cooperation between the modules, so that the performance of the different modules can be enhanced by combining all the available information.

The proximity module provides support to all other methods. Since it is the most accurate module, but also the one with the most scarce information, its main role is to re-calibrate other modules, preventing drift and providing an accurate position, whenever available.

The altitude module provides the hybrid module with information on the floor that the user is on. Over long enough periods, there will be drift in measured atmospheric pressure. To correct this problem, the hybrid module gives the information it receives from the proximity modules and re-calibrates this drift.

The PDR module is the complementary module to the multilateration method, providing a position relative to the last iteration. It receives from the hybrid module information about the positions of the last few iterations to correct the drift caused by the minor errors in each step.

The multilateration module provides the position of the user most of the time. It receives information about the floor that the user is on to reduce the dimension of the search space.

4.6.2 Module Selection

The main job of the hybrid module is to choose the best method based on all the received information.

We use proximity-based localization whenever possible because it is the most accurate and less prone to errors. However, it is not possible to always use this module due to its nature. It is highly inconvenient to not only place QR codes everywhere but to scan the code every other step as well.

When there is no other option, we elect the PDR as the main localization module. The inputs to this module inputs are integrated into the smartphone, so it is always possible to use PDR to obtain the relative position of the user.

Finally, when there is information from at least three APs, we use the multilateration module, where we also combine the information of the movements from PDR module using a Kalman filter.

In some cases, we were able to obtain estimated distances to the APs, but the obtained distances were inaccurate. We conducted experiments to understand under which conditions the distances were inaccurate. Based on the results, when the measurements are above a certain value d_τ, we classify the distance as invalid and only used in the cases where we could compute an absolute position in the last 30 seconds.

We also noticed that sometimes the RTT measurements can be collected in a single instance and most of the time the information acquired in these flashes are not reliable enough to locate a user and is only used in cases when there was no other absolute location in the last 60 seconds.

4.6.3 Kalman Filter

We fuse the information from the multilateration and PDR module using a Kalman filter where we use the PDR information is used for the prediction state.

The state transition function is defined as:

$$X_t = A\,X_{t-1} + B\,l_t \tag{4.8}$$

The observation model uses the outputs from the multilateration module:

$$Y = H\,X_t \tag{4.9}$$

where l_t is the distance walked, obtained from the PDR module, and A, B and H are identity matrices.

The Kalman filter is divided into two processes — the prediction process:

$$X_{t_p} = A\,X_{t-1} + B\,l_t \tag{4.10}$$

$$P_{t_p} = A\,P_{t-1}\,A^T + Q \tag{4.11}$$

and the update process:

$$K = \frac{P_{t_p}\, H}{H\, P_{t_p}\, H^T + R} \tag{4.12}$$

$$X_t = X_{t_p} + K\left(Y - H\, X_{t_p}\right) \tag{4.13}$$

$$P_t = (I - K\, H)\, P_{t_p} \tag{4.14}$$

where Q and R are the process covariance matrix noise and measurement matrix noise.

The Kalman gain K is the amount of trust that we have in the new measurements. When the gain is small, new measurements will have a low impact in the final position, as shown in equation 4.13. The new process covariance is then the inverse of the Kalman gain of the current process covariance prediction.

Chapter 5

Experiments and Tests

In this chapter, we explain the setups used in the experiments, the evaluation method and compare the performance of the different implementations of the localization algorithms in different settings. We then evaluate the positioning accuracy using data fusion of the modules and the improvements obtained in comparison to the localization modules without fusion.

5.1 Experiment Setup

We used two different smartphones, a Pixel 4 and a Xiao Mi 9T to capture the information at Departamento de Ciência de Computadores of Faculdade de Ciências Universidade do Porto. More specifically, we used a space that is 30.25m by 22.5m near the cafeteria. We deployed Access Points (APs) in two different setups to simulate different coverage areas.

For each setup, we marked synchronization location where we know the precise time at the exact position, marked with pink dots in figures 5.1 and 5.2. These synchronization points will be later used to measure the accuracy of the estimated positions of each module.

5.1.1 Setup 1

In the first setup, we simulate a case where the APs are not optimally positioned because of physical limitations or aesthetic reasons. The purpose of this setup is to test the localization capabilities when there are areas where APs cannot be reached to perform multilateration and test the ability of the hybrid localization module to perform its job when APs are poorly placed.

In Figure 5.1, we marked the position of the routers using blue dots and the synchronization points with pink dots. The path we walked is shown by the green line.

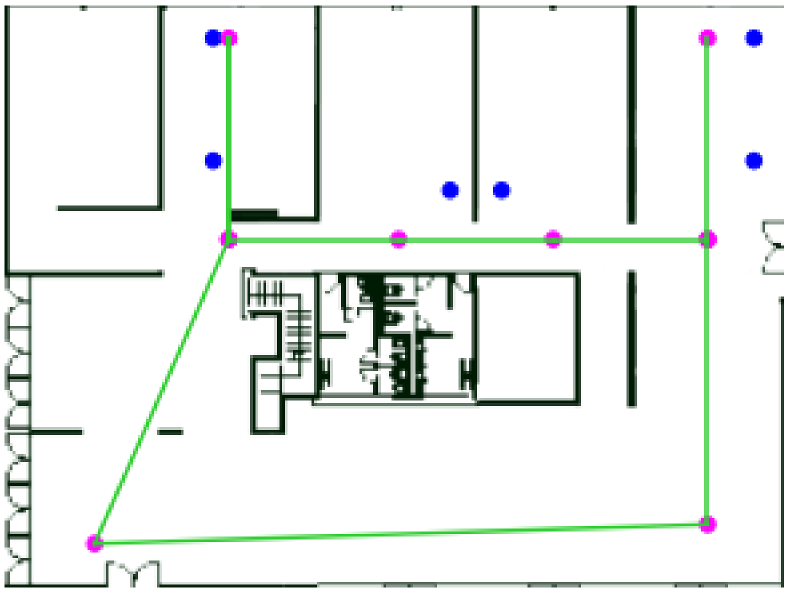


Figure 5.1: Setup 1

5.1.2 Setup 2

In the second setup, the APs are located along the walked path. In this setup, most of the time we have at least 3 APs in line of sight, possibly reducing the error in the distance measured by the Round Trip Time (RTT).

In figure 5.2 we marked the position of the routers using blue dots, the synchronization points with pink dots, and the walked path with a green line. The routers at the top could be placed a little bit better, but due to the corridor being narrow and the fact that we wanted Line of Sight (LOS), the APs ended up almost colinear.

5.2 Evaluation method

We evaluate the accuracy of each module. The accuracy is calculated using the euclidean distance between the position estimated by the module and the real position. The real position is computed using the weighted average from the synchronization points as in equation 5.1. Then the coordinates of the expected real position are calculated using interpolation and assuming constant speed between those points, as shown in equation 5.2.

$$ratio = \frac{t_{cur} - t_{prevSync}}{t_{nextSync} - t_{prevSync}} \tag{5.1}$$

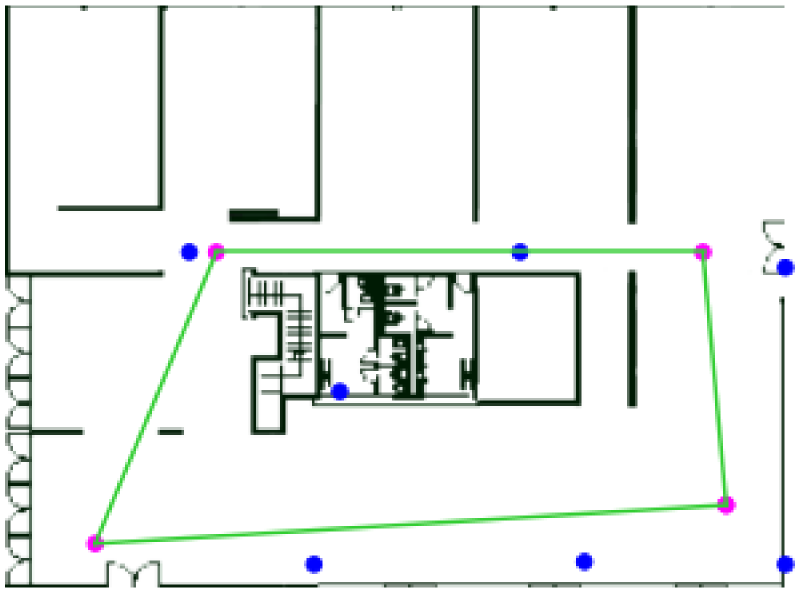


Figure 5.2: Setup 2

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = (1 - ratio) \begin{bmatrix} x_{prevSync} \\ y_{prevSync} \\ z_{prevSync} \end{bmatrix} + ratio \begin{bmatrix} x_{nextSync} \\ y_{nextSync} \\ z_{nextSync} \end{bmatrix} \tag{5.2}$$

5.3 Result analysis

In this section, we discuss the experimental results concerning the performances of the different modules. We will first discuss the Pedestrian Dead Reckoning (PDR) module in isolation, in section 5.3.1. Then in section 5.3.2 discuss the multilateration module in isolation. Finally, in section 5.3.3, we discuss the results using data fusion to combine the information from both modules.

In figures 5.3 and 5.4, we stitched together the position estimated by multilateration module in blue dots, PDR module in red dots and the hybrid module with yellow dots. The green lines represent the walk paths that we took in the experiments.

5.3.1 PDR

In this section, we talk about the performance of the PDR module, which uses only the measurements from the motions sensors of the smartphone.

In figure 5.5, we plotted the distance between the estimated position and real position over time. We can see a general trend of the error accumulating, which clearly shows that this module

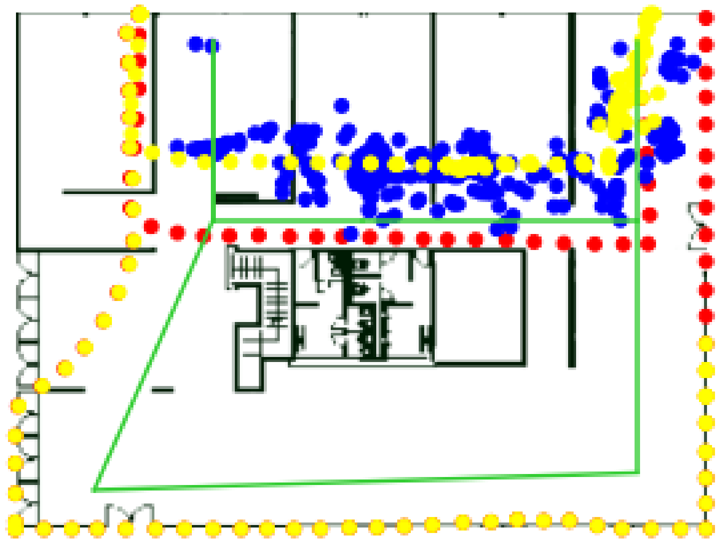


Figure 5.3: Position obtained from every localization module with setup 1

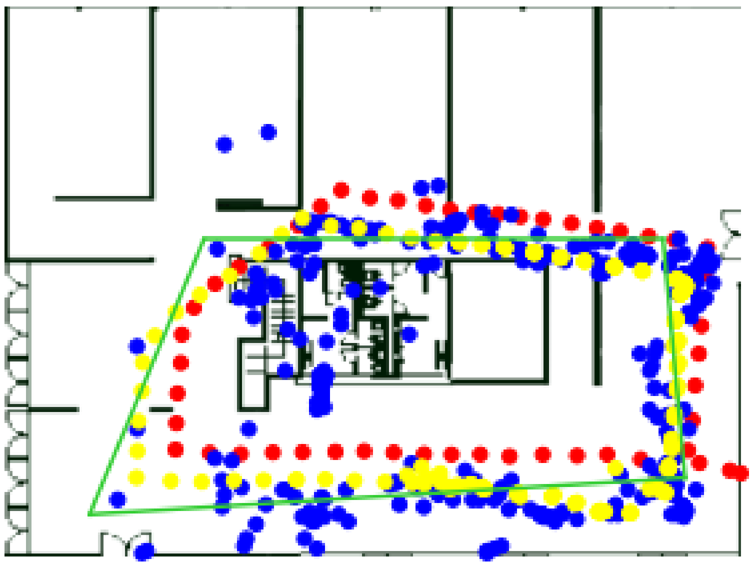


Figure 5.4: Position obtained from every localization module with setup 2

in isolation suffers from drift. As expected the longer the PDR module runs the larger the drift, and, consequently, the error. In both setups, the performance was similar, because this method is not affected by the change in the environments.

In figures 5.6 and 5.7 the results were again similar and the results are almost a linear growth, as expected from the errors caused by drift.

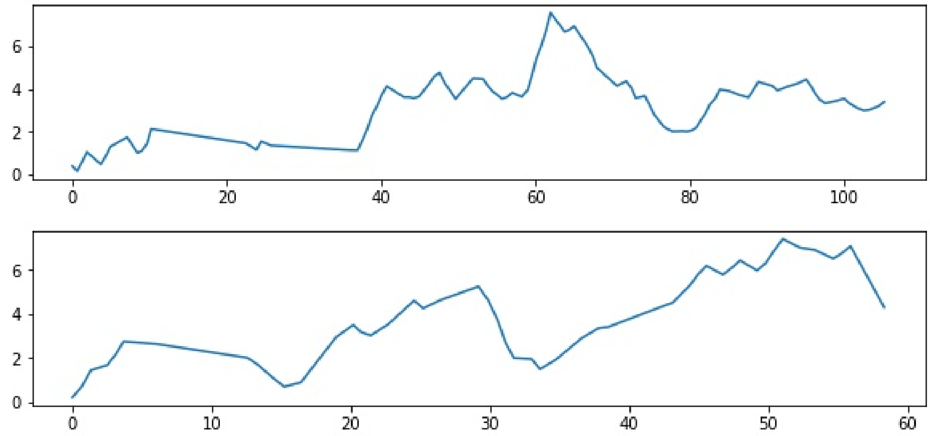


Figure 5.5: Position estimation error overtime using PDR for setup 1(top) and setup 2 (bottom)

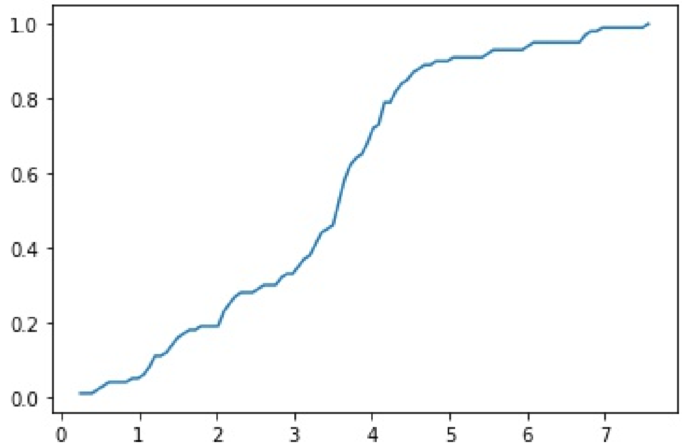


Figure 5.6: CDF of error of position estimation using PDR for setup 1

5.3.2 Multilateration

In this section, we discuss the multilateration module based on Bayesian Grid, where we only use the distance measurements from RTT to each AP.

In figure 5.8, the first until the penultimate is the plots regarding the information of each AP where the orange line is the real distance to the corresponding AP, the blue line is the distance between the estimated position and AP, finally, the green line is the distance reported by each AP the last plot is the error of the estimation which is the distance between the estimated position and real position. Charts on the left side are for setup 1 and charts on the right side are for setup 2. The red shaded area corresponds to the moments when the hybrid module selected the PDR module as the main localization method. While the others areas are when there was enough information to perform multilateration and combine every information available to estimate a position.

For the first setup, we can see in figure 5.9 the CDF of the error of position estimation. The

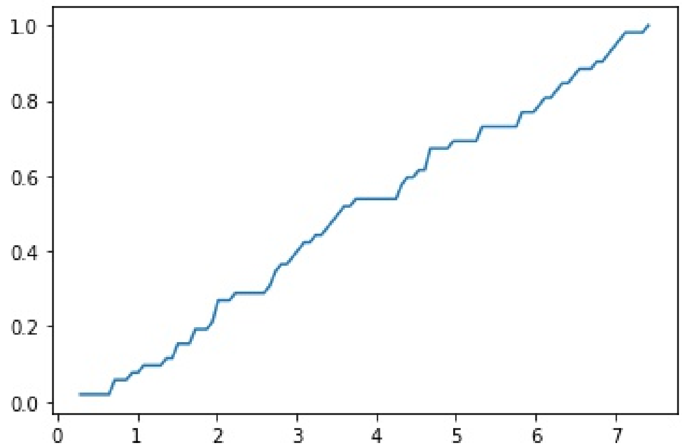


Figure 5.7: CDF of error of position estimation using PDR for setup 2

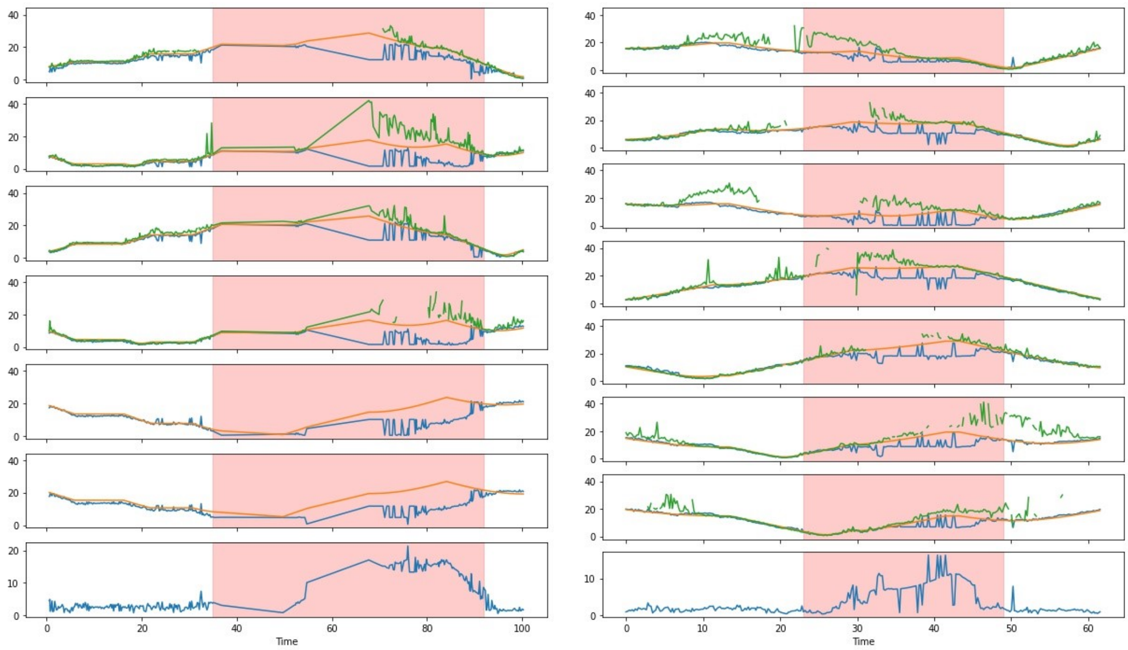


Figure 5.8: Distance measurement, real distance and distance from position estimation to each of AP and position estimation error using WiFi RTT for setup 1(left) and setup 2(right)

results are not great, but we need to keep in mind that in this particular setup the conditions are extremely unfavourable for RTT based localization as about half of the time we did not have communication with at least 3 APs.

For the second setup, whose results are shown in figure 5.10, we achieved slightly better results due to the better placements of the APs, but there was still an area that was not optimally covered by at least 3 APs, corresponding to about 40% of the walking path.

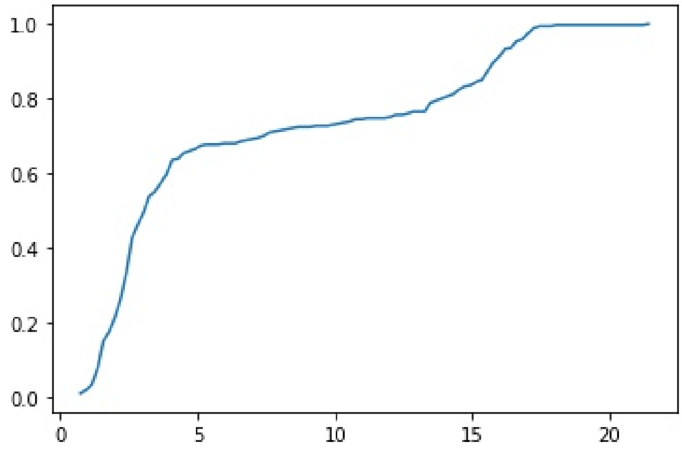


Figure 5.9: CDF of error of position estimation using WiFi RTT for setup 1

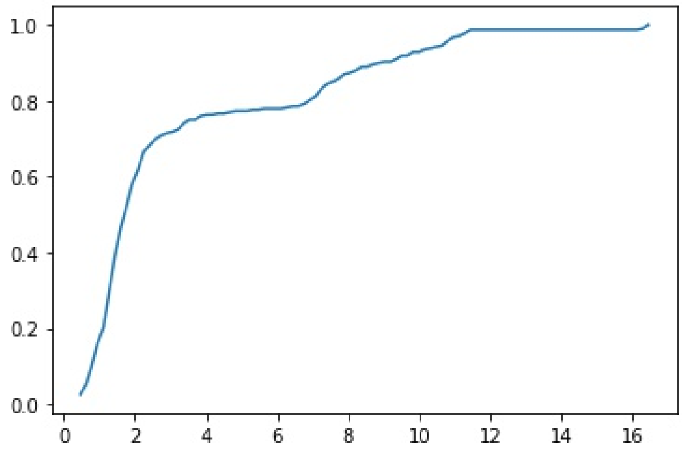


Figure 5.10: CDF of error of position estimation using WiFi RTT for setup 2

5.3.3 Hybrid

Finally, we tested our hybrid module, where we had access to all collected information.

In figure 5.11, we made similar plots to those previously shown in figure 5.8 but for the hybrid module instead of the multilateration module alone. we plotted, in each chart, the distance of the estimated location to each AP, the real distance, and the distance measured using WiFi RTT. The bottom charts show the distance from the estimated location to the ground truth. Notice, however, that the bottom charts, showing the localization error, are on different scales for the y axis.

We can see vast improvements regarding the position estimation in the shaded area where we had an improvement of over 250% for the worst-case scenario in setup 1 and an improvement of over 350% in setup 2 compared to the multilateration module.

In figures 5.12 and 5.13, we show CDFs of the error. In these charts, we can see even

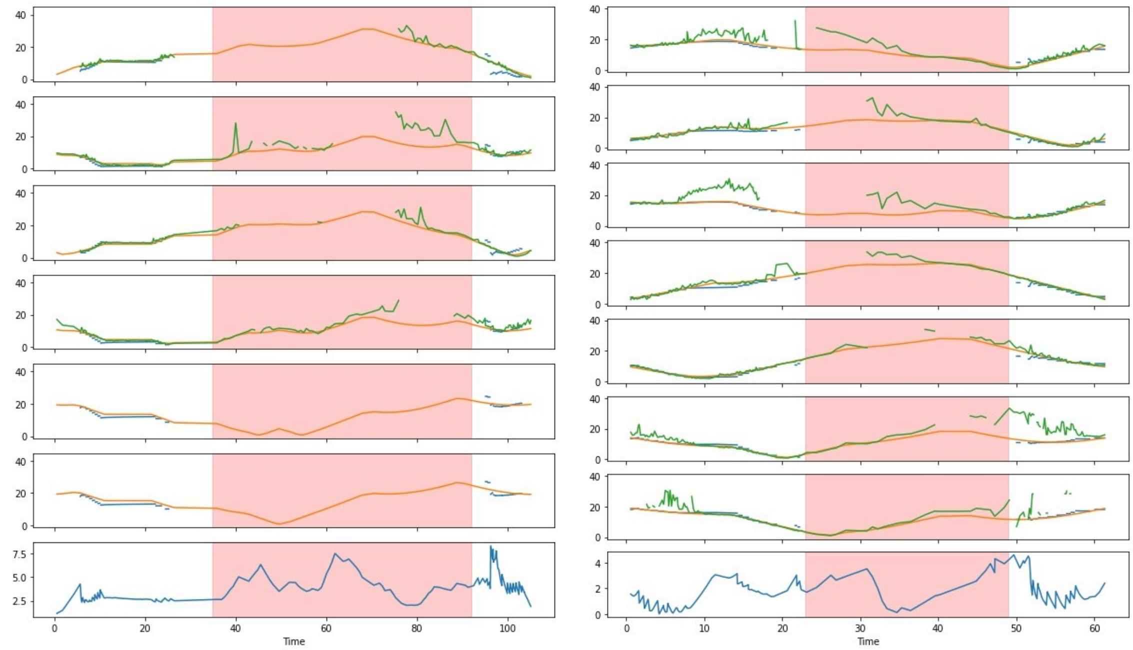


Figure 5.11: Distance measurement, real distance and distance from position estimation to each of AP and position estimation error using hybrid module for setup 1(left) and setup 2(right)

more clearly the improvements brought from the data fusion, especially in setup 2, where the placements of APs were much more suited for indoor localization.

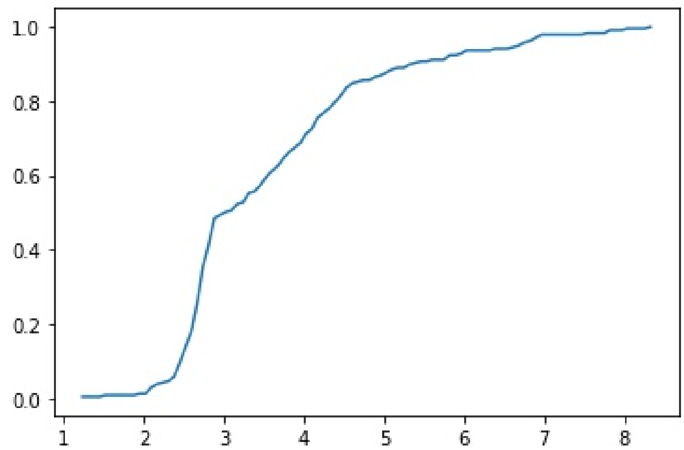


Figure 5.12: CDF of error of position estimation using hybrid module for setup 1

5.4 Comparison

In this section we compare the results in the 2 environments of experiments, where we evaluate the errors presents in the estimations at 70% and at 90% of the time.

In table 5.1, we compare the exact values that each module can achieve at 70% and 90%

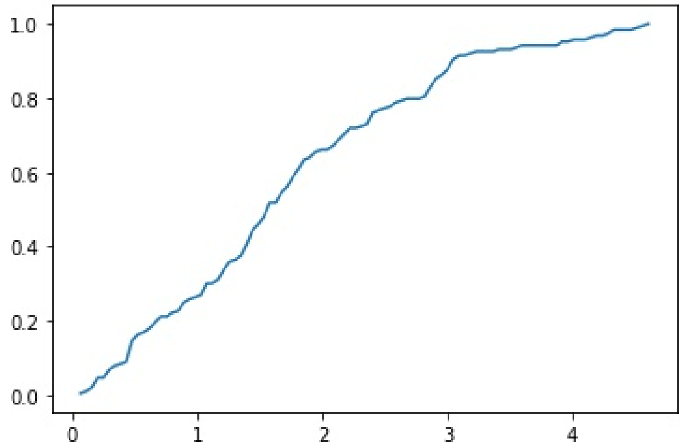


Figure 5.13: CDF of error of position estimation hybrid module for setup 2

of the time in setup 1. While at first glance it looks like the PDR achieved better results, it is not entirely true due to the walking path being near the boundaries, thus hard blocking the estimated position from exceeding the bounds. This setup was also particularly harsh against RTT based localization due to the poor placements of the APs. We should take this as the worst-case scenario performance of RTT based localization.

Percentage	PDR	Multilateration	Hybrid
70	3.95m	7.41m	3.98m
90	4.78m	15.78m	5.33m

Table 5.1: Percentage of each module for setup 1

In table 5.2, we do a similar comparison of the results of the different modules in setup 2. This is a more realistic setup, where the placement of the APs had consideration for indoor localization. This time, the walking path was closer to the centre, so the results of the PDR module were not adulterated by artificially keeping them within the bounds. The PDR module achieved a worse result at 70% than the multilateration module. While there were areas not covered by APs, thus worsening the 90% percentage of multilateration module, the hybrid module took the best of both worlds and achieved better results.

Percentage	PDR	Multilateration	Hybrid
70	5.16m	2.59m	2.16m
90	6.72m	8.80m	3.03m

Table 5.2: Percentage of each module for setup 2

Based on the results, we can say there were improvements regarding the estimates of the position. Although due to the position mainly relying on multilateration as the main method of estimation, the worst-case scenario for the measurements of RTT the position estimate could be better. But it is also very difficult to generalise the situations as the measurements do not provide information about the environment, and can only be custom made for each case.

Chapter 6

Conclusion

Based on the state of the art, where we learned the strong points and shortcomings of each technology we devised a good plan to compensate for those weaknesses.

We successfully implemented a multi-modular indoor localization system with a focus on the multilateration module with WiFi Round Trip Time (RTT), with the Bayesian Grid we were able to correct the systematic error present in the measurements of RTT and a reasonable place Access Point (AP) we could achieve results with an error under 2.6 meters from the real position 70% of the time.

We also developed a Pedestrian Dead Reckoning (PDR) module using the sensors, such as an accelerometer to simulate a pedometer, gyroscope and magnetometer to provide the heading direction. This method provided great support when there wasn't enough information to perform multilateration. It also provided consistent and relatively good results.

We also implemented auxiliary modules like the QR code-based proximity module and altitude measurements derived from the atmospheric pressure readings from the barometer. That could assist the main methods in performing better in terms of computational time or recalibrating the modules.

In a later stage, we evaluated the performance of each module in two different setups. The first one, where the APs are placed poorly and we test the worst-case scenario. The second one, where the placements of APs were more suitable for indoor localization, where most places had Line of Sight (LOS) with at least 3 APs.

For the first setup, the module that performed the best was PDR, not a surprising result because of the poor placements of the APs, although at 70% and 90% the hybrid module was just slightly worse than PDR module.

In the second setup, the multilateration module performed better due to the placements of APs, and the improvements were reflected in the performance of the hybrid module.

In summary, the placements of the APs are just as important as the implementation of the localization module. When the environments are more suited for indoor localization, the hybrid module was able to take the strength of each module and cover the weakness of others.

6.1 Future work

For future work, we plan to add more modules with more technologies, such as Ultra Wide Band (UWB) where the Time Of Flight (ToF) measurements are more accurate than WiFi RTT. We also want to pass the positions through a particle filter, where we take into consideration the floor plan, further increasing the accuracy of the position estimations.

Bibliography

[1] Android Developers. Wi-fi location: ranging with rtt, . Accessed: 2021.

[2] Android Developers. Android sensors overview, . Accessed: 2021.

[3] Android Developers. Camerax, . Accessed: 2021.

[4] Android Developers. Motion and position sensors, . Accessed: 2021.

[5] Zhenghua Chen, Han Zou, Hao Jiang, Qingchang Zhu, Yeng Chai Soh, and Lihua Xie. Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization. *Sensors*, 15(1):715–732, 2015.

[6] Marco Cominelli, Paul Patras, and Francesco Gringoli. Dead on arrival: An empirical study of the bluetooth 5.1 positioning system. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pages 13–20, 2019.

[7] Frank Van Diggellen, Roy Want, and Wei Wang. How to achieve 1-meter accuracy in android, Oct 2018. Accessed: 2021.

[8] Sinan Gezici, Ismail Guvenc, and Zafer Sahinoglu. On the performance of linear least-squares estimation in wireless positioning systems. In *2008 IEEE International Conference on Communications*, pages 4203–4208. IEEE, 2008.

[9] Global Stats. Browser, os, search engine including mobile usage share. Accessed: 2021.

[10] Google. Google home, . Accessed: 2021.

[11] Google. Wifirttscan app, . Accessed: 2021.

[12] Google Developers. Scan barcodes with ml kit on android. Accessed: 2021.

[13] Fuqiang Gu, Xuke Hu, Milad Ramezani, Debaditya Acharya, Kourosh Khoshelham, Shahrokh Valaee, and Jianga Shang. Indoor localization improved by spatial context—a survey. *ACM Computing Surveys (CSUR)*, 52(3):1–35, 2019.

[14] Berthold KP Horn. Observation model for indoor positioning. *Sensors*, 20(14):4027, 2020.

[15] Wonho Kang and Youngnam Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal*, 15(5):2906–2916, 2014.

[16] Zhenyu Liu, Bin Dai, Xiang Wan, and Xueyi Li. Hybrid wireless fingerprint indoor localization method based on a convolutional neural network. *Sensors*, 19(20):4597, 2019.

[17] Markets and Markets. Indoor location market. Accessed: 2021.

[18] Stefania Monica and Federico Bergenti. Hybrid indoor localization using wifi and uwb technologies. *Electronics*, 8(3):334, 2019.

[19] Alwin Poulose, Jihun Kim, and Dong Seog Han. A sensor fusion framework for indoor localization using smartphone sensors and wi-fi rssi measurements. *Applied Sciences*, 9(20): 4379, 2019.

[20] Roland Stull. *Practical Meteorology.* Amsterdam University Press, Amsterdam, Netherlands, 2018.

[21] Paula Tarrío, Ana M Bernardos, and José R Casar. Weighted least squares techniques for improved received signal strength based localization. *Sensors*, 11(9):8569–8592, 2011.

[22] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016.

[23] Harvey Weinberg. Using the adxl202 in pedometer and personal navigation applications. 2002.

[24] Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M Ni. A survey on wireless indoor localization from the device perspective. *ACM Computing Surveys (CSUR)*, 49(2):1–31, 2016.

[25] Zhuoling Xiao, Hongkai Wen, Andrew Markham, and Niki Trigoni. Indoor tracking using undirected graphical models. *IEEE Transactions on Mobile Computing*, 14(11):2286–2301, 2015.

[26] Zheng Yang, Zimu Zhou, and Yunhao Liu. From rssi to csi: Indoor localization via channel response. *ACM Computing Surveys (CSUR)*, 46(2):1–32, 2013.

[27] Yue Yu, Ruizhi Chen, Liang Chen, Guangyi Guo, Feng Ye, and Zuoya Liu. A robust dead reckoning algorithm based on wi-fi ftm and multiple sensors. *Remote Sensing*, 11(5):504, 2019.

[28] Faheem Zafari, Athanasios Gkelias, and Kin K Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599, 2019.