Universidade Estadual de Campinas
Instituto de Computação

Tiago Moreira Trocoli da Cunha

A comparative study of indoor localization methods based on RSS

Um estudo comparativo de métodos de localização em ambientes fechados.

CAMPINAS
1500

Tiago Moreira Trocoli da Cunha

A comparative study of indoor localization methods based on RSS

Um estudo comparativo de métodos de localização em ambientes fechados.

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Dr. Lucas Francisco Wanner

Este exemplar corresponde à versão da Dissertação entregue à banca antes da defesa.

CAMPINAS
1500

Na versão final esta página será substituída pela ficha catalográfica.

De acordo com o padrão da CCPG: “Quando se tratar de Teses e Dissertações financiadas por agências de fomento, os beneficiados deverão fazer referência ao apoio recebido e inserir esta informação na ficha catalográfica, além do nome da agência, o número do processo pelo qual recebeu o auxílio.”
e
“caso a tese de doutorado seja feita em Cotutela, será necessário informar na ficha catalográfica o fato, a Universidade convenente, o país e o nome do orientador.”

Na versão final, esta página será substituída por outra informando a composição da banca e que a ata de defesa está arquivada pela Unicamp.

Acknowledgements

My special thanks to the Universidade Virtual do Estado de São Paulo (Univesp) who financially support me through a scholarship during my master research. My thanks to my colleges of the Laboratory of Computer Systems (LSC) from the University of Campinas (Unicamp) for the support. Thanks to my father, mother, supervisor and all others who helped me made this realization come true.

Resumo

A localização em ambientes fechados é uma aplicação de IoT importante e abrangente. Em dispositivos de baixo custo, os métodos que dependem da Força do Sinal Recebido (RSS, sigla em inglês) em pacotes de rádio para estimar a distância e a posição dos dispositivos são amplamente usados, uma vez que não requerem hardware adicional e fornecem precisão razoável. Embora muitos métodos de localização em ambientes fechados baseada em RSS tenham sido desenvolvidos, os estudos comparativos para esses métodos geralmente se concentram principalmente na precisão, ignorando métricas importantes, como tempo de resposta e consumo de energia. Em dispositivos com restrição de energia, em particular, a precisão não pode ocorrer em detrimento da redução da vida útil da bateria e, portanto, maximizar a precisão isoladamente pode não ser desejável. Neste trabalho, apresentamos um estudo comparativo de técnicas de localização baseadas em RSS implementadas em hardware comum. Comparamos nove métodos de localização em um ambiente de escritório do mundo real com forte interferência de sinal e em cômodos de uma casa com fraca interferência de sinal. Além da precisão, apresentamos o consumo de energia e o tempo de resposta para cada método. Nossos resultados mostraram que o modelo lognormal de estimativa de distância comumente usado teve um desempenho ruim em termos de precisão nesses ambientes, mas eles tinham o menor custo de energia; que as variações do método Fingerprinting tiveram valores de consumo de energia, o tempo de resposta e a precisão mais balanciados e melhores; e implementações mais simples obtiveram resultados melhores do que as complexas, no geral. Os resultados experimentais mostraram que a precisão foi amplamente impactada pela posição dos nós âncora, corroborando a necessidade de posicionamento cuidadoso desses dispositivos. Finalmente, a pesquisa mostrou que o descarte seletivo de informações de alguns nós âncoras podem levar a uma melhor precisão quando comparado ao uso de todos ais nós disponíveis. No entanto, a seleção do subconjunto de âncora ótimo para cada estimativa é uma tarefa difícil.

Abstract

Indoor localization is an important and wide-ranging IoT application. In low-cost devices, methods relying on the Received Signal Strength (RSS) in radio packets to estimate distance and position of nodes are widely used since they require no additional hardware and provide reasonable accuracy. While many methods for RSS-based indoor localization have been developed, comparative studies for such methods have typically focused primarily on the accuracy, ignoring important metrics such as performance and energy consumption. In energy-restricted devices, in particular, localization accuracy cannot come at the expense of reducing battery lifetime, and so maximizing accuracy in isolation may not be desirable. In this work, we present a comparative study of RSS-based localization techniques implemented in commodity hardware. We compare nine localization methods in a real-world office environment with strong radio signal interference and in rooms of a house with weak interference. In addition to accuracy, we present energy consumption and response time for each method. Our results showed that the commonly used lognormal model of distance estimation performed poorly in terms of accuracy in such an environment, but it had the lowest energy cost; that fingerprinting variations were the best compromise between energy consumption, response time, and accuracy; and simpler implementations attained better results than complex ones, as a whole. Experimental results showed that accuracy was widely impacted by locations of anchor nodes, corroborating the need for careful placement of these devices. Finally, the research showed that selectively discarding information from some sensor nodes can lead to better accuracy when compared to using all available anchors. However, the selection of the optimal anchor subset for each estimation is a difficult task.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Indoor localization consists of estimating a position of people and devices in indoor environments, such as tunnels, airports, shopping malls, warehouses and smart factories. Indoor localization for mobile devices and people is a difficult problem that GPS and other global localization technologies are highly inaccuracy or fail entirely [40]. A workaround consists of deploying sensor nodes that emit signals to mobile devices. Mobile devices can use received signal strengths (**RSS**) from such nodes along with numerical methods to estimate their position, as shown in figure 1.1. That is the essence of localization based on RSS: use RSS as a means to estimate positions. In this thesis, we prefer to call mobile and anchor nodes instead of sensor nodes and mobile devices. Because **anchor nodes** generalize any device capable of emitting signals. Therefore, it could be NodeMCU, smartphones, Raspberry Pi and so on. Signals could be of any kind, but in this project are Wi-Fi signals. **Mobile nodes** are anything which the indoor localization system needs to localize. They must have devices that can measure RSS and communicate with the system, such as a smartphone carried by a person. Mobile nodes do not necessarily move all the time, they could be items or goods in factories which remains unmoved occasionally.

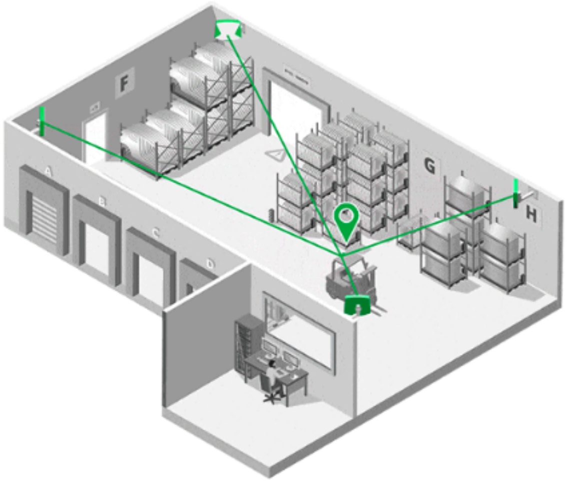


Figure 1.1: Example of a real indoor localization system of a company from the Czech Republic [32]. Anchor nodes (in green) are employed to localize forklifts (mobile nodes).

However, indoor localization of mobile nodes is a difficult task since objects, such as furniture, can block, reflect, attenuate or strengthen signals emitted from anchor nodes which could decrease localization estimation [43, 12]. Even people can disturb signals [37]. Moreover, signals from anchor nodes can overlap which provokes more interference. Despite that, indoor localization methods emerged such as multilateration and fingerprinting. Nowadays, non-classic ones apply machine learning and adaptive filters with brand new measurement hardware [6, 30]. Despite that, they are numerical methods which are prone to numerical errors aggravating even more the estimation of positions. Moreover, although this research concerns only received signal strength (RSS) of Wi-Fi, there are other approaches such as time-of-flight (ToF) and time differential of arrival (TDoA) [13] that are briefly discussed in Background chapter.

Indoor localization applications are found in commercial, military, retail, and inventory tracking industries [19, 22, 45]. For example, it can improve automation in warehouses by helping robots and forklifts navigation [27]. Smartphones can be employed as a guide in museums as long as they know their location and also as guidance to visually impaired people [24]. It could guide people inside shopping centers and they can even receive advertisements from nearby stores and restaurants, technology known as location-based ads [33]. It could be applied to track assets in warehouses. In underground mining, positioning and monitoring can be used to identify endangered personal in real-time [25]. A similar application could be employed in power plants to ensure workers' safety. In sports, mobile nodes could be the players that wear a smartwatch and the ball, thus, it is possible to evaluate players' performances regarding running distance, ball possession, running speed and so on [34]. These data can benefit coaches to devise a strategy. Also, localization systems can be integrated into a factory to speed up the production flow [26]. Figure 1.2 depicts a real localization system.

The thesis is organized into five chapters. This chapter is about motivations, objective and contributions. The following shows some related works about localization methods. The third one describes localization methods and related techniques. The fourth chapter explains two experiments, their findings, and results. The last one is the conclusion of this master's research.

1.1 Motivations and Objective

As we can check in chapter 2, research tends to focus on accuracy while putting aside other important metrics, such as response time and energy consumption. However, depending on the application, those metrics could be important as well.

Response time is the time elapse of a method to provide an estimated position. Response time is imperative in real-time embedded systems and important in localization systems in general. A localization method that localizes mobile nodes with a considerable delay can cause accidents if their movement decision is based solely on such a method. Shops with location-based advertisements could lose potential clients. A system that guides people inside shopping centers and museums could not be worth having it. Delays also could cause problems in positioning systems integrated with smart factories.

Energy consumption is the energy consumed of a method during its response time.

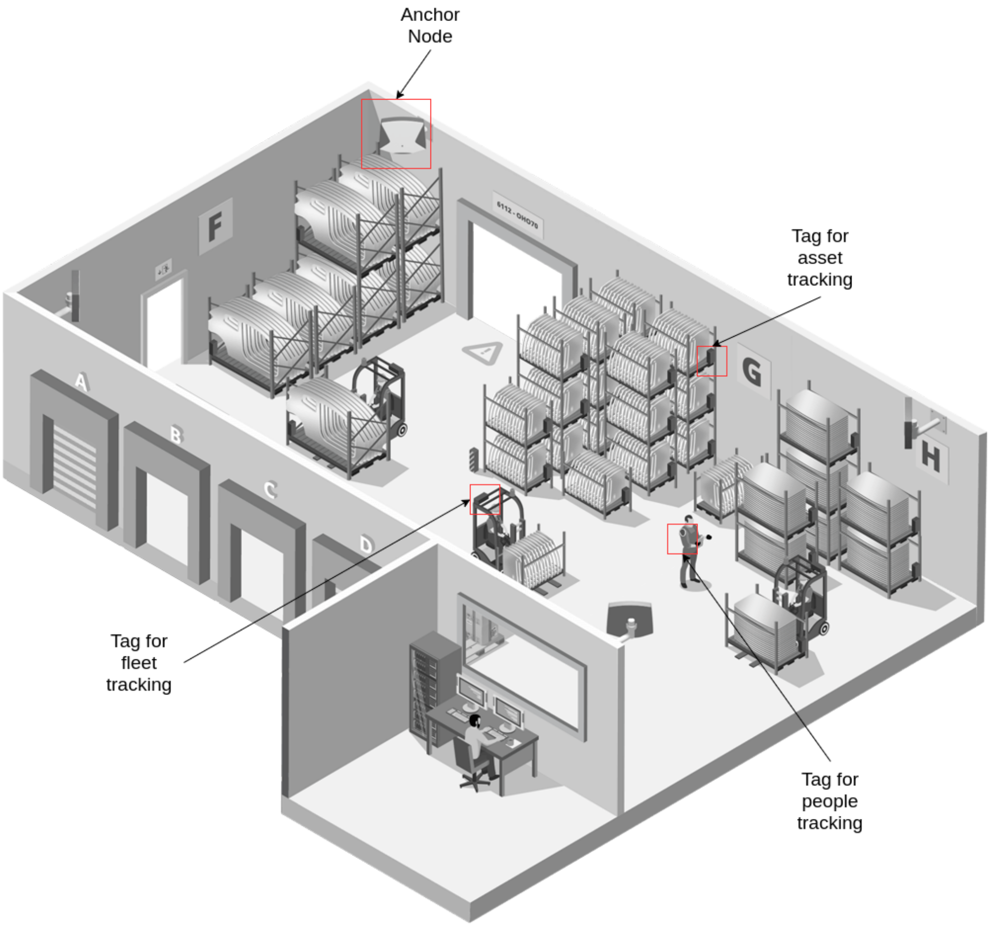


Figure 1.2: Example of a real indoor localization system of a company from the Czech Republic [35]. Assets, personnel, and fleet have tags making it possible to localize them using tablets that run a software integrated with the system. Anchor nodes are routers and mobile ones are personnel, assets, and fleet with tags.

It is important for indoor localization systems especially because they employ embedded systems and power-constrained devices. Thus, in some cases, it is imperative to take into consideration energy consumption for indoor localization systems [44, 10]. For instance, some systems need to consume less energy as much as possible when they employ smartphones, for example, to guide people in shopping centers or museums. They need to be battery friendly to increase the battery's life span of sensor nodes spread in smart factories.

Accuracy, which is also known as average localization error, is the error between the estimated position calculated by a localization method and the true position. It is the most important metric of all, so it is no surprise that research tends to focus on that metric. The primary goal of indoor localization methods is to be accurate. All applications mentioned in the previous section need to be accurate. An inaccurate system can cause many accidents, side effects and turn the system to be underused.

We observed that research tends to be interested in accuracy, which motivated us to carry out research considering other metrics to have a broader analysis. Thus, the objective of this research is a comparative study of indoor localization methods on the accuracy, energy consumption and response time. To do that, we conducted an experiment in a laboratory at the State University of Campinas. The second one was carried out in two rooms of a house.

1.2 Contributions

Since comparative studies focus on accuracy, this motivated us to carry out a novel study that measured other metrics. The contributions of this research were:

1. Implementation of 9 indoor methods which are variations of multilateration and fingerprinting. Some of them applied standard numerical methods, while other machine learning.

2. Results and findings of the comparison of these methods concerning the accuracy, energy consumption and response time in a real-world office environment with a strong radio signal interference and in rooms of a house with a weak interference.

3. The use of a different model from lognormal path-loss one to improve multilateration accuracy.

We carried out the experiment at Laboratory of Computer System of the University of Campinas, which had a strong signal interference and in another indoor environment with weak signal interference. We implemented nine methods, which are variations of multilateration and fingerprinting. To calibrate them, we measured the RSS of various locations in the environment, and we tested them using other positions. The objective of the methods was to estimate the coordinates of those positions. We applied a polynomial model and compared it with the standard lognormal path-loss one, which is the commonly used model for multilateration, and results showed that methods using the former model had better accuracy than the latter. We also measured energy consumption and response time, and we compared the methods based on their values. Fingerprinting variations stood out as the best technique, in general. The research reassured the need to carefully select positions of anchor nodes since results showed their impact on the accuracy of methods. We tested the accuracy of many subsets of anchor nodes for each method instead of using all of them available. So, the accuracy of the methods presented in this research belongs to the local optimal subset, which was the subset that provided the best accuracy among all that were tested. The research showed that selectively discarding information from some anchor nodes can lead to better accuracy when compared to using all available ones. Thus, it could be advisable to test subsets of anchor nodes to find one that gives the lowest accuracy.

Chapter 2

Related Works

This section describes research about comparative studies of indoor localization methods. Among them, the most significant one relates to the Microsoft indoor localization competitions which more than 20 teams participated using various kinds of technologies and methods. We can distinguish two types papers. One~~s~~ that compare methods that apply machine learning and adaptive filter, such as Kalman and particle filters for robot localization [11, 2, 16]. Others compared traditional techniques, such as multilateration and fingerprinting. However, these techniques can overlap each other, that the case of multilateration with Singular Value Decomposition (SVD) [41]. Nevertheless, all papers presented here discussed technologies that focus on accuracy.

Dimitrios Lymberopoulos and Jie Liu [20] present results, experiences, and lessons learned of Microsoft competitions from 2014 to 2017. These competitions use four places to conduct experiments, one for each year as shown in figure 2.1. Their sizes range to $300m^2$ to $2000m^2$. Each competition had approximately 20 teams divided into two categories: infrastructure-free and infrastructure-based. The first one did not require any custom hardware besides Wi-Fi infrastructure, while the other one did require the deployment of custom hardware. They concluded that lidar-based technology consistently achieved the lowest localization error. However, the cost, as well as the size and power requirement of lidar sensors, prevent them to be the mainstream indoor localization technology. In infrastructure-free technology, the best team achieved accuracy between 1.1m to 1.9m. The systems that perform better are based on Fingerprinting, this technique does not overcome the temporal variation in wireless signals that may occur in realistic environments, such as airports or shopping centers. Infrastructure-based technology applies six core technologies: Time-of-Flight, UWB, visible light communication, magnetic resonators and 2.4Ghz phase offset. They noted that UWB seems to rapidly rise as accurate and most popular technology. Teams that applied UWB achieved accuracy ranging from impressive 0.17m in 2017 to 0.39 in 2014.

Moreover, Dimitrios Lymberopoulos, Jie Liu et. al [21] also published a paper specific for Microsoft competition in 2014 that showed a detailed analysis of practical experiments of indoor localization techniques. In total, 22 state-of-the-art solutions were presented, most of them applied the fingerprinting method with Wi-Fi technology. The experiments took place in a $300m^2$ area having 20 evaluation points across two rooms and a hallway. To introduce a realistic scenario, rooms were equipped with furniture, such as tables and

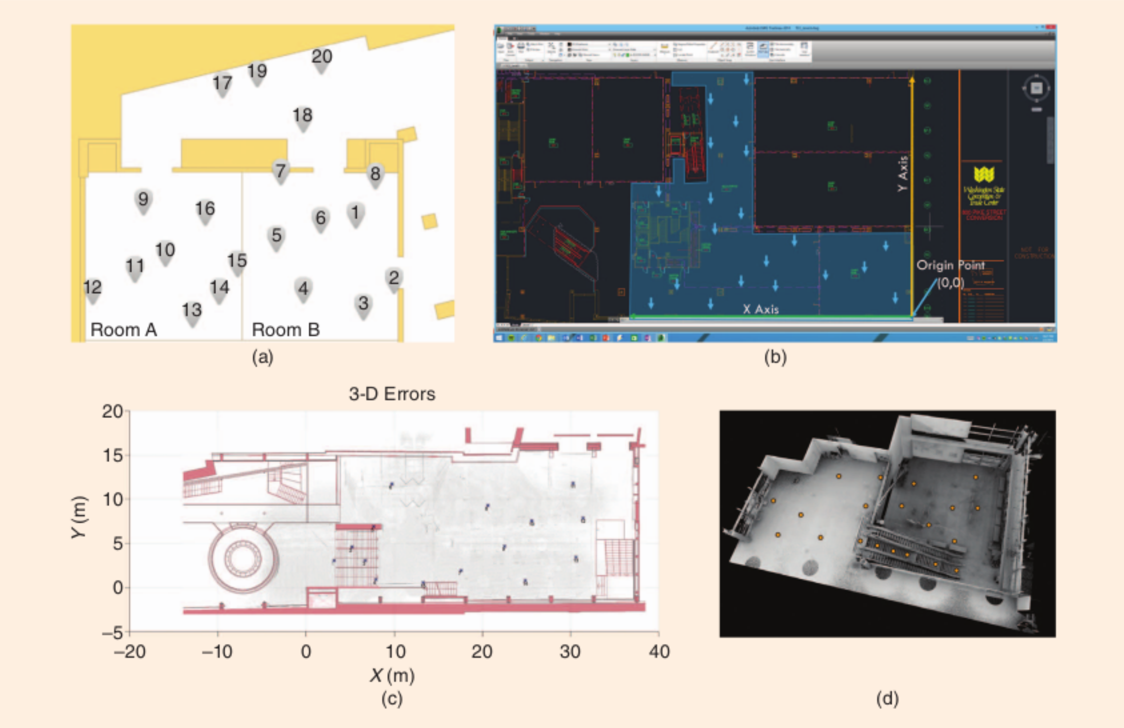


Figure 2.1: The different evaluation areas used in the last four Microsoft Indoor Localization Competitions [20].

chairs. There was a considerable level of wireless interference on the first day which was reduced on the second day. The ground truth measurements of evaluation points were taken using laser range finders. They show that WiFi-based approaches can achieve close to $1m$ accuracy and in general localization accuracy degrades by as much as $3m$ caused by setup and environmental changes, such as RF interference and furniture movement. The best solution had $0.72m$ average error with less than $1m$ standard deviation as shown in figure 2.2. While the worst one had $10.22m$ average error with almost $4m$ standard deviation.

Paulo Tarrio, Ana Bernardos, and Jose Casar made a comparative study of four types of multilateration [39] using Wi-Fi, Bluetooth, and WSN technologies. Among these four, two were new methods developed by the researchers which applied a weighted covariance

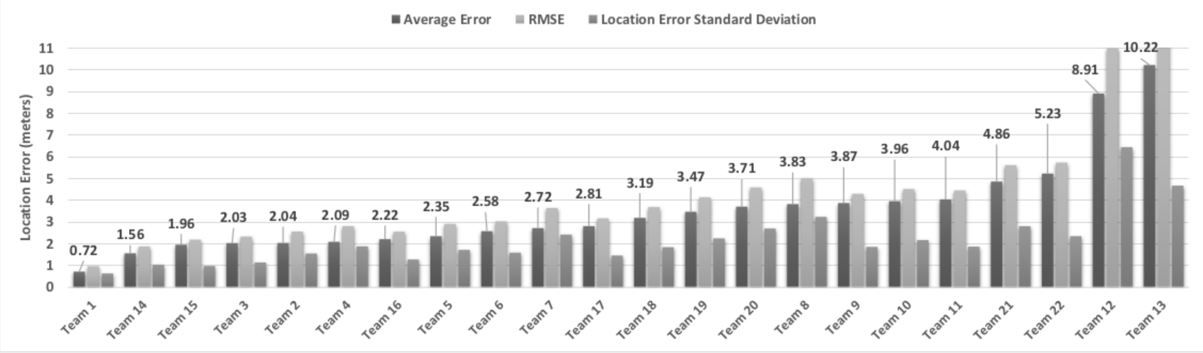
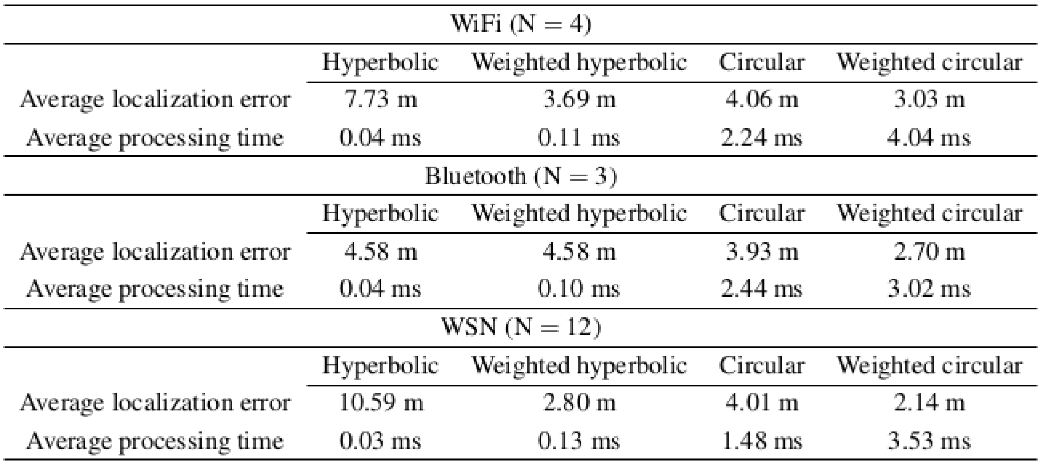


Figure 2.2: Average location error, root mean square error (RMSE), and the standard deviation of the location error for all 22 teams. [21].



WiFi (N = 4)				
	Hyperbolic	Weighted hyperbolic	Circular	Weighted circular
Average localization error	7.73 m	3.69 m	4.06 m	3.03 m
Average processing time	0.04 ms	0.11 ms	2.24 ms	4.04 ms
Bluetooth (N = 3)				
	Hyperbolic	Weighted hyperbolic	Circular	Weighted circular
Average localization error	4.58 m	4.58 m	3.93 m	2.70 m
Average processing time	0.04 ms	0.10 ms	2.44 ms	3.02 ms
WSN (N = 12)				
	Hyperbolic	Weighted hyperbolic	Circular	Weighted circular
Average localization error	10.59 m	2.80 m	4.01 m	2.14 m
Average processing time	0.03 ms	0.13 ms	1.48 ms	3.53 ms

Figure 2.3: Average localization error and average processing time for the different positioning algorithms in the three experiments ~~which~~ where N is the number of anchor nodes [39].

matrix. They deployed a smartphone to calibrate the lognormal model that multilateration used, collecting in total 689 RSS for each anchor node. Two indoor environments were used in the experiments, one with dimensions 15 × 15m and other with 12 × 15m. The most accurate method achieved an average localization error of 2.8m with a time response of 0.13ms. While the worst had an accuracy of 10.59m with a processing time of 0.03s. Methods' accuracy is shown in figure 2.3.

N. Li, J. Chen and Y. Yuan, and C. Song [18] argued that many indoor tracking systems suffer from low accuracy and high time delay caused by the complexity of indoor environments and by the time consumption of positioning algorithms. Thus, it proposed a new tracking indoor algorithm based on particle filter and an improved k-nearest neighbor, called PF-IKNN and compared it to particle filter (PF), k-nearest-neighbor (KNN) and PF-KNN. The authors conducted a realistic experiment on the third floor of the School of Automation, Beijing Institute of Technology that had an area of $20m \times 15m$ including a hallway and five office rooms. The tracking software was installed on a mobile phone, which auto-scans the nearby Wi-Fi signals and showed the estimation tracking positions. The mobile phone had a scan rate of 2 times per minute, and people's velocity was about $1m/s$ to $2m/s$. A total of 139 reference points were selected in the offline phase with a distance of $2m$ between neighboring points. They first wanted to know what was the best number of particles for the algorithm and they showed that 700 had the best results of an experiment between 100 and 1000 particles. In the second experiment, they compared their new algorithm with PF, KNN, and PF-KNN by altering the number of access points (AP) on the floor. PF-IKNN had the second-best mean location error with three access points but was the best when the number of them was equal and greater than four. With the results of other experiments, they concluded that PF-IKNN only provides a higher tracking performance than other methods, but also decreases the positioning time.

S. Shen, C. Xia et al. [36] compared three different tracking algorithms, namely, Standard Kalman Filter (SKF), Extended Kalman Filter (EKF), and Modified Kalman Filter (MKF) in terms of accuracy and latency for a ranged-based indoor tracking system.

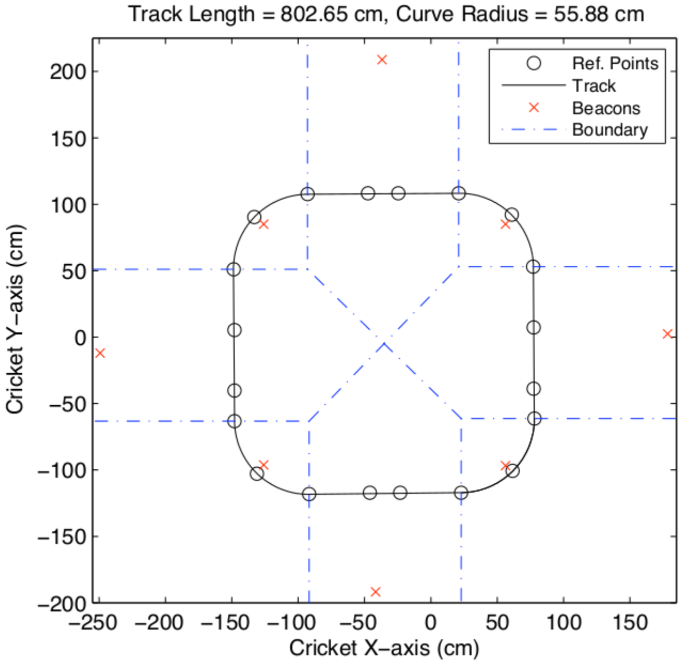


Figure 2.4: Layout Experiment Environment [36].

Figure (corrigir em todas as ocorrências de figure)

As shown in figure 2.4, in the experiment, an electronic model train moves around a track with a low and high speed set up to 15.43cm/s and 51.98cm/s, respectively. There are 8 beacons on the room that can estimate the localization of the train using linear Least Square (LSQ). The video recordings are also used to retrieve the actual positions of the train concerning time by applying the Kalman filter. The experimental results show that the tracking techniques improve the estimation accuracy when the target is moving rapidly. The SKF, EKF, and MKF have similar performance in terms of accuracy at low speeds.

J. Chóliz and M. Eguizabal [8] evaluate the performance of different locations and tracking algorithms on a realistic indoor scenario and with a specific Ultra-Wideband UWB indoor ranging model. The algorithms are Trilateration, Weighted Least Square with Multidimensional Scaling (WLS-MDS), Least Square with Distance Contraction (LS-DC), Extended Kalman Filter (EKF) and Particle Filter (PF). The simulation scenario is a representation of a 50$m \times$ 50m indoor area. Two scenarios with 10m and 12.5m between anchors are considered, which result in 36 and 25 anchors of a UWB network, respectively as shown in figure 2.5. Ten mobile targets move along the scenarios with a random speed and predefined probabilities of going forward and backward. The evaluation is in terms of average absolute positioning error. The position update rate is one update per second. Experiment results show that EKF and trilateration had the worst performance in both scenarios. The PF was the best followed by WLS-MDS in both scenarios.

2.1 Related works: Methods and Metrics

Table 2.1 shows a comparison between this research to other ones. We can see that other articles focus on accuracy, except the [39]. And the authors of articles about the Microsoft

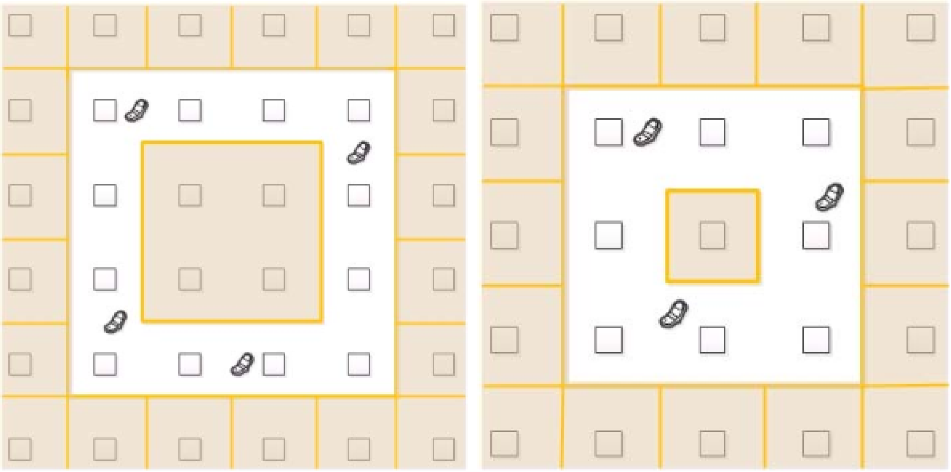


Figure 2.5: Simulation scenarios with 10m and 12.5m between anchors. Anchor nodes are depicted as small squares and mobile targets as cellphone icons. [18].

Article	**Number of methods**	**Technology**	**Type of methods**	**Acc.**	**Energy Cons.**	**R. Time**
This research	9	Wifi RSS	Multilateration / Fingerprinting	✓	✓	✓
Paulo Tarrio, Ana Bernardos et al. [39]	4	Wifi and bluetooth RSS	Multilateration	✓	✗	✓
S. Shen, C. Xia et al. [36]	3	RSS	Kalman Filter	✓	✗	✗
Microsoft competitions [20, 21]	∼20	Many	Many	✓	✗	✗
N. Li, J. Chen and Y. Yuan et al. [18]	4	Wifi RSS	Fingerprinting, Particle Filter	✓	✗	✗

Table 2.1: Table of articles that shows what metrics each one measured. Acc., Energy Cons. and R. Time mean accuracy, energy consumption and response time, respectivelly.

competitions had the opportunity to compare a large number of methods that several teams developed but, they neglected energy consumption and response time metrics. Besides, many technologies were used in Microsoft competitions, such as Wifi RSS, Wifi + IMU, modulated LED, Zigbee, Lidar, Wifi + magnetic + IMU, as well as many types of methods such as fingerprinting, multilateration, adaptive filters, and neural networks.

~~2.2 Related works: Main Results~~

Table 2.2 shows the results of some methods developed in this research and other ones. Some of them did not require additional hardware besides commercial off-the-shelf ones. This hardware could be sensors (such as gyro, accelerometer) and ones that emit radio signals such as internet access points. But other solutions did need custom hardware such as UWB or LIDAR.

Regarding the results, we can clearly see the large difference of Fingerprinting + KNN between the first and second experiments. That happened because the first had a high signal interface, while the second did not. The company RealEarth developed an extremely accurate solution, however, according to [20], the solution was for academic

Article	**Method**	**Additional hardware**	**Acc. (m)**	**Energy Cons. (dj)**	**Response Time (ms)**
This Research (2nd exp)	Fing. + KNN	No	0.97	0.09	0.47
This Research (1st exp)	Fing. + KNN	No	2.19	0.09	0.47
Paulo Tarrio, Ana Bernardos et al. [39]	Weighted Circular	No	2.14	✗	3.53
RealEarth [20, 21]	not provided	LIDAR	0.033	✗	✗
Cork Institute of Technology [20, 21]	Fing.	No	1.56	✗	✗
Quantitec Intranav [20, 21]	not provided	UWB	0.168	✗	✗
Biocontrol [20, 21]	not provided	UWB	3.22	✗	✗
Fraunhofer Research Inst. in Portugal [20, 21]	Fing.	Magnetic + Inertial	8.49	✗	✗

Table 2.2: Table of results of each solution. Some of them did not provide the methods used.

reasons since it was commercially unfeasible because LIDAR is a relatively big device that consumes much power. Quantitec Intranav applied UWB, a commercially viable enabling technology in which many companies that offer IoT solutions use nowadays. However, that was the best result using UWB technology in Microsoft competitions, the worst one (Biocontrol) had an accuracy value of 3.22m that was worse even for no additional hardware solution. Fraunhofer Research Inst. in Portugal was an example of a solution that did not work out well despite the usage of two additional hardware, which could imply that hardware alone (with no good integration of the software part) may not be sufficient to provide good results. Finally, we can see that solutions with no hardware provided similar results.

Chapter 3

Background

This chapter describes the main knowledge required to carry out this master research, which comprises hardware components, localization methods of indoor environment and distance estimation techniques.

3.1 Anchor Node

Anchor nodes are deployed to estimate the positions of mobile nodes. They can be any device that can emit signals or has a means of communication. They are usually sensor nodes or simply sensors. Sensor nodes are devices composed of sensor readings (electronic components that measure physical phenomena), micro-controller and other components. These components are grouped in three subsystems as depicted in figure 3.1 [10]: sensor, processor, and communication. The sensor subsystem is responsible for collecting data from natural phenomena, via sensor readings, in the form of analog signals and transform them into digital signals through an analog-digital converter (ADC). The processor subsystem is comprised of a micro-controller, a volatile (RAM) and non-volatile memory (Flash ROM). It is responsible for controlling functionalities of the sensor node, such as executing instructions and communication. The communication subsystem is responsible to transmit and receive data, it is made up of transceiver and co-processor devices.

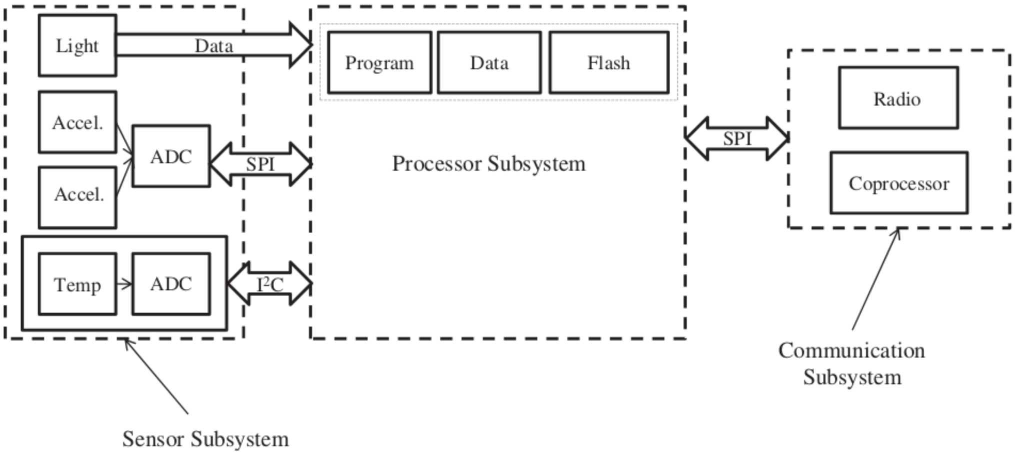


Figure 3.1: Architecture of a wireless sensor node [10, pg.48].

3.2 Multilateration

Multilateration [10] estimates a mobile node position based on distances from it to anchor nodes. But how many anchors are needed to do that? Using only one, the mobile node's position could be anywhere around the circle, as show in figure 3.2a. Deploying two, the estimation falls to two positions (figure 3.2b), however, it needs at least three or more to find the mobile node's location (figure 3.2c). The method that deploys exactly three anchor nodes is called lateration. Multilateration is a generalization of lateration which uses more than three of them [10].

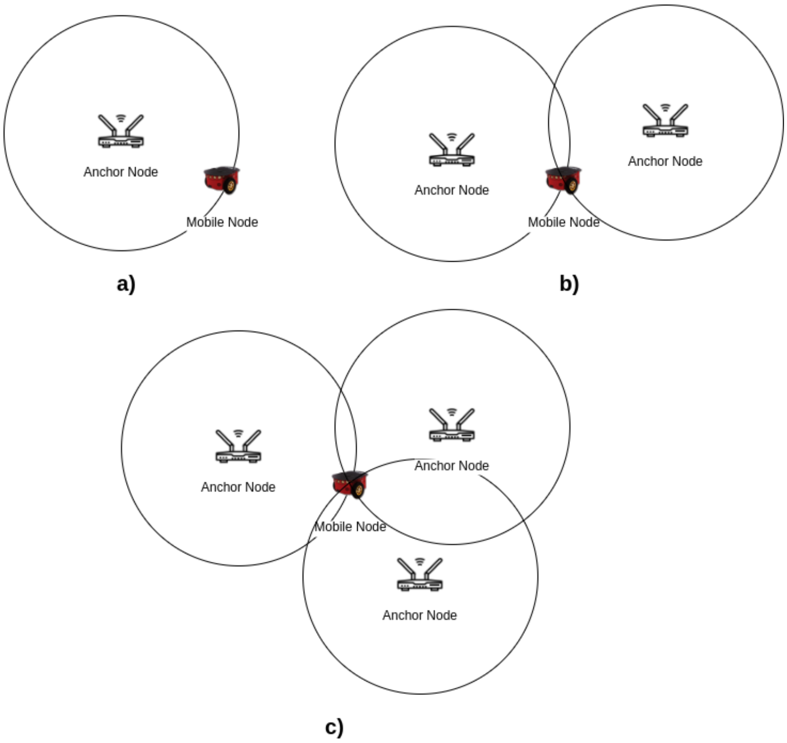


Figure 3.2: a) The robot could be in any position around the circle, b) it could be in two intersection points and c) it could be in the only one intersection point.

However, the situation as in figure 3.2c is unlikely to happen in practical environments due to signal interference since in indoor environments, signals can deflected, absorbed, attenuated or strengthened. The mobile node finds its position (x, y) using known positions from anchor nodes $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ with their estimated distances $d_1, ..., d_n$ from the mobile node. The distances are estimated by Wi-Fi RSS. The mobile node position could be found by solving the system of equations 3.1.

$$\begin{bmatrix} (x - x_1)^2 + (y - y_1)^2 \\ (x - x_2)^2 + (y - y_2)^2 \\ \vdots \\ (x - x_n)^2 + (y - y_n)^2 \end{bmatrix} = \begin{bmatrix} d_1^2 \\ d_2^2 \\ \vdots \\ d_n^2 \end{bmatrix}. \tag{3.1}$$

In this master project, distances are determined by received signal strength (RSS) described in section 3.4. Besides, this project applied three techniques to solve system

3.1: linear least squares (LLS), non-linear least squares (NLS) and weighted non-linear least squares [3].

3.2.1 Non-linear Least Squares

Rearranging the system 3.1, results:

$$\begin{bmatrix} ((x-x_1)^2+(y-y_1)^2-d_1^2)^2 \\ ((x-x_2)^2+(y-y_2)^2-d_2^2)^2 \\ \vdots \\ ((x-x_n)^2+(y-y_n)^2-d_n^2)^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.2}$$

However, in real-world environments, system 3.2 cannot be solved due to sensor measurement errors, computer numerical error and Wi-Fi signal interference. So, system 3.2 results in

$$\begin{bmatrix} ((x-x_1)^2+(y-y_1)^2-d_1^2)^2 \\ ((x-x_2)^2+(y-y_2)^2-d_2^2)^2 \\ \vdots \\ ((x-x_n)^2+(y-y_n)^2-d_n^2)^2 \end{bmatrix} = \begin{bmatrix} e_1^2 \\ e_2^2 \\ \vdots \\ e_n^2 \end{bmatrix}. \tag{3.3}$$

Non-linear Least Squares (NLS) method estimates (x, y) by minimizing the sum of errors (e_i^2), that is the difference between the estimated distance $\hat{d}_i = (x - x_i)^2 + (y - y_i)^2$ and the distance d_i^2 of each equation from system 3.3. In order words:

$$(\hat{x}, \hat{y}) = \underset{(x,y)}{\operatorname{argmin}} \frac{1}{w_T} \sum_{i=1}^{n} w_i((x_i - x)^2 + (y_i - y)^2 - d_i^2)^2, \tag{3.4}$$

Problem 3.4 cannot be solved analytically, but using an optimization algorithm, such as methods of Quasi-Newton family, Gradient Descent or Levenberg-Marquardt [38].

3.2.2 Linear Least Squares

Subtracting the last equation of 3.1 from all previous ones and, after some rearrangements, 3.1 results in system of equations 3.5.

$$\begin{bmatrix} 2(x_n - x_1) & 2(y_n - y_1) \\ 2(x_n - x_2) & 2(y_n - y_2) \\ \vdots & \vdots \\ 2(x_n - x_{n-1}) & 2(y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 - y_1^2 + x_n^2 + y_n^2 \\ d_2^2 - d_n^2 - x_2^2 - y_2^2 + x_n^2 + y_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 - y_{n-1}^2 + x_n^2 + y_n^2 \end{bmatrix} \Rightarrow A\mathbf{x} = b. \tag{3.5}$$

Linear least square method solves system 3.5 by minimize the function

zing

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \, ||A\mathbf{x} - b||^2, \tag{3.6}$$

whose estimated solution is $\hat{\mathbf{x}} = (A^T A)^{-1} A^T b$ [10], as long as $A^T A$ has inverse. Therefore, if $A^T A$ has no inverse, LLS cannot estimate a position. To avoid this, a common approach is to minimize an alternative function

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}}\ ||A\mathbf{x} - b||^2 + \lambda||\mathbf{x}||^2, \tag{3.7}$$

whose solution is $\hat{\mathbf{x}} = (A^TA + \lambda I)^{-1}A^Tb$ [14], which $\lambda > 0$ and I is an identity matrix. The inverse of $(A^TA+\lambda I)$ exists. Although, LLS is solved analytically, to find the inverse of matrix requires an optimization algorithm. So, it is an iterative method like the NLS.

Mathematical notes

This subsection shows that the solution of problem 3.7 is indeed $\hat{\mathbf{x}} = (A^TA + \lambda I)^{-1}A^Tb$. We can prove the existence of $(A^TA+\lambda I)^{-1}$ by proving that $(A^TA+\lambda I)$ is positive-definite since all positive-definite has inverse [46].

1. For any vector $\mathbf{v} \in \mathbb{R}^2$, $\mathbf{v}A^TA\mathbf{v} = ||\mathbf{v}A|| \geq 0$. Thus the matrix A^TA is positive semi-definite.

2. Let μ_i be the eigenvalue of A^TA. Then, $A^TA\mathbf{v} = \mu_i\mathbf{v}$, thus $(A^TA+\lambda I)\mathbf{v} = (\mu_i+\lambda)\mathbf{v}$. So, $(\mu_i + \lambda)$ is the eigenvalue of $(A^TA + \lambda I)$.

3. By definition $\lambda > 0$. Moreover, $\mu_i \geq 0$ since A^TA is positive semi-definite. As result, $\mu_i + \lambda > 0$, for every i which implies that $(A^TA + \lambda I)$ is positive-definite.

We can prove the global minimum of the problem 3.7. By definition, stationary points are ones which the first derivative equals zero. After applying the first derivative of the function $J(\mathbf{x}) = ||A\mathbf{x} - b||^2 + \lambda||\mathbf{x}||^2$, we can find its stationary points, which is only one:

$$\nabla J(\hat{\mathbf{x}}) = \mathbf{0} \Rightarrow A^TA\hat{\mathbf{x}} - A^Tb + \lambda I\hat{\mathbf{x}} = \mathbf{0} \therefore$$

$$\hat{\mathbf{x}} = (A^TA + \lambda I)^{-1}A^Tb.$$

To prove that $\hat{\mathbf{x}}$ is the global minimum, note that $J(\mathbf{x})$ is a strict convex function since its hessian matrix is positive-definite, that is,

$$\nabla J(\mathbf{x}) = A^TA\mathbf{x} - A^Tb + \lambda I\mathbf{x} \Rightarrow \nabla^2 J(\mathbf{x}) = A^TA + \lambda I,$$

but we already know that $A^TA + \lambda I$ is positive-definite. Thus, $\hat{\mathbf{x}}$ is in fact the global minimum, because every stationary point is also a global minimum in a convex function and it is unique since the function is strict convex [5][pg. 120-123].

3.2.3 Weighted Multilateration

There are variations of multilateration in literature, one of them adds weights to the system 3.3, that is:

$$\frac{1}{w_T}\begin{bmatrix} w_1((x-x_1)^2+(y-y_1)^2-d_1^2)^2 \\ w_2((x-x_2)^2+(y-y_2)^2-d_2^2)^2 \\ \vdots \\ w_n((x-x_n)^2+(y-y_n)^2-d_n^2)^2 \end{bmatrix} = \begin{bmatrix} e_1^2 \\ e_2^2 \\ \vdots \\ e_n^2 \end{bmatrix}. \tag{3.8}$$

Optimization algorithms solves system 3.8, as discussed in subsection 3.2.1, by minimizing the function:

$$\min_{(x,y)} \quad \frac{1}{w_T}\sum_{i=1}^{n} w_i((x_i-x)^2+(y_i-y)^2-d_i^2)^2. \tag{3.9}$$

in which $w_T = w_1 + w_2 + ... + w_n$, so w_T normalize all weights ensuring that they reside in between 0 and 1. In this research, weights are the inverse of RSS variance of a anchor node during a time interval. Let rss_{ik} be the sample k of RSS during a time interval Δt_i of node i to the mobile node in position (x, y), thus, the weight of node i is:

$$w_i = (\frac{1}{K}\sum_{k=1}^{K}(rss_{ik}-\mu_i)^2)^{-1}, \tag{3.10}$$

in which μ_i is the mean of the K samples.

3.3 Fingerpriting

Fingerpriting [4, 12] is another localization method in indoor environments that uses received signal strengths. The whole idea is that the closer the positions are, the more similar their received signals are (emitted by anchor nodes). To do that, the environment is divided into small grids. After knowing the RSS (data) of each grid, it is possible to estimate the unknown position of a mobile node by applying a machine learning method. Fingerprinting is composed of setup and execution phase.

1. Setup phase: the environment is divided into m squared cells. In each cell i, a signal receiver collects RSS from all n anchor nodes and sends them to a database. Let $\mathbf{rss}_i = \langle rss_{1i}, rss_{2i} ..., rss_{ni}\rangle$, be the vector of RSS of n anchor nodes corresponding to cell i. In the end of this phase, the database has all vectors related to m cells, that is, $\mathbf{rss}_1, \mathbf{rss}_2, ..., \mathbf{rss}_m$.

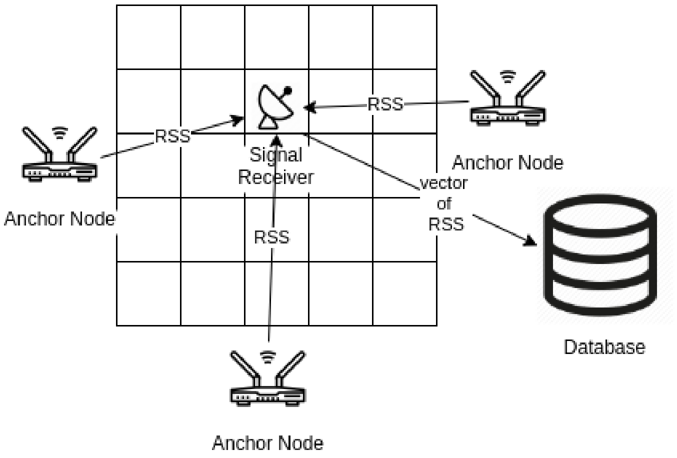


Figure 3.3: The figure depicts three anchor nodes emitting signals to a signal receiver which sends them to a database.

2. Execution phase: after collecting RSS of all cells, it is possible to estimate the position of a mobile node using machine learning techniques. In this research, machine learning algorithms were ran in a central computer ~~which~~ where was located the database. The whole process is shown in figure 3.4.

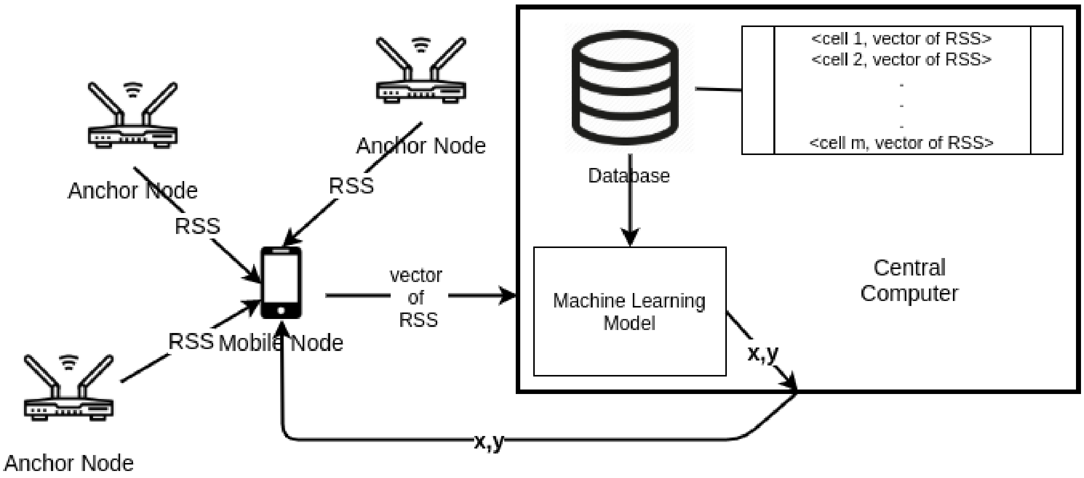


Figure 3.4: Architecture of fingerprinting method.

As mentioned before, fingerprinting applies a machine learning technique to estimate a mobile object's position. In this master research, we applied three of them: K-Nearest-Neighbors, Weighted K-Nearest-Neighbors and Neural Network. Also, in this project fingerprinting methods were executed in a central computer while multilateration was ran in a mobile node.

3.3.1 K-Nearest-Neighbors (KNN)

Let's define the similarity of a cell **i**, whose $\mathbf{x}_i \in \mathbb{R}^2$ is its middle position, to a point $\mathbf{p} \in \mathbb{R}^2$ in a map as the Euclidean distance of their RSS, that is:

$$d(\mathbf{rss}_i, \mathbf{rss}_p) = \sqrt{(rss_{1i} - rss_{1p})^2 + ... + (rss_{ni} - rss_{np})^2}, \tag{3.11}$$

in which $\mathbf{rss}_i$ and $\mathbf{rss}_p$ are the vectors of cell i and the point $\mathbf{p}$, respectively. Define the set S as the positions of cells which have the k smallest Euclidean distance values among all m cells from point $\mathbf{p}$. KNN [14] estimates the localization of a mobile node in $\mathbf{p}$ by the arithmetic average positions in S, that is,

$$\hat{\mathbf{p}} = \frac{1}{k}\sum_{i=1}^{N} \mathbf{x}_i, \text{ if } \mathbf{x}_i \in S.$$

3.3.2 Weighted K-Nearest-Neighbors (weighted KNN)

Weighted KNN [29, 12] applies weighted arithmetic mean of positions in S, that is:

$$\hat{\mathbf{p}} = \frac{1}{w_T}\sum_{i=1}^{N} w_i\mathbf{x}_i, \text{ if } \mathbf{x}_i \in S.$$

in which $w_i = 1/(d(\mathbf{rss}_i, \mathbf{rss}_m) + \epsilon)$ is the weight of cell i, ϵ is a small value to avoid division by zero, and $w_T = w_1 + w_2 + ... + w_k$.

3.3.3 Neural Network (NN)

Neural Network estimates a mobile node's location by applying a multi-layer perceptron regressor in which its inputs are RSS of each anchor node and outputs are the estimated position, as depicted in figure 3.5. This type of NN has input, intermediate and output layers [15]. Each layer has a neuron that has an activation function. That function has weighted inputs from the previous layers and it sends its output to neurons from the next layer. The weights are calibrated using an optimization algorithm with the database which is set up in first phase of fingerpriting.

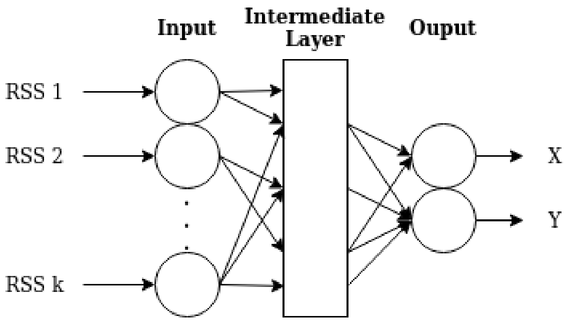


Figure 3.5: Architecture of the Neural Network.

3.4 Distance Estimation Techniques

This section describes how to the estimate distance between anchor and mobile nodes, since it is required for some localization techniques such as multilateration. Distance could be estimated through measurements of certain characteristics of signals exchanged between nodes, such as signal propagation time, signal strength and angle of arrival. As described below, there are five main methods to estimate distance [9]: one-way time of arrival (ToA), two-way ToA, time differential of arrival (TDoA), angle-of-arrival (AoA), and received signal strength (RSS), as shown in figure 3.6.

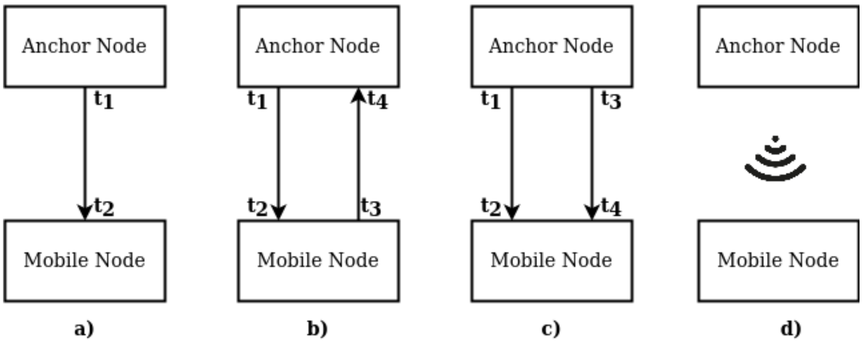


Figure 3.6: Comparison of different distance estimation techniques: a) one-way ToA, b) two-way ToA, c) TDoA and d) RSS. Nodes exchange messages (depicted as arrows) or RSS (signal icon) in t_1, t_2, t_3 and t_4 time stamps.

3.4.1 One-way time of arrival

The one-way time of arrival (one-way ToA) measures the difference of time of a signal from a anchor to a mobile node, as shown in figure 3.6a. So, in ToA the distance could be estimated using the signal velocity and its propagation time, that is, $dist = \Delta t \times v$. For example, let's suppose a anchor node emits a radio waves, $v = 3 \times 10^8 m/s$, that takes only $30 \times 10^{-9} s$ to reach a 9m distant mobile node. Thus, this method requires a highly accurate synchronization between the sender and receiver, that could be impractical for many scenarios.

3.4.2 Two-way time of arrival

The two-way time of arrival (two-way ToA) method is preferred over one-way ToA.It measures the round-trip time of a signal, that is, the time of a message leaves and returns to a anchor node. From figure 3.6b, the estimated distance is $dist = \frac{((t_4 - t_1) - (t_3 - t_2)) \times v}{2}$, v is the signal velocity. Two-way ToA is more precise than one-way ToA, since the receiver and sender calculates the time difference locally, however, its has more transmission which increases energy consumption.

3.4.3 Time differential of arrival

The time differential of arrival (TDoA) uses two types of signal to estimate the distance between nodes. For example, the first signal could be a radio signal (velocity v_1), followed by an acoustic signal (speed v_2) after a interval of t_{wait}, shown in figure 3.6c. The distance could be estimated by the equation $dist = (v_1 - v_2) \times (t_4 - t_2 - t_{wait})$. This method is much more precise than the previous ones at the expense of monetary cost since each anchor node requires another device to generate a second signal.

3.4.4 Angle-of-arrival

In angle-of-arrival (AoA), anchor nodes are embedded with antenna arrays which is possible to estimate the direction of signal propagation. For example, for acoustic signals, microphones are used to receive a single signal and the differences in arrival time are used to estimate the arrival angle. Similar to the previous one, this method is better than the first two at the expensive of monetary cost since each anchor node needs a directional antenna.

3.4.5 Received signal strength (RSS)

This method applies mathematical models that estimate distance between anchor and mobile nodes based on RSS. It does not require additional hardware to generate another signal such as TDoA, and also does not require clock synchronization like ToA. However, objects between receivers and senders could attenuate, reflect or block radio waves which increases measurement inaccuracy. The ideal scenario has no obstacles which the equation [1] is

$$P_r(d) = P_t G_t G_r (\frac{\lambda}{4\pi d})^2, \tag{3.12}$$

in which d is the distance (in meters) traveled by a radio signal with transmitted strength (P_t) in dBm, the received strength (P_r) in dBm, G_t is the gain of the transmitting antenna in dBm, G_r is the gain of the receiving antenna in dBm, and λ is the signal wavelength in meters. However, for real-world indoor environments it is common to use the log-normal distance path loss model that predicts the strength received and whose equation depends on the Gaussian norm, that is,

$$P_r(d) \sim \mathcal{N}(\overline{P_r(d)}, \sigma^2), \tag{3.13}$$

where $\overline{P_r(d)}$ is the average received signal strength (RSS) and σ^2 (in dBm) is the variance which is related to the random effect of shadowing [17]. The log-normal distance path loss model is

$$\hat{P}_r(d) = P_r(d_0) - 10 n_p \log_{10}(\frac{d}{d_0}) + X_\sigma \quad (dBm), \tag{3.14}$$

where $P_r(d_0)$ is a known RSS in dBm at a reference distance d_0, n_p is the path loss exponent that depends on the environment, $\hat{P}_r(d)$ is the estimated RSS and X_σ is a normally distributed random variable with zero mean and σ standard deviation. By selecting the reference distance to 1m, the equation 3.14 results to

$$\hat{P}_r(d) = P_r(1) - 10 n_p \log_{10}(d) + X_\sigma \quad (dBm), \tag{3.15}$$

therefore, this model can be expressed as

$$\hat{P}_r(d) = -\eta \log_{10}(d) + C \quad (dBm), \tag{3.16}$$

but the problem requires the inverse of 3.16, that is

$$\hat{d} = 10^{-\frac{P_r(d)-C}{\eta}} \quad (meters), \tag{3.17}$$

where $\eta = 10n_p$. $C = P_r(1)$ and the estimated distance $\hat{d}$. Thus, equation 3.17 has two unknown parameters, n_p and C, that are found by applying regression which is a process called calibration of the model. Calibration requires data, which are RSS of various distances from an anchor node. Note that calibration is necessary for each anchor node and every time it changes position. The detailed description of calibration is in section 4.4.

Log-normal path-loss is not the only model for distance estimation based on RSS. For example, polynomial model (equation 3.18) could also be applied to, which uses polynomial regression [42] to calibrate parameter $a_1, a_2, ..., a_n$.

$$\hat{d} = a_0 + a_1 P_r(d) + ... + a_n P_r(d)^n \quad (meters). \tag{3.18}$$

Chapter 4

Set-ups and implementations

This chapter describes the setups and implementations of two experiments. The first one is the main experiment that was conducted in the Computer System Laboratory, Institute of Computing at the State University of Campinas where there were a high radio signal interference. In contrast, the second is a minor one that was performed in a much less noisy place, two rooms of a house. We deployed six Raspberry Pi as anchor nodes and one to run the methods. All anchor nodes emitted Wi-Fi signals only. In these experiments, methods and models were evaluated according to the following measurements:

- Localization error (in meters) is the Euclidean distance between the predicted position $(\hat{x}, \hat{y})$ and the true one (x, y):

  $$e = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2} \quad (m),$$

  and sections also mention average localization error (accuracy).

- The root mean square error (RMSE) of a model is

  $$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{d}_i - d_i)^2} \quad (m),$$

  which $\hat{d}_i$ and d_i is the estimated and true distance from an anchor to a mobile node, for n observable distances.

- Response time is the time elapse of a method to provide an estimated position, in milliseconds.

- Energy consumption is the energy consumed of a method, in decijoules (dJ), during its response time. To measure the average energy consumption of a method, Raspberry Pi executes it constantly until the battery runs out while generating a log of timestamps and the number of executions. The battery had a nominal capacity of 46000 joules. The average energy consumption in decijoules (dJ) is:

  $$\frac{battery\ capacity}{num.\ of\ executions} = \frac{460000dJ}{num.\ of\ executions}$$

SoC	Broadcom BCM2837
CPU	Quad-core ARM Cortex-A53 @ 1.2GHz
RAM	1GB LPDDR2 (900 MHz)
Storage	32 GB SanDisk Ultra microSD
Operating System	Raspbian Buster Lite 2020-02-13
Linux kernel version	4.14.79 armv7l

Table 4.1: Raspberry Pi specifications.

CPU	Intel Core i5-5200U CPU @ 2.20GHz × 4
RAM	8.192GB DDR3 (1600 MT/s)
Storage	512GB of SSD
Operating System	Ubuntu 18.04.3 LTS
Linux kernel version	4.15.0-96-generic ×86_64

Table 4.2: Central computer specifications.

It was assumed that the nominal and actual capacity of the battery were equal. Although it is not true, it did not compromise the comparison, since the same assumption was made for all methods and all of them ran using the same battery.

Methods use **training** and **test points**. Multilateration utilizes the former to calibrate the model while the latter to evaluate it. Fingerpriting does not need calibration, instead it estimates position of test points using training ones. Table 4.1 shows the device specifications that executed all methods. Fingerprinting methods used a central computer with the specification shown in table 4.2. All methods were implemented in Python version 3+, used numpy, scipy, matplotlib, and sklearn packages. All the codes, tables and images are stored in the following GitHub link:

`https://github.com/tiagotrocoli/Master-Project`

4.1 The local optimal subset

During the experiments, we searched for a subset of anchor nodes that gave the lowest average localization error for each method. To do that, we tested many subsets and chose the one with the lowest localization error which is called the local optimal subset. It is "local" since it may not be the best overall, but the best among the tested subsets. Each method had its local optimal subset that was different from one another. So, the localization error values shown in this research refer to the local optimal subset of each method which was not the set of all 6 anchor nodes available.

4.2 First experiment setup

As mentioned previously, the first experiment was in a real-world office environment with a strong radio signal interference, figure 4.1, which was a fully furnished room having office-chairs, tables, writing desks, computers, people and Wi-Fi routers as the main source of signal interference. Six Raspberry Pi were deployed as anchor nodes, and one run the methods. The room had 85.$8m^2$, 13m of length and 6.6m of width. As motioned previously, all anchor nodes emitted Wi-Fi signals. We also collected RSS from 42 training points (table A.2) and 18 test points (table A.1). For each point, we collected the average value of 10 RSS for every anchor node. In total, there were 2520 and 1080 RSS of training and test points, respectively. Moreover, nine techniques were compared which can be grouped into multilateration and fingerprinting.

4.3 Second experiment setup

We carried out a minor experiment in two fully furnished rooms of a house, figure 4.2. One had $16m^2$ and the other $29.624m^2$. Six Raspberry Pi were deployed as anchor nodes, and one run the methods in each test point. All anchor nodes emitted Wi-Fi signals. There were 37 training points (table B.2) and 17 test points (table B.1). We did not collect the average values of RSS for each test point since the Wi-Fi signals did not fluctuate. It happened because the second experiment had a weaker external signal interference.

4.4 Implementations

Calibration of anchor nodes is a process which finds a mathematical function (model) that estimates the distance between an anchor and a mobile node given an RSS. And multilateration requires that calibration to execute. Given the model below:

$$\hat{d}_{ij} = g(\mathbf{a}_i, rss_{ij}),$$

in which $g(\mathbf{a}_i, rss_{ij})$ is a model of k parameters ($\mathbf{a}_i \in \mathbb{R}^k$), rss_{ij} and $\hat{d}_{ij}$ are the RSS and the estimated distance of anchor node i to the training point j, respectively. Calibration of anchor node i estimates the best parameters $\mathbf{a}_i$ given all RSS $(rss_{i1}, rss_{i2}, ..., rss_{im})$ of training points (table A.2 (first exp.) or table B.2 (second exp.)) to minimize differences between the actual and estimated distances from anchor node i to all these points. To do that, least squares was applied:

$$\hat{\mathbf{a}}_i = \underset{\mathbf{a}_i}{\operatorname{argmin}} \sum_j (d_{ij} - g(\mathbf{a}_i, rss_{ij}))^2,$$

in which d_{ij} is the real distance from anchor node i to training point j. Since there were 6 anchor nodes, 6 calibrations were done using polynomial (equation 4.2) and 6 for lognormal shadowing path-loss model (equation 4.1). Note that, all RSS of anchor node i is one of AN_1, ..., AN_6 columns from table A.2 (first exp.) or B.2 (second exp.) . All RSS of a training point j is the row j of these tables. Also, rss_{ij} is the mean value of

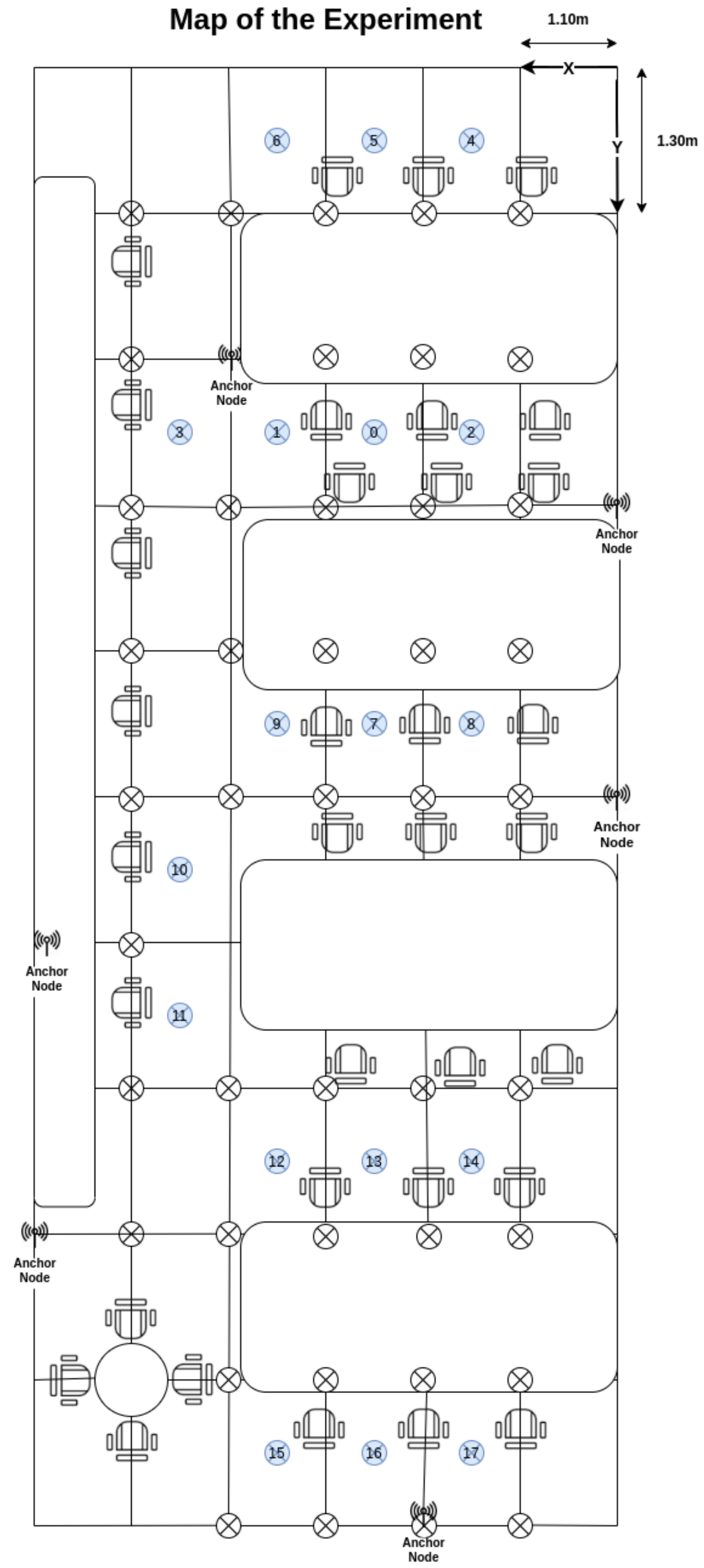


Figure 4.1: The first experiment map with anchor nodes that emitted Wi-Fi signals. Empty crossed circles represent training points, filled ones are test points, rounded squares are tables and writing desks with office chairs. The image does not show computers and other furniture.

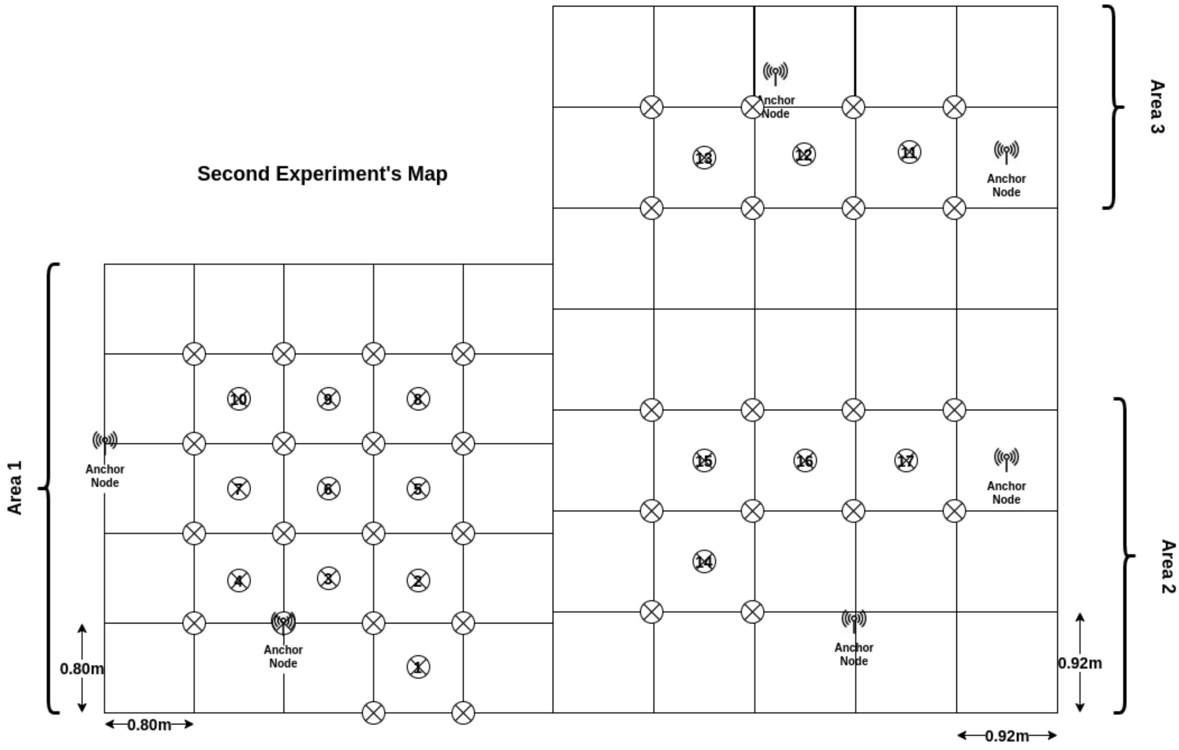


Figure 4.2: Two fully furnished rooms where the second experiment took place. The crossed circles with a number are test points, the others are training points. The image does not show furniture.

N samples of RSS from anchor node i to training point j, that is:

$$rss_{ij} = \frac{1}{N}\sum_{k=1}^{10} rss_{ijk},$$

in which rss_{ijk} is the k-th sample. In the first experiment $N = 10$ while the second $N = 1$ since RSS did not fluctuate as explained previously.

- Lognormal shadowing path-loss model [7]:

$$g(\mathbf{a}_i, rss_{ij}) = 10^{-\frac{A_i - rss_{ij}}{10 n_{p_i}}} \quad (meters), \tag{4.1}$$

  in which $\mathbf{a}_i = \{A_i, n_{p_i}\}$ (n_{p_i} is the path-loss exponent) are the known parameters of anchor node i model which are found through calibration, and rss_{ij} is the variable.

- Polynomial model [23]:

$$g(\mathbf{a}_i, rss_{ij}) = a_{0i} + a_{1i} rss_{ij} + ... + a_{ni} rss_{ij}^{n} \quad (meters), \tag{4.2}$$

  in which $\mathbf{a}_i = \{a_{0i}, a_{1i}, ..., a_{ni}\}$ are known parameters of anchor node i model which are found through calibration, and rss_{ij} is the variable. In this research, $n = 4$.

The project applied all the algorithm below. The only one which is not shown is Fingerprinting with a multi-layer perceptron regressor that applied MLPRegressor of the sklearn package [31]. The neural network had 10 neurons in the hidden layer, 6 in the input layer (one for each anchor node) and 2 in output layer $(\hat{x}, \hat{y})$. Its activation function was the identity with LBFGS solver. Note that multilateration techniques can use the lognormal path-loss or polynomial model.

In the following algorithms, $n = 6$ (6 anchor nodes), $\lambda = 0.1$, $\epsilon = 10^{-6}$, and N is the number of training points. Therefore, N is 42 (first experiment) or 37 (second experiment). Define the vector $\mathbf{rss}_j = \langle rss_{1j}, rss_{2j} ..., rss_{nj} \rangle$, that is the j row of table A.2 (fist exp.) or B.2 (second exp.).

```
Algorithm 1: NLS Multilateration
  Input: The test point $j = (x, y)$, the parameter model $\mathbf{a}_i$, $rss_{ij}$ and position of
         $(x_i, y_i)$ of each anchor node, $1 \leq i \leq n$.
  Output: The estimated position $(\hat{x}, \hat{y})$ of test point $j$.
  /* Estimate the distance from anchor nodes to the test point.     */
  for $i = 1, ..., n$ do
    $\hat{d}_{ij} = g(\mathbf{a}_i, rss_{ij})$
  end
  /* Estimate the position of test point using the BFS solver [28].
     */
      $(\hat{x}, \hat{y}) = \underset{(x,y)}{\operatorname{argmin}} \quad \frac{1}{n} \sum_{i=1}^{n} ((x_i - x)^2 + (y_i - y)^2 - \hat{d}_{ij})^2.$
```

```
Algorithm 2: LLS Multilateration
  Input: The test point $j = (x, y)$, the parameter model $\mathbf{a}_i$, $rss_{ij}$ and position of
         $(x_i, y_i)$ of each anchor node, $1 \leq i \leq n$. The $\lambda > 0$.
  Output: The estimated position $\hat{\mathbf{x}}$ of test point $j$.
  /* Estimate the distance from anchor nodes to the test point.     */
  for $i = 1, ..., n$ do
    $\hat{d}_{ij} = g(\mathbf{a}_i, rss_{ij})$
  end
  /* Let $A$ and $b$ be the same as equation 3.5.  The estimated position
     is                                                               */
      $\hat{\mathbf{x}} = (A^T A + \lambda I)^{-1} A^T b.$
```

Algorithm 3: Weighted Multilateration

```
Input: The test point $j = (x, y)$, the parameter model $\mathbf{a}_i$, $rss_{ij}$ and position of
       $(x_i, y_i)$ of each anchor node, $1 \leq i \leq n$
Output: The estimated position $(\hat{x}, \hat{y})$ of test point $j$.
/* Estimate the distance from anchor nodes to the test point.     */
for $i = 1, ..., n$ do
    $\hat{d}_{ij} = g(\mathbf{a}_i, rss_{ij})$
    $w_i = (\frac{1}{K}\sum_{k=1}^{K}(rss_{ijk} - \mu_i)^2)^{-1}$ (see subsection 3.2.3)
end
Define $w_T = w_1 + w_2 + ... + w_n$.
/* Estimate the position of test point using the BFS solver [28]  */

    $$(\hat{x}, \hat{y}) = \underset{(x,y)}{\text{argmin}} \quad \frac{1}{w_T}\sum_{i=1}^{n} w_i((x_i - x)^2 + (y_i - y)^2 - \hat{d}_{ij})^2.$$
```

Algorithm 4: Fingerprinting with KNN

```
Input: The test point $\mathbf{b} \in \mathbb{R}^2$, the vector $\mathbf{rss}_i$ and position $\mathbf{p}_i \in \mathbb{R}^2$ of each
       training point, $1 \leq i \leq N$. The test point $b \in \mathbb{R}^2$ and its vector $\mathbf{rss}_b$.
Output: The estimated position $\hat{\mathbf{x}}$ of test point $b$.
/* Calculate Euclidean distance values.                           */
for $i = 1, ..., N$ do
    $d(\mathbf{rss}_i, \mathbf{rss}_b) = ||\mathbf{rss}_i - \mathbf{rss}_b||$. (see eq. 3.11)
end
Define the set $\mathbf{S}$ to be the coordinates of $k$ training points with the lowest
  Euclidean distance values.
/* Estimate the position of the test point.                       */

    $$\hat{\mathbf{x}} = \frac{1}{k}\sum_{i=1}^{N} \mathbf{p}_i, \text{ if } \mathbf{p}_i \in \mathbf{S}.$$
```

Algorithm 5: Weighted Fingerprinting

```
Input: The vector $\mathbf{rss}_i$ and position $\mathbf{p}_i \in \mathbb{R}^2$ of each training point, $1 \leq i \leq N$.
       The test point $b \in \mathbb{R}^2$ and its vector $\mathbf{rss}_b$.
Output: The estimated position $\hat{\mathbf{x}}$ of test point $b$.
/* Calculate Euclidean distance values.                                  */
for $i = 1, ..., N$ do
    $d(\mathbf{rss}_i, \mathbf{rss}_b) = ||\mathbf{rss}_i - \mathbf{rss}_b||$. (see eq. 3.11)
end
Define the set $\mathbf{S}$ to be the coordinates of $k$ training points with the lowest
  Euclidean distance values.
Define the weight $w_i = 1/(d(\mathbf{rss}_i, \mathbf{rss}_o) + \epsilon)$, $\forall i$ such that $\mathbf{p}_i \in \mathbf{S}$.
Define $w_T$ as the some of all weights.
/* Estimate the position of the test point.                              */
        $$\hat{\mathbf{x}} = \frac{1}{w_T} \sum_{i=1}^{N} w_i \mathbf{p}_i, \text{ if } \mathbf{p}_i \in \mathbf{S}.$$
```

Chapter 5

Experimental Results

5.1 First Experiment

This section concerns the results of the first experiment, which was the major one having a strong signal interference, and it took place in the Computer System Laboratory, Institute of Computing at the University of Campinas.

5.1.1 Multilateration

This section is divided into two parts, since we applied two models: lognormal and polynomial. We decided to apply the latter since we thought we could improve the results of the former. And in fact, in the next sections, experiments proved our prediction. Three methods were implemented:

- Multilateration using NLS.
- Multilateration using LLS.
- Weighted Multilateration.

Polynomial model approach

The calibration of the six anchor nodes was done using the polynomial model. The RMSE values ranged from 2.3m to 1.38m. The different positions of anchor nodes caused the variation of RMSE values since the distribution of data (RSS) depends on the position. Thus each anchor node had different parameters values of the polynomial model.

Figure 5.5 depicts the polynomial model for each anchor node with 42 RSS (data). Note that, the data represents the average values of 10 RSS for each test point. Moreover, the RSS did not necessarily decrease as distance increase, that was probably the effect of furniture and Wi-Fi interference from other devices inside the room. That effect degraded the models which worsened localization estimation of multilateration techniques. Note also that data distributions were different because the two anchor nodes were in different locations.

Figures 5.1 and 5.2 show the results of the polynomial multilaterations that used the local optimal subset of anchor nodes (explained in section 4.1). The most accurate one was multilateration with NLS. Note that weighted multilateration complexity did not make up for accuracy, so much so that even the simple LLS had a better performance. LLS was the fastest one and had the the lowest energy consumption, it was not a surprise since LLS was the simplest method.

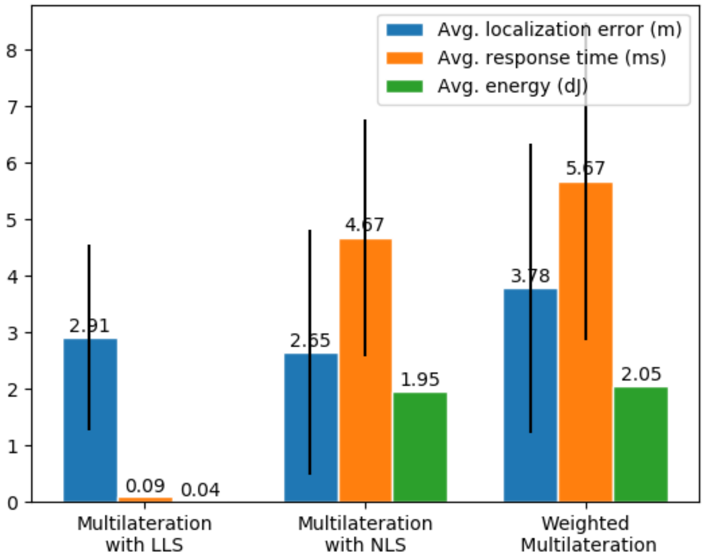


Figure 5.1: Results of methods that applied polynomial model. Methods used the local optimal subset of anchor nodes.

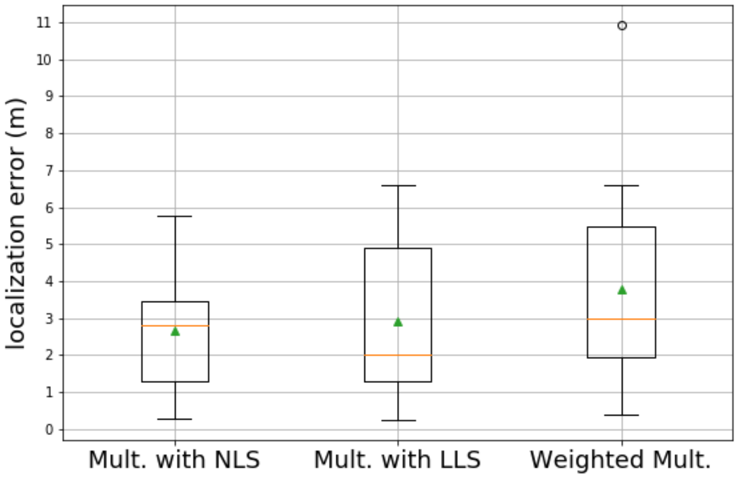


Figure 5.2: Box plots of polynomial multilaterations.

Some test points were difficult to estimate as shown in figure 5.3, specially 4, 5 and 6. They were located in the upmost part of the map, figure 4.1. That was the most obliterated region, having tall furniture near the wall, full of objects and is out-of-sight of all anchor nodes. Conversely, points 9, 10, 12 and 13 had the lowest errors, they were located in the line of sight of at least two anchor nodes and surrounded with few or no furniture. Note that weighted multilateration was the worst method, its lack of accuracy mostly comes down to its inaccurate estimation of test point 1. NLS had the best average estimation and also the smallest localization error (test point 9).

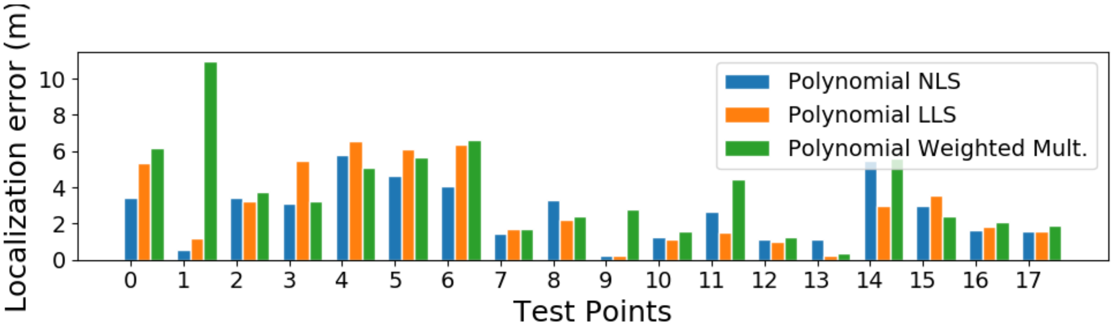


Figure 5.3: Error for each test point separated by methods. Test points surrounded by furniture had worst localization errors.

This paragraph describes an interesting experiment involving the selection of anchor nodes and its impacts on accuracy which brought unexpected results. Figure 5.4 shows four multilateration, each one utilized different anchor nodes. *A* used 5 anchor nodes and was the most accurate of all. *B* used 4 anchor nodes leaving aside two of them with the greatest RMSE. *C* utilized all except two with the lowest RMSE. *D* employed all anchor nodes. We might expect that removing anchor nodes with the worst RMSE could be a good way to increase accuracy, but that could not be true, as the experiment showed. Conversely, removing nodes with the lowest RMSE may not lead to good results. And, using all of them could not be the best alternative. Thus, selecting anchor nodes might not be as straightforward as we may lead to believe.

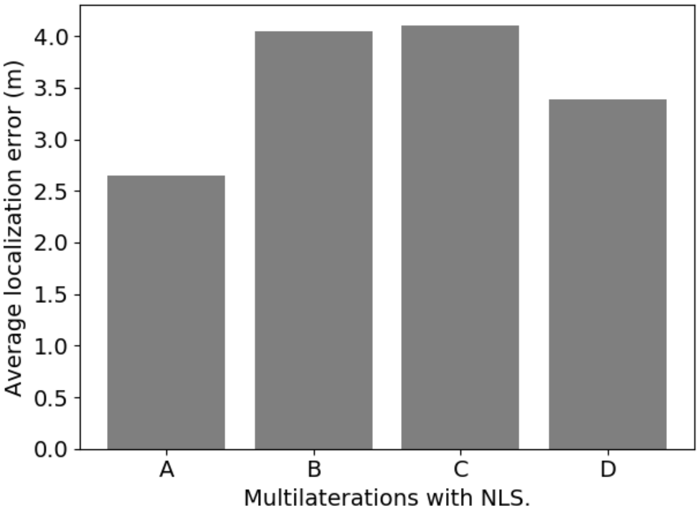
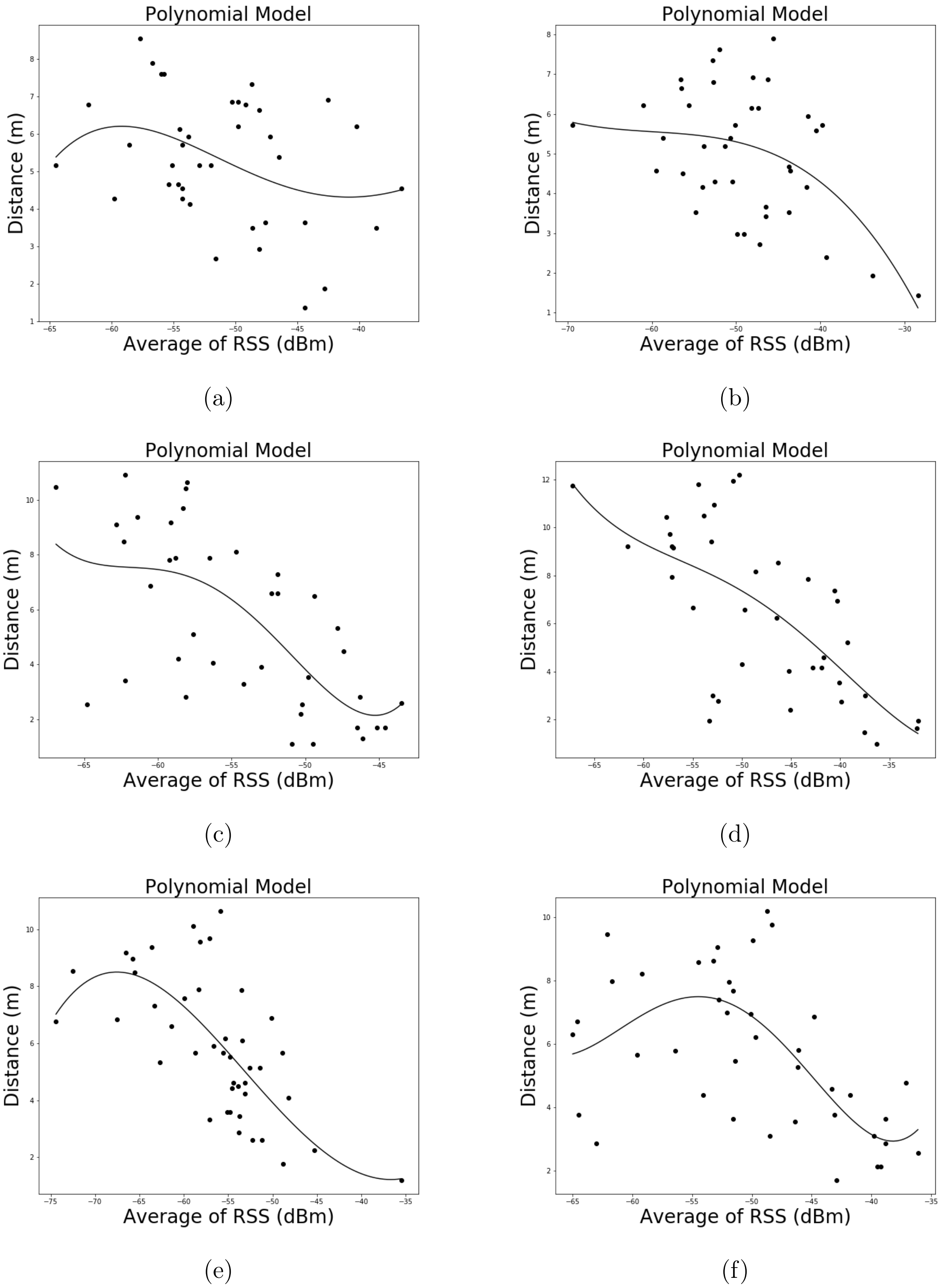


Figure 5.4: Results of NLS multilaterations using different subsets of anchor nodes.



(a)

(b)

(c)

(d)

(e)

(f)

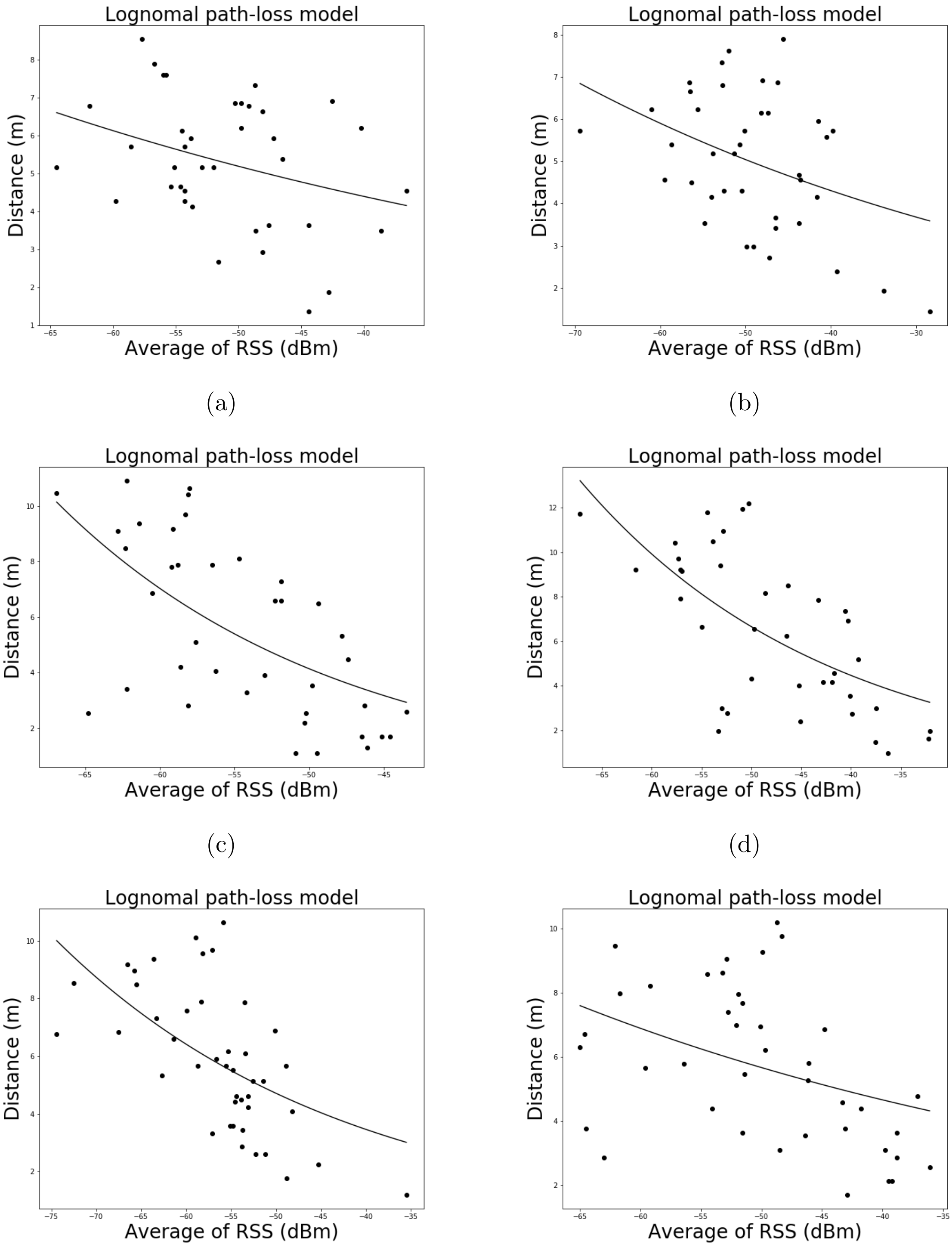
Figure 5.5: Graphs of RSS (dBm) × distances of 6 anchor nodes. Different positions of them caused different data (RSS) distributions and the interference on the data degraded the localization estimation. Polynomial model fit the data better than lognormal one (fig. 5.6).

Lognormal model approach

Likewise in section 5.1.1, we calibrated the lognormal model using a dataset of RSS from 42 training points. Its RMSE values of ranged from 2.43m to 1.47m. In general, RSME values were worst than of polynomial one, that was the reason methods that applied the former had a worst estimation than those which applied the latter. Likewise the previous section, figures 5.6 show graphs displaying 42 RSS (data) and the lognormal functions for each anchor node. As previously explained, data (RSS) distribution varies according to location of anchor nodes, that is the reason the two images show different data distributions.

Figures 5.7 and 5.8 show the results of the lognormal multilaterations. Each method used a local optimal subset of anchor nodes as explained in section 4.1. Note that accuracy differences were small, even though LLS was simpler than NLS and weighted methods. LLS outperformed them over response time and energy consumption, since it was the simplest method. Note also that standard deviations of response time of LLS was so small that were not shown, it was less than 10^{-3}. Note as well that the average response time of NLS was about 1.06ms greater than weighted multilateration which was a more complex method, however, the former consumed 0.42dJ less than the latter. We believe that these differences were due to 4 outliers (test points) of the latter that increased the average response time a little bit but was not enough to increase the average energy consumption.

Methods that used lognormal model, had similar localization error trends to polynomial ones, as shown in figure 5.9. Like polynomial methods, points 4, 5 and 6 had the worst error. As mention previously, they were located in the upmost part of the map, figure 4.1. That was the most obliterated region, having tall furniture near the wall, surrounded with objects and was out-of-sight regarding all anchor nodes. Conversely, points 9, 10, 12 as well 13 had the lowest errors, and they were located in the line of sight of at least two anchor nodes and surrounded with few or no furniture.



(a) (b)

(c) (d)

(e) (f)

Figure 5.6: Graphs of RSS (dBm) × distances of 6 anchor nodes. Different positions of them caused different data distributions.

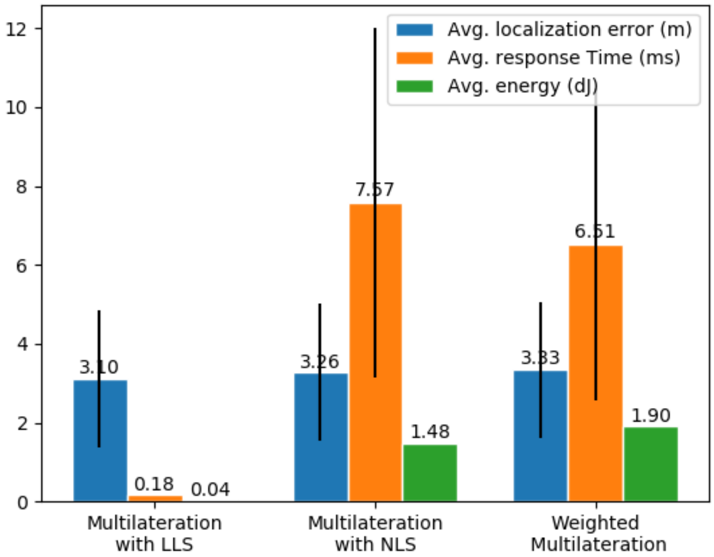


Figure 5.7: Results for each method that applied lognormal path-loss model. They were worst than methods that used polynomial models.

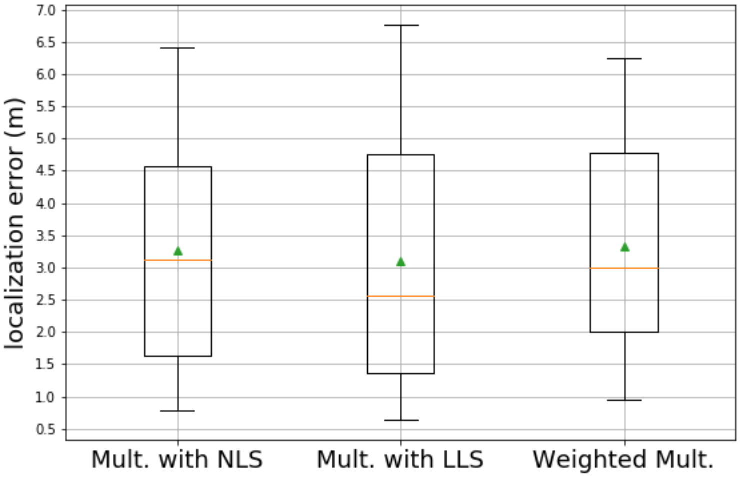


Figure 5.8: Box plots of lognormal multilaterations.

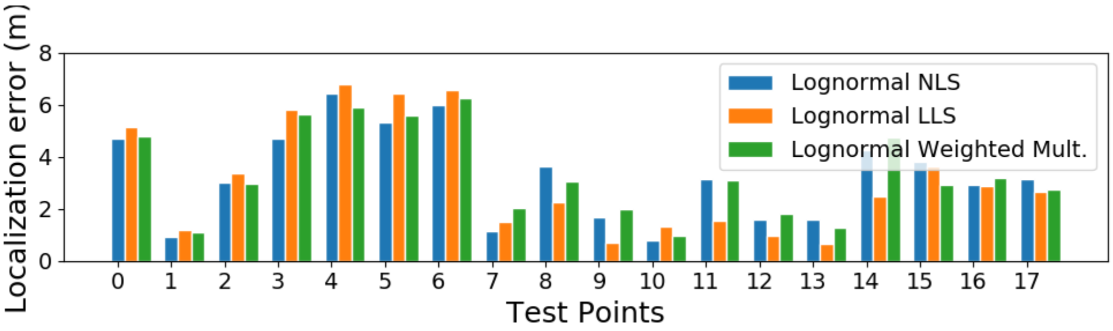


Figure 5.9: Error for each test point separated by methods.

5.1.2 Fingerpriting

This section compares three types of fingerpriting methods:

- Fingerpriting with KNN.
- Fingerpriting with weighted KNN.
- Fingerpriting with NN.

This section also shows that fingerpriting methods are more accurate than multilateration ones, but this advantage comes with a price, the setup phase (see section 3.3) maybe time-consuming and could require much more RSS data. However, in this research both shares the same data, that is, Fingerpriting utilized the same 42 training points of Multilateration. But Fingerprinting techniques of this research need to communicate via Wi-Fi with a central computer to estimate its position, as mentioned in section 3.3.

Figures 5.10 and 5.11 show the results of the fingerprinting methods that used the local optimal subset of anchor nodes (explained in section 4.1). As we can see in figure 5.10, fingerpriting with KNN ($k = 4$) was the most accurate method among the three and, after analysing all multilateration ones as well, we concluded that it was also the most accurate overall. Besides, it was faster than multilateration even with the drawback of connection delay. However, the central computer was faster than Raspberry Pi so it made up for that delay. We can also verify that neural network caused a big overhead and its complexity did not make up for accuracy. In terms of energy consumption, we can verify the substantial difference between the standard fingerprinting to multilateration. The most energy demanding method of the former had an average value of 540% less than the most demanding of the latter. Even NN consumed very little energy, that because the central computer executed them which saved energy of Raspberry Pi.

Figure 5.12b shows average localization errors of different subset of 4 out of 6 anchor nodes. Specifically, there exist 15 groups (combinations), but the image shows 8 of them. We can see that localization errors vary for each subset. Thus, during the setup of a indoor localization solution, it could be a good approach to analyze different subsets to pick out one with the lowest error. Figure 5.12a shows the optimal subset for each number of anchor nodes. So, number of nodes in a subset also affects the accuracy of fingerprinting. Expanding our analysis, it could be a good practice to test different subset varying its size also to find one that gives good results.

Some test points were difficult to estimate as shown in figure 5.13, specially points 8, 3 and 14. That means training points around them were not chosen by fingerprinting. Conversely, points 7, 9 and 17 have the smallest errors. As noted before, everything went down to which training points were chosen by fingerprinting. And sometimes, it made a bad choice because some of them were "similar" to the test points, although they were distant to each other. That bad choices were probably due to furniture and interference between Wi-Fi signals.

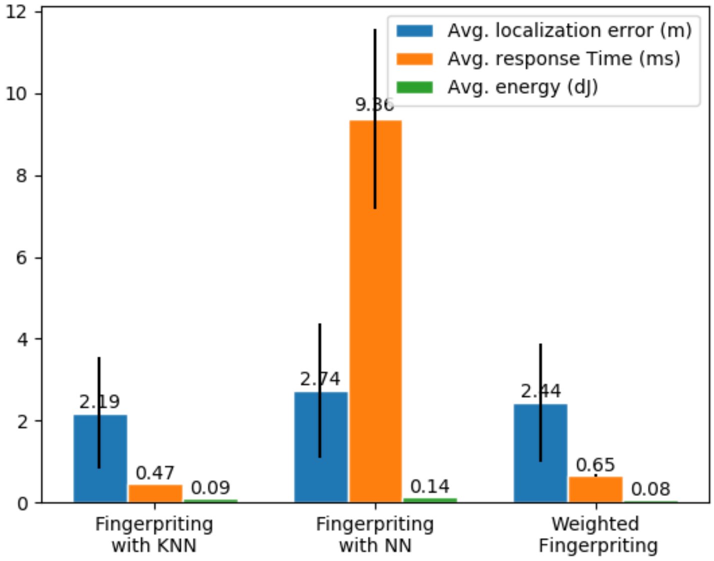


Figure 5.10: Results for each type of fingerprinting method. They achieved better results than multilateration ones.

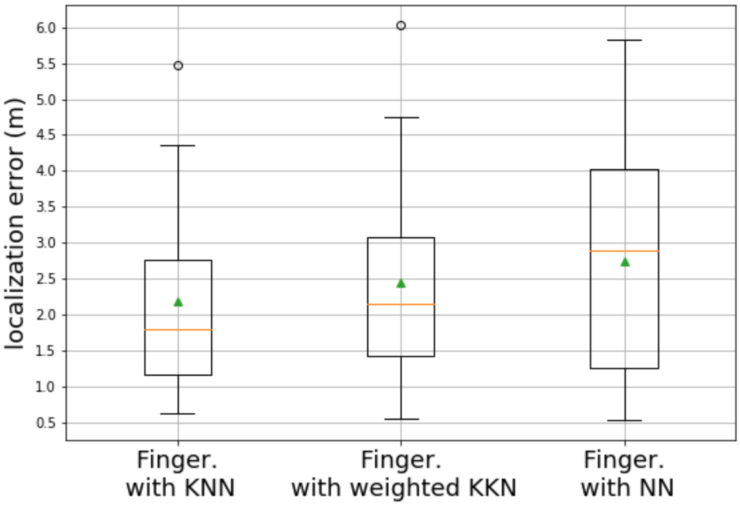
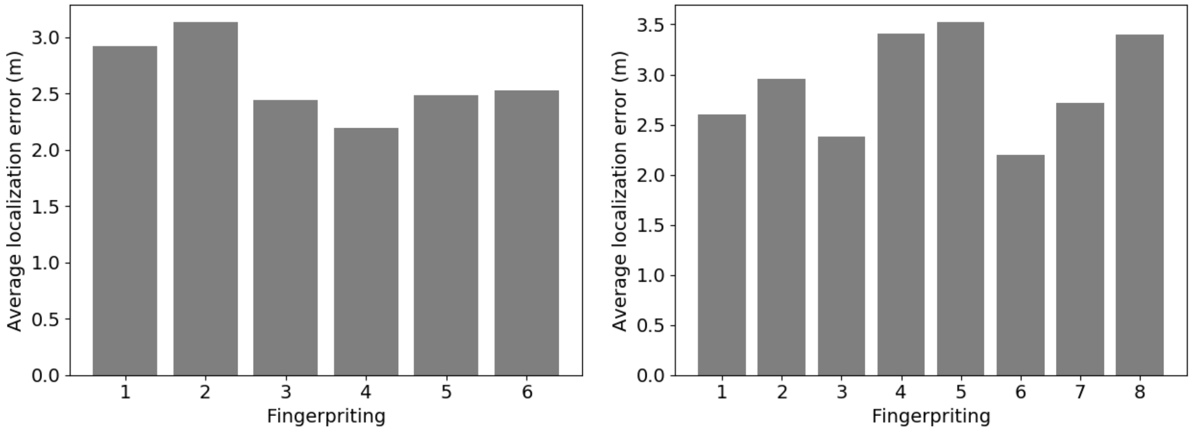


Figure 5.11: Box plots of fingerprinting methods.



(a)

(b)

Figure 5.12: Both images relates to fingerpriting with KNN. Image (a): the 1st column corresponds to fingerprinting method that employed the optimal subset of only 1 anchor node. Likewise the 2nd to the 6th that correspond to the optimal subset with 2 to 6 anchor nodes, respectively. Image (b): results of average errors by different subsets of 4 among 6 anchor nodes of the same fingerprinting method.

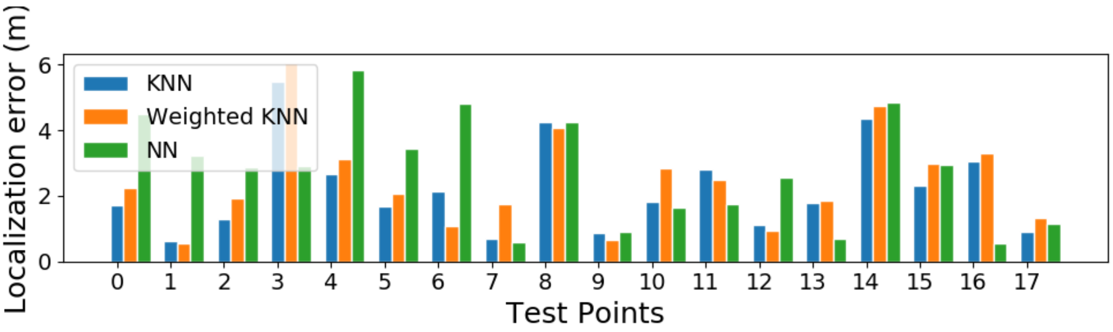


Figure 5.13: Localization error for each test point separated by Fingerprinting methods.

5.1.3 Overall Comparison

The table 5.1 shows the results of all methods of the first experiment. Results attested that fingerprinting variations had the best overall performance in which the winner was the KNN since KNN had the smallest average localization error and was in 3rd place in terms of energy consumption and response time. The polynomial model increased the accuracy of multilateration (except for the weighted one) and also sped up the response time but increased the energy consumption. Probably it made easier for optimization algorithms to find a solution in the expensive of more power consumption, after all, it is more complex than the lognormal model.

	Localization error (m)	Response Time (ms)	Energy Consumption (dJ)
Polynomial Multilateration			
LLS	2.91	0.09	0.04
NLS	2.55	4.67	1.95
Weighted	3.78	5.67	2.05
Lognormal Multilateration			
LLS	3.10	0.18	0.04
NLS	3.26	7.57	1.48
Weighted	3.33	6.51	1.90
Fingerprinting			
KNN	2.19	0.47	0.09
NN	2.74	9.36	0.14
Weighted	2.44	0.65	0.08

Table 5.1: Average results of all methods of the first experiment.

5.2 Second experiment

This section concerns the results of the second experiment which was conducted in two rooms of a house. Initially, the experiment would take place in the same area as the previous one. However, it was changed due to the pandemic caused by COVID-19. Thus, the objective of it changed as well. Its objective is to analyze methods in a controlled place, which differentiates the first one where the place is a laboratory with people walking around, Wi-Fi routers and other devices that can cause strong signal interference.

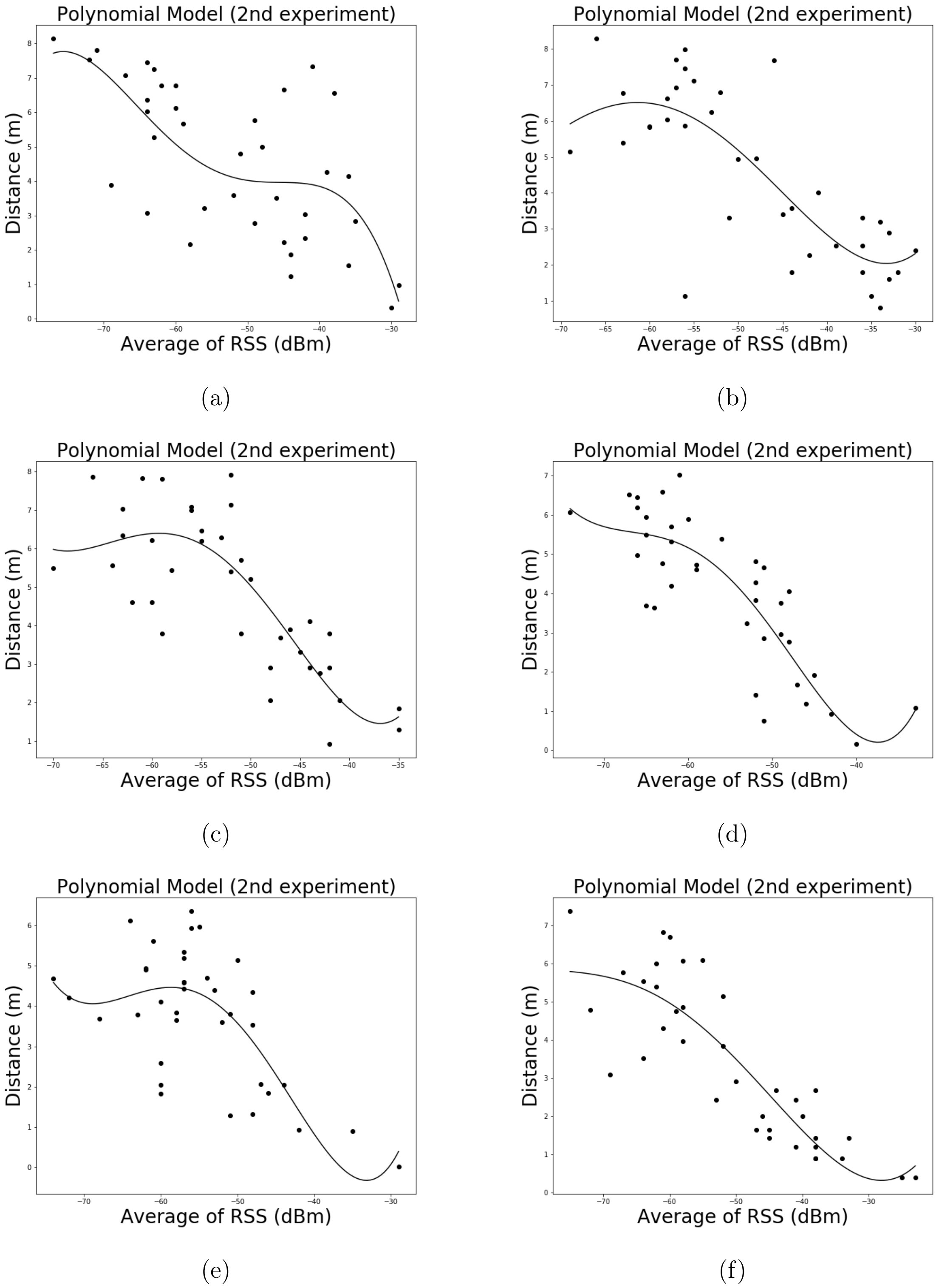
Table 5.2 shows the results of all methods implemented in the second experiment. Again, the fingerprinting methods outperformed the multilateration variations and the polynomial model improved a bit the average localization error. Notice also that the results of the second experiment were much better than the first, that is because the former had much less external signal interference than the latter and had less furniture as well. In terms of anchor nodes calibration, five of them had lower RSME values than

Polynomial Multilateration		
	LLS	NLS
Avg. localization error (m)	1.20 ± 0.75	1.23 ± 0.68
Lognormal Multilateration		
	LSS	NLS
Avg. localization error (m)	1.39 ± 0.92	1.38 ± 1.05
Fingerprinting		
	KNN	Weighted
Avg. localization error (m)	0.97 ± 0.31	0.99 ± 0.34

Table 5.2: Results of all methods of the second experiment.

the first experiment. RMSE ranged from 1.10m to 4.91m, but four were less than 2.0m. Figures 5.15 and 5.14 show the graphs displaying 38 RSS (data) and polynomial and lognormal model, respectively.

Note the absence of weighted multilateration and neural network (NN). The former was not implemented since the weights depend on the variance of RSS in a given test point, but there was no fluctuation of RSS. Because of that, we only collected one sample of RSS for each test point, that data was not enough to train the neural network properly. Since the methods were the same, their response time values were about the same from the first experiment. Methods that used optimization algorithms could have a different response time, but the number of anchor nodes of local optimal subsets differed only one or two from the first experiment which was not enough to cause a different response time. Likewise the energy consumption.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.14: Graphs of RSS (dBm) × distances of 6 anchor nodes. The second experiment had much less signal interference which improved methods' accuracy.

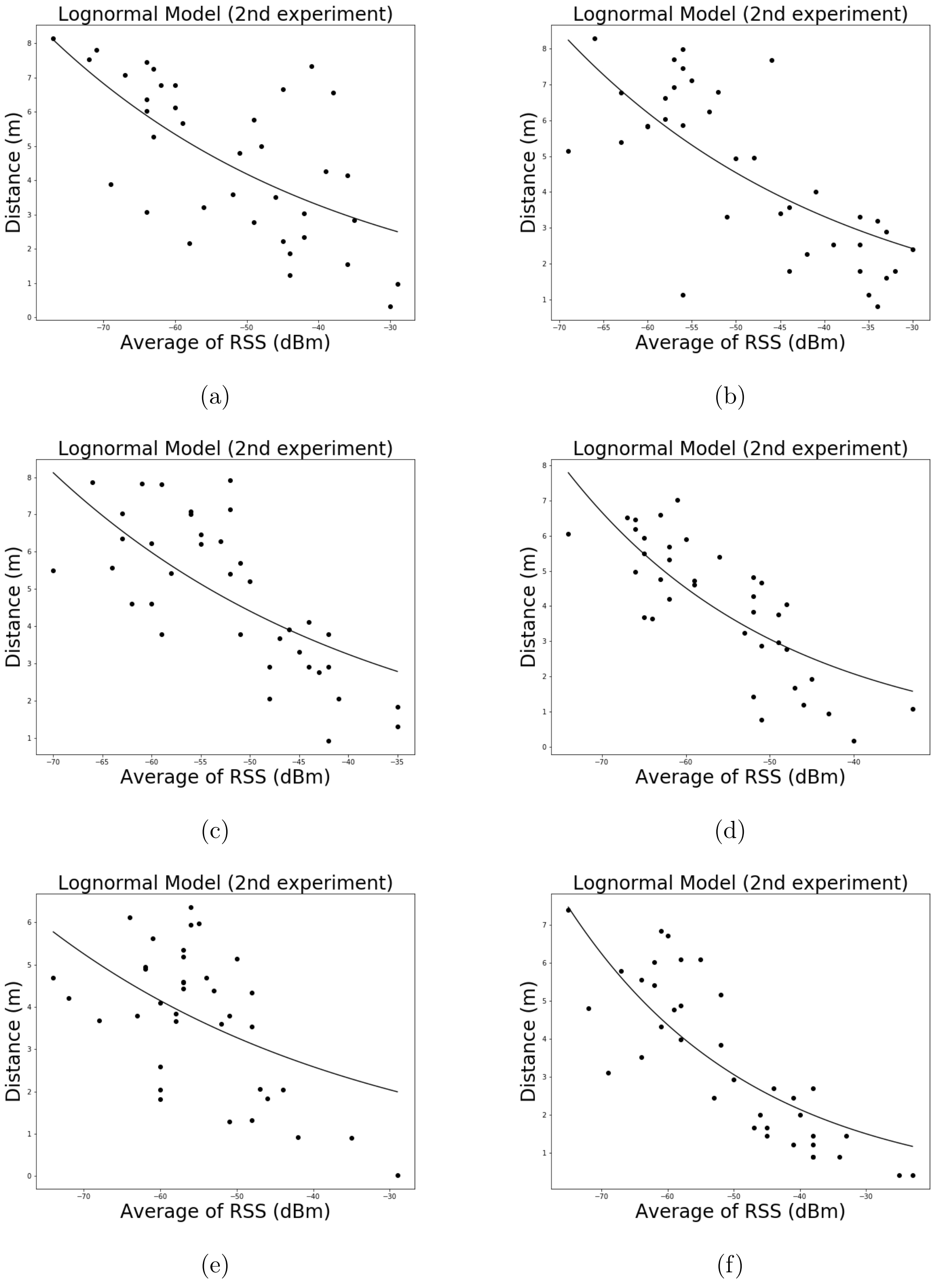


Figure 5.15: Graphs of RSS (dBm) × distances of 6 anchor nodes. The second experiment had much less signal interference which improved methods' accuracy.

Chapter 6

Conclusion and future works

In this research, we showed a comparative study of 9 methods concerning the accuracy, energy consumption, and response time in two different indoor environments. One had a high signal interference while the other had weak interference. The 9 methods were variations of multilateration and fingerprinting. Results showed that the latter outperformed the former, in general. The research also applied a polynomial model and results verified that it surpassed the lognormal path-loss model in terms of accuracy, which is a commonly used model for multilateration. The research reassured the need to carefully select positions of anchor nodes since results showed their impact on the accuracy of methods. It also showed that selectively discarding information from some sensor nodes can lead to better accuracy when compared to using all available ones. That is the reason it is advisable to test subsets of anchor nodes to find one that gives the lowest accuracy (optimal subset). But we found that search for an optimal subset presented as a difficult task and counter-intuitive since discarding anchor nodes with the highest RMSE may not lead to a lower accuracy let alone an optimal subset. So, instead of finding the optimal subset, we can find the local optimal one. Finally, the results reaffirmed that signal interference degrades indoor localization methods.

It was a challenging project which had many unforeseen situations. I had to code many scripts besides implementing the localization methods. However, I have a good background in mathematics, especially in terms of optimization and a pretty knowledge of machine learning which sped up the development of the research. Despite that, I still had to read many papers and a book to understand the whole indoor localization topic and distributed computing. To increase knowledge of mathematics required for that research I also took courses of linear algebra for machine learning and nonlinear optimization.

There were some tasks I wanted to do but were not possible due to time and resource constraints as well as unforeseen situations, such as the pandemic. I wanted to carry out the second experiment in the same place as the first because. However, it became a minor experiment in two rooms of a house. I also wanted to conduct an experiment in an empty room, but unfortunately, it did not happen. I wish I had measure energy using a proper device but, due to the pandemic, it was not possible.

If I have to conduct new experiments in the future I probably would implement other methods with new hardware. I would take courses about adaptive filters, neural networks, and signals and systems to implement Kalmar filter, particle filter, multilateration with

Kalman filter and their variations. I would deploy a robot and brand new hardware used nowadays in corporations and factories such as ultra-wideband (UWB) that has shown promising results.

Bibliography

[1] Raida Al Alawi. Rssi based location estimation in wireless sensors networks. In *2011 17th IEEE International Conference on Networks*, pages 118–122. IEEE, 2011.

[2] Ehab I Al Khatib, Mohammad A Jaradat, Mamoun Abdel-Hafez, and Milad Roigari. Multiple sensor fusion for mobile robot localization and navigation using the extended kalman filter. In *2015 10th international symposium on mechatronics and its applications (ISMA)*, pages 1–5. IEEE, 2015.

[3] Howard Anton and Chris Rorres. *Elementary linear algebra: applications version.* John Wiley & Sons, 2013.

[4] Chaimaa Basri and Ahmed El Khadimi. Survey on indoor localization system and recent advances of wifi fingerprinting technique. In *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*, pages 253–259, Marrakech, Morocco, 2016. IEEE, IEEE.

[5] Amir Beck. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB.* SIAM, 2014.

[6] 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.

[7] Yong Soo Cho, Jaekwon Kim, Won Y Yang, and Chung G Kang. *MIMO-OFDM wireless communications with MATLAB.* John Wiley & Sons, Singapure, 2010.

[8] Juan Chóliz, Miguel Eguizabal, Ángela Hernández-Solana, and Antonio Valdovinos. Comparison of algorithms for uwb indoor location and tracking systems. In *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pages 1–5, Yokohama, Japan, 2011. IEEE, IEEE.

[9] Davide Dardari, Pau Closas, and Petar M Djurić. Indoor tracking: Theory, methods, and technologies. *IEEE Transactions on Vehicular Technology*, 64(4):1263–1278, 2015.

[10] Waltenegus Dargie and Christian Poellabauer. *Fundamentals of wireless sensor networks: theory and practice.* John Wiley & Sons, 2010.

[11] Luigi D'Alfonso, Walter Lucia, Pietro Muraca, and Paolo Pugliese. Mobile robot localization via ekf and ukf: A comparison based on real data. *Robotics and Autonomous Systems*, 74:122–127, 2015.

[12] Shahin Farahani. *ZigBee wireless networks and transceivers.* Newnes, USA, 2011.

[13] Anum Hameed and Hafiza Anisa Ahmed. Survey on indoor positioning applications based on different technologies. In *2018 12th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, pages 1–5. IEEE, 2018.

[14] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, USA, 2009.

[15] Simon Haykin. *Neural Networks and Learning Machines, 3/E.* Pearson Education India, 2010.

[16] Hao Lang, Tiancheng Li, Gabriel Villarrubia, Shudong Sun, and Javier Bajo. An adaptive particle filter for indoor robot localization. In *Ambient Intelligence-Software and Applications*, pages 45–55. Springer, 2015.

[17] William CY Lee et al. *Integrated wireless propagation models.* McGraw-Hill Education, 2015.

[18] Nan Li, Jiabin Chen, Yan Yuan, and Chunlei Song. A fast indoor tracking algorithm based on particle filter and improved fingerprinting. In *2016 35th Chinese Control Conference (CCC)*, pages 5468–5472, Chengdu, China, 2016. IEEE, IEEE.

[19] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.

[20] Dimitrios Lymberopoulos and Jie Liu. The microsoft indoor localization competition: Experiences and lessons learned. *IEEE Signal Processing Magazine*, 34(5):125–140, 2017.

[21] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, and Souvik Sen. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th international conference on information processing in sensor networks*, pages 178–189, Seattle, WA, USA, 2015. ACM, ACM.

[22] Luca Mainetti, Luigi Patrono, and Ilaria Sergi. A survey on indoor positioning systems. In *2014 22nd international conference on software, telecommunications and computer networks (SoftCOM)*, pages 111–120, Split, Croatia, 2014. IEEE, IEEE.

[23] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, New Jersey, USA, 2012.

[24] Nanjing Woxu Wireless Co.,Ltd. Rtls in exhibition, 2020. [Online; accessed 12-May-2020].

[25] Nanjing Woxu Wireless Co.,Ltd. Rtls in mine, 2020. [Online; accessed 12-May-2020].

[26] Nanjing Woxu Wireless Co.,Ltd. Rtls in smart factory, 2020. [Online; accessed 12-May-2020].

[27] Nanjing Woxu Wireless Co.,Ltd. Rtls in warehouse, 2020. [Online; accessed 12-May-2020].

[28] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[29] Yiran Peng, Wentao Fan, Xin Dong, and Xing Zhang. An iterative weighted knn (iw-knn) based indoor localization method in bluetooth low energy (ble) environment. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 794–800, Toulouse, France, 2016. IEEE, IEEE.

[30] Ahmed H Salamah, Mohamed Tamazin, Maha A Sharkas, and Mohamed Khedr. An enhanced wifi indoor localization system based on machine learning. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2016.

[31] scikit-learn. Multi-layer perceptron regressor, 2020. [Online; accessed 06-May-2020].

[32] Sewio Networks, s.r.o. Maximize your efficiency and safety with real-time location system, 2020. [Online; accessed 12-May-2020].

[33] Sewio Networks, s.r.o. Real-time localization system in retail., 2020. [Online; accessed 12-May-2020].

[34] Sewio Networks, s.r.o. Real-time localization system in sports., 2020. [Online; accessed 12-May-2020].

[35] Sewio Networks, s.r.o. Sewio uwb real-time location system (rtls), 2020. [Online; accessed 12-May-2020].

[36] Shuo Shen, Chen Xia, Robert Sprick, Lance C Perez, and Steve Goddard. Comparison of three kalman filters for an indoor passive tracking system. In *2007 IEEE International Conference on Electro/Information Technology*, pages 284–289, Chicago, IL, USA, 2007. IEEE, IEEE.

[37] Gaotao Shi and Keqiu Li. *Signal interference in WiFi and ZigBee networks*. Springer, 2017.

[38] Tilo Strutz. Data fitting and uncertainty. *A practical introduction to weighted least squares and beyond. Vieweg+ Teubner*, 2010.

[39] 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.

[40] Wikipedia contributors. Indoor positioning system, 2020. [Online; accessed 12-May-2020].

[41] Jihoon Yang, Haeyoung Lee, and Klaus Moessner. Multilateration localization based on singular value decomposition for 3d indoor positioning. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2016.

[42] Derek Scott Young. *Handbook of regression methods.* CRC Press, 2018.

[43] 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.

[44] Dakai Zhu, Rami Melhem, and Daniel Mossé. The effects of energy management on reliability in real-time embedded systems. In *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, pages 35–40. IEEE, 2004.

[45] Lingling Zhu, Aolei Yang, Dingbing Wu, and Li Liu. Survey of indoor positioning technologies and systems. In *Life system modeling and simulation*, pages 400–409. Springer, Shanghai, China, 2014.

[46] Zoya Byliskii. Generalized linear regression with regularization, 2015. [Online; accessed 12-May-2020].

Appendix A

Tables of the first experiment

X	Y	AN 1	AN 2	AN 3	AN 4	AN 5	AN 6
2.75	3.25	-54.9	-45.7	-45.7	-59.6	-52.5	-51.5
3.85	3.25	-37.3	-57.9	-49.1	-63.1	-44.4	-43.8
1.65	3.25	-58.8	-55.4	-50.8	-58.3	-46.9	-41.1
4.95	3.25	-47.9	-47.9	-52.7	-54.3	-57.8	-60.2
1.65	0.65	-55.6	-50.6	-51.3	-61.1	-52.6	-45.3
2.75	0.65	-45.4	-50.5	-49.3	-62.0	-49.5	-43.0
3.85	0.65	-51.8	-51.2	-47.1	-65.0	-55.8	-42.9
2.75	5.85	-55.7	-51.8	-52.8	-53.2	-42.7	-50.0
1.65	5.85	-63.4	-45.6	-33.8	-60.8	-50.3	-46.2
3.85	5.85	-55.0	-50.6	-43.4	-54.5	-34.1	-37.5
4.95	7.15	-56.7	-52.7	-42.2	-50.4	-49.7	-39.1
4.95	8.45	-54.8	-46.4	-52.5	-50.7	-40.1	-54.3
3.85	9.75	-60.9	-37.5	-49.4	-46.6	-57.6	-49.8
2.75	9.75	-56.4	-38.8	-46.8	-54.5	-49.2	-49.7
1.65	9.75	-47.5	-49.7	-48.4	-57.1	-58.0	-44.5
3.85	12.35	-62.9	-43.9	-55.8	-57.3	-55.8	-45.6
2.75	12.35	-61.9	-37.2	-48.8	-51.7	-52.8	-49.1
1.65	12.35	-60.6	-32.4	-52.2	-62.0	-54.3	-47.0

Table A.1: Average values of RSS of all anchor nodes (AN) for each test point (x, y).

X	Y	AN 1	AN 2	AN 3	AN 4	AN 5	AN 6
1.1	1.3	-48.8	-49.5	-50.7	-55.9	-48.5	-57.7
1.1	6.5	-57.6	-51.3	-28.4	-74.4	-39.8	-53.9
1.1	9.1	-51.9	-42.8	-49.9	-48.9	-51.4	-54.7
1.1	13.0	-62.2	-31.0	-56.5	-53.4	-49.9	-56.0
2.2	1.3	-64.8	-67.2	-50.1	-58.9	-51.6	-56.7
2.2	3.9	-50.2	-57.0	-43.7	-53.5	-36.1	-53.8
2.2	6.5	-47.4	-49.7	-39.3	-56.6	-38.8	-54.6
2.2	13.0	-58.0	-36.3	-48.0	-52.6	-62.1	-49.8
3.3	1.3	-44.6	-59.4	-55.6	-57.1	-54.1	-48.7
3.3	3.9	-49.0	-57.1	-52.5	-63.3	-46.4	-52.9
3.3	6.5	-55.4	-58.7	-46.5	-51.4	-41.8	-47.6
3.3	13.0	-66.9	-44.1	-52.8	-53.1	-48.3	-40.2
4.4	1.3	-38.3	-50.9	-46.2	-63.6	-56.3	-42.5
4.4	3.9	-53.9	-53.1	-49.7	-50.1	-43.3	-36.6
4.4	5.2	-43.5	-48.6	-43.7	-55.6	-37.1	-38.6
4.4	6.5	-53.0	-40.3	-56.3	-53.9	-36.0	-46.6
4.4	9.1	-49.4	-41.7	-52.9	-52.3	-50.1	-56.6
4.4	13.0	-58.1	-45.1	-45.6	-53.7	-48.7	-58.6
5.5	1.3	-41.3	-50.3	-52.0	-66.5	-52.3	-48.1
5.5	2.6	-49.5	-52.8	-49.5	-58.3	-50.4	-46.5
5.5	3.9	-46.5	-57.3	-48.2	-61.4	-59.6	-53.7
5.5	5.2	-46.3	-46.3	-41.4	-62.7	-41.8	-54.8
5.5	6.5	-57.1	-40.6	-40.5	-48.2	-47.1	-40.5
5.5	7.8	-47.8	-46.5	-38.2	-53.8	-44.8	-44.4
5.5	9.1	-51.9	-39.3	-47.4	-48.8	-51.6	-45.1
5.5	10.4	-58.8	-50.0	-55.9	-35.5	-53.2	-41.4
1.1	10.4	-62.3	-53.0	-41.6	-54.8	-64.6	-53.4
1.1	11.7	-58.3	-53.3	-58.7	-58.7	-61.7	-61.9
1.1	2.6	-54.2	-54.2	-54.0	-58.2	-39.2	-55.8
1.1	3.9	-50.8	-61.6	-49.1	-72.5	-42.9	-49.2
1.1	5.2	-58.6	-58.6	-33.8	-59.9	-39.5	-55.6
2.2	10.4	-54.7	-52.4	-43.6	-54.6	-52.1	-64.5
2.2	11.7	-61.4	-32.2	-69.4	-53.1	-59.2	-47.2
2.2	2.6	-50.3	-57.7	-59.5	-65.8	-38.8	-50.3
2.2	5.2	-62.2	-43.3	-47.2	-67.5	-63.0	-52.0
2.2	9.1	-60.5	-45.2	-54.8	-54.4	-56.4	-55.4
3.3	10.4	-56.5	-37.5	-51.9	-57.1	-52.8	-54.3
3.3	11.7	-59.1	-32.1	-61.0	-55.1	-54.5	-55.1
3.3	2.6	-50.9	-53.6	-55.8	-65.5	-43.1	-49.8
3.3	5.2	-58.1	-55.6	-46.5	-55.3	-64.5	-59.8
3.3	9.1	-52.3	-41.9	-50.4	-54.8	-65.0	-44.4
4.4	10.4	-59.2	-40.1	-41.5	-45.3	-51.9	-48.6
4.4	11.7	-62.8	-39.9	-56.6	-51.2	-52.9	-54.3

Table A.2: Average values of RSS of all anchor nodes (AN) for each training point (x, y).

A.1 Multilateration

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. time**	**Localization Error**
2,75	3,25	1,46830068	7,76160636	0,00305653	4,69013274
3,85	3,25	4,40596575	4,00921821	0,01105714	0,94101552
1,65	3,25	1,9648108	6,22774758	0,00399351	2,99434241
4,95	3,25	2,39397133	7,19515492	0,00357223	4,70080099
1,65	0,65	1,54620834	7,07008761	0,00388122	6,42092654
2,75	0,65	2,68921202	5,98000397	0,00378704	5,3303506
3,85	0,65	1,77843303	6,26230504	0,00448108	5,98242071
2,75	5,85	2,15253528	6,8461981	0,0094769	1,16162591
1,65	5,85	0,53898672	9,30799205	0,00932479	3,63208749
3,85	5,85	2,43838294	6,77727723	0,01266694	1,68893629
4,95	7,15	4,3578916	6,64852395	0,00494552	0,77593208
4,95	8,45	1,94973894	7,57738658	0,01622605	3,12458326
3,85	9,75	2,24382624	9,72671314	0,01599145	1,60634256
2,75	9,75	1,85965172	8,4285383	0,00332332	1,59341805
1,65	9,75	4,29235976	6,42743743	0,0103848	4,24517221
3,85	12,35	1,16018426	9,67517026	0,00900173	3,793392
2,75	12,35	1,65699816	9,63925034	0,00301504	2,92280973
1,65	12,35	1,01518831	9,2884321	0,00810289	3,12668897

Table A.3: Results of lognormal multilateration with NLS.

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. time**	**Localization Error**
2,75	3,25	2,20745715	7,61805954	0,00020742	5,12928046
3,85	3,25	4,36072764	4,2510725	0,00017357	1,16623597
1,65	3,25	4,36451538	6,40499919	0,00017452	3,36634555
4,95	3,25	1,69461853	7,31087655	0,00017834	5,82710829
1,65	0,65	3,39670288	6,93749086	0,00017667	6,77220944
2,75	0,65	3,18441936	6,15624373	0,0001905	6,42790205
3,85	0,65	3,82864689	6,35790706	0,00018001	6,58506332
2,75	5,85	2,62735037	6,89871076	0,00016594	1,49724119
1,65	5,85	2,92834749	8,84251177	0,00017476	2,23462752
3,85	5,85	3,0233008	6,77050618	0,00016689	0,67980278
4,95	7,15	3,95929467	6,97588974	0,00017834	1,31104425
4,95	8,45	1,32793015	7,62933677	0,00017953	1,56251032
3,85	9,75	2,24650576	9,59546102	0,00017357	0,97198658
2,75	9,75	1,63317882	8,4413454	0,00016832	0,63774399
1,65	9,75	3,49568118	6,80144971	0,00017786	2,47792489
3,85	12,35	3,09631518	9,16852763	0,00017571	3,6151227
2,75	12,35	2,04097795	9,55734327	0,00017786	2,89693267
1,65	12,35	2,11286856	9,14753968	0,00017357	2,67101891

Table A.4: Results of lognormal multilateration with LLS.

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. time**	**Localization Error**
2,75	3,25	2,37743165	8,0348139	0,00512266	4,79929695
3,85	3,25	4,56610986	4,07530078	0,01086164	1,09267319
1,65	3,25	4,42424241	4,31241488	0,01288915	2,97071479
4,95	3,25	1,86041391	7,98163249	0,00265312	5,65100772
1,65	0,65	2,01309285	6,51794148	0,00343251	5,87916436
2,75	0,65	4,42324764	5,96261655	0,00585055	5,569888
3,85	0,65	5,36583714	6,71729996	0,00233674	6,25379014
2,75	5,85	1,90314338	7,69738967	0,0036335	2,03224376
1,65	5,85	0,36333717	8,61316256	0,00322151	3,0480434
3,85	5,85	2,26443038	7,06045348	0,00905848	1,9948004
4,95	7,15	4,90376335	6,20146799	0,00541663	0,94965826
4,95	8,45	1,92807437	7,83666566	0,00926137	3,08353913
3,85	9,75	2,04486599	9,64267267	0,01506829	1,80832186
2,75	9,75	1,9741835	8,70913453	0,00822663	1,29818804
1,65	9,75	5,4790111	6,97964534	0,00221682	4,72611796
3,85	12,35	2,12016378	10,02125574	0,00460982	2,90092798
2,75	12,35	1,38272462	9,48419053	0,01050806	3,17526469
1,65	12,35	1,82572723	9,59119003	0,00274086	2,76440093

Table A.5: Results of lognormal weighted multilateration.

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. time**	**Localization Error**
2,75	3,25	4,4664402	6,2508356	0,004076	3,45704806
3,85	3,25	3,87455345	3,79642531	0,00279355	0,54697668
1,65	3,25	4,82738045	4,57320642	0,00781846	3,44189218
4,95	3,25	5,31115351	6,35617067	0,00503159	3,12709579
1,65	0,65	5,3770369	5,03956045	0,00309539	5,7583891
2,75	0,65	4,69736044	4,83262718	0,0036056	4,61373848
3,85	0,65	5,0291783	4,5685964	0,00329852	4,09217047
2,75	5,85	4,11072885	5,36712233	0,00262117	1,44386766
1,65	5,85	0,40326436	8,89943037	0,00725412	3,29444614
3,85	5,85	3,93090674	5,59343413	0,00810266	0,26902035
4,95	7,15	5,46835439	6,01841648	0,00314522	1,2446576
4,95	8,45	2,97677377	6,70142019	0,00324059	2,63650396
3,85	9,75	3,00388143	10,50956774	0,00334311	1,13703992
2,75	9,75	1,73768075	10,24369298	0,00378132	1,12628727
1,65	9,75	5,40569635	5,76276995	0,00476789	5,47752303
3,85	12,35	1,49129688	10,54094837	0,00457406	2,97256593
2,75	12,35	2,47396821	10,75258837	0,01003623	1,62108528
1,65	12,35	0,72698196	11,0588967	0,0035615	1,58710744

Table A.6: Results of polynomial multilateration with NLS.

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. time**	**Localization Error**
2,75	3,25	1,97367551	8,54029026	0,02258015	5,3469478
3,85	3,25	3,71973821	4,45063787	0,00092745	1,2076835
1,65	3,25	4,07607572	5,41909713	0,00097704	3,2543549
4,95	3,25	2,80812814	8,30663068	0,00097823	5,49155069
1,65	0,65	2,89949638	7,11491749	0,00096178	6,58455764
2,75	0,65	2,93293443	6,77001745	0,00088501	6,1227509
3,85	0,65	2,41712535	6,87366873	0,00097752	6,38648434
2,75	5,85	4,46855274	5,93540793	0,00101161	1,72067371
1,65	5,85	1,01164686	7,94650988	0,00102425	2,19154019
3,85	5,85	3,97216588	6,05138425	0,00099921	0,23554218
4,95	7,15	4,08173702	6,45271342	0,00099277	1,11359291
4,95	8,45	4,01398499	7,28250666	0,00097823	1,4963839
3,85	9,75	3,1640375	10,49186205	0,00098896	1,01039786
2,75	9,75	2,5121747	9,86752056	0,00103903	0,26527713
1,65	9,75	3,34584189	7,26621308	0,00093699	3,00750348
3,85	12,35	2,92099218	8,92802379	0,00088358	3,54583935
2,75	12,35	2,71946381	10,52974377	0,00089741	1,82051234
1,65	12,35	1,45508729	10,782431	0,00089574	1,57964032

Table A.7: Results of polynomial multilateration with LLS.

X	Y	$\hat{X}$	$\hat{Y}$	R. time	Localization Error
2,75	3,25	2,12421342	9,40610668	0,00659299	6,18783147
3,85	3,25	-0,3412331	13,35896748	0,00930381	10,94338423
1,65	3,25	5,29695219	4,00173557	0,00416541	3,72362278
4,95	3,25	5,74445294	6,36685279	0,00327849	3,21650848
1,65	0,65	2,66242001	5,64023995	0,00964141	5,09190426
2,75	0,65	6,05868035	5,21248715	0,00277805	5,63592536
3,85	0,65	6,7356341	6,6022151	0,0025866	6,61481283
2,75	5,85	1,65469817	7,16867599	0,01042485	1,71423233
1,65	5,85	0,21980721	7,80476086	0,00330973	2,42209443
3,85	5,85	1,67863556	7,60130419	0,00330663	2,7896039
4,95	7,15	6,30837879	6,30581614	0,00298572	1,59932465
4,95	8,45	0,77450529	6,91518668	0,00929666	4,44864114
3,85	9,75	2,71101051	10,28039573	0,00509071	1,25643014
2,75	9,75	2,36346666	9,723881	0,00776291	0,38741479
1,65	9,75	6,23056472	6,49108252	0,00321913	5,62157596
3,85	12,35	2,89970305	10,15464541	0,00770116	2,39220527
2,75	12,35	3,0904157	10,27907896	0,00752163	2,09871313
1,65	12,35	2,48745121	10,64859206	0,0031004	1,89634214

Table A.8: Results of polynomial weighted multilateration.

A.2 Fingerprinting

X	**Y**	**$\hat{X}$**	**$\hat{Y}$**	**R. time**	**Localization Error**
2,75	3,25	3,3	6,83	0,000470161437988	1,70293863659264
3,85	3,25	4,12	1,95	0,000469207763672	0,641404708432983
1,65	3,25	3,58	3,9	0,000469446182251	1,27769323391806
4,95	3,25	3,58	5,53	0,000474214553833	5,4665711373767
1,65	0,65	2,75	5,85	0,000528573989868	2,65363901086791
2,75	0,65	3,85	2,27	0,000478029251099	1,6786303941011
3,85	0,65	4,67	4,55	0,000475883483887	2,12162673437153
2,75	5,85	2,48	3,58	0,000471115112305	0,707742891168819
1,65	5,85	3,3	8,78	0,00047779083252	4,2573465914816
3,85	5,85	3,03	5,52	0,000468254089355	0,880227243386615
4,95	7,15	3,58	8,45	0,000476598739624	1,82915280936285
4,95	8,45	3,58	6,5	0,000473737716675	2,81440935188895
3,85	9,75	4,12	11,05	0,000452280044556	1,12378823627942
2,75	9,75	3,57	10,4	0,000471830368042	1,77341478509682
1,65	9,75	3,58	4,55	0,000475168228149	4,35142505393348
3,85	12,35	3,3	11,05	0,000471353530884	2,2959311836377
2,75	12,35	3,58	12,03	0,000472545623779	3,0356712601993
1,65	12,35	2,48	12,02	0,000471115112305	0,893196506934504

Table A.9: Results of fingerprinting with KNN.

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. time**	**Localization Error**
2,75	3,25	3,05	7,74	0,0973961353302	4,50422654693714
3,85	3,25	4,34	0,06	0,009112119674683	3,23054249462045
1,65	3,25	2,27	6,05	0,008601665496826	2,87245294105496
4,95	3,25	4,14	6,06	0,003299236297607	2,92368848531063
1,65	0,65	2,63	6,39	0,009617567062378	5,82760782309516
2,75	0,65	3,75	3,94	0,002700805664063	3,43342976012678
3,85	0,65	2,86	5,37	0,002586364746094	4,81924947986295
2,75	5,85	3,04	6,35	0,00525689125061	0,576463161572026
1,65	5,85	2,14	10,06	0,003094434738159	4,23514014549845
3,85	5,85	3,12	6,37	0,002801179885864	0,895075183429919
4,95	7,15	3,34	7,46	0,003719806671143	1,64313692350257
4,95	8,45	3,47	7,49	0,0030517578125	1,76065645299256
3,85	9,75	3,56	12,28	0,002740144729614	2,54272278402335
2,75	9,75	3,41	9,93	0,002907514572144	0,686498208017827
1,65	9,75	3,99	5,50	0,003307580947876	4,84754539236819
3,85	12,35	2,23	9,89	0,00282096862793	2,94194603460984
2,75	12,35	3,04	11,90	0,002960443496704	0,53752314801518
1,65	12,35	2,54	11,63	0,002646207809448	1,13945253614775

Table A.10: Results of fingerprinting with neural networks.

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. time**	**Localization Error**
2,75	3,25	3,4	7,98	0,00058913230896	2,22611769679862
3,85	3,25	4,18	2,06	0,000620126724243	0,563205113613149
1,65	3,25	3,47	3,81	0,000660181045532	1,93961336353408
4,95	3,25	3,52	5,4	0,000661373138428	6,03530446622207
1,65	0,65	3,18	5,05	0,000745534896851	3,1104019032916
2,75	0,65	4,07	2,18	0,000674724578857	2,08086520466848
3,85	0,65	4,86	4,26	0,000670909881592	1,06962610289764
2,75	5,85	3	3,59	0,000665664672852	1,75559106855782
1,65	5,85	3,4	9,15	0,000628232955933	4,05690768936144
3,85	5,85	2,79	5,28	0,000661849975586	0,653681879816168
4,95	7,15	3,09	9,41	0,000667333602905	2,85119273287514
4,95	8,45	3,76	5,86	0,000635862350464	2,49377224300857
3,85	9,75	3,83	11,38	0,000671625137329	0,939148550549911
2,75	9,75	3,78	10,68	0,000684261322021	1,84024454896625
1,65	9,75	3,89	4,1	0,00062370300293	4,75143136328412
3,85	12,35	2,81	10,1	0,000676155090332	2,97060599878207
2,75	12,35	3,62	11,81	0,000630617141724	3,29387310016643
1,65	12,35	2,62	11,68	0,000632047653198	1,31209755734854

Table A.11: Results of weighted fingerprinting.

Appendix B

Tables of the second experiment

X	**Y**	**AN 1**	**AN 2**	**AN 3**	**AN 4**	**AN 5**	**AN 6**
1,2	0,4	-64	-34	-47	-62	-56	-34
1,2	1,2	-59	-35	-57	-59	-56	-37
2	1,2	-59	-32	-43	-69	-63	-37
2,8	1,2	-60	-41	-50	-62	-59	-28
1,2	2	-61	-38	-60	-62	-59	-44
2	2	-64	-38	-53	-64	-59	-58
2,8	2	-69	-34	-66	-63	-55	-48
1,2	2,8	-60	-38	-63	-55	-59	-51
2	2,8	-57	-34	-60	-57	-69	-46
2,8	2,8	-57	-28	-52	-62	-68	-40
-3,22	5,06	-70	-55	-41	-51	-64	-73
-2,3	5,06	-40	-55	-50	-41	-63	-57
-1,38	5,06	-37	-55	-55	-50	-69	-59
-1,38	1,38	-42	-44	-51	-52	-42	-53
-1,38	2,3	-64	-55	-49	-49	-44	-43
-3,22	2,3	-52	-49	-46	-51	-39	-51
-2,3	2,3	-54	-43	-42	-55	-53	-44

Table B.1: Average values of RSS of all anchor nodes (AN) for each test point (x, y).

X	Y	AN 1	AN 2	AN 3	AN 4	AN 5	AN 6
0,8	0,8	-60	-44	-70	-65	-48	-38
1,6	0,8	-60	-33	-53	-65	-48	-23
2,4	0,8	-64	-42	-56	-66	-50	-25
3,2	0,8	-77	-36	-66	-61	-56	-41
0,8	1,6	-59	-36	-52	-52	-52	-33
1,6	1,6	-64	-36	-55	-62	-53	-38
2,4	1,6	-67	-44	-56	-60	-57	-38
3,2	1,6	-71	-35	-59	-67	-55	-38
3,2	2,4	-72	-34	-61	-74	-64	-46
2,4	2,4	-62	-33	-63	-56	-57	-45
1,6	2,4	-64	-30	-60	-63	-57	-47
0,8	2,4	-63	-34	-58	-62	-58	-40
0,8	3,2	-48	-51	-64	-64	-72	-44
1,6	3,2	-49	-39	-63	-52	-62	-53
2,4	3,2	-38	-32	-52	-66	-61	-41
3,2	3,2	-41	-56	-52	-62	-56	-38
0,8	0	-45	-41	-51	-66	-58	-45
1,6	0	-63	-45	-55	-63	-57	-34
-0,92	0,92	-51	-69	-42	-59	-60	-50
-1,84	0,92	-39	-58	-48	-59	-35	-52
-2,76	0,92	-69	-57	-48	-51	-29	-59
-0,92	1,84	-36	-48	-47	-52	-44	-69
-1,84	1,84	-46	-56	-43	-65	-51	-58
-2,76	1,84	-42	-52	-35	-49	-42	-58
-3,68	1,84	-49	-57	-42	-48	-48	-67
-0,92	2,76	-52	-50	-59	-49	-60	-64
-1,84	2,76	-35	-60	-44	-48	-60	-61
-2,76	2,76	-45	-63	-41	-51	-46	-52
-3,68	2,76	-44	-46	-35	-53	-47	-62
-1,84	4,6	-58	-53	-46	-43	-63	-62
-2,76	4,6	-44	-55	-45	-46	-68	-55
-3,68	4,6	-30	-56	-42	-45	-51	-61
-1,84	5,52	-42	-58	-62	-40	-74	-58
-2,76	5,52	-36	-56	-44	-51	-57	-60
-3,68	5,52	-29	-66	-51	-47	-54	-75
-0,92	4,6	-64	-63	-60	-52	-60	-72
-0,92	5,52	-56	-60	-50	-33	-62	-64

Table B.2: Average values of RSS of all anchor nodes (AN) for each training point (x, y).

X	Y	$\hat{X}$	$\hat{Y}$	R. time	Localization Error
1,2	0,4	0,94904648	1,66322989	0,003582	1,28791592
1,2	1,2	0,80619328	2,21988166	0,00297976	1,09327139
2	1,2	1,56691112	0,14336719	0,00305986	1,14194521
2,8	1,2	1,207565	1,84645132	0,004215	1,7186473
1,2	2	1,19585151	2,00741809	0,00419378	0,0084993
2	2	1,35476776	2,09104373	0,0016613	0,65162382
2,8	2	0,70510793	1,33396134	0,00253844	2,19822207
1,2	2,8	0,64418817	3,26697104	0,00350285	0,72593991
2	2,8	1,60375047	3,57586249	0,00135636	0,87119246
2,8	2,8	1,966021	2,7367203	0,00218797	0,83637628
-3,22	5,06	-3,84862181	4,89549848	0,00534058	0,6497893
-2,3	5,06	0,07843559	4,32139455	0,00485182	2,49048065
-1,38	5,06	0,77487357	4,73641415	0,00290418	2,17903371
-1,38	1,38	-0,74572898	2,59692322	0,00367188	1,37229802
-1,38	2,3	-0,27718182	2,65543448	0,00492501	1,15868098
-3,22	2,3	-0,7220032	2,5390597	0,00568318	2,5094098
-2,3	2,3	0,33697282	2,62681083	0,00129437	2,65714715

Table B.3: Results of lognormal multilateration with NLS.

X	Y	$\hat{X}$	$\hat{Y}$	R. Time	Localization Error
1,2	0,4	1,6	1,4	9,51290130615234E-05	1,08
1,2	1,2	1,6	2,2	0,0001540184021	1,08
2	1,2	1,4	1,8	0,000126123428345	0,85
2,8	1,2	1,6	0,8	0,000155687332153	1,26
1,2	2	2,4	1,6	0,000123977661133	1,26
2	2	1,4	1,8	0,000124216079712	0,63
2,8	2	2,6	1,8	0,000124454498291	0,28
1,2	2,8	2	2,2	0,000125408172607	1
2	2,8	1,6	2,6	0,000128269195557	0,45
2,8	2,8	2	2,6	0,000129461288452	0,82
-3,22	5,06	-2,3	4,37	0,00016188621521	1,15
-2,3	5,06	-1,84	4,37	0,000125646591187	0,83
-1,38	5,06	-1,61	4,14	0,000123739242554	0,95
-1,38	1,38	-0,35	1,75	0,000128269195557	1,09
-1,38	2,3	-2,76	2,76	0,000126123428345	1,45
-3,22	2,3	-2,07	1,38	0,000125646591187	1,47
-2,3	2,3	-1,64	2,93	0,000155925750732	0,91

Table B.4: Results of fingerprinting with KNN.

X	**Y**	$\hat{\mathbf{X}}$	$\hat{\mathbf{Y}}$	**R. Time**	**Localization Error**
1,2	0,4	1,64	1,53	0,000202655792236	1,21
1,2	1,2	1,35	2,17	0,000188827514648	0,98
2	1,2	1,66	2,07	0,000231027603149	0,93
2,8	1,2	1,33	0,45	0,000207185745239	1,65
1,2	2	1,97	1,81	0,000290870666504	0,79
2	2	1,23	1,3	0,000345468521118	1,04
2,8	2	2,94	1,21	0,000186681747437	0,8
1,2	2,8	2	2,53	0,000185489654541	0,84
2	2,8	1,59	2,67	0,000220775604248	0,43
2,8	2,8	2,01	2,77	0,000184774398804	0,79
-3,22	5,06	-2,35	4,35	0,000186443328857	1,12
-2,3	5,06	-1,87	4,53	0,000195741653442	0,68
-1,38	5,06	-1,67	4,07	0,00018572807312	1,03
-1,38	1,38	-0,62	1,79	0,000225305557251	0,86
-1,38	2,3	-2,58	2,5	0,000173330307007	1,22
-3,22	2,3	-1,46	1,64	0,000184297561646	1,88
-2,3	2,3	-1,84	2,75	0,000170707702637	0,64

Table B.5: Results of weighted fingerprinting.