Universidade do Minho
Escola de Engenharia

Cristiano Gonçalves Pendão

FastGraph - Unsupervised Location and Mapping in Wireless Networks

Programa Doutoral em Telecomunicações (MAP-tele) das Universidades do Minho, Aveiro e Porto

Outubro de 2018

Universidade do Minho
Escola de Engenharia

Cristiano Gonçalves Pendão

FastGraph - Unsupervised Location and Mapping in Wireless Networks

Tese de Doutoramento
MAP-tele - Doctoral Programme in Telecommunications

Universidade do Minho

universidade de aveiro

U.PORTO

Trabalho efectuado sob a orientação de
Professor Doutor Adriano Moreira

Outubro de 2018

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, _____

Full name: _____

Signature: _____

Acknowledgment

I would like to express my sincere gratitude to my supervisor Professor Adriano Moreira for his valuable advices, comments and suggestions provided throughout this journey. I would like to thank him for giving me the opportunity to work with him along these years, and for all that I have learned from him.

I am grateful to my family, especially to the love of my life, Catarina, for her emotional support, constant motivation, and for always making me believe that I can achieve my goals. I dedicate this work to her.

I also thank Ivo Silva for the long talks, support and valuable help in the preparation of the experimental setups.

I would like to acknowledge the Centro Algoritmi and PIEP (Pólo de Inovação em Engenharia de Polímeros), for hosting my research work.

Finally, I would like acknowledge the committee of the MAP-Tele Doctoral Program and FCT (Fundação para a Ciência e Tecnologia) for the provided PhD grant[1], that allowed this research work.

[1]PD/BD/105865/2014

Abstract

FastGraph - Unsupervised Location and Mapping in Wireless Networks:

Wi-Fi Based Indoor Positioning solutions normally require complex and time-consuming deployment processes or have limited accuracy. Fingerprinting Matching is one of most used techniques for indoor positioning, which relies on a radio map that is normally created offline in a calibration phase by manual site survey. Wi-Fi Fingerprint can be used to locate regular mobile devices, such as smartphones, using only software and can be used in any indoor environment without being necessary to deploy additional infrastructure, relying only on the existing Wi-Fi infrastructure. However, in most cases the manual site survey is unpractical and involves a significant effort, even for small scale spaces. Moreover, due to the dynamic nature of radio environments, to maintain the system performance, the site survey has to be repeated often to keep the radio map updated. This process is not feasible in large spaces, and compromises the scalability of this type of approach.

Solutions have been proposed to reduce the calibration effort, using collaborative site survey to create and maintain the radio maps, or by using Model-Based methods to approximate a radio map. However, the reduced calibration effort usually implies a lower positioning accuracy and higher computational requirements.

In this context, FastGraph is proposed as a new solution able to provide unsupervised positioning using different devices, such as smartphones or autonomous machines, while automatically creating and maintaining a Radio Map. A 3D Force-Directed Graph is used to rapidly model the radio environment. The 3D Graph is iteratively constructed with data collaboratively collected by several devices. Orientation and motion information, obtained from different sensors, can be used to improve the Graph constrains.

FastGraph is able to operate shortly after its deployment, without previous knowledge about the environment. The proposed solution uses a novel algorithm to automatically provide location while simultaneously updating the radio map; and estimate the position of the Access Points (APs) and location-specific radio propagation parameters. In addition, FastGraph does not rely on expensive hardware or requires high computational effort.

The FastGraph approach may be used in different contexts. In addition to the indoor

positioning, the radio maps created by FastGraph include supplementary information that can be used to automatically map the interference in Wi-Fi networks and even to automatically map the physical space.

The described solution was deployed and evaluated in two very distinct real world spaces, an industrial environment and a regular office building. The experiments, in these two spaces, evaluated the several aspects of FastGraph, and considered scenarios where only Wi-Fi data is available and when the Wi-Fi can be combined with data from other sensors. The results suggest that the proposed solution has potential to provide interference information in wireless networks and provide positioning in different indoor scenarios, from regular buildings, to autonomous vehicles in industrial environments, with the possibility of being also extended to outdoor spaces using data from cellular networks, especially considering the characteristics of the upcoming 5G networks.

Resumo

FastGraph - Localização e Mapeamento Automáticos em Redes sem Fios:

Os sistemas de posicionamento *indoor* baseados em Wi-Fi normalmente implicam processos de implementação complexos e longos, ou são limitados em termos de exactidão. *Fingerprinting Matching* é uma das técnicas mais utilizadas para posicionamento *indoor*, utilizando mapas de rádio normalmente criados *offline*, numa fase de calibração, através de mapeamento manual do ambiente rádio. O Wi-Fi fingerprinting dispensa a instalação de uma infra-estrutura (usa a infra-estrutura Wi-Fi existente) e pode ser utilizado para localizar dispositivos móveis comuns só com recurso a software. No entanto, na maioria dos casos o mapeamento rádio manual não é prático e envolve um esforço significativo, mesmo em espaços pequenos. Além disso, devido à natureza dinâmica dos ambientes rádio, para manter a desempenho do sistema, a fase de calibração tem de ser repetida frequentemente para manter o mapa de rádio atualizado. Por estes motivos, este processo não é praticável em espaços de grandes dimensões, comprometendo a escalabilidade deste tipo de abordagem.

Por este motivo têm vindo a ser propostas soluções para reduzir o esforço de calibração, recorrendo a mapeamento colaborativo do ambiente rádio para criar e manter os mapas de rádio, ou utilizando métodos baseados em modelos para obter um mapa de rádio aproximado. Contudo, a redução do esforço de calibração normalmente traduz-se numa redução da precisão no posicionamento, ou no aumento dos requisitos computacionais.

Neste contexto é proposto o FastGraph, um novo sistema capaz de fornecer automaticamente posicionamento utilizando diferentes dispositivos, como por exemplo smartphones ou veículos autónomos, e capaz de em simultâneo criar e manter um mapa de rádio. É utilizado um Grafo 3D (*3D Force-Directed Graph*) para rapidamente modular o ambiente de rádio. O Grafo 3D é criado iterativamente com os dados recolhidos colaborativamente por múltiplos dispositivos. Dados de orientação e movimento, obtidos através de diferentes sensores, podem ser usados para melhorar as restrições espaciais do Grafo.

O sistema FastGraph é capaz de operar rapidamente após ser instalado, sem necessitar de qualquer informação prévia sobre o ambiente rádio ou espaço. A solução proposta implementa um algoritmo original para fornecer posicionamento enquanto automaticamente mantêm o mapa

de rádio atualizado; estima a posição dos Pontos de Acesso Wi-Fi e parâmetros de propagação específicos para cada local do espaço. Além disso, a solução FastGraph não se baseia em *hardware* dispendioso e não implica elevado esforço computacional.

A solução FastGraph pode ser utilizada em diferentes contextos. Além do posicionamento *indoor*, os mapas de rádio criados pelo FastGraph incluem informação suplementar que pode ser utilizada para mapear automaticamente a interferência em redes Wi-Fi, e até para mapear automaticamente o espaço físico.

A solução descrita foi instalada e testada em dois espaços reais com características muito distintas, um espaço industrial e um edifício de escritórios. As experiências nestes dois espaços avaliaram os diferentes aspetos da solução e consideraram cenários onde apenas é possível utilizar dados Wi-Fi, e cenários onde os dados Wi-Fi podem ser combinados com dados de outros sensores. Os resultados sugerem que a solução proposta tem potencial para fornecer informação relativa à interferência em redes sem fios assim como posicionamento em diferentes ambientes *indoor*, com aplicações desde do posicionamento em edifícios comuns à navegação de veículos autónomos em ambientes industriais, podendo também ser possível a extensão da solução para ambientes exteriores utilizando dados das redes celulares, em especial considerando as características das novas redes 5G.

Contents

List of Figures

List of Tables

Acronyms

AHS Active Heading Stabilization.

AoA Angle of Arrival.

AP Access Point.

AR Augmented Reality.

BPSK Binary Phase-Shift Keying.

BSSID Basic Service Set Identifier.

CCI Co-Channel Interference.

cF Combined Force.

CL Current Length.

CSV Comma-Separated Values.

DS Direct Sequence.

DSSS Direct Sequence Spread Spectrum.

GA Genetic Algorithm.

GD Gradient Descent.

GLONASS GLObalnaja NAvigatsionnaja Sputnikovaja Sistema.

GNSS Global Navigation Satellite System.

GP-LVM Gaussian Process Latent Variable Model.

GPS Global Positioning System.

IMU Inertial Measurements Unit.

IoT Internet of Things.

IPIN Indoor Positioning and Indoor Navigation.

IR Infrared Light.

ISM Industrial, Scientific, and Medical.

kNN k-Nearest-Neighbour.

LDPL Log-Distance Path Loss Model.

LoS Line of Sight.

MAC Media Access Control.

MDS Multi-Dimensional Scaling.

MIMO Multiple-Input Multiple-Output.

MLP MultiLayer Perceptron.

MTU Moving Testing Unit.

MWM Multi-Wall Model.

NL Natural Length.

NLoS Non Line of Sight.

OFDM Orthogonal Frequency Division Multiplexing.

QPSK Quadrature Phase-Shift Keying.

RF Radio Frequency.

RFID Radio-Frequency IDentification.

RPi Raspberry Pi.

RSA Received Signal Attributes.

RSS Received Signal Strength.

RToF Roundtrip Time of Flight.

SIR Signal-to-Interference Ratio.

SLAM Simultaneous Localization and Mapping.

SMP Smallest M-vertex Polygon.

SNIR Signal-to-Noise-Plus-Interference Ratio.

SNN Shared Nearest Neighbor.

SNR Signal-to-Noise Ratio.

SSID Service Set Identifier.

SVM Support Vector Machine.

TDoA Time Difference of Arrival.

ToA Time of Arrival.

TTU Tripod Testing Unit.

WLAN Wireless Local Area Network.

Chapter 1

Introduction

The research presented in this thesis is focus on the creation of a method to automatically create and maintain radio maps, that can be used in different contexts, such as indoor positioning or radio environment analysis. A radio map can be seen as a database containing fingerprints collected at several locations of a space. A fingerprint or radio signature consists in a sample collected at a specific position containing information about the APs detected. Each fingerprint includes for each AP detected, the AP identification (Media Access Control (MAC) Address) and the Received Signal Strength (RSS), between other information.

This chapter provides the research context and describes the motivation. The main objectives and most relevant contributions are also detailed, and the main hypothesis is presented. Finally, is outlined the structure of this thesis.

1 Research Context and Motivation

Wireless Local Area Network (WLAN) and cellular networks infrastructures are spread across most areas, providing constant access to digital services to millions of users. The widespread of WLANs, leaded to the research of different Wi-Fi based positioning systems, mainly in indoor environments, where Global Navigation Satellite System (GNSS) systems can't be used or are unreliable.

Wi-Fi based positioning in indoor spaces has associated several challenges, and different approach have been proposed in the recent years. These approaches normally rely on complex and time consuming deployment processes and requirements, or impose significant limitations in scalability, space heterogeneity and location accuracy. However, areas such as Internet of Things (IoT) [2], Augmented Reality (AR) [3], monitoring of patients with degenerative dementia [4] or autonomous vehicles and drones indoor navigation [5], are only a few examples of the increasing number of applications requiring an accurate and easily scalable indoor positioning system.

Wi-Fi Fingerprinting Matching is a well known and one of most used techniques for indoor positioning, which relies on a previous site survey. Other approaches rely on propagation models or on geometric relations between the transmitters and the receivers. Some Simultaneous Localization and Mapping (SLAM) solutions explore the techniques used in robotics, to provide positioning while mapping the space, taking advantage from sensors' data, such as orientation and displacement. This type of data is also frequently combined with the Wi-Fi RSS measurements to increase the positioning accuracy.

In addition, the technologies that are expected to integrate the upcoming 5G cellular standard provide interesting prospects for positioning. The 5G characteristics such as the small cells, the millimeter wave, massive Multiple-Input Multiple-Output (MIMO) [6], and the device-to-device communication, may contribute to new positioning systems as well as benefit from it [7]. As example, an accurate awareness of the mobile stations positions could be used by cellular networks to significantly increase the spectrum efficiency, as the frequency slots may be re-used for the mobile stations that are distant from each other [8].

Another relevant topic related to WLAN networks is the interference, that degrades the quality of service. In contrast to cellular networks that operate in regulated spectrum bands, and where the deployment is controlled and planned, the deployment of Wireless LAN networks infrastructures is unregulated and uncontrolled. The reason is the unlicensed nature of the bands used by Wi-Fi. This uncontrolled and most of the times unplanned deployment leads

to competition for resources among a large number of access points, hotspots, and other types of devices, such as laptops and smartphones, which transmit on the same unlicensed band. All of these devices contribute to high levels of interference, which dramatically decreases the throughputs [9]. The wireless interference is in fact one of the major cause of degradation of capacity in Wi-Fi wireless networks, and has been well-documented in literature [9–12].

The energy consumption in large WLAN networks is also a problem. Modern access points already include dynamic transmit-power control. In addition, strategies to intelligent optimizing of the energy consumption, independent from manufactures, are also being investigated [13]. However, the problem begins with the inefficient planing and deployment of Wi-Fi access points, which in addition to interference, also results in unnecessary energy consumption. These problems are frequent, even in large institutions such as universities campus, where the deployment of Wi-Fi is expected to be planned. Methods to easily identify the areas with hight levels of interference are therefore essential to more efficient Wi-Fi networks, specially in large scale and Wi-Fi dense environments.

2 Proposed Hypothesis

This research work intend to assess if it is possible to automatically create and maintain radio maps, using an approach based on organic Force-Directed 3D Graphs.

The main idea is that a 3D Graph can be used to represent a radio environment, where the access points and the fingerprints (samples unsupervisedly collected) are the nodes of the Graph, and the edges represent the communications channels between the devices collecting samples and the access points, and can be described by a propagation model. In addition, the samples can be connected by edges, established based on motion information. Using a Force-Directed approach, the Graph is able to evolve, and the position of the samples and the access points can be automatically estimated. With more samples being added, the Graph can improve, and can dynamically adjust to changes in the radio environment, automatically maintaining a radio map updated.

The main hypothesis formulated for this research was therefore the following: **A dynamic 3D Graph that is built iteratively and adjusted by a Force-Directed approach, that can be used to model radio environments and to automatically create and maintain Radio Maps, and that can be used in different contexts.**

3 Research Strategy and Objectives

The research work presented here followed a deductive approach and the engineering design process. As expected, the research and development was an iterative process, with recursive testing and modification. Controlled tests, using synthetic data, allowed to find and isolate the problems, by making sure that only one factor or condition was changed at each time. A radio environment simulator, that was developed and will be detailed on this document, was essential during this process.

Although the hypothesis presented may be valid for different radio technologies, this research focused mainly in Wi-Fi, for being an ubiquitous and well establish technology in the indoor positioning field. The proposed hypothesis was validated by experimentation using real world data. A prototype was developed to deploy the solution in real world scenarios. The experiments were conducted in two very distinct spaces, in order to validate the solution in environments with different characteristics.

Main Research Objectives:

- To assess if it is possible to create an easily deployable, and self maintained method, to provide positioning and to automatically create and maintain radio maps in heterogeneous indoor spaces.

- Discuss and evaluate how the radio maps created by this method can be used in other applications, especially the analysis of interference in Wi-Fi networks.

- Discuss possible applications for the developed solution, and how it can be extended to outdoor environments and to upcoming technologies such as 5G networks.

4 Contributions

The most relevant contribution of this research is the definition of a new method for unsupervised indoor positioning and radio mapping.

In contrast with other approaches, the proposed solution don't rely on manual site survey or calibration to create radio maps, or require previous knowledge about the access points positions. Moreover, no assumptions are made about the propagation characteristics being

uniform for a given space, or that propagation parameters for an access point are equal in all areas of the space.

The contributions in the positioning field may be valid to other wireless technologies, such as Bluetooth or cellular networks, and possibly extended to outdoor environments, contributing for a seamless indoor and outdoor positioning system.

The radio maps created by the proposed method may also be useful for analysis and planing of Wi-Fi networks, contributing to reduce interference, improving the communication's quality and the energy efficiency. In addition, this research can contribute to positioning methods for 5G cellular networks, which can be used not only by users, but also by the network to improve aspects such as spectrum efficiency or to analyze how the mobile networks services are been used in outdoor and indoor environments.

This research work also resulted in scientific publications already published and others that are being prepared to be published.

Publications:

- Cristiano Pendão, Adriano Moreira, "FastGraph - Unsupervised Location and Mapping". *Journal Paper describing the FastGraph full solution and including the enhanced positioning results with motion information.* Submitted to IEEE Transactions on Mobile Computing (TMC) Journal

- Cristiano Pendão, Adriano Moreira, *Regular Paper covering the interference mapping and exploring the relation between the interference and the positioning performance.* (In Progress)

- Cristiano Pendão, Adriano Moreira, "FastGraph - Organic 3D Graph for Unsupervised Location and Mapping" 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nates, 2018. (Accepted, Presented and Waiting to be Published in IEEE Xplore)

- A. Moreira, I. Silva, F. Meneses, M. J. Nicolau, C. Pendão and J. Torres-Sospedra, "Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning," 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, 2017, pp. 1-8. doi: 10.1109/IPIN.2017.8115914

- S. Conceição, C. Pendão, A. Moreira and M. Ricardo, "Evaluation of medium access and a positioning system in wireless underground sensor networks", 2016 Wireless Days (WD), Toulouse, 2016, pp. 1-6. doi: 10.1109/WD.2016.7461482

5 Thesis Structure

This chapter have defined the context and the research scope of this PhD thesis. The motivation and the research objectives were presented.

The next chapter reviews the state of the art related to Wi-Fi networks and how they have been used to provide indoor positioning. Relevant topics of related works are discussed, including the algorithms and technologies explored for positioning in indoor and outdoor spaces.

Chapter three, describes the 3D RF Environment Simulator, that was designed and implemented to generate controlled synthetic data. This data was essential to develop and test different components of the FastGraph algorithm.

The fourth chapter, presents the FastGraph. The fundamental principles of the proposed solution are explained, detailing the Graph based approach and the several challenges that had to be overcome.

Chapter five, describe the two real world spaces, an industrial building and a regular University building, where the FastGraph prototype was deployed and evaluated. In this chapter are also described the components and devices developed for the experiments, and the setup to obtain ground truth data.

In chapter six the results obtained from the real world experiments are presented and discussed. In addition, these results are compared with state of the art solutions.

Chapter seven, describes how the FastGraph approach can be used and extended to other applications, especially for additional automatic mapping features.

Chapter eight, concludes this thesis, resuming the key aspects and providing topics for future work. Some broader indicators for further research are also provided.

Chapter 2

State of the Art Review

This chapter presents a literature survey of the state of the art research exploring Wi-Fi networks to create radio maps and to provide indoor positioning. The advantages and challenges related to the different approaches are discussed.

The first section introduces the main characteristics of WLANs, regarding the radio spectrum usage, the deployment types and coverage.

Second section overviews the different positioning systems and the different type of location that each system provides.

Section three focus on research related to Wi-Fi based positioning, discussing the different approaches and relevant projects in this field.

1 WLAN Radio Channel

Wireless LAN based on the IEEE 802.11 operate in 2.4 GHz and 5 GHz bands that are allocated on unlicensed Industrial, Scientific, and Medical (ISM). The lower frequencies have limited bandwidth resulting in lower transmission rates, but the range is superior to the higher frequencies that are more attenuated by objects [14].

1.1 Wi-Fi Radio Spectrum

Wi-Fi connections can be dropped or the throughput lowered by having other devices operating in the same band in the same area. Many 2.4 GHz (802.11b and 802.11g) Access Points are configured by default to operate in the same channel on initial startup, contributing to congestion on certain channels. Other devices in the 2.4 GHz band, such as ZigBee and Bluetooth devices or even microwave ovens, also contribute to the interference.

In addition, an excessive number of APs in the area, especially on neighboring channels, can prevent access and interfere with other APs. Overlapping channels in the 802.11g/b spectrum also decrease the Signal-to-Noise Ratio (SNR). This is becoming a problem in high-density areas, such as large apartment complexes or office buildings with many Wi-Fi APs, and also when municipalities or other large entities (such as universities) seek to provide coverage in a large area.

The 802.11a physical layer operates in the 5GHz band, avoiding the interference from the devices operating in the 2.4 GHz, but the different frequency range used leads to incompatibility with the 802.11b or 802.11g standards [14].

Spectrum assignments and operational limitations are not consistent worldwide. Australia and Europe allow for two additional channels beyond the 11 permitted in the United States for the 2.4 GHz band (1 to 13), while Japan has three more (1 to 14). The 802.11a physical layer uses the 5 GHz band, which, for much of the world, offers at least 23 non-overlapping channels rather than the 2.4 GHz ISM frequency band, where adjacent channels overlap (Figure 2.1), leading to channel interference.

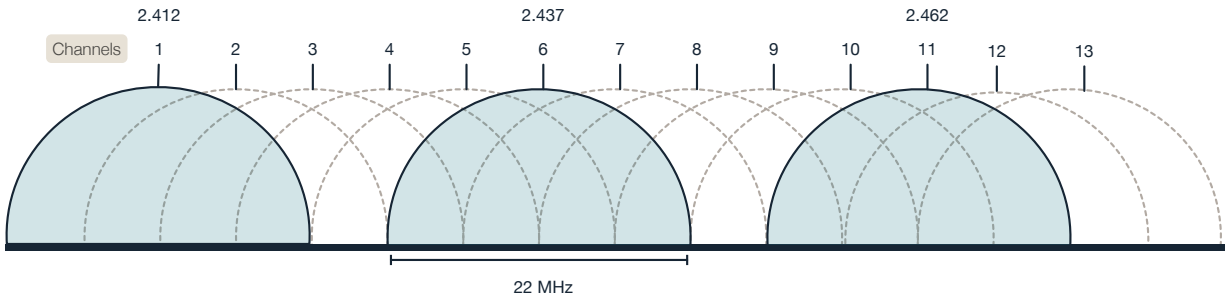

Figure 2.1: Wi-Fi Channels 2.4 GHz (Europe)

The two technologies (802.11b/g and 802.11a) are complementary and will continue to be used in enterprise environments [14]. When implementing these technologies it is necessary to decide between deploying only one type or a combination of both. The 802.11a technology cannot be deployed for an existing 802.11b network and obtain the same coverage in the same areas as with the previous technology. The difference between the coverage of the two bands do not allow this type of approach [14].

1.1.1 Spectrum Usage Techniques

Direct Sequence Spread Spectrum (DSSS) provides redundancy to the Radio Frequency (RF) signal, allowing better chances for a successful pack reception when there is noise or interference on the channel. The modulation used are Binary Phase-Shift Keying (BPSK) and Quadrature Phase-Shift Keying (QPSK).

IEEE 802.11b uses Direct Sequence (DS) Channels, with 14 channels defined. A DS channel has a bandwidth of 22 MHz, and the channels are separated by only 5 MHz, which can results in interference between the signals from neighboring channels. In planned deployments the APs are usually installed using the same approach of cellular deployments, allocating the adjacent APs in non overlapping channels.

IEEE 802.11a uses Orthogonal Frequency Division Multiplexing (OFDM). The OFDM multi-carrier provides high spectral efficiency by allowing sub-channels to overlap. In addition, in OFDM the modulation technique used is more efficient than the spread spectrum used in the 802.11b [14].

The encoding techniques on the wireless link provide better data rates by allowing data to be less affected by noise. However, the minimum required reliable data rate is influenced by the number of APs, the power setting, the antenna gain, and the location. Also, when there are multiple clients, the client with lower rate takes longer to transmit and this affects the data throughput of the higher rate clients [14].

1.2 Different Deployment Types of Overlapping WLAN Coverage

The overlapping in WLAN coverage depends on the services provided by the network. Wireless networks can be deployed for different applications (location management, voice, data-only networks, or a combination). The difference is the APs distribution pattern, and the RF overlap level. The update of a WLAN deployment to support additional services may imply additional site analysis and change the positions of the existing APs.

For a data-only deployment the overlap necessary is low and is defined by the required WLAN data rate. Voice deployments require more overlap than data-only, therefore the APs are closer, ensuring better alternative APs for voice clients. Cisco recommends that two APs on non overlapping channels to be available at all times, for redundancy and load balancing purposes. Is important to obtain a reasonably high energy density within the cell but also a low noise background, being the 35–50 mW range a good power baseline for the AP, but usually requiring around 15% more APs than at 100 mW [14].

Location-based services deployments requires excellent cell coverage and optimal location of APs, and this is the most complex deployment. This type of deployment use the WLAN infrastructure to track thousands of devices simultaneously, using for example Wi-Fi tags. The AP separation estimation must take into account the objects that affect RF coverage, since the coverage results vary depending on the RF environment. Space surveys help identifying and characterizing problematic areas and potential sources of interference, for example existing WLANs, or non-802.11 interference from other sources. Also, surveys after the deployment should be considered to ensure that the coverage model was well planned [14].

2 Positioning Systems

Current positioning systems provide a wide range of services to the general population, to the industry and to the governmental organizations.

In outdoor environments, most positioning services are based on GNSS, with a dedicated satellites infrastructure called constellation. Examples of GNSS systems are the United States Global Positioning System (GPS) and the Russian GLObalnaja NAvigatsionnaja Sputnikovaja Sistema (GLONASS) and the European Galileo.

The general population uses GNSS services for daily routines and tasks, such as car travel navigation. Professionals such as firefighters and rescuers rely on GNSS for their daily oper-

ations, such as locating emergency scenarios. In large urban areas, the GNSS coverage and accuracy are affected by the city geometry, with tall buildings blocking the GPS signals. This effect is commonly known as urban canyons. In addition severe weather conditions also affect the GNSS systems, attenuating the signals.

In indoor spaces, GNSS is very unreliable and most of the times unusable due to the week signals, attenuated by the walls and ceiling and also due to multi-path. This lead to the research and development of approaches for indoor spaces based on alternative technologies. Over the past decade the research in indoor positioning enabled a new set of applications, ranging from location-based services to indoor robot navigation. For each specific application, different requirements prompted the development of different techniques, ranging from high-accuracy laser based approaches to Radio-Frequency IDentification (RFID) solutions [15–17].

Indoor positioning technologies can be based on dedicated infrastructures. Some of this type of systems offers high accuracy, but the required dedicated hardware and infrastructure increases the costs, limiting large scale applications [18, 19]. Examples of technologies explored in this context are Infrared Light (IR) [20], RFID [16], Acoustic Waves [21], Visual Analysis [22], and Bluetooth [23, 24]. The solutions based on these technologies are usually based on proximity algorithms, that provide symbolic relative location, normally relying on a grid with a large number of antennas with known and fixed positions. The mobile device position can be determined by a single antenna, or considering the strongest signal when multiple antennas are detected. RFID and IR are perhaps the most common examples of systems that are based on this approach [25]. These systems have been used for example to track assets and inventory management [25] or for security purposes in commercial spaces.

In recent years, radio signals from existing wireless infrastructures has been also explored for indoor positioning. For example, a system can leverage from the existing infrastructures of mobile cellular networks to estimate the position for outdoor mobile clients. However, the accuracy using cellular signals is generally low (50–200 m), also the accuracy is higher in densely covered spaces such as urban areas and much lower in rural environments [25].

The ubiquity of Wi-Fi networks in indoor spaces attracted the attention of many research teams around the world, seeking to explore the existing Wi-Fi infrastructures to provide low-cost indoor location services [19, 25–27]. The majority of the previous works on this field explore the RSS of Wi-Fi received from multiple APs, as a metric for location determination [28]. Positioning based on Wi-Fi signals for indoor spaces has a better accuracy than systems based on cellular signals, due to the lower range of coverage and higher density of APs in indoor spaces. Different approaches with different requirements and accuracy have been developed.

Current positioning systems provide different types of location information, and each system is often designed to be used for outdoor or for indoor spaces. A system that can provide positioning seamlessly for indoor and outdoor spaces with high performance is still not available. This type of solution is important for example to provide seamless indoor and outdoor navigation and location for emergency response teams.

The demand for this type of solution is therefore high, and the possible applications are vast and will increase with the upcoming technologies. Internet of Things (IoT), Augmented Reality (AR), and Autonomous Navigation are only a few examples of the areas being developed that can benefit from a solution able to provide positioning with high accuracy in indoors and outdoors.

2.1 Location Types and Performance Metrics

Positioning systems can provide different types of location, depending on the application requirements and objectives. Four types of location are commonly established: physical, symbolic, absolute or relative [15, 25].

Physical location identify a point on a 2D or 3D referential and is expressed by a set of coordinates, as it is in the GPS system. Symbolic location represents a location as natural language, such as: Building A, floor five, room two. The absolute location is based on a reference grid for all located objects, and the relative location information is typically based on the proximity to known references, for example base stations [25].

The type of location that a system provide also define the system requirements, such as precision, accuracy, scalability, and robustness. Other important parameters for a positioning system are the complexity and cost [15, 25]. The accuracy can be represented as the average Euclidean distance between the estimated location and the true location, and is considered the most important requirement for a position system. Other frequently used metrics include the 75th percentile of the error, the max error, metrics that consider correct detection rate of the building and floor (e.g. Indoor Positioning and Indoor Navigation (IPIN) conference competition), or even a metric that is not based on the Euclidean distance [1]. The precision measures how consistently the positioning system works. The accuracy and precision are related. If the system requirements allow less accuracy it is possible to trade it for more precision [15].

The complexity is related to the hardware, software and operation complexity required. A system is robust if it can operate even when some of the requirements are unavailable, for example if a satellite of GPS system fails, the system has other as replacement. The scalability is related to the scope, if a system can be used in large scale scenarios and applications, and

includes for example the number of concurrently clients or terminals that can obtain positioning information simultaneously. The cost includes installation, deployment and also maintenance and operational costs, related to infrastructures and hardware, time, space, weight, and energy [15, 25].

3 Wi-Fi based Positioning

As mentioned before, the ubiquitousness of WLANs has been opening the opportunity for many different positioning systems, especially for indoor environments. Generally, we can separate Wi-Fi positioning techniques in two main groups: Wi-Fi Fingerprinting Matching and Model-Based/Geometric. In order to achieve low positioning errors, several solutions propose to combine odometry and orientation information with Wi-Fi RSS measurements. This type of data is also explored in SLAM approaches.

3.1 Received Signal Strength (RSS) Based

In RSS-based approaches a position is estimated based on RSS measurements included in samples obtained by scanning the available Access Points in the environment. In this context, in a single $sample_i$ can be described as:

$$sample_i = (deviceid, t, rs : ((ap_1, rss_1, ch_1), ..., (ap_n, rss_n, ch_n)), p : (x, y, z), he) \tag{2.1}$$

where, *deviceid* is the identification of the device that collects the sample (for example the MAC address), t is the timestamp when the sample was collected, and rs is a list containing information about the detected Access Points, including for each AP detected, the identification (usually the AP MAC address), the measured RSS at the position where the sample was collected, and other information such as the AP channel. In addition, a sample can also include supplementary information such as the position (p) where it has been collected and the device current orientation (he).

Different methods can be applied such as Approximate Perception, Trilateration or Fingerprinting Matching.

3.1.1 Approximate Perception

Approximate Perception is a relative simple approach where the position of the AP with the strongest signal is used as the user position [25]. In order to improve accuracy additional methods such as Antenna cluster, and Cell-ID have been explored [29]. Since this method relies on the AP position and coverage area, provides low accuracy positioning (around 100 meters).

3.1.2 Trilateration

Trilateration is a simple and well known method to estimate a position based on the distance to transmitters with known positions. Trilateration is the fundamental principle of GNSS systems to estimate position, using the spatial distances to the satellites, which are estimated from the received signals.

To trilaterate a device, at least three transmitters are needed. Trilateration is also frequently used to obtain an estimated position of a mobile station in cellular networks (Figure 2.2).

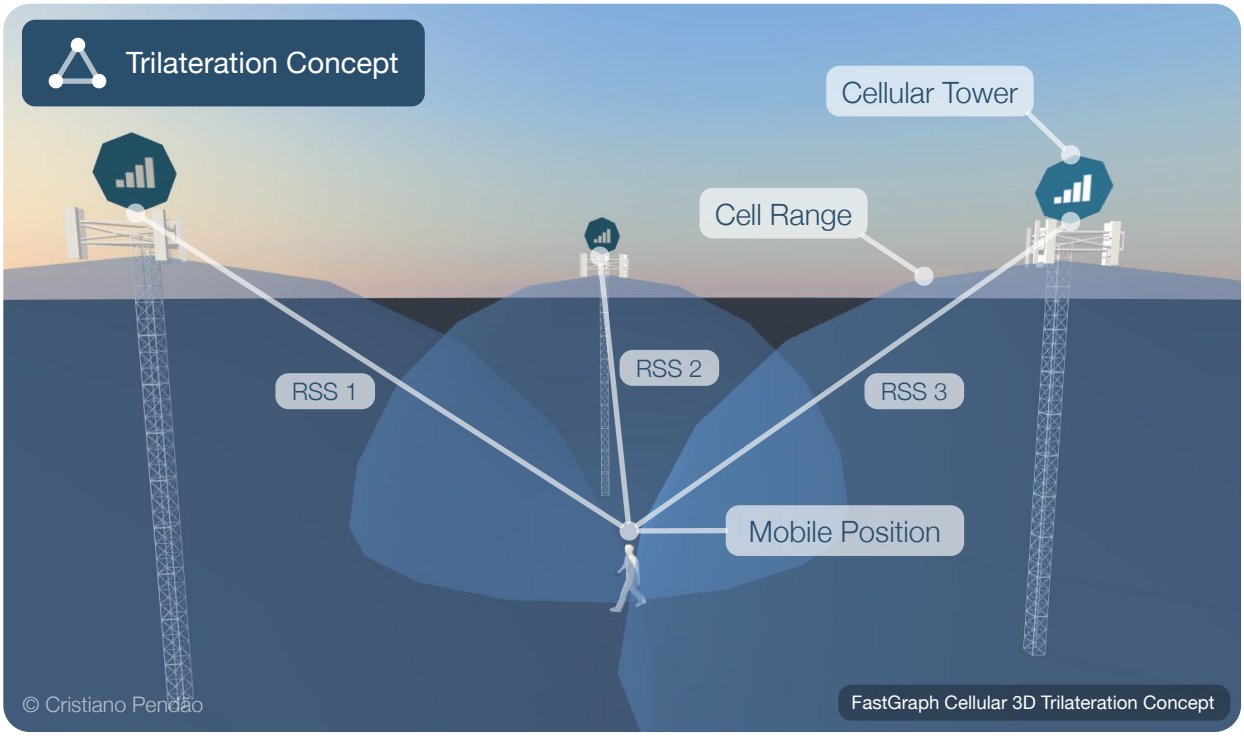

Figure 2.2: Cellular Trilateration in 3D

Similar trilateration methods can be used with Wi-Fi, applying the geometric properties of triangles to the space relations between the target device and the APs to estimate a position. The spatial distance to three or more APs can be estimate based on the RSS or based on the Received Signal Attributes (RSA) at a specific position. Each AP position is the center of a circle (or a sphere in 3D) with the radius defined by the estimated spacial distance (Figure 2.3). The intersection of three or more circles define the possible position for the device. To obtain a

good position estimation using a traditional trilateration solution it is necessary to previously know the position of the transmitters. However, this is a problem in most Wi-Fi infrastructures, since the deployment of Wi-Fi APs is uncoordinated and their position is unknown. Another problem is how to estimate the distance between each AP and the position where the sample has been collected.

In addition, the GNSS system is designed to work with unobstructed Line of Sight (LoS) between the device and the satellites. In indoor environments it is impossible to ensure LoS, therefore the propagation of RF signals is affected by different effects due to the indoor layout and materials, that attenuate and create multi-path effects. Due to the indoor propagation complexity, a traditional trilateration approach alone don't provide satisfactory results, but can be part of more advanced solutions.

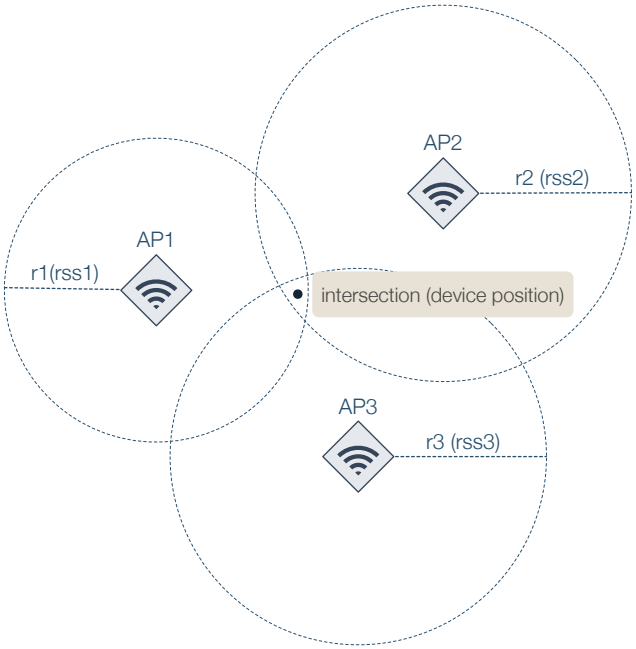

Figure 2.3: Trilateration principle

3.1.3 Fingerprinting Matching

The Wi-Fi Fingerprinting Matching is based on the assumption that the Wi-Fi RSS measurements can be used as an unique signature to identify a specific location in the environment, and this unique signature is known as Wi-Fi fingerprint [30]. Therefore, fingerprinting solutions rely on scene analysis, known as the offline/calibration phase, to map the Wi-Fi radio signatures (fingerprints) at several locations and build a fingerprint database, commonly known as Radio Map (Figure 2.4). In the online/location phase, the position of a device is estimated by comparing the currently observed fingerprints against the radio map.

A fingerprint can be based on a single observation (sample) obtained from a single scan or,

as has already been attempted [31–33], by combining multiple samples (RSS measurements) collected at the same location to minimize the impact of RSS fluctuations. This can be done simply by using the average of the RSS measurements of 10 samples collected at each 2 seconds at the same position. An alternative is to combine multiple samples collected simultaneously at the same position by multiple Wi-Fi interfaces, exploring the week correlation between the RSS values obtained from independent interfaces [30]. Another approach is to generate virtual fingerprints where the RSS are not measured but predicted using methods such as interpolation or propagation modeling [34]. The RSS prediction attempts to reduce the offline calibration effort to create the radio map.

Considering this, the format of a fingerprint fp_i is therefore similar to the format defined before for a sample (Eq. 2.1). The difference is that a sample represents a single observation obtained by a scan (similar to a single screenshot of the radio environment), and a fingerprint can be represent the information of one or multiple samples combined.

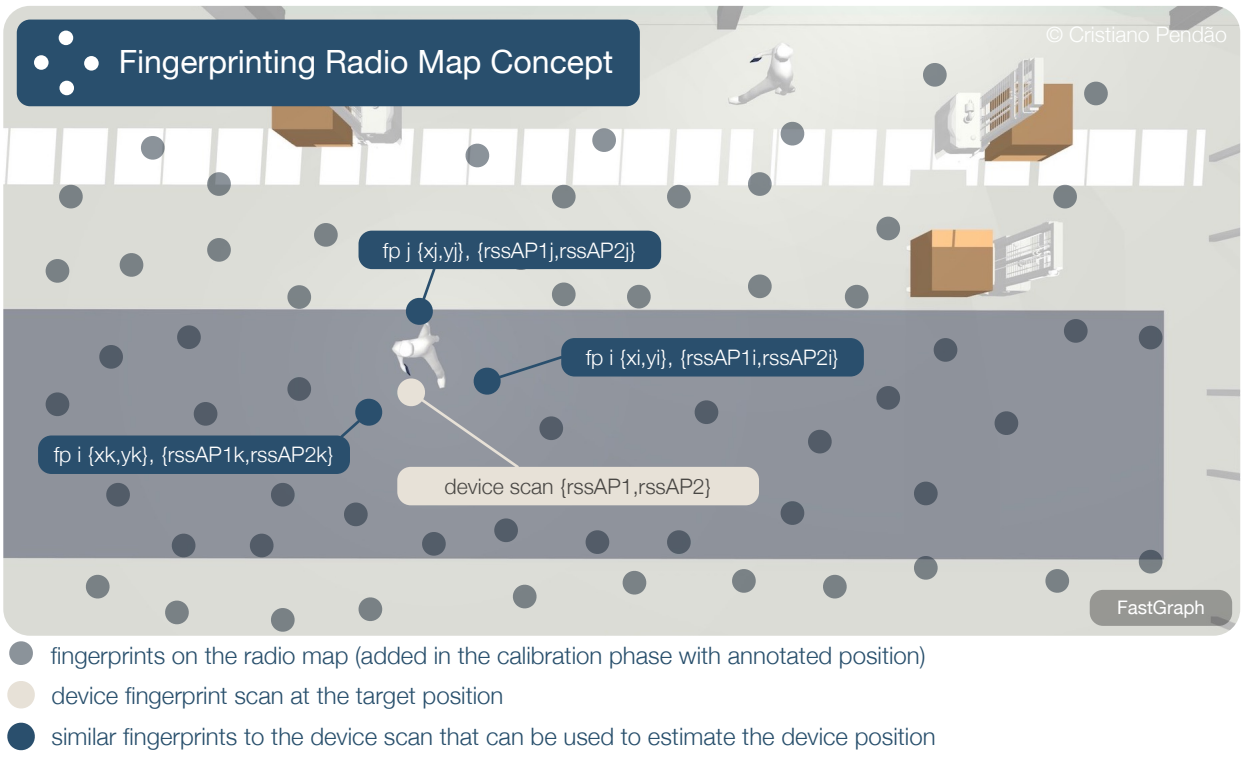

Figure 2.4: Wi-Fi Fingerprinting Radio Map Concept

Radar [35] is a pioneer and well-known work in Wi-Fi Fingerprinting. Fingerprinting techniques became popular because they can provide positioning without additional hardware, knowledge about the APs' positions, or the space layout. One advantage of this type of strategy is that it can be applied to heterogeneous indoor spaces, including underground [36].

Different approaches have been proposed for the matching process, leading to different levels

of accuracy [17]. There are fingerprinting positioning algorithms exploring pattern recognition techniques such as Probabilistic Methods, k-Nearest-Neighbour (kNN), Smallest M-vertex Polygon (SMP), Support Vector Machine (SVM), and Neural Networks.

- The **k-Nearest-Neighbour** averaging method rely on the online RSS to discover the k closest matches known positions from the database previously built, according to the root mean square errors principle. RADAR [35], adopts a similar technique (nearest neighbors in signal-space).

- The **Probabilistic Method** considers positioning as a classification problem. In Horus [37] each RSS measurements (corresponding for a position and AP) are represented as a probability distribution and using the maximum likelihood criterion for matching. SurroundSense [38] extends this concept to build a map using other features found in indoor spaces in addition to Wi-Fi RSS, such as ambient sound, light, color. Several other improvements have been proposed to the basic RF fingerprinting, such as the incorporation of mobility constraints [39] and an extension to outdoor settings [40]. Nibble [41], also uses a probabilistic approach. This approaches usually rely on the space analysis to construct a training dataset with the RSS measurements at known positions [18].

- **Neural Networks** are also used. In the offline phase the Neural Network is trained, using RSS measurements as inputs, and the sampling positions as targets. The weights are obtained after the training phase. For this type of positioning approach, a MultiLayer Perceptron (MLP) network is normally used. Battiti et al. [42] used a neural network based classifier to provide a location determination method.

- The **Bayesian Approach** explores probabilistic modeling techniques and may need active learning, calibration, error estimation, and tracking with history. Tracking-assisted positioning based on Bayesian approaches has been proposed [43–46] .

- The **Support Vector Machine** can be explored for data classification and regression, being used as a tool for machine learning and statistical analysis, with high performance in several classification and regression applications [47].

- The **Smallest M-vertex Polygon** [25] rely on the online RSS values to discover locations in signal space in relation to each signal transmitter separately.

In fingerprinting approaches, the fingerprints density and distribution on the space, as well as the position annotation accuracy, all affect the radio map quality, which has direct impact

in the positioning performance. In addition, near locations can have similar fingerprints which typically results in accuracy within a few meters (about 1 to 10 m [25,48,49]). A fair comparison is however difficult due to the heterogeneity of the testing environments.

For large buildings, manually collecting the thousands of samples required to build a good radio map, is a demanding task that may take a long period of time, being a scalability constrain. As a simple example let us consider that to create a good radio map it is necessary:

- Collected fingerprints in the positions defined by grid of points spaced by 1 meter;
- At each point are collected 10 samples (to minimize the impact of signal fluctuations);
- The samples are collected at intervals of 2 seconds;

In these realistic conditions, for a shopping mall with two floors of 100000 m^2, and considering only the effective scan time to collect the 10 samples in each position, the time required to build the radio map is around 1111 hours ($\approx$ 46,2 days). To this time has to be added the time required to create the referential grid and to move between positions. This illustrates how unpractical this process can be in real world deployments.

Moreover, the radio map becomes outdated due to RSS variations [50, 51] and due to maintenance and updates to the Wi-Fi infrastructure. The radio signature at a specific position can change rapidly due to interference, obstacles being moved, or even due to open/closed doors and different density of people inside the space. For this reasons, in addition to the initial calibration, keeping the radio map updated requires frequent recalibration. More detail about Wi-Fi Fingerprinting in general can be found in [51–53].

The calibration phase imposes significant limitations on the scalability of fingerprinting solutions, as is one of the main challenges. Collaborative approaches have been proposed to replace the required professional site survey for calibration.

3.1.4 Collaborative Radio Maps

In collaborative systems the radio map is built and maintained with the fingerprints collected and explicit annotated by the users [17, 47, 53].

The devices used to build a radio map can be controlled or uncontrolled.

- **Controlled:** Devices where the movement can easily be predicted or controlled, in order to cover a given area. The sensors on vehicles or robots inside a factory that keep the same paths are examples of this type of devices.

- **Uncontrolled:** Sensors carried by humans, where we can not predict its motion paths can be defined as uncontrolled devices.

Despite the challenges, using uncontrolled devices such as personal smartphones has the advantage of being easily available in most scenarios. In this context, several crowdsourced or organic solutions have been proposed to build radio maps.

Herecast [54] is a pioneer collaborative solution to build radio maps based on users inputs. The system relies on users inputs to associate the unknown access points to symbolic locations. The problems related to the environment changing are referred by the authors but not addressed. Molé [17] improves the quality of the fingerprints collected by using the accelerometer's data. This solution aims to provide room level accuracy (symbolic location). The accelerometer data allow a faster room detection by preventing some bad fingerprints to be added to the radio map. Redpin [55] extends this type of approach to build radio maps not only from Wi-Fi, but also to Cellular and Bluetooth. This solution also aims to provide room-level symbolic location on smartphones. The users manually provide the names to the places they visit. In [56], the authors propose an interpolation-based fingerprinting technique, based on users' feedback. Their system is built upon Bluetooth, and uses a feedback-weight assigning model, which assigns relative weights to user feedbacks, fine-tuning an under-trained positioning system, to achieve better accuracy. The authors refer the importance of users be well-behaved, in order to construct good radio maps. Another proposed user feedback model is presented in [47]. An important aspect brought up by the authors is the validation of this type of solution using different mobile devices, with different Wi-Fi chips. The authors argue that the system performance could be improved if the diversity of Wi-Fi chips in different mobile devices is considered, and suggest that an RSS compensation mechanism can be integrated to automatically adjust RSS patterns among different mobile devices. FreeLoc [57] is a solution that aims to be a calibration-free indoor localization scheme using Wi-Fi infrastructures which facilitates the extraction of accurate fingerprint values from short RSS measurement times, handle different devices, and keep a single fingerprint for each location in the radio map, independent of the number of fingerprints uploaded.

Collaborative systems can be used to create radio maps using for example personal smartphones. This approach has the advantage of using a very large number of devices to contribute to the same propose. However some strategies can be affected by the quality of the users' feedback, resulting in poor position accuracy. In addition, requiring explicit user interaction is inconvenient and unpractical leading to a low level of participation. For these reasons, some

other approaches explore inertial sensors and interfaces integrated in mobile phones to take advantage of the user motion patterns, in order to construct the radio map without explicit user' interaction [17, 18, 28, 58]. However, with this approach the system performance is also affected by the inertial sensors' noise and the users uniform motion pattern, which tend to follow a routine, limiting the surveyed areas. Moreover, constant sampling of several interfaces and sensors, which is required to track the users and to collect the fingerprints, result in high energy consumption, which has to be addressed as discussed in [59].

3.2 Model Based Approaches

Model-Based techniques try to minimize or replace the calibration effort of scene analysis by using, for example, radio propagation models.

Several indoor propagation models can be found in literature [60], where the propagation models are commonly classified as deterministic or empirical. Deterministic or semi-deterministic models are based on physical principles (electromagnetic wave propagation theory). Models based on geometrical optics are known as ray tracing, where the radio wave propagation can be seen as optical rays. The outputs of deterministic models require detailed description of the scenario (constitutive material parameters and 3D geometry) to obtain high accuracy, which is not easy to obtain in detail. More sophisticated predictions are time consuming, which limit the real applicability of deterministic models.

The empirical and semi-empirical models are based on representative measurements processed statistically. The prediction is usually based on simple closed expressions providing easy and fast application of this type of models. However, the propagation loss can only be predicted without high accuracy, but less specific description inputs of the environment are required.

The One-Slope and Multi-Wall are the most popular empirical models. The One-Slope Model (1SM) [61] does not require detailed knowledge of the building to compute the average signal level. The path loss in dB is a function of the distance between the antennas of the transmitter and receiver. The prediction is controlled by the reference loss value and a power decay factor as empirical parameters for a given environment.

The indoor environment structure type has high influence on the value of the power decay factor. The 1SM provide a rough estimation and the selection of suitable power decay factor is fundamental. Regardless of the 1SM dependence on used empirical parameters, it is useful when no information about the indoor space is available or when is needed a fast draft.

A semi-empirical Multi-Wall Model (MWM) [61] provides better accuracy than 1SM. The results are site-specific, but is also necessary a floor plan description as an input. This models

considers the number of walls and floors between the transmitter and the receiver and the attenuation factor for each wall and floor (considering a specific type). Despite the several building materials, when considered the wall attenuation factors statistical nature, only a few wall types are necessary to be define for MWM. The MWM is site-specific because during the prediction particular walls are considered. A significant improvement of the site-specific accuracy is observed when comparing to 1SM [62].

3.2.1 Log-Distance Path Loss Model (LDPL)

The Log-Distance Path Loss Model (LDPL) is well known and frequently used [18, 63–66] to model signal propagation. The LDPL relates the RSS at a given position with the distance to the transmitter, such as a Wi-Fi AP (Figure 2.5).

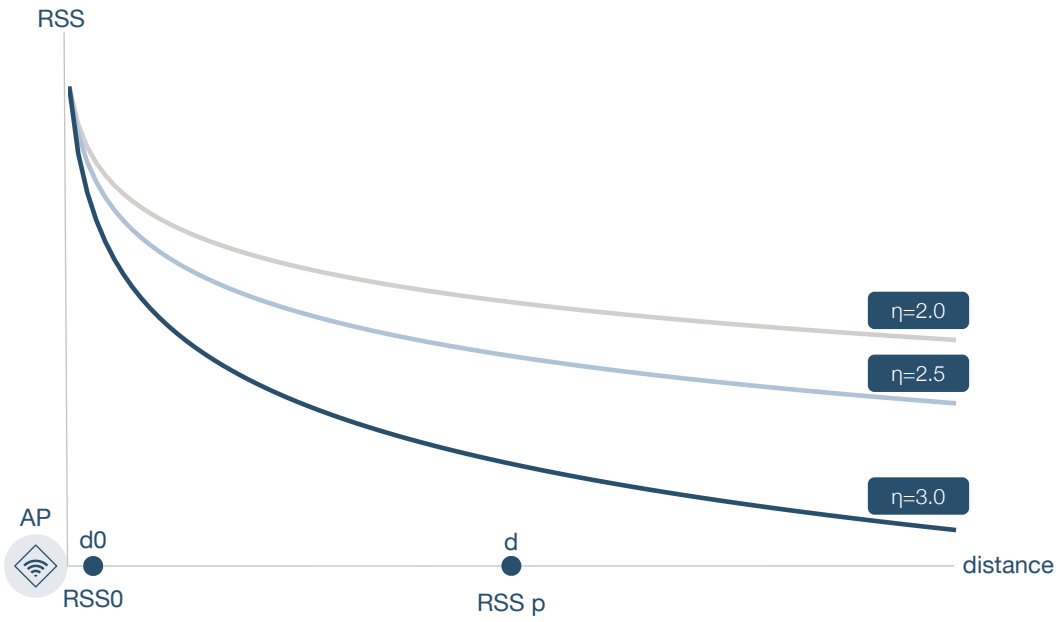

Figure 2.5: RSS signal over distance LDPL

The RSS at a position p is given by:

$$RSS_p = RSS_0 - \left(10 \times \eta \times log10\left(\frac{d}{d_0}\right)\right) \tag{2.2}$$

RSS_p is dependent on the RSS_0 (measured at d_0, usually at 1 meter), the distance to the transmitter (d) and the path loss exponent (η), that reflects the fall rate of the RSS signal from an AP at a specific position.

The difference between the RSS measurements of the same AP at different known positions can be used to approximate the AP's position. Then the device's position can be estimated for

example based on the RSS-based weighted centroid, considering the approximated positions of the observed APs. This type of technique is affected by the indoor propagation complexity. For example severe multi-path effects, which vary with the space layout, objects displacement, different reflective materials and due to low probability of Line of Sight.

The reduced calibration effort required by model-based approaches is typically translated into worse positioning performance. In addition, the floor plans or the APs' locations are needed for most of these systems. However, it is unpractical and very difficult to know the access points positions, the location of obstacles and even more how each type of obstacle affects a signal from a specific access point at a given location.

Radar [35] has a model-based version, which uses knowledge about the AP locations, transmit powers and floor map to estimate the RSS at several locations. EZ, a system described in [63], uses the geometric spatial constrains of the Wi-Fi propagation, given by the LDPL, with opportunistic GPS fixes of some devices, and estimate the location based on the RSS relative signal measurements, without requiring previous knowledge of the RF environment, floor plan or APs locations. A R variable is used to account the variations from noise and multi-path.

EZ treats the signal measurements RSS_i, the 2D positions (of APs and target devices), and a path loss exponent for each AP (eta_{AP}) as unknowns, resulting in a set of simultaneous equations for each RSS observation.

The authors explain that the resulting set of LDPL equations are a system of simultaneous non-linear equations, and they don't found an analytical solution to analyze them. Instead, a Genetic Algorithm (GA) and optimization technique are used to solve the system of equations and find a solution that minimizes the least mean absolute error. This process requires high computational effort, which can range from minutes to several hours, depending on the space and the problem size. The time required to solve the equations is related to the number of unknowns (locations and APs' parameters), the nature of data (data that fits the model well is solved faster), the number and placement of known locations, the distribution of the unknown locations, and the choice and number of initial solutions that are randomly picked for the GA. All of this increases the computational effort when computing the RF model due to the large space of possibilities and the evaluation of solutions fitness. In order to reduce the search space and compute a solution more efficiently, a set of empirical conditions and constrains are defined, such as the possible values for the η of an AP. EZ only considers 2D positions and select a set of APs. Estimating 3D positions for both user and APs adds more unknowns to the LDPL equations and therefore increases the complexity of solving them, increasing the computational effort. Considering all the APs heard, also has impact in the computational performance, and

the authors suggest that the benefits are small.

In addition, this process has to be repeated periodically. The authors suggest that the parameters of the RF model are likely to be subject to diurnal variations and may be need to estimate them for different times of the day. The authors also define a set of rules and conditions where the LDPL equations can be uniquely solved, reporting also a set of situations where it is impossible to uniquely determine the $RSS0$ and n for an AP. EZ must also avoid measurements that have almost the same RSS from the same AP. The authors refer that determining the necessary conditions under which a set of LDPL equations has a unique solution is still an open problem.

EZ considers that the path loss exponent η is the same for one AP in all areas of the space while, in practice, η is different from location to location, due for example to different obstacles between each location and the AP. The EZ reliance on GPS fixes to convert the obtained positions into true positions can be a problem in indoor spaces.

Despite the challenges, EZ reports results close to conventional fingerprinting solutions, without requiring extensive calibration. The AP's position estimation error is not mentioned.

- **Path Loss Exponent (η)**

The practical values usually considered for the path loss exponent in indoor environments range between 1.5 and 6.0 [63]. As mentioned before some approaches [63, 64] also assume that the parameters of an AP ($RSS0$ and η) are homogeneous across a space. In fact the $RSS0$ of and AP is a fixed characteristic, and can only be different from AP to AP. The η is however more difficult to define. As explained before the path loss factor (η) reflects the fall rate of the RSS signal from an AP at a specific area. In free space we may say that the η is always the same for a given AP. However, this is not true in a real world space, even in open spaces the η will vary from location to location, due for example to reflections in the floor and celling. Even the radiation diagram may be different from AP to AP, due to AP installation position or the AP antennas design, which can lead to different RSS measures at the same distance from the AP. To obtain the correct distance to an AP using the LDPL it is necessary to use an η, that compensates the loss introduced by all of this effects.

The communication channel between an AP and a device at a given position has therefore singular propagation characteristics, and in addition that propagation characteristics may even change over time. Even thought some aspects that affect the signals propagation are fixed for a space, such as the attenuation introduced by the walls, some other aspects can change, such as density of people in a given area, increasing the complexity of applying the correct η. Moreover,

a small difference on the used η may result in a significant error in the estimated distance. The Figure 2.6 represents the distance obtained, using the LDPL (Eq. 4.8), for the same RSS measurement applying a η=2 and a η=1.5. As we can see in the next figure, decreasing the η in 0.5 results in 10 meters of difference for the same RSS value.

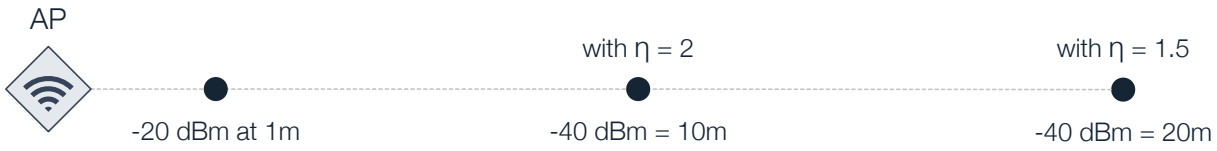

Figure 2.6: Free Space Path Loss

When an obstacles such as a wall (with 26 dB of attenuation) is added between the AP and the target position (Figure 2.7), the attenuation (A_{dB}) is given by:

$$A_{dB} = 10 \times \log_{10}\left(\frac{P_b}{P_a}\right) \tag{2.3}$$

where, P_b and P_a is the power with and without the obstacle, respectively.

With the additional attenuation of 26 dB the RSS measurement at the same position (10 meters) will now be -66 dBm instead of -40 dBm.

When applying the same η from the previous scenario ($\eta = 2$) the calculated distance is now 199 meters ($p1'$). In this case a new value of η has to compensate the obstacles's attenuation, in order to obtain the correct 10 meters distance ($p1$).

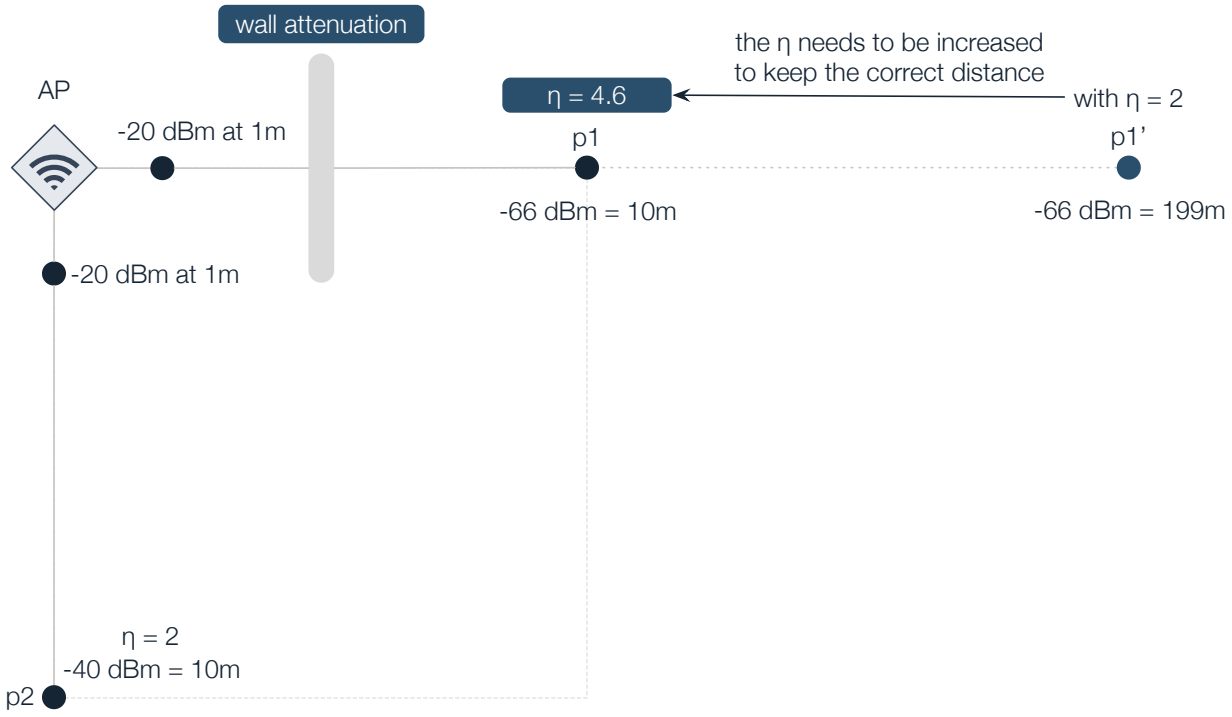

Figure 2.7: Free Space Path Loss with Obstacle Attenuation

This simple example illustrates the importance of finding the correct η value, which is crucial to accurately estimate the distance between a device and an AP. In order to obtain the correct distance when we have an obstacle attenuating the signal, the η value has to be adjusted (in this case increased). This also explains why assuming the same value of path loss (η) for an AP in all positions of a space is not realistic. In this case although we have two positions at the same distance from the AP ($p1$ and $p2$), the propagation is different due to the obstacle, therefore a different η has to be used since the received RSS at the same distance will be different. However, considering all the variables and effects that influence the fall rate of a signal in indoor spaces, estimating the correct η for a specific area, in a total unsupervised and automatic way, is a very complex challenge. The space layout, the build material, furniture, and temperature, all influence how the signals propagate. The difference in materials and dimension of obstacles will introduce different attenuations to a signal. All this attenuations have to be reflected in the path loss exponent, that has high impact in the LDPL model.

Despite the fact that indoor propagation modeling is one of the most complicated tasks in this field, it is highly relevant as it may be used to replace or complement the site survey techniques [62].

3.3 Received Signal Attributes (RSA) Based (Time And Space)

RSA-based methods estimate a position using the attributes of the received signals, such as the signal's propagation time from the sender to the receiver, or the angle at which the signal is received [28]. In most of these methods the position of the transmitters must be known a priori. In WLANs this requirement is difficult to ensure due to the uncoordinated deployment of APs. The propagation effects, due to obstacles, signal fluctuations, and interference also affects this type of solutions.

3.3.1 Lateration Techniques:

- Approaches based on Time of Arrival (ToA), such as PinPoint [67], explore the relation between the distance from the receiver (mobile target) to the transmitter, which is directly proportional to the propagation time. This approach requires that all transmitters and the receiver be perfectly synchronized, and a timestamp must be used in order to measure the distance traveled by the signal.

- Instead of estimating the absolute arrival time, as in the ToA, the Time Difference of Arrival (TDoA) analyses the difference between the signal's arriving time at multiple measuring units to estimate the relative position of the mobile transmitter. With this

approach there are no requirements on the mobile device, however all measuring units must be synchronized by a precise time reference and reference signals. Cricket [68] and NearMe [69] are two examples of systems based on TDoA.

- Roundtrip Time of Flight (RToF) approaches [70] measure the time of flight of the signal from the transmitter to the receiver and back. The synchronization precision required for RToF is lighter that for ToA.

3.3.2 Angulation Techniques:

- In Angle of Arrival (AoA) approaches [71, 72] the angle of arrival of the signals is used to obtain a location estimation by intersecting multiple pairs of direction angle lines, each represented by a circular radius from a emitting station to the mobile devices. The position may be determined with only two or three measuring units for 2D or 3D positioning respectively, and the measuring units don't need to be synchronized. However, this method has complex hardware requirements, and as the mobile target moves away from the measuring unit the location estimation performance degrades [25].

With these methods a spatial relationship between the transmitters and the receivers can be obtained. Then trilateration is usually used to estimate a position, which requires three or more transmitters with known positions to perform well. Some of this techniques also require specific hardware or modified transmitters (such as APs), which is unpractical in most applications with already existing infrastructures, involving additional costs. In addition, this approaches are affected by the same indoor propagation complexity, such as multi-path effects and interference, that affect the RSS-based Geometric and Model methods.

3.4 Wi-Fi SLAM

In robotics, the Simultaneous Localization and Mapping (SLAM) approach is used to construct the space map while locating the robot in the space. Some works have been done to apply this approach to Wi-Fi. WiFi-SLAM [73] uses the Gaussian Process Latent Variable Model (GP-LVM) to build Wi-Fi RSS Maps without requiring Inertial Measurements Unit (IMU) data. The limitations of the WiFi-SLAM are the computational efficiency, and relying on assumptions on a signature uniqueness and human walking patterns. Wi-Fi GraphSLAM [74] improves the techniques based on GP-LVM, in terms of computational efficiency and the assumption on signature uniqueness. In robotics, the GraphSLAM technique is used for building the space map while estimating the trajectory. Wi-Fi GraphSLAM uses gyroscope and pedometer data,

and similar Wi-Fi RSS observations to detect that the user has returned to a previously surveyed physical location. In WiSLAM [75] data from a foot mounted IMU is combined with the Wi-Fi RSS. The RSS measurements are translated into distance using a log-distance propagation model. The application scenarios are limited by the required foot-mounted sensor and also by fixing the path loss exponent to 2. However, for indoor spaces it is not realistic to assume a fixed and homogenous path loss exponent value for all areas of the space.

3.5 Dimension Reduction and Multi-Dimensional Scaling

A problem of dimensionality reduction arrises from large volumes of high-dimensional data, such as Wi-Fi fingerprints for a 3D space. For this reason, finding meaningful low-dimensional structures hidden in the high-dimensional data is important.

As mentioned before, the need associating fingerprints to real-world coordinates is probably one of major challenges when building radio maps. Non-linear dimension reduction techniques started to be explored in this context [27]. This technique try to represent the radio map using a low-dimensional, non-linear manifold, enabling this way better statistical modeling of the signal properties in complex multi-path environments.

Manifold learning methods are applied to high-dimensional data sets to reduce the dimensions or number of features of the data to a lower level. This is done by finding the defining features in the data, and relies in the assumption that the low-dimensional manifold embedded in the high-dimensional space contains most of the variability. After the manifold is constructed the observation points attached to the manifold are mapped into the geographical coordinates.

Multi-Dimensional Scaling (MDS) is used to explore similarities or dissimilarities in data using a set of related statistical techniques. An MDS algorithm begins with a matrix of item to item dissimilarities, then assigns a location to each item in D-dimensional space, where D is specified a priori. The resulting locations may be displayed in a 2D or 3D structure. Considering a WLAN network, the proximity is the simplest measure of dissimilarity, describing if two devices are in communication range. Is known that according to the radio propagation the distance has an exponentially relation to the RSS. [76]

LiFS [28] use MDS to create a high dimension space according to the inter-fingerprint distances, in which fingerprints are represented by points, and their mutual distances are preserved. LiFS measures walking steps to help estimate the distance between two locations in a floor plan. The authors propose a stress-free floor plan, which maps real locations into a high dimension space by MDS, such that the geometrical distances between the points in the high dimension space reflect their real walking distances.

In traditional approaches, fingerprints are geographically unrelated, losing the possibility of building a fingerprint space. Zee [18] leverages from inertial sensors and MDS to resolve ambiguity in location during crowdsourcing.

The knowledge about the position of the Wi-Fi Access Points can be used to estimate or to improve the position for a target device based on the RSS measurements.

In [76] the authors use MDS techniques to determine a geographic configuration of Wi-Fi APs, without ground truth information. The authors are able to distinguish APs on different floors. In this approach, AP to AP distances are estimated by a radio attenuation model. Other works also adopt MDS to estimate the locations of wireless devices [28,77,78]. Serendipity [66], aims to locate Wi-Fi APs using radio observations from users smartphones in an unsupervised process. A MDS technique is used to estimate the relative positions of the APs based on the dissimilarities between all pairs of Wi-Fi APs. Absolute positions are obtained using additional observations at known positions. The AP's positions estimated are used to positioning the smartphones. The authors quantize the received signal strength into several levels to avoid the conversion of the received signal strength to a real distance and the estimation of propagation model parameters. They statically determine the quantization thresholds based on domain knowledge practical values, as example the authors define a normal maximum value of an RSS around -20 dBm, and the minimum around -90 dBm. They also propose a dynamic quantization method, in order to automatically decided thresholds for a place.

Pulkkinen et al. [27] presented a WLAN positioning approach where high-dimensional signal fingerprints are represented as points on a two-dimensional manifold, using a semi-supervised manifold learning technique for building accurate radio maps. The authors used the Isomap algorithm for the manifold learning phase. The Isomap algorithm [79] is based on the same principles of MDS. It tries to find a lower dimensional representation of the data with the minimum distortion possible between the points. The Isomap algorithm constructs a neighborhood graph where each point xi is connected to K nearest neighbors. The distance dX(i, j) between two points is calculated as the sum of edge lengths along the shortest path connecting them. The manifold learning phase presented by the authors is based on observing plain RSS vectors without their geographical coordinates. To fix the mapping to geographical coordinates from the coordinate system of the manifold are used a sample of key points whose location is known. In this case it is assumed that the signal characteristics are determined by the location of the receiver. Is referred that if the possible locations are constrained to a flat two-dimensional surface, the resulting manifold is then two-dimensional as well. The authors say that when mapping the fingerprints from the high-dimensional signal space to the low-dimensional mani-

fold is important to maintaining the pairwise distances between the fingerprints.

The described contributions of this work include a method for mapping points on the Isomap manifold to a geographical coordinate system using a relatively small subset of the fingerprints with precise known location. The authors mentioned that a urgent topic for investigation is the balance between the accuracy and the number of labelled examples (deployment cost).

Other methods use Laplacian regularized least squares regression [80–82], without explicitly constructing a low-dimensional manifold, with some drawback related to highly sensitive outcomes.

3.6 Wi-Fi Positioning Performance, Problems and Challenges

In this chapter different Wi-Fi positioning solutions were discussed, each of them with different performance, requirements and limitations. As mentioned before, the positioning performance comparison between solutions is difficult due to the heterogeneity of the testing environments. Considering the IPIN conference competition as example, where the solutions are compared in the same conditions, in the same environment, and considering the same accuracy metric (75th percentile of error), the position performance of state of the art indoor positioning solutions can range between about 9 meters to 30 meters, when tested online and on-site in real world spaces, and between about 3.5 to 4.5 meters when evaluated offline and out-site. [1].

Most of the Wi-Fi-based positioning systems are affected by the RF propagation complexity inside indoor spaces, where RF signals are affected by different types of interference. Sources of performance degradation are: complex environment characteristics, dynamics inside indoor spaces, changes in Wi-Fi infrastructure and devices heterogeneity.

- **Complex Environment Characteristics:** Wi-Fi Based positioning systems suffer from multi-path, fading and shadowing effects in indoor spaces. Radio propagation in such environments is affected by all physical elements, such as walls and furniture. In addition, different types of materials will introduce different attenuation to the RF signals, and will produce different multi-path effects. Each space has different propagation characteristics, even if considering only the static element of the space. Therefore, modeling the propagation in indoor environments is difficult.

- **Environment Dynamics:** Static characteristics are only one part of the complexity. Indoor spaces are highly dynamic environments, adding complexity to the RF propagation. The elements of the space can also change with open/closed doors, furniture being

moved or people moving around.

- **Wi-Fi Infrastructure:** Scene Analysis techniques rely on the radio environment analysis to map physical positions with radio signatures observed at that position. The radio signatures observed at a given position are strongly linked to the wireless infrastructure configuration, number of APs and their position. In addition, the radio infrastructure maintenance and update, where APs are replaced or even changed to another location, degrades the performance of the positioning system.

  The number of APs also affect several approaches. A low number of APs may result in poor results. However, some researcher found that with a large number of APs the accuracy of some positioning systems also decreases, suggesting that a different number of APs must be selected in different environments in order to obtain the better results. In addition, selecting the right set of APs can also impact the system performance, when for example considering also APs from nearby buildings. Another problem is the mobile Wi-Fi Hotspots that can be observed at different locations. In fingerprinting approaches this type of APs can be associated to a given location in the radio map during the calibration phase, and later in the positioning phase that same Hotspot can be observed at a different location, introducing error to the estimated position.

- **Devices Heterogeneity:** RSS-based methods such as fingerprinting approaches are also affected by the heterogeneity of Wi-Fi devices. Different smartphones have different Wi-Fi chips, operating systems, and wireless networks handling processes. All of this influence the signals strength measurements, introducing variations in the fingerprints collected by different terminals at the same position, reducing the positioning performance [39, 83, 84]. Also when a collaborative approach is adopted to build the radio maps, if the data provided by users is not controlled, it can also compromise the system performance, mainly when these data is collected using different devices and the position annotated manually.

- **System Complexity:** Some approaches require dedicated hardware or infrastructures, such as the deployment of modified APs. The requirements of the target devices are also different for each approach. Some solutions require dedicated sensors to track the motion, but in some cases this can be replaced by the internal sensors of a smartphone, at the cost of some accuracy. Other approaches require more complex hardware to capture specific parameters such as the received signal attributes.

Chapter 3

3D RF Environment Simulator

This chapter describes the 3D RF Environment Simulator that was designed primarily to simulate Wi-Fi radio environments, but that can be easily extended to other types of Wireless technologies such as Bluetooth or Zigbee.

This simulator was used to generate synthetic data, that was essential during the FastGraph's development process. Developing this simulator from scratch allowed flexibility and full control over all aspects of the simulation and over the generated data.

A 3D virtual space can be created, and 3D objects such as walls can be added to create a layout. Multiple virtual devices can also be configured and added to the space. These devices collect samples with Wi-Fi Fingerprints and can also collect motion and orientation data. Several parameters of the simulation session can be configured, allowing generating controlled synthetic datasets in different conditions.

Section I describes the available configurations for a simulation, which includes defining the space, objects and devices. Section II explains the simulation session and the available configurations.

1 Simulation Configuration

To run a simulation session a 3D virtual space (vs_i) must be configured. First the space dimensions $vsd : (w, l, h)$ are defined:

$$vs_i = (vsd : (w, l, h)) \tag{3.1}$$

After defined the space 3D dimensions, different elements can be added to the virtual space (Figure 3.1).

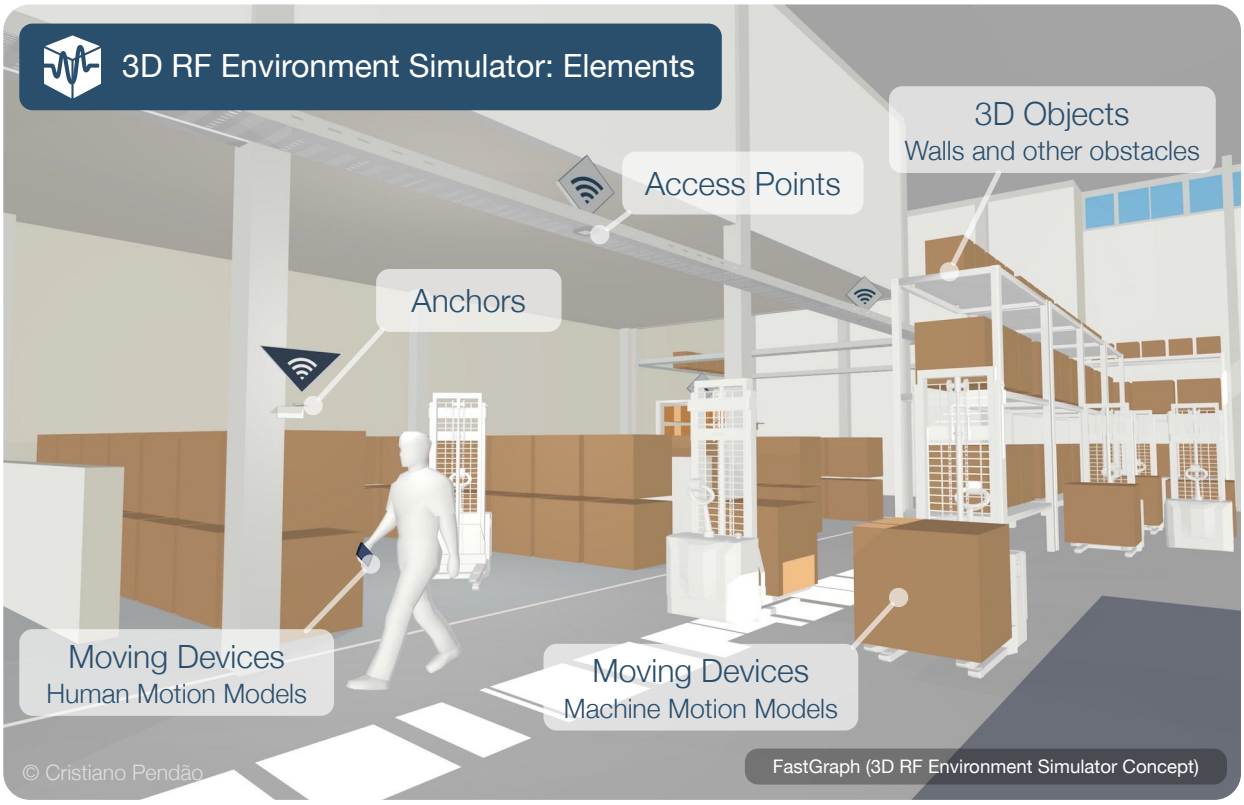

Figure 3.1: Elements available in the Simulator

1.1 3D Objects

A specific layout can be created by adding 3D objects as walls. These objects can also be used to represent furniture or other type of obstacles. A 3D object o_i is defined by:

$$o_i = (p : (x, y, z), od : (w, l, h), \alpha) \tag{3.2}$$

where, p is the position of the object in the space, od is the object dimensions (width,length,height) and α is the attenuation factor for the signals crossing the object. The attenuation can be configured to match the attenuation of a given material, such as metal, concrete, glass.

At this point the obstacles can be added to the virtual space vs_i:

$$vs_i = (vsd : (w, l, h), o : (o_1, ..., o_n)) \tag{3.3}$$

1.2 Wi-Fi Access Points

With the virtual space dimensions set, the layout defined and obstacles added, virtual transmitters can be added, in order to simulate radio emitters, such as Wi-Fi Access Points. A virtual AP ap_i can be described as:

$$ap_i = (ssid, bssid, rss0, p : (x, y, z)) \tag{3.4}$$

where, $ssid$ is the name of the WLAN, $bssid$ is the mac address of the AP, $rss0$ is the transmit power at 1 meter and the p is the position of the AP in the space.

After adding the APs, the virtual space vs_i is defined as:

$$vs_i = (d : (w, l, h), o : (o_1, ..., o_n), ap : (ap_1, ..., ap_n)) \tag{3.5}$$

1.3 Anchors

Anchors are fixed devices that can be used to measure the radio signals transmitted by emitters, therefore the samples from these devices can be used as references. An Anchor can be defined as:

$$a_i = (aID, p : (x, y, z), ts) \tag{3.6}$$

where aID is the unique identifier of the Anchor, p is the Anchor position in the space, ts is the sampling period.

The Anchors only collect Wi-Fi fingerprints. Each sample collected by an Anchor is therefore defined as:

$$s_i = (aID, t, fp : ((ap_1, rss_1, ch_1), ..., (ap_n, rss_n, ch_n)), p : (x, y, z)) \tag{3.7}$$

where aID is the unique identifier of the Anchor that collected the sample, t is the timestamp when the sample was collected, fp is a fingerprint with the rss measurement, channel ch and identification of each AP in range. The p is the position where the sample was collected, in this case is the position of the Anchor.

1.4 Moving Devices

Moving devices travel through the virtual space, and can be used to simulate smartphones carried by humans or to simulate autonomous machines. Moving devices follow a configurable motion model and respect the space configuration.

The random walk motion is commonly used to simulate human walking, however at indoor spaces, a random model doesn't characterize well the human walking patterns. Therefore, different motion models, which will be explained in more detail later, where implemented and are available in the simulator.

A moving device md_i is described by:

$$md_i = (mdID, p : (x, y, z), ts, motion) \tag{3.8}$$

where $mdID$ is the unique identifier of the device, p is the current position of the device in the space, ts is the sampling period, $motion$ defines the motion model to be used.

The samples collected by moving devices can have additional information regarding the travelled path, such as the orientation and displacement since the last sample. A sample s_i collected from a moving device, can be described as:

$$s_i = (mdID, t, fp : ((ap_1, rss_1, ch_1), ..., (ap_n, rss_n, ch_n)), dis, he, sp : (x, y, z), n) \tag{3.9}$$

where $mdID$ is the identifier of the virtual device that collected the sample, t is the timestamp when the sample was collected, fp is a fingerprint with the rss measurement, channel ch and identification of each AP in range. The dis is the linear displacement since the previous sample, he is the current heading and sp is the position where the sample was collected. The n is the path loss exponent (η) for the sampling position.

The difference between the sp in the samples from moving devices, and the p in the samples collected by the Anchors, is that in real world applications, a sample from moving devices has no information about the exact position where it was collected. Therefore, the sp and n are used only as ground truth for testing and evaluation purposes.

1.4.1 Motion Models

A moving device can use different motion models, which are suitable to simulate human and machine motion patterns (Figure 3.2).

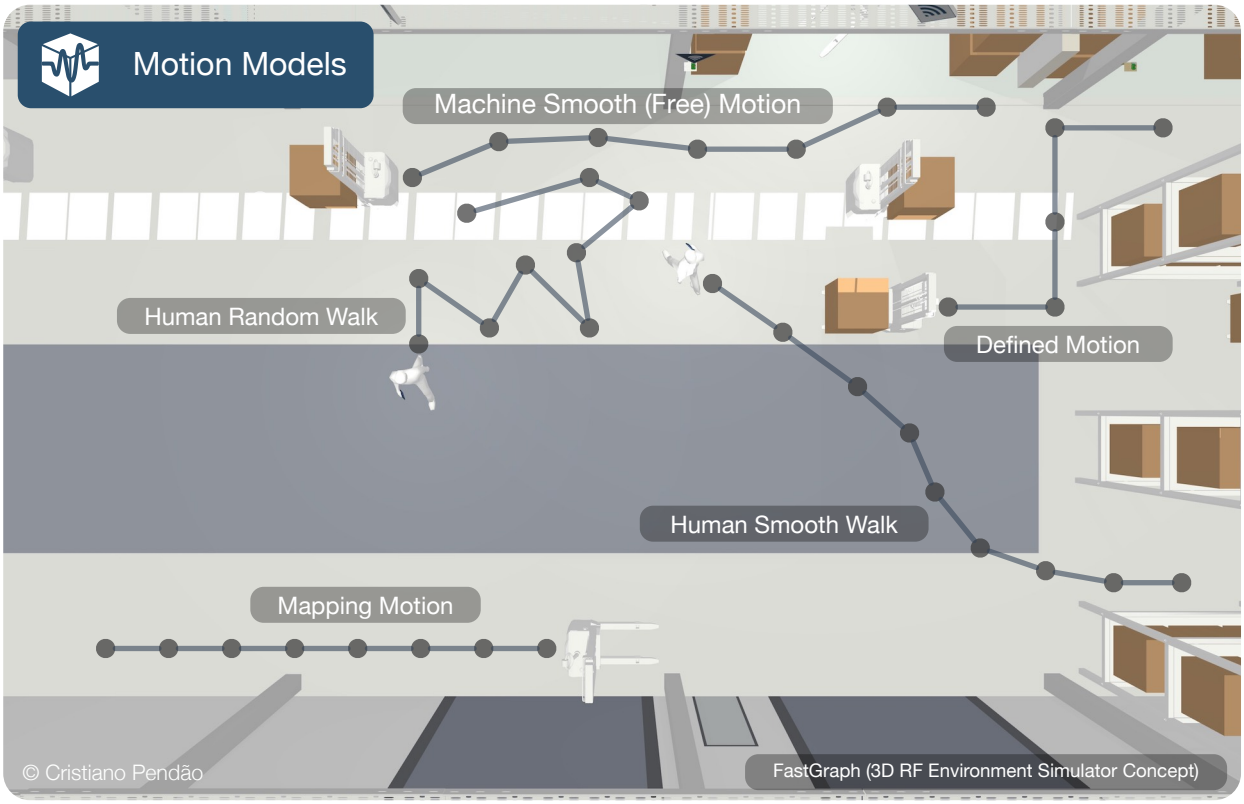

Figure 3.2: Moving Devices Motion Models

To simulate human motion patterns, the Random Walk or the Smooth Walk models can be used. These motion models are useful to simulate a scenario where a smartphone collects data while carried by a person.

The Defined motion model, where the moving device follows a specific path, and the Smooth motion model, where the device moves freely in the space, are useful to simulate for example an autonomous machine moving in an industrial environments.

The Mapping motion model allows to collect data across the entire space.

Random Motion Model: The device will move in a random way.

Each new position (p_{i+1}) for the device will be:

$$px_{i+1} = px_i + (vel \times t) \times cos(he) \tag{3.10}$$

$$py_{i+1} = py_i + (vel \times t) \times sin(he) \tag{3.11}$$

$$pz_{i+1} = pz_i \tag{3.12}$$

where, (px_i, py_i, z_i) defined the current position, $vel = random(0, maxV)$ gives the device

moving velocity, which is a random value between 0 and $maxV$, t is the elapsed time from the previous sample, and $he = random(0, 360)$ is the moving orientation. In this motion model the pz does not change (Figure 3.3).

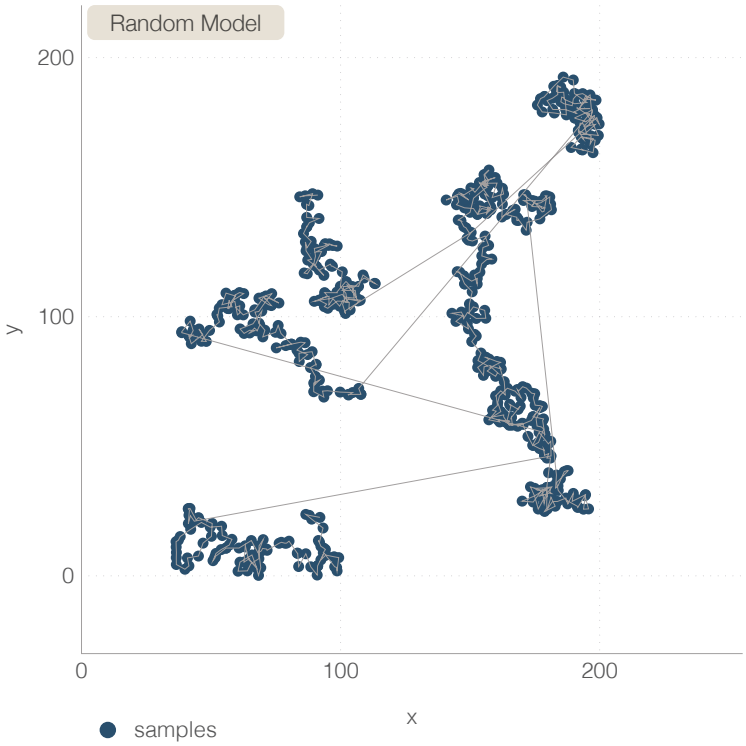

Figure 3.3: Samples generated using the Random Model

Smooth Motion Model: With this motion model (Figure 3.4) a device motion is more realistic, with smoother direction changes. Each new position is defined by the same equation of the *Random* model, however the new orientation is defined by:

$$he_{i+1} = he_i + random(-20, 20) \tag{3.13}$$

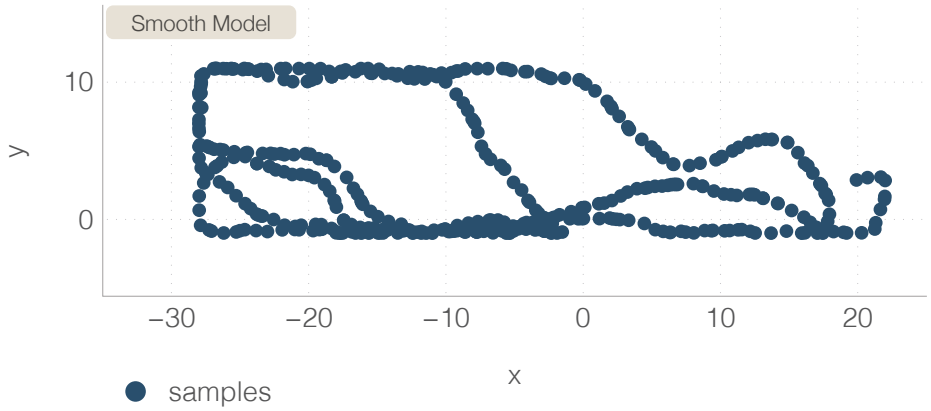

Figure 3.4: Samples generated using the Smooth Model

This means that the moving orientation is defined by the current orientation with a variation between -20 and 20 degrees, and not a random value between 0 and 360 degrees.

Mapping Motion Model: In this configuration the device will map the space by moving in a matrix of positions (Figure 3.5). Samples will be collect in a grid of positions spaced by 1 meter in x and y, and restricted by the space dimensions. The result is a grid of samples for the entire space.

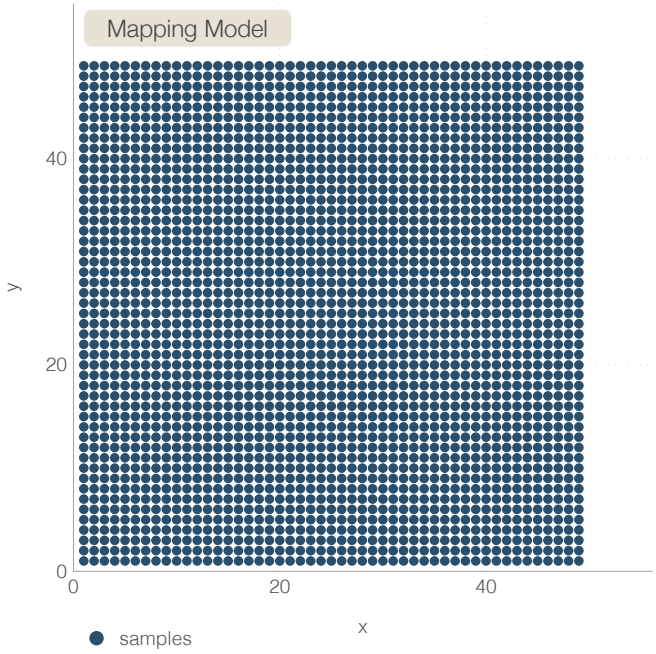

Figure 3.5: Samples generated using the Mapping Model

This model is useful to create, for example, a dense radio map for the entire space.

Defined Motion Model: When using this configuration, the device will follow pre-defined paths for the simulation session (Figure 3.6). The device will sequentially visit each position of each defined path.

With this motion model is possible to simulate a human or a machine traveling in fixed and controlled paths, allowing for example to define a specific trajectory and sharp or smooth turns.

A path based on this model can be described as:

$$path_i = (pathID, p : (p_1 : (x, y, z), ..., p_n : (x, y, z))) \tag{3.14}$$

where, $pathID$ is the path identifier, and p is a list of positions in the space to be visited in sequence. A list of paths $(paths : (path_1, ..., path_n))$ can be passed to a moving device before a simulation session, and the device will follow these paths if the motion model $Defined$ is

selected. The time interval between the positions of the path can also be configured.

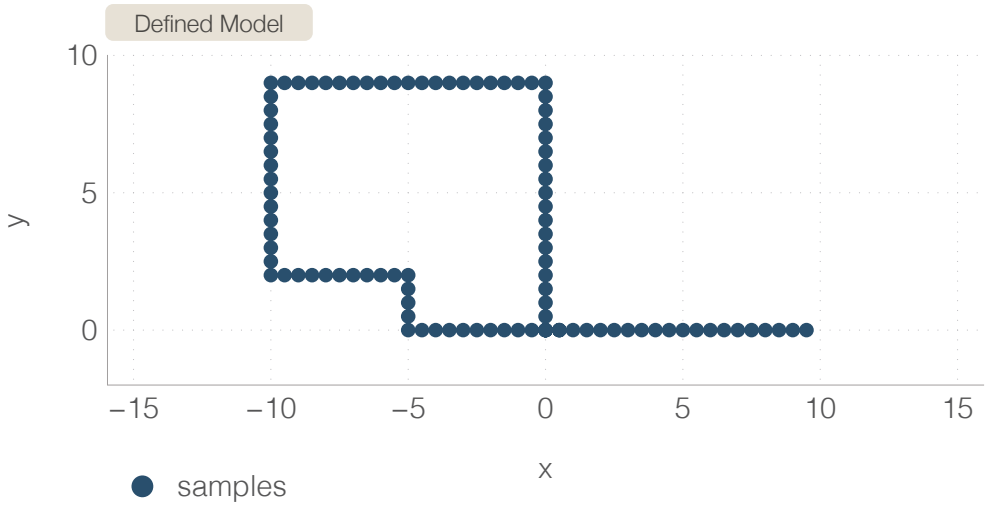

Figure 3.6: Samples generated using the Defined Model

2 Simulation Session

After the initial configurations, simulation sessions can be run to generate synthetic data. At this point, and after all the devices added, a virtual space vs_i is described by:

$$vs_i = (d : (w, l, h), o : (o_1, ..., o_n), ap : (ap_1, ..., ap_n), a : (a_1, ..., a_n), md : (md_1, ..., md_n)) \quad (3.15)$$

where, a are the Anchors and the md are the moving devices added to the space.

Before starting a simulation session, a few more parameters can be defined:

- **Simulation Duration:** Defines the total number of samples to be collect in the simulation session.
- **Number of Samples per Group:** Samples collected in intervals of less than 30 seconds can be related by time or motion information. In real world scenarios, devices can go offline or leave the space, appearing later in a different position. In order to simulate this behavior, a number of samples to be collected by a moving device before it goes offline can be configured.
- **Number of Groups:** This parameter defines the number of times that a device will go offline and online, collecting each time the defined number of samples per group. When

a device returns to online mode, it starts in a different and random position in the space. After collecting the defined number of groups the device's collection process ends.

- **Wi-Fi Noise (γ):** This parameter adds variations or noise to the RSS readings collected by all Wi-Fi devices.
- **Orientation Noise (ϕ):** Adds noise to the orientation readings collected by the moving device.
- **Distance Noise (σ):** This parameter adds noise to the displacement readings collected by the moving device.
- **Standard Path Loss Exponent (η):** Defines a standard path loss exponent for the LDPL model, which will be applied to the virtual space. If obstacles are added, this value will be affected and vary across the space due to the obstacles attenuation.

The Groups of Samples are only applied to the *Random* and *Smooth* motion models. A simulation session $session_i$ can be described as:

$$session_i = (sid, ns, sgroup, groups, vs_i) \tag{3.16}$$

where, sid is the session identifier, ns is the number of samples to be collected (total), $sgroup$ is the number of samples per group, $groups$ is the number of groups per session, and vs_i is the virtual space to simulate.

2.1 Sampling

As shown before a sample (s_i) collected by a device can contain Wi-Fi fingerprints and motion information. The samples are collected at each position visited by the device.

2.1.1 Wi-Fi Fingerprints (fp)

As described before, each Wi-Fi fingerprint (fp) contains RSS measurements for the APs in the range of the device. The range is defined by the device sensitivity, which can be configured. The standard sensitivity is -92 dBm ($RSS > -92$). Each rss measurement can be affected by noise and attenuation. The noise ($rssN_i$) added to each RSS measurement is defined by:

$$rssN_i = random(Gaussian) \times \gamma \tag{3.17}$$

where, a random gaussian value is multiplied by the Wi-Fi noise standard deviation (γ).

The attenuation due to obstacles ($rssOA_i$) between a device and an AP is given by:

$$rssOA_i = \sum_{j=1}^{n} oI_j \tag{3.18}$$

where, oI are the obstacles between the AP and the device. A line between the device and the AP is defined (Figure 3.7) and the attenuation of each obstacle (oI_j) intersected by that line is added to $rssOA_i$.

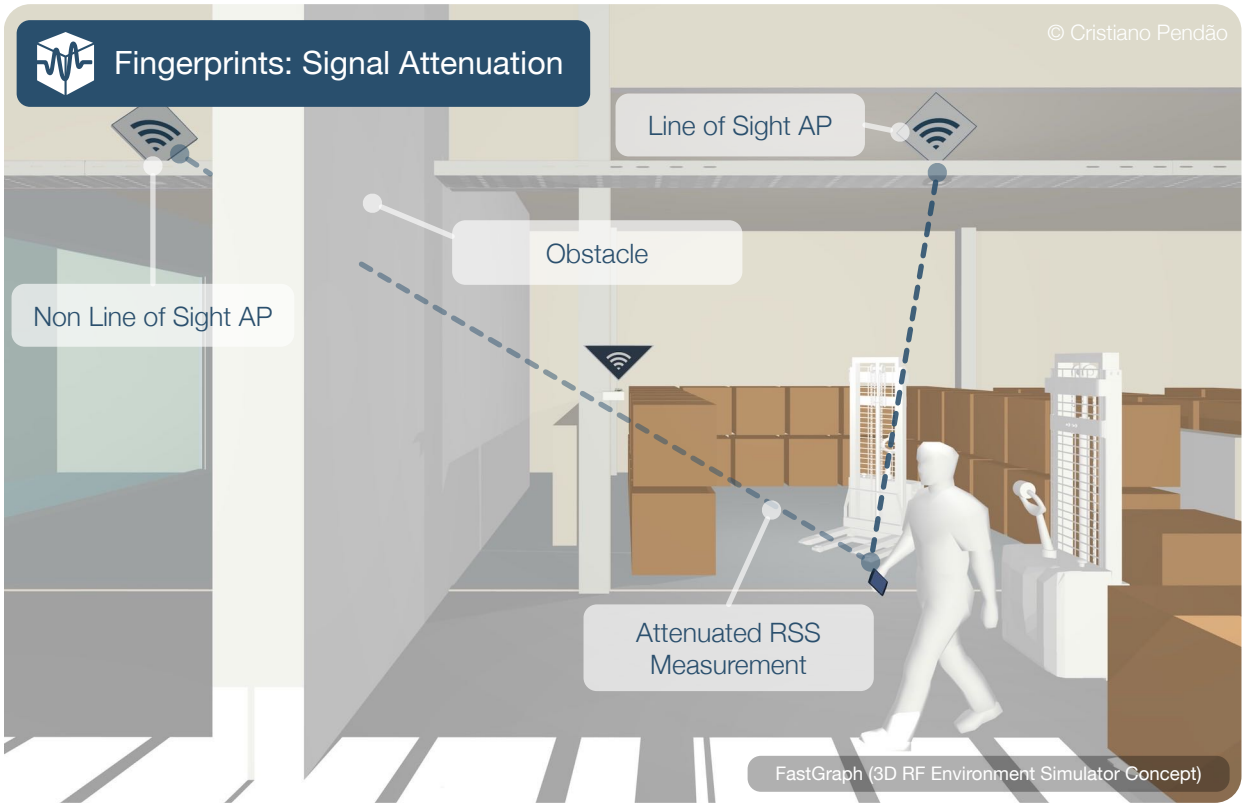

Figure 3.7: Fingerprints: RSS Obstacle Attenuation

Then the attenuation and noise parameters are used to affect the RSS measurement (rss_i) for an AP, using the Log-Distance Path-Loss (LDPL) propagation model (Page 21) such as:

$$rss_i = rss0 - \left(10 \times \eta \times log10\left(\frac{d}{d_0}\right) + rssN_i + rssOA_i\right) \tag{3.19}$$

2.1.2 Motion Data

When a device collects motion data, to each sample will be also added the current device orientation (he), and the displacement (dis) in relation to the previous sample. In order to better simulate the behavior of the sensors in real devices, such as smartphones, this two parameters, can also be affected by a configurable amount of noise, that vary for each sample collected (Figure 3.8).

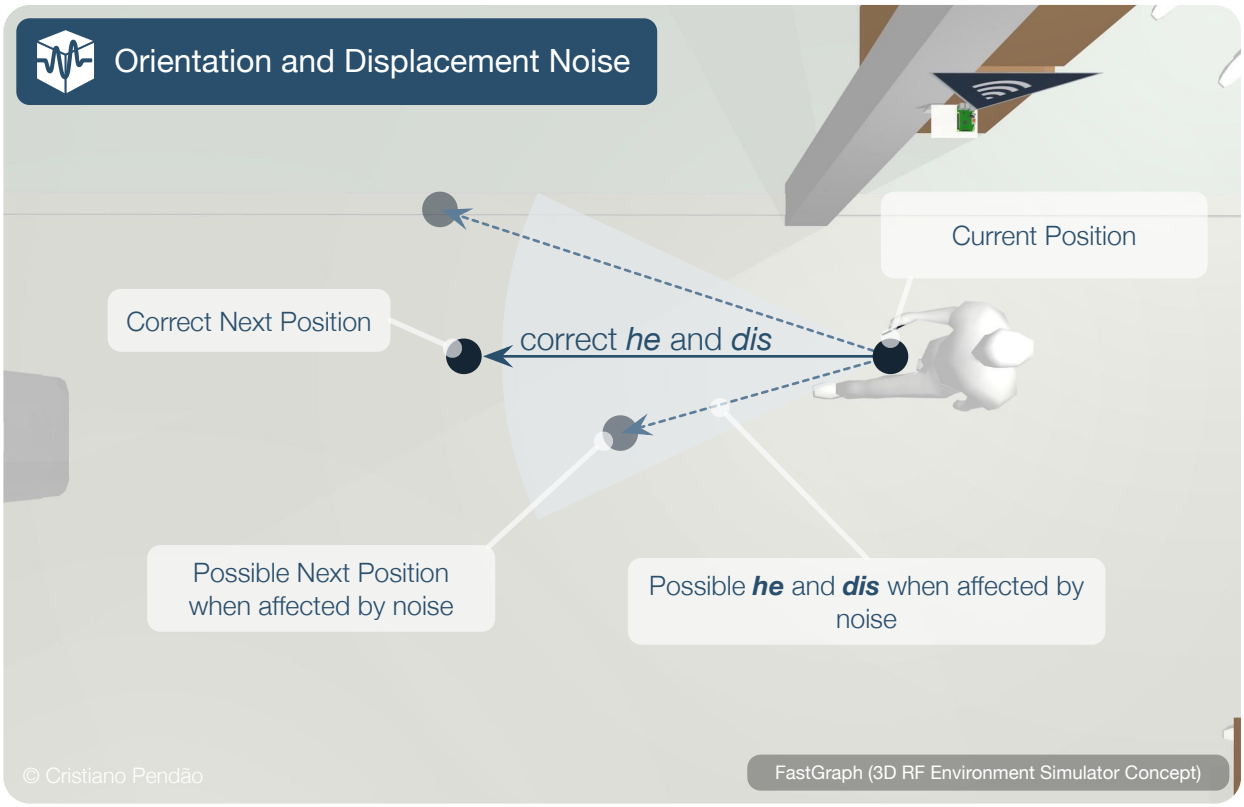

Figure 3.8: Motion Data: Orientation and Displacement

Therefore the orientation he_i affected by noise is given by:

$$he_i = he_i + (random(Gaussian) \times \phi) \tag{3.20}$$

where, ϕ is the orientation noise standard deviation.

The distance dis_i affected by noise is given by:

$$dis_i = dis_i + (random(Gaussian) \times \sigma) \tag{3.21}$$

where, σ is the displacement noise standard deviation.

2.2 Synthetic Dataset

When the simulation session ends, a new synthetic dataset has been generated, containing the samples collected by the virtual devices, including the ground truth information for samples and APs. Information about the simulation session is also available, such as the used noise values and the obstacles added. Therefore, a synthetic dataset ($dataset_i$) is defined by:

$$dataset_i = (sessioninfo, samples : (s_i, ..., s_n), aps : (ap_i, ...ap_n), objects : (o_i, ..., o_n)) \tag{3.22}$$

3 Implementation

This simulator was implemented in Java, and as a library can be easily extended or integrated in other Java programs or applications. Since no additional libraries are used or required, and the synthetic data is stored in SQLite databases, which is standard in Android, the simulator can even be integrated in mobile applications. This allows for example a direct feed of synthetic data to test mobile positioning applications directly in the smartphone.

The generated dataset stored in the SQLite database, that can also be exported to Comma-Separated Values (CSV) files, contains the following information:

- **Info:** ID, Space Dimensions, Motion Model, APs Count, Anchors Count, Wi-Fi Noise, Direction Noise, Distance Noise, Sampling Period, Obstacles Count, Standard Path Loss Exponent;
- **Samples:** Sample ID, Device ID, Timestamp, xPosition, yPosition, zPosition, Direction, Distance;
- **Detected APs:** AP ID, Sample ID, AP MAC, SSID, RSS, Distance, xPosition, yPosition, zPosition;
- **Objects (Obstacles):** Attenuation Factor, xDimension, yDimension, zDimension, xPosition, yPosition, zPosition;
- **Path Loss:** Sample ID, AP ID, Path Loss Exponent;

4 Summary and Discussion

In this chapter was described the developed RF environment simulator, which was used to generate valuable synthetic data, to test the proposed solution during the research and development process.

The simulator allows to create a 3D virtual space, with full control over the space characteristics, such as dimensions, level of noise and propagation parameters. Multiple 3D objects can be added to the virtual space, to define the space layout or to simulate different types of objects. For each object, can be configured a tridimensional position, the dimensions and the attenuation factor. The attenuation parameter allows to simulate different types of building

materials, which will have a different effect in the signals crossing the obstacles. The signals can be affect by multiple obstacles, creating realistic path loss variations across the space.

Multiple virtual radio transmitters can also be added to the space. These devices allow to simulate for example Wi-Fi APs, with a configurable transmission power. The simulator can easily be extended to support other radio technologies, such as Bluetooth and ZigBee beacons or cellular base stations.

Different types of sampling devices can be used to collect data. Fixed devices can be added to work as Anchors, collecting reference radio fingerprints at fixed and known positions. Moving devices allow to collect radio fingerprints and also orientation and displacement information about the travelled path.

Multiple motion models are available for the moving devices and can be used to simulate distinct motion patterns for humans or machines. The paths followed during a simulation session can be random or manually defined.

Different simulation sessions can be run in the defined virtual space. In each session, parameters regarding the devices behaviors, or the level of noise affecting the Wi-Fi and motion readings can be configured.

A simulation session generates a synthetic dataset, that in addition to the collected samples, also includes ground truth data, such as APs positions and sampling positions as well as information regarding obstacles, noise and propagation parameters.

The simulator was implemented in Java without using additional libraries, therefore can be easily integrated in other programs and even in Android mobile applications, which is useful for direct testing purposes.

Chapter 4

The FastGraph Approach

This chapter presents the approach behind the FastGraph solution. The base for FastGraph, in a simplistic way, can be seen as a solution for a high-dimensional trilateration problem, where RSS values are used to obtain the distances between the sample points and the detected APs using the Log-Distance Path Loss Model (LDPL) model. The core algorithm implements a method inspired in a Force-Directed 3D Graph to solve a set of equations in a way similar to Multi-Dimensional Scaling (MDS), providing unsupervised positioning without relying in a manual calibration, radio maps, or previous knowledge of the APs' positions. The 3D graph is continuously updated with the latest fingerprints, that can either come from crowdsourcing (unlabelled) of from Anchor nodes (georeferenced). With the proposed approach one argue that the more gathered data through the whole scenario the more consistent will be the graph and by consequence the lower will be the positioning error. Motion and orientation data can also be used to establish additional spacial constrains between the collected fingerprints in the Graph.

In addition, the algorithm is not based on assumptions such as that the propagation parameters are uniform across the space or even for the same AP, or that RSS measurements are unique for a specific area of the space. The only requirement in the FastGraph is the deployment of a limited number of Anchor nodes, that are based on the low-cost Raspberry Pi (RPi). The effort of installing them is much lower than the effort and costs of generating a dense radio map and keep it updated over a long period of time. In the absence of Anchors, the system may use a few labelled fingerprints as reference points. Another key aspect of FastGraph is that the system is ready to provide location estimations just a few minutes after installing the anchors.

The proposed solution can be applied to different areas, some of them will be discussed and evaluated in the subsequent chapters.

1 The Fundamental Principles

The fundamental idea for FastGraph is that a dynamic 3D Graph can be used to represent a physical and radio environment (Figure 4.1), that can be used for positioning and as an enhanced radio map. In this Graph, nodes represent Access Points (APs), reference nodes such as Anchors, or Samples from Moving Devices (see Figure 4.2).

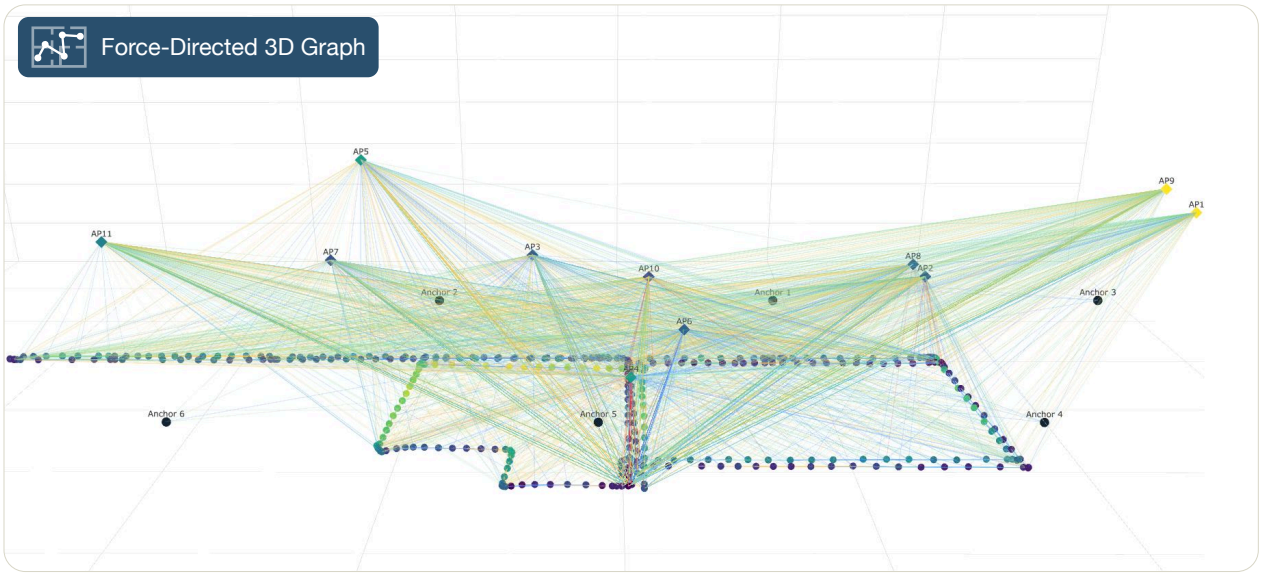

Figure 4.1: Example of a 3D Force-Directed Graph for a Radio Environment

Only the position of the reference nodes or Anchor nodes, is assumed to be known a priori. After each new sample is added, the Graph is adjusted to a minimum energy state. Each sample contains a fingerprint (Wi-Fi RSS measurements of the nearby APs) collected at a specific position. In addition, these samples can also contain motion and orientation information. A generic sample s_i is described as:

$$s_i = (device_{ID}, t, ((AP_1, RSS_1), (AP_2, RSS_2), ...), dis, he) \tag{4.1}$$

where $device_{ID}$ is the unique identifier of the device that collected the sample, t is the timestamp when the sample was collected. The list of (AP_i, RSS_i) pairs is the RSS measurements for the visible APs, dis is the linear displacement since the previous sample from the same device and he is the current heading.

1.1 The Graph Concept

Graphs can be used in different contexts and practical applications, for example to provide a more understandable view about connected networks or to represent possible relations between

different types of objects or data.

In a Graph there are nodes that are connected by edges, and can be undirected or directed from one node to another. Information can be represented in a Graph, and it is this graphical representation which helps us understand many of their properties.

In the proposed approach, the Graph can be seen as representation of a physical space and the radio environment in that space, where the nodes are the Wi-Fi APs and the samples collected by fixed or moving devices. This nodes are connected by edges that represent the spatial relation between them, and that work as physical constrains for the Graph. The edges length is estimated from the radio signal observed by a device at a given position, for example from an Wi-Fi Access Point, or by other information such as odometry based on other sensors.

Therefore the Graph in the proposed approach has different types of nodes and different types of edges that connect the nodes (see Figure 4.2).

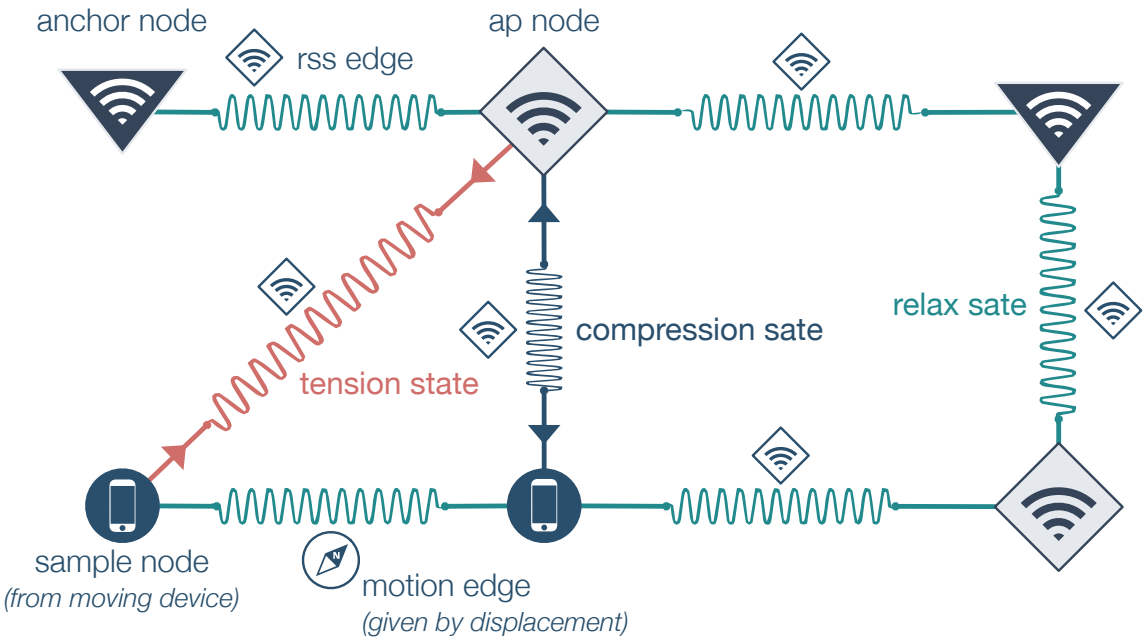

Figure 4.2: FastGraph Approach: Graph Elements

A node on the Graph can be:

- **AP node:** $node_i = (AP_i, p : (x, y, z))$, represents an Access Point$_i$ with unknown initial position p.
- **Anchor node:** $node_i = (A_i, p : (x, y, z))$, represents Anchor$_i$ with fixed and known position p. A Graph built without reference nodes can be rotated or translated in relation to the physical space. The reference samples from Anchors' are used to adjust the graph to the space configuration, in a process similar to Multi-Dimensional Scaling. In addition,

since the Anchors are fixed devices, the length of the edges that connect then to APs can be more accurately estimated, as will be explained later.

- **Samples node:** $node_i = (S_i, p : (x, y, z))$, each sample collected by a moving device originates a new sample node in the Graph, with unknown initial position. Each sample collected by a moving device is added to the Graph as a new node. A new node representing a new sample from a moving device can be connected to previous nodes of the same device by edges is the elapsed time is short. The length of these edges is establish using dead reckoning information, if available, or using the time difference to estimate displacement.

The position (p_{node}) of nodes on the Graph are subject to constrains:

- **Anchor nodes:** Known and fixed positions.
- **AP nodes and Sample nodes:** p_{node} must be within the 3D space and not complex (e.g. p_{node} cannot be underground or APs positions cannot be above the celling, and xi, yi, zi must be real numbers).

An edge in the Graph connects two nodes and can be seen as a spring with an elastic factor. The Natural Length (NL) of an edge can be obtained based on different types of information, depending on the two nodes that the edge connects. The edges can be:

- **RSS-based edge:** A RSS-based edge can be seen as representing the radio communication channel with an AP:
  - Anchor $\leftrightarrow$ AP: $e = ((node_i, node_j), NL, CL, ke1, \eta)$
  - Sample $\leftrightarrow$ AP: $e = ((node_i, node_j), NL, CL, ke2, \eta)$

  where NL is the Natural Length of the edge estimated based on the measured RSS using a propagation model, CL is the Current Length of the edge, $ke1$ and $ke2$ are the elastic constants, and η is the path loss exponent (LDPL).
- **Motion edge:** Connects two consecutive Sample nodes from the same moving device. The NL of this type of edge can be estimated through:
  - **Time-based:** $e = ((node_i, node_j), NL, CL, ke3)$, where the NL is the maximum displacement based on the time difference between the two samples. The maximum displacement is estimated considering a possible maximum velocity multiplied by

the time difference. A value for the maximum velocity can be assumed for each type of device, considering for example the motion characteristics of different vehicles or smartphones carried by humans.

- **Odometry/Inertial-based:** $e = ((node_i, node_j), NL, CL, ke4)$, where NL is based on more accurate displacement measurements between consecutive samples.

The edges based on accurate motion data, such as odometry, can be used to improve the natural length estimations, as they provide significantly more accurate displacement information, which results in better spatial constrains between samples.

An edge can be in one of three different states (like a spring):

- **Relax state:** $CL = NL$, means that the two nodes are at the correct distance.
- **Compression state:** $CL < NL$, means that the two nodes should be far from each other.
- **Tension state:** $CL > NL$, means that the two nodes should be closer.

1.2 Building the Graph

The Graph is built iteratively, based on new samples, one sample at a time.

- If the new sample is from an Anchor:
  1. Create an Anchor node to represent the Anchor (if not already exists).
  2. Create a AP node for each new AP, visible in the sample.
  3. Add or update the edges from the Anchor node to the APs visible in the sample.
- Otherwise if the new sample is from a moving device:
  1. Create a new Sample node.
  2. Create a AP node for each new AP, visible in the sample.
  3. Create one edge from each AP visible in the sample.
  4. Create an edge to previous Sample nodes of the same device (from time or odometry information).

1.3 Adjusting the Graph to a Minimum Energy State

The position of all nodes (except Anchor nodes) is updated, using a Force-Directed inspired method. This means that each node is subject to a Combined Force vector $c\vec{F}$, resulting from the individual forces applied by each edge connected to the node.

2 Building the Graph Iteratively

After deploying the anchors in the physical space, the system starts processing the samples collected by the Anchors to initiate the 3D Graph, ignoring the samples from moving devices. The Anchors' data is used to automatically estimate an initial position of the AP nodes and to adjust the graph to the real-world space configuration. When the algorithm detects that the position of the APs has stabilized, which normally takes only a few minutes, then the system begins to process also the samples of the moving devices, providing positioning. When the system is already operating, the additional samples provided by the Anchor nodes and the moving devices make the Graph to automatically evolve and keep the radio map updated for the whole environment.

2.1 Adding Nodes

When a new node is created and added to the Graph, its initial position p must be defined. This initial position is based on the type of the node, and based on the relations with the nodes already on the Graph. The position can also be calculated using different approaches, depending on the information available.

2.1.1 Anchor nodes

Anchor nodes are created at known positions when the first sample from the Anchor is processed. Subsequent samples from the same Anchor are associated to the previously created node.

2.1.2 AP nodes

AP nodes are created when they are observed for the first time, and they are initially placed according to the first detected RSS measurements, that are used to obtain a set of distances. This distance is used to define a sphere of possible positions for the AP. As at this point no other information is available to define the direction[1] or the hight of the AP. Therefore a random

[1]Direction as seen from the position where the sample was collected, relative to the X axis.

direction and a 2 meters Z position, in relation to the current floor, are initially defined. Then the initial XY coordinates are calculated (Figure 4.3). The initial position will be improved when the Graph is adjusted. The floor detection has been addressed in multiple works [85, 86] and can be achieved using air pressure readings obtained from barometric sensors. The Anchors in multiple floors can also work as floor reference.

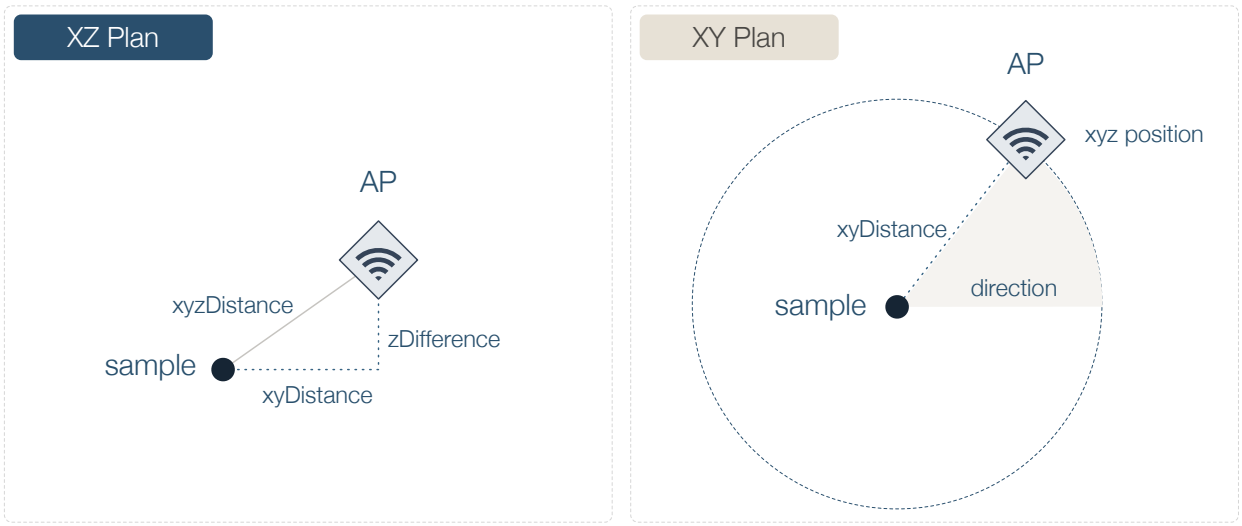

Figure 4.3: Access Point Initial Position Estimation

The initial position for an AP node is given by:

$$\begin{aligned} ap.x &= xyDistance \times \cos(direction) + sample.x \\ ap.y &= xyDistance \times \sin(direction) + sample.y \\ ap.z &= apInitZ \end{aligned} \tag{4.2}$$

where $xyDistance = \sqrt{\|xyzDistance^2 - (zDifference)^2\|}$, $apInitZ = 2$, $zDifference = apInitZ - sample.z$, and $xyzDistance$ given by the equation 4.9.

2.1.3 Sample nodes

Created for every new sample from moving devices. The node initial position on the Graph is dependent on the information available on the samples:

- **Independent sample:** A new sample that has no relation to previous sample nodes from the same device. When the elapsed time between a new sample and the device's previous sample on the Graph is high, the new sample is treated as an independent sample, as there is no information to establish a spatial relationship between the two.

- **Group samples:** When a sample is part of a sequence of samples from the same device, and is collected within a short period of time in relation to the previous sample.

For independent samples, the initial position is estimated based on the APs referenced in the sample that are already on the Graph, denominated shared APs.

The initial position of independent samples is defined by using one of the following methods:

- Computing an weighted centroid using the distances to the AP's (weight given by the RSS measurements) on the Graph and visible on the sample (Shared APs). The initial Z position is considered to be 1 meter above the ground level of the current floor.
- The same approach used for placing APs (if only one single AP is represented in the Graph).
- At the origin (0,0,1) if no other method can be applied.

In addition, when the Graph already contains a large number of samples and no Shared APs exist, fingerprinting techniques can be used by using the sample nodes previously added to the Graph as a radio map.

The weighted centroid method (Figure 4.4) is applied when two or more AP's visible on the sample are already on the Graph (Shared APs).

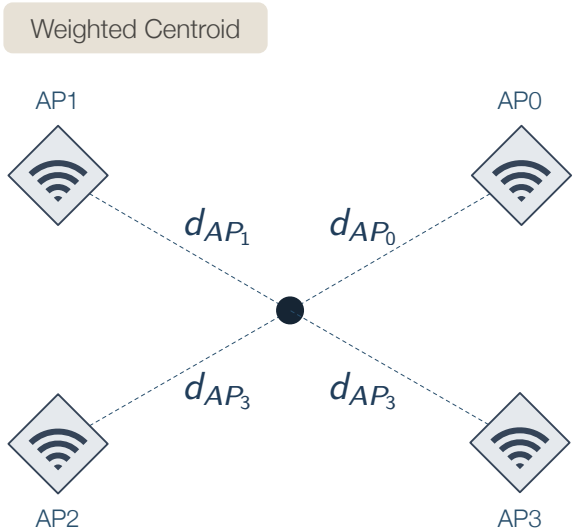

Figure 4.4: Independent Samples Initial Position Estimation (2 or more Shared APs)

The weight is given by:

$$k = \frac{1}{\sum_{i=0}^{n} \frac{1}{d_{AP_i}}} \tag{4.3}$$

where n is the number of shared APs, d_{AP_i} is the distance to AP_i given by eq. 4.9.

The initial position (x, y, z) for the sample node $node_j$ is then given by:

$$\begin{aligned} node_j.x &= k \times \sum_{i=0}^{n} \frac{1}{d_{AP_i}} \times AP_i.x \\ node_j.y &= k \times \sum_{i=0}^{n} \frac{1}{d_{AP_i}} \times AP_i.y \\ node_j.z &= k \times \sum_{i=0}^{n} \frac{1}{d_{AP_i}} \times AP_i.z \\ \vec{node_j} &= \{node_j.x, node_j.y, node_j.z\} \end{aligned} \tag{4.4}$$

Group Samples: Samples that are collected by the same device within a short interval of time, can be related by displacement measurements or by the time difference.

When a sample has direction and displacement information, dead reckoning techniques can be used to define the initial position of the sample, in relation to the previous sample collected by the same device (Figure 4.5). This method can be applied if the time difference between the two samples is small.

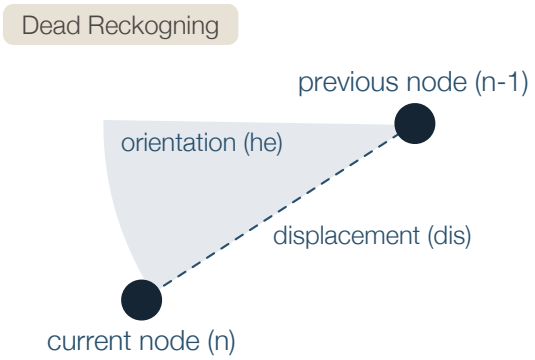

Figure 4.5: Group sample node initial position

1. The initial position of a sample node ($node_n$) with direction (he) and distance (dis) in relation to the previous sample node $node_{n-1}$ of the same device is given by:

$$\begin{aligned} node_n.x &= node_{n-1}.x + node_n.dis \times \cos(node_n.he) \\ node_n.y &= node_{n-1}.y + node_n.dis \times \sin(node_n.he) \\ node_n.z &= node_{n-1}.z \\ \vec{node_n} &= \{node_n.x, node_n.y, node_n.z\} \end{aligned} \tag{4.5}$$

2. When no motion information is available on a sample from a moving device, but that sample was collected within a short interval of time, in relation to the previous sample node ($node_{n-1}$) from the same device, the displacement (dis) from the previous sample can be defined based on the elapsed time ($node_n.t - node_{n-1}.t$). To obtain a distance from the elapsed time, the device is considered to be moving at a velocity between zero and the max velocity ($maxVel$), which depends on the moving device type:

$$dis = random(0, maxVel) \times (node_n.t - node_{n-1}.t) \tag{4.6}$$

The initial direction is a random value between -20 and 20 degrees in relation to the previous sample node:

$$node_i.he = node_{i-1}.he + random(-20, 20) \tag{4.7}$$

The [-20,20] interval gives an initial direction similar to the previous sample node, which is then corrected by the Graph adjustment process.

The node initial position (x,y,z) is defined using the same approach that is used when the distance and direction is known (Eq. 4.5).

2.2 Edges

With the exception of the Anchor nodes, the position of the nodes on the Graph will be influenced by the spatial constrains defined by the Natural Length (NL) of the edges.

2.2.1 RSS Edges

Sample node to AP node edges: Created when a new sample from a moving device is processed and the Sample node is created. These edges are RSS-based, therefore the Natural Length (NL) is set to the distance estimated from the measured RSS value. To obtain a distance from the RSS value the Log-Distance Path Loss Model (LDPL) propagation model is used. The LDPL model is frequently used to model the propagation of Wi-Fi signals [18, 28, 63, 64, 87, 88]. Using the LDPL model the RSS at a given position p at a distance d from an AP is given by:

$$RSS_p = RSS_0 - \left(10 \times \eta \times log10\left(\frac{d}{d_0}\right)\right) \tag{4.8}$$

RSS_p is dependent on the RSS_0 (measured at d_0, generally 1 meter), the distance to the transmitter (d) and the path loss exponent (η) that reflects the fall rate of the RSS signal from

an AP at a specific position.

Therefore, the NL of an edge e can be given by:

$$e_{NL} = 10^{\left(\frac{RSS0 - RSS_p}{10 \times \eta}\right)} \tag{4.9}$$

In contrast to many works in the field, where the path loss exponent of each AP is empirically set, the FastGraph approach considers that this parameters depends not only on the AP, but also on the location of the sample. It is well-known that getting the exact sources of attenuation in indoor radio propagation is not an easy task, as it involves many factors such obstacles or building materials. For that reason, it is considered that the communication channel from a given position to a specific AP, that is represented by an edge in the Graph, has singular propagation characteristics, and therefore a particular path loss exponent (η).

The value of η has a strong influence in the estimated natural length of the edges, and by consequence in the nodes' positions. For this reason, after adding a new sample node to the Graph, an initial estimation of the value of η is done for each RSS-based edge of the sample node. This initial estimation is obtained running a Gradient Descent Algorithm. This process is usually unable to find the optimal η, however provides a close estimation, that is later improved as more samples are processed and the Graph evolves. The complexity in estimating the path loss exponent, and the approach followed, will be detailed later in this chapter.

Anchor node to AP node edges: These are also RSS-based edges, and are created when an AP is observed for the first time by an Anchor. Existing edges are updated as new samples involving the attached nodes are gathered. In contrast to the Sample nodes, the Anchor nodes represent a set of observations. Therefore, the RSS measurements of the same AP can be combined, for example by averaging a given number of previous RSS values. Using combined measurements leads to more robust estimation of the NL with the LDPL model. Therefore the RSS_p for an Anchor edge is the arithmetic mean of all RSS values of the Anchor with reference to the AP:

$$Anchor \leftrightarrow AP : RSS_p = \frac{1}{n} \sum_{i=0}^{n} RSS_i \tag{4.10}$$

With the arithmetic mean of a set of RSS measurements, the error affecting this type of edges is smaller than the error affecting the edges where the length is calculated based on a single RSS measurement. Therefore, this type of edges can have more influence on the overall Graph, as we have an higher degree of certainty on their length.

All the samples collected by an Anchors are therefore represented by a single node in the

Graph, and the edges for that node are updated with the new arithmetic mean of the RSS values, when a new sample from that Anchor is processed. For this reason, these edges store a fixed size list of the latest RSS values observed by an Anchor. When the max size of this list is reached, the oldest values are eliminated. This allows the Graph to adjust to the changes in the radio environment. Moreover, this approach also helps to improve the algorithm performance by reducing the number of the nodes and edges in the Graph.

2.2.2 Motion Edges

Sample to Sample edges: Created between samples of the same moving device, when possible. The natural length is given by the displacement obtained from odometry or inertial data, or based on the time elapsed between samples.

- **Odometry/Inertial Motion Edges:** The length of a motion edge e_{NL} from Odometry/Inertial data is given by the actual sensor measurement:

$$e_{NL} = odometryDisplacement \tag{4.11}$$

- **Time-Based Motion Edges:** If no explicit distance measurements are available, the edge natural length is calculated based on the time different between the two samples, when they belong to the same device and are separated by a short interval of time.

  The natural length of a motion edge e_{NL}, is a random value between zero and the max possible distance travelled by the device, considering the elapsed time since the previous sample node:

$$e_{NL} = random(0, maxDistance) \tag{4.12}$$

  with:

$$maxDistance = maxVelocity \times eslapsedTime \tag{4.13}$$

  where, $maxVelocity$ is a configurable parameter that defines how much a device can move in a specific amount of time (e.g. 1 m/s) and $elapsedTime$ is the time difference between two consecutive samples ($s_i.time - s_{i-1}.time$). A specific value for $maxVelocity$ can be assumed for each type of device, considering for example the motion characteristics of different vehicles or smartphones carried by humans.

The accuracy of Time-Based natural length is understandably lower than the natural length based on explicit distance measurements by sensors. However when explicit measurements are not available this helps to restrict the possible initial position of the node to a smaller area. Moreover, the negative impact of the error in Time-Based estimations can be limited, since the uncertainty in the length of these edges can be controlled by using a lower elasticity (ke).

3 Adjusting the Graph

The 3D Graph evolves when a new sample is processed and the Graph is adjusted. The position of the APs and the position of the moving devices (Sample nodes) is estimated by adjusting the Graph, in order to respect the 3D spatial constrains defined by the edges. The Anchor nodes provide spatial references, in order for the Graph to keep the correct rotation and scale. To adjust the Graph to a minimum energy state a force-directed approach is used [89].

Following the spring analogy, each edge has associated a force vector that is calculated based on the difference between the Natural Length (NL) and the Current Length (CL) of the edge, that is then multiplied by an elastic constant (ke), that gives the edge more a less elasticity. As explained before, each edge can be in relax state, in tension or in compression, depending on the current values of NL and CL. The NL defines the desired length for an edge, however as explained, the NL of each edge can be estimated with different degrees of accuracy, depending on the data source. An individual edge can be forced to compress or to extend to allow a node to move to a position that suits better the majority of the edges. In these cases the length of the edge will be different from it's natural length, and is named Current Length (CL). Therefore, the CL is the actual 3D distance between a $node_i$ and a $node_j$ on the Graph at a specific time, and is given by:

$$\vec{e_{CL}} = node_i(\vec{x}, y, z) - node_j(\vec{x}, y, z) \tag{4.14}$$

$$e_{CL} = \|\vec{e_{CL}}\| \tag{4.15}$$

As will be explained in more detail later, the difference between the natural length and current length is used to define a force associated to the edge.

The elastic constant (ke) is different for each type of edge, allowing them to have different contributions to node Combined Force ($c\vec{F}$), that is the sum of all forces applied by each edge

of the node, and by consequence different influence in the nodes' positions. As mentioned before, edges connecting Anchors and APs are less affected by signal fluctuations in the RSS measurements, and for this reason have an higher value of ke. In the edges connecting two sample nodes the principle is the same. In the odometry or inertial based edges, the length is defined by distances measured by actual sensors. Therefore, it is natural that these edges have higher ke than the edges defined by the temporal difference between samples.

In addition, since the intensity of Wi-Fi signals decreases with the distance following a logarithmic function (Figure 4.6), variations in lower values of RSS translates in higher differences in the calculated distance.

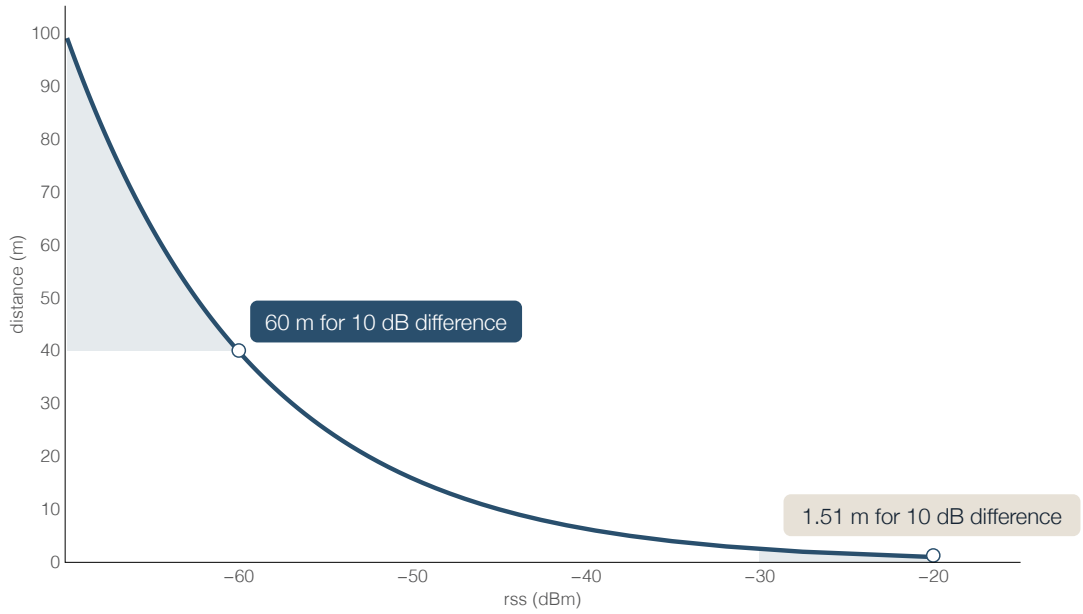

Figure 4.6: Distance variation with RSS ($\eta = 2.5$)

Considering this, the error affecting a distance calculated from a low RSS value will be probably significantly higher than the error affecting the distance calculate based on a higher RSS value. In addition, lower RSS values can mean that the signal was heavily attenuated by obstacles and affected by noise, reducing even more the confidence on the estimated distance.

Hence, the ke value for RSS edges is adjusted based on the RSS value, giving more strength to edges with higher RSS.

The adjusted elasticity constant for an RSS edge (ke') is a function of the initial elastic constant ke and the Natural Length (NL) of the edge, and is given by:

$$ke' = ke \times 10^{(ke \times \|NL\|)} \tag{4.16}$$

The tension magnitude (te) applied by an edge e on the connected nodes is given by:

For a RSS edge:

$$te = ke' \times (e_{CL} - e_{NL}) \tag{4.17}$$

For a Motion edge:

$$te = ke \times (e_{CL} - e_{NL}) \tag{4.18}$$

The force vector ($e\vec{F}$) is then given by:

$$e\vec{F} = (\vec{distance} \times \hat{u}) \times signum(te) \times log\,(1 + \|te\|) \tag{4.19}$$

where, $\vec{distance} \times \hat{u}$ is the distance vector between the positions of the two nodes ($\vec{p_{node1}} - \vec{p_{node2}}$) times the unit vector, $signum(te)$ is the signal, positive or negative, of the tension magnitude, and $log\,(1 + \|te\|)$ is the logarithmic of the tension module.

The total force applied to a $node_j$ ($c\vec{F}_j$) is the vectorial sum of all the forces associated to each edge ($e\vec{F}_i$) connected to the node (Figure 4.7):

$$c\vec{F}_j = \sum_{i=1}^{n} e\vec{F}_i, \text{for all node edges} \tag{4.20}$$

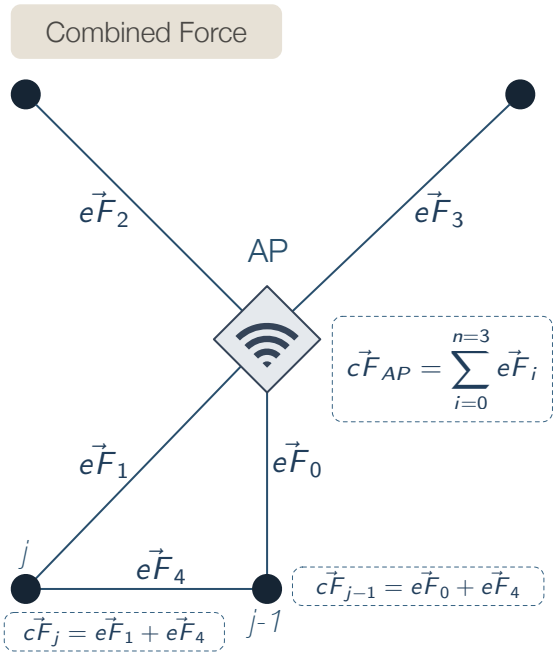

Figure 4.7: Combined forces

The Graph adjustment can affect all nodes or a specific sub-set of nodes. The Graph adjustment is an iterative process that is performed when a new sample is processed. The adjustment process stops when all the nodes on the Graph move less than a defined threshold, meaning that the Graph reached the minimum energy state. In each iteration the nodes are

moved based on a velocity vector ($\vec{vel}_j$) that is the linear combination between the current combined force ($c\vec{F}_j$) and the previous combined force ($c\vec{F}_{j-1}$), in order to preserve part of the momentum from the previous iteration. Hence the velocity vector of a node in an iteration j is given by:

$$\vec{vel}_j = \beta_1 \times c\vec{F}_j + (1 - \beta_1) \times c\vec{F}_{j-1} \tag{4.21}$$

where β_1 is a parameter.

Each node is moved in order to minimize the forces applied to it. The new position is calculated by adding to the current position the velocity vector multiplied by a factor T:

$$\vec{pos} = \vec{pos} + \vec{vel} \times T \tag{4.22}$$

The T factor defines how much the node moves in a single iteration in the direction defined by the velocity vector. This controls how fast the node will converge to the position that minimizes all forces.

As explained before the $c\vec{F}$ is a vectorial sum of all the forces applied by the edges connected to a node. Therefore, more edges connected to a node results into an higher $c\vec{F}$ magnitude. This is the case of AP nodes, since they are connected to all Anchor nodes and Sample nodes within radio communication range. Easily, the number of edges can grow to hundreds or even thousands. This can lead the node to enter into an oscillatory motion between positions (Figure 4.8), increasing the time needed for the Graph to converge to a minimum energy state. In addition, this oscillatory motion can even prevent the Graph to reach the minimum energy state, even if only one node is oscillating.

In specific conditions, the oscillatory motion can even produce forces that increase progressively in magnitude, with edges changing between tension and compression states when the node oscillates between two opposite positions.

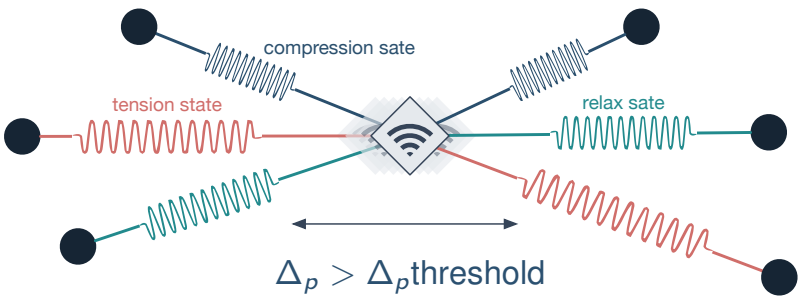

Figure 4.8: Unstable node in oscillatory motion

This problem appears when a large number of mixed forces push and pull a node, the $c\vec{F}$

vector can have opposite directions from one adjustment iteration to another, due to the node move beyond the correct position due the high magnitude of the $c\vec{F}$ vector. In the next iteration, the additional movement in relation to the correct position produces higher forces that will pull the nodes back, and this will make the node to go even more beyond the correct position in the opposite direction, staying in an increasing oscillatory motion. This oscillatory motion prevents the Graph to reach the minimum motion threshold that defines the minimum energy state. Reducing the motion threshold would not solve the problem, and in normal conditions will make the adjustment process to stop before the minimum energy state is reached.

The solution is to keep an high elastic potential energy and momentum, moving the node faster when it is far from the target position (that results in the minimum energy state for the node), and reduce the elastic potential energy, moving the node slower when the node is near the target position, in order to avoid the node to go beyond the target position and enter into oscillatory motion.

To accomplish this, a process was designed to adjust the magnitude of the $c\vec{F}$ depending on the overall nodes' acceleration. The influence of the total combined force on a node position is therefore adjusted depending on the acceleration of all nodes in the Graph (Figure 4.9).

This is controlled by dynamically adjusting the value of T at each iteration.

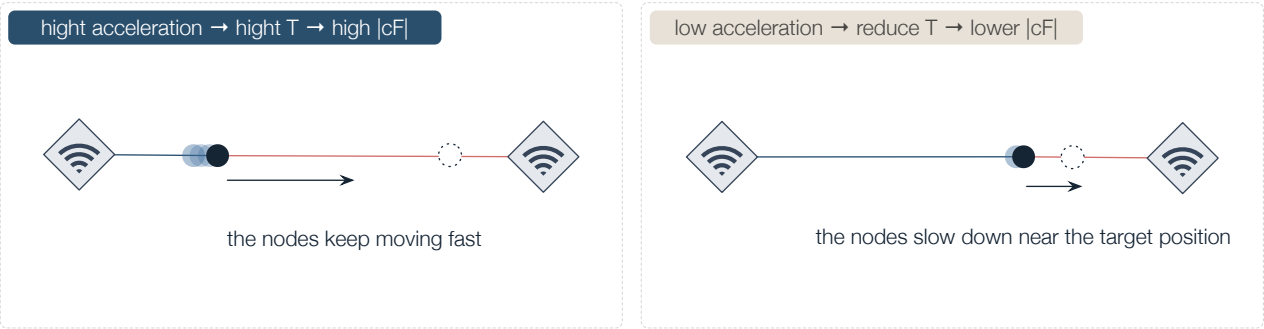

Figure 4.9: Dynamic Acceleration

First the position variation, from one iteration to the next, for all nodes is calculated:

$$\Delta pos_{nodes} = \frac{1}{n} \times \sum_{i=1}^{n} \Delta pos_{\vec{node_i}} \tag{4.23}$$

The new value of T (see equation 4.22) to be used in the next adjustment iteration is then given by:

$$T = \frac{2}{\gamma + \Delta pos_{nodes} \times \Delta_t} \tag{4.24}$$

where γ is the amortization factor, and Δ_t is the time taken by the previous iteration.

The T factor increases when we have higher acceleration of the nodes, and this way forces have more impact on the node position. When the overall nodes acceleration is lower the T decreases, resulting in finer adjustments to the node position, reducing the oscillatory motion.

The same T is applied to all nodes, otherwise nodes affected by the same forces could move more than others. This method keeps the adjustment process only while there is significant acceleration in the Graph (nodes moving), and not if the movement is generated by only a few unstable nodes in oscillatory motion. Moreover, with this solution the Graph converges to the minimum energy state faster, which improves the performance of the algorithm in real time applications.

3.1 Nodes Adjustment Approaches

The nodes of the Graph can be moved using different approaches:

- **Independent Node Adjustment:** Each node of the Graph is moved based on its individual combined force vector ($c\vec{F}$). In addition, depending on the nodes affected by the adjustment this can be:
  - **Full adjustment:** All the nodes of the Graph are adjusted in each iteration.
  - **Partial Adjustment:** Only a subsequent set of nodes connected to a new node just added to the Graph are adjusted.
- **Group Adjustment:** Each node belonging to a group is moved based on the individual $c\vec{F}$ vector but also taking into account the group combined force vector ($g\vec{F}$).

3.1.1 Independent Node Adjustment

Full Adjustment: In a full adjustment all nodes in the Graph will be adjusted, therefore for each node in the Graph the combined force vector ($c\vec{F}$) is calculated and, then based on the $c\vec{F}$ vector, the node will be moved. In this approach the adjustment process starts from the oldest node in the graph. This way the position of the APs and the previous nodes will be less affected by the possible error in the initial position of a new node. The nodes can be moved immediately after their $c\vec{F}$ have been calculated, or only after all nodes' $c\vec{F}$ have been calculated. Each one of the approaches can be useful, depending on the information available in the sample processed.

Moving a $node_i$ immediately after calculating the $c\vec{F}_i$ will influence the $c\vec{F}$ of all nodes connected to the node moved in the same adjustment iteration. Each adjustment iteration is preformed in n steps, one step for each node (Figure 4.10). The new position of the $node_i$

produces new forces in the edges connecting to the other nodes. Therefore, the $c\vec{F}$ of the following node to be adjusted $node_{i-1}$ will be calculated taking into account the new state of the Graph. This can produce problems when we use high values for the elastic constant between sample nodes ($ke1$), because in this case the force produced by this edges can overcome the forces of the edges connecting to the AP's, and even lead to oscillatory motion.

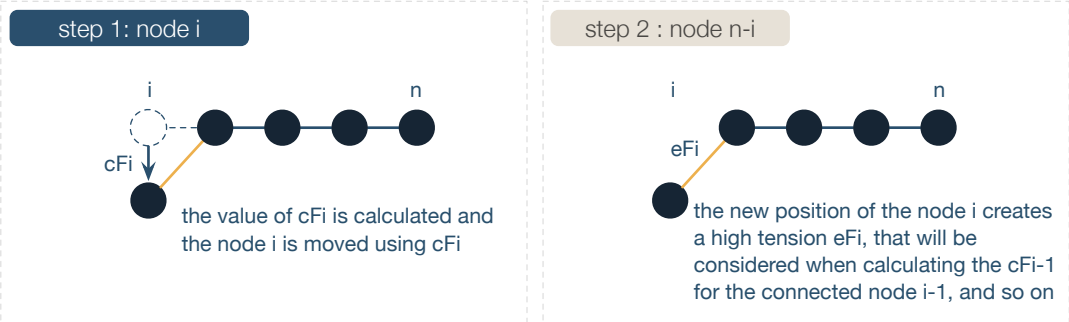

Figure 4.10: Calculate a node $c\vec{F}$ and move the node

An alternative approach is to move the nodes only after all $c\vec{F}$ forces have been calculated, therefore each adjustment iteration will have only two steps in total (Figure 4.11). The new node will affect only the nodes connected to it directly, and the node influence will propagate slower on the Graph, which can be desirable in order to propagate less the effect of one node in their node chain. The overall node move will be also lower as only the new node and the nodes directly connected will be moved in the current iteration.

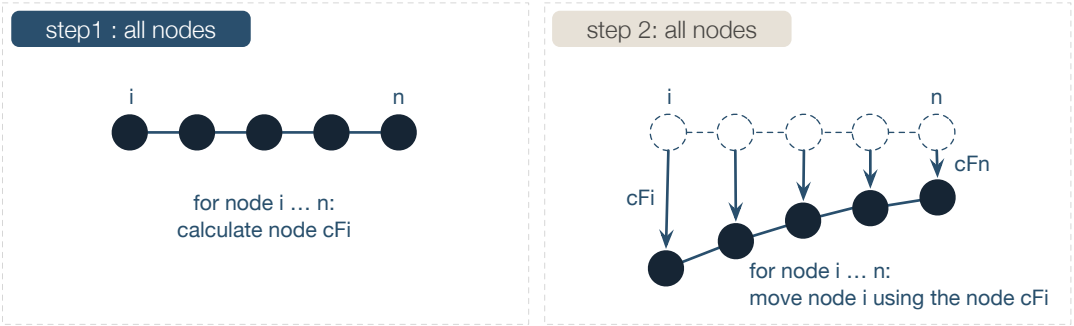

Figure 4.11: Calculate all nodes $c\vec{F}$ and them move the nodes

After testing, it was concluded that the second approach has more advantages when performing full graph adjustments in a Graph with a large number of nodes and forces. The advantage of performing full adjustments is that we can improve the position of all the nodes in the Graph, even the very old ones, when adding more sample nodes. However, the adjustment time in full adjustment depends on the number of nodes in the Graph, and for this reason full adjustments may not be ideal as the main adjustment process for Graphs with a large number of nodes.

Partial Adjustment: With this in mind, a partial adjustment method was designed to improve convergence speed in order to improve scalability, and to reduce the impact of the samples' initial position error on the Graph. The partial adjustment can be applied to obtain a rapid position estimation. In Graphs representing a large space and with a large number of nodes, adding a new sample node in one area will not affect the nodes in other areas, therefore not all nodes need to be adjust. In addition, as explained before, depending on the information available on each sample, the initial position estimation can be more or less affected by error, therefore the nodes already on the Graph can be used to improve the initial position of the new sample node, without this having limited impact on the other nodes.

For these reasons, in this method the adjustment process starts on the most recent node added to the Graph. Then a configurable range defines how many levels of nodes will also be adjusted (Figure 4.12). As an example, with a range $r = 0$ only the current node will be adjusted. With a range of $r = 1$ only the nodes directly connected to the new node will be adjusted, such as the APs or the previous sample node. In this method the previous sample nodes of a sequence will help to quickly correct the new node position, mainly when we have odometry/inertial based edges.

A partial adjustment is therefore more efficient than performing a full adjustment. It only affects a set of nodes depending on their connections and the defined range. On other hand, when using a small range, the positions of old samples are not affected, and no further improvements are done to their position based on the information from the newer samples.

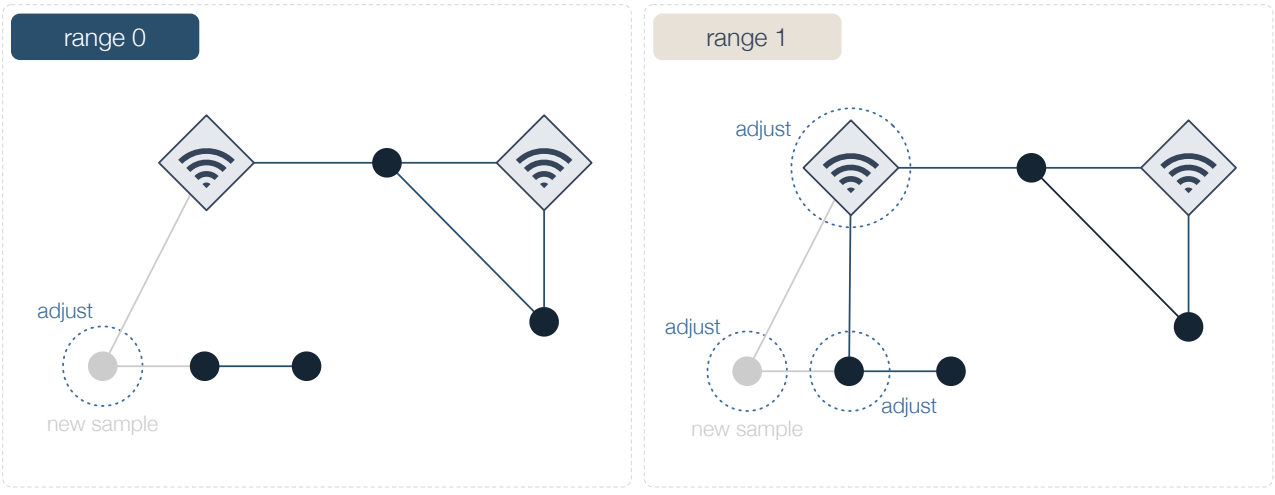

Figure 4.12: Partial adjustment with different ranges

3.1.2 Group Adjustment

As explained previously, the Independent Node Adjustment approach moves the nodes based on their individual $c\vec{F}$. However, in applications where motion information is available, this approach don't take full advantage of the potential of the Motion-Edges.

When processing sample nodes connected by Motion-Edges the orientation and distance information are used to establish spacial constrains between the samples. The algorithm can use these additional constrains to improve the APs positions and the edges path loss exponents. As explained before, the Sensor-Based Motion edges are less affected by error, and for that reason the *ke* of this type of Motion-Edges is higher (lower elasticity) than the *ke* of the RSS-based edges between sample nodes and APs. When the *ke* for the RSS edges is high, the error in the edges' length can degrade a good position estimation that was obtained based on the motion information, using dead reckoning techniques (Figure 4.13).

Considering this, it seams obvious to maintain the *ke* of the Motion edges significantly higher than the *ke* of RSS edges. However, this leads to another problem. The dead reckoning process takes into account the orientation and displacement in relation to a previous position, and for that reason, although it can place accurately a sample node in relation to the previous node, all the nodes positions will be dependent on the position of the first sample of the sequence.

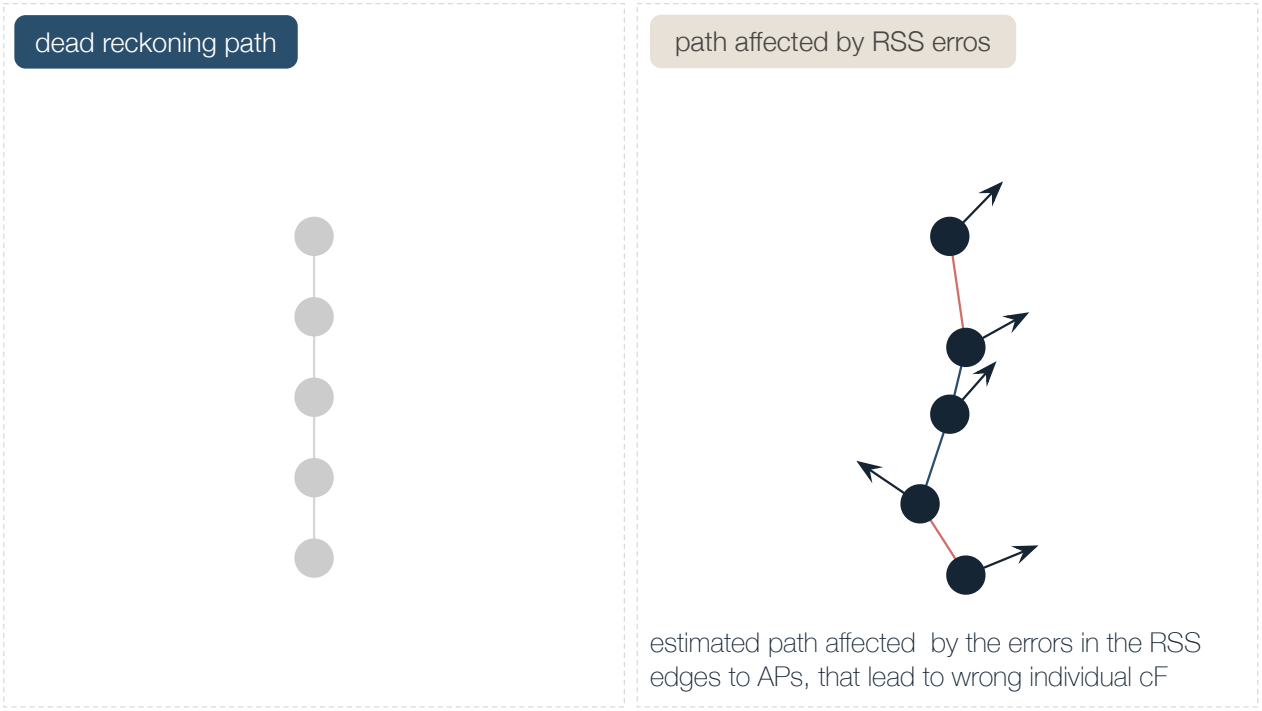

Figure 4.13: Path degradation due to RSS noise

The error in the first sample node, will therefore propagate for the next nodes if not rapidly corrected (Figure 4.14). This can have a significant impact because the initial position of the

first sample of a sequence is estimated only based on noisy RSS measurements and therefore more susceptible to errors. In addition, a low number of shared APs on the Graph, the error in the AP's position that may be still converging to a correct position, and the error in the path loss exponent initial estimation (η), all contribute to the error in initial position of the first sample.

Considering all these aspects, in fact it makes sense to maintain the Motion-Edges ke higher than the ke of the RSS edges to APs, in order to prevent the path (given by dead reckoning) to be destroyed by noisy RSS-based edges. But is also necessary to keep the RSS-based edges influence to correct errors in the first sample initial position.

Solving this problem relying only on a good balance between the ke values proved to be very difficult, usually leading to a destroyed path or to a whole path affected by the error of the first sample.

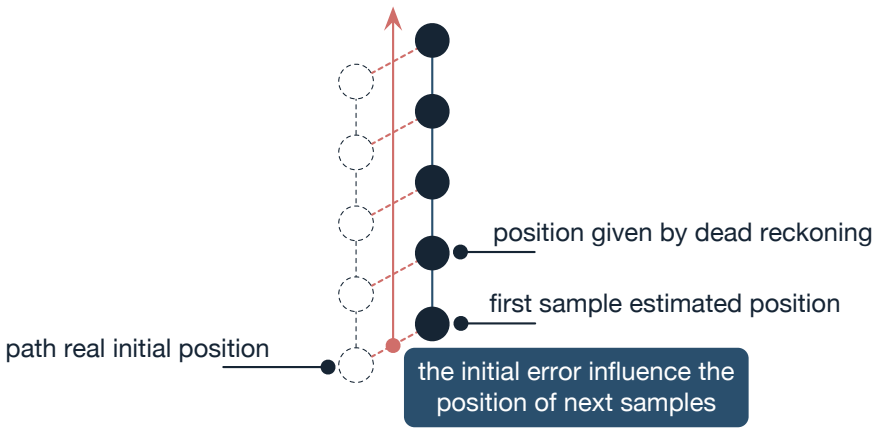

Figure 4.14: First sample error propagation on a path

To address this problem, the algorithm should be able to explore the information from both types of edges, taking advantage of the strengths of each one. Let's consider the characteristics of the two types of edges.

The Motion edges:

- Can preserve accurate spacial constrains between samples (paths), that were defined by dead reckoning.
- The motion edges alone can't position a sample in the space. In addition, the dead reckoning process is based on a previous sample position, therefore the error of previous sample nodes will propagate to the next nodes.

The RSS edges:

- Are based on less accurate and noisy measurements than the Motion edges, therefore can destroy the spacial constrains defined by the Motion edges.
- But can position the sample nodes in the space, and help to correct problems in the dead reckoning process.

The Group Concept concept: Considering these characteristics leaded to the Group Adjustment concept. A sequence of sample nodes from the same device connected by Motion edges can be seen as a Group, and the algorithm can take advantage of the Group, to improve the position estimation for each node of the Group. In addition, improving the Group also improves the initial position estimation when a new sample node is added to that Group, as this estimations are based on the previous related samples. Therefore, the Group Adjustment method moves each node belonging to a Group, based not only on their individual $c\vec{F}$ vector, but also taking into account in the Group Force vector ($g\vec{F}$).

The Group Force: The $g\vec{F}$ is the vectorial sum of all the Group' nodes individual $c\vec{F}$ (Figure 4.15).

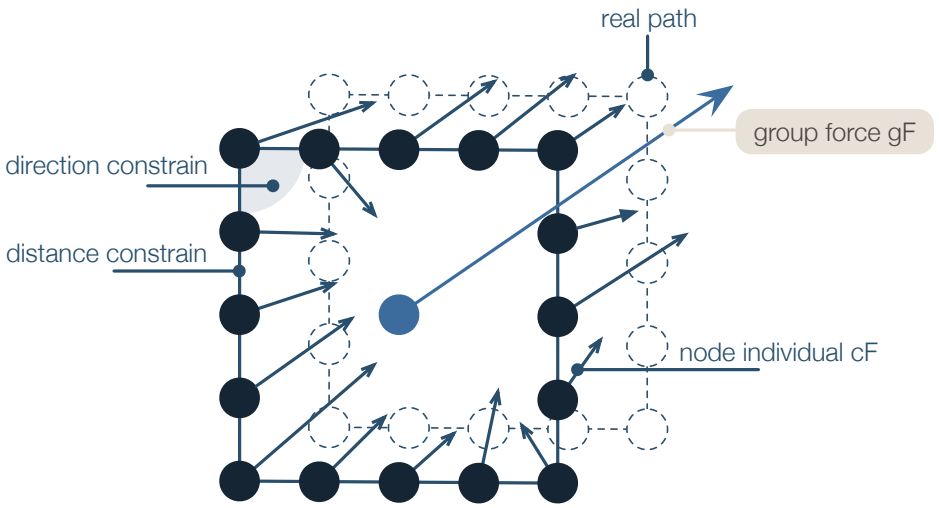

Figure 4.15: Group Force

For each $node_i$ of the Group, the $c\vec{F}_i$ is calculated and added to the $g\vec{F}$, that is normalized using the number of nodes (n) in the Group:

$$g\vec{F} = \frac{1}{n}\sum_{i=0}^{n} c\vec{F}_i \tag{4.25}$$

Then for each $node_i$ a linear combination between the node individual force $c\vec{F}_i$ and the

group force $g\vec{F}$ provides the node final combined force vector $c\vec{F}_i'$:

$$c\vec{F}_i' = c\vec{F}_i \times (1 - \beta_2) + g\vec{F} \times \beta_2 \tag{4.26}$$

The value of β_2 controls the percentage of each force that will be used to move the node. For example $\beta_2 = 1$ means that only the group force vector ($g\vec{F}$) is used to move the node, and $\beta_2 = 0$ means that only the node individual $c\vec{F}$ is used.

This linear combination allows the nodes to be moved as a Group, rapidly correcting the error in the initial position and preserving the path shape. Each node will benefit from the Group but can also keep some adjustment freedom as an individual element. The sample nodes will not be totally crystallized as a Group. This is important in order to allow individual corrections in the nodes of a path, that can be affected by errors for example in the dead reckoning information, such as the direction drift, as will be explained later.

After the final force $c\vec{F}'$ is calculated for all nodes of the Group, the nodes are moved using their $c\vec{F}'$:

$$p\vec{o}s = p\vec{o}s + c\vec{F}' \times T \tag{4.27}$$

The contribution from each node is used to improve the other nodes position, even the previous nodes. The next nodes to be added to the group will also leverage from the correction made to the group.

This method allows a sequence of nodes to be moved in the Graph without changing the path shape that was given by the dead reckoning information. In addition, moving the nodes based also on the Group force minimizes the impact of individual errors. Moreover, this approach also helps to improve the position of the APs. By preventing the related samples spatial constrains (for example a path) to be destroyed, provides additional information to adjust the APs. Another advantage is that the Group constrains help to progressively learn and adjust the path loss exponent (η) of each edge. The progressive path loss learning will be addressed further on.

3.1.3 Orientation Drift

Sensors used to measure orientation, such as Inertial Measurement Units (IMUs) are often affected by drift. The drift impact increases with the time and the distance travelled (Figure 4.16).

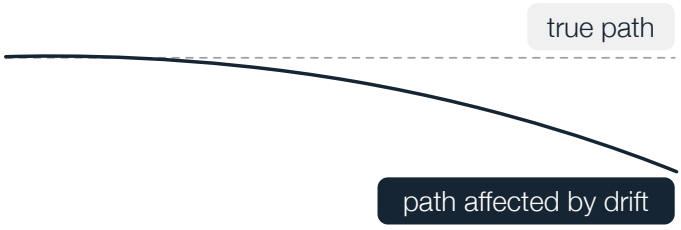

Figure 4.16: Drift Example

Trajectory tracking approaches based in pure dead reckoning are highly affected by drift. In the FastGraph the dead reckoning is only a part of the solution, but the orientation is an important parameter to define the travel trajectory and the position of samples belonging to a path. The drift effect can degrade this estimations, leading to lower positioning performance.

In order to evaluate the impact of the drift in the FastGraph approach, experiments were done using controlled synthetic data, generated by the developed simulator. In a first experiment, the graph algorithm was configured to operate as a pure dead reckoning solution. Then the orientation readings were affected by drift. This experiment provide an idea of the impact of the drift when the Wi-Fi has no influence over the sample nodes positions.

The drift evolution can be described as increasing linearly over time [90]. Following this, the drift added to the orientation angle of each sample (θ_i) was based on a linear function given by:

$$\theta_i = \theta_{i-1} + \theta_{drift} \times i \tag{4.28}$$

The position of each sample of the sequence is given by the orientation and distance in relation to the previous sample position. The previous sample position may already be affected by drift. Considering this, a position where all the previous n samples are affected by drift can be given by:

$$\begin{aligned} x_n &= x_0 + \sum_{i=0}^{n} \vec{displacement}_i \times cos(\theta_{i-1} + drift \times i) \\ y_n &= y_0 + \sum_{i=0}^{n} \vec{displacement}_i \times sin(\theta_{i-1} + drift \times i) \end{aligned} \tag{4.29}$$

This means that even with a small amount of drift in each sample, the impact in the estimated position will increase over time. This effect is shown in the Figure 4.17, that is the result of a pure dead reckoning solution, where the Wi-Fi information has no effect on the samples position. Based on the Equation 4.28 was added drift (with $\theta_{drift} = 0.5°$) to the orientation readings.

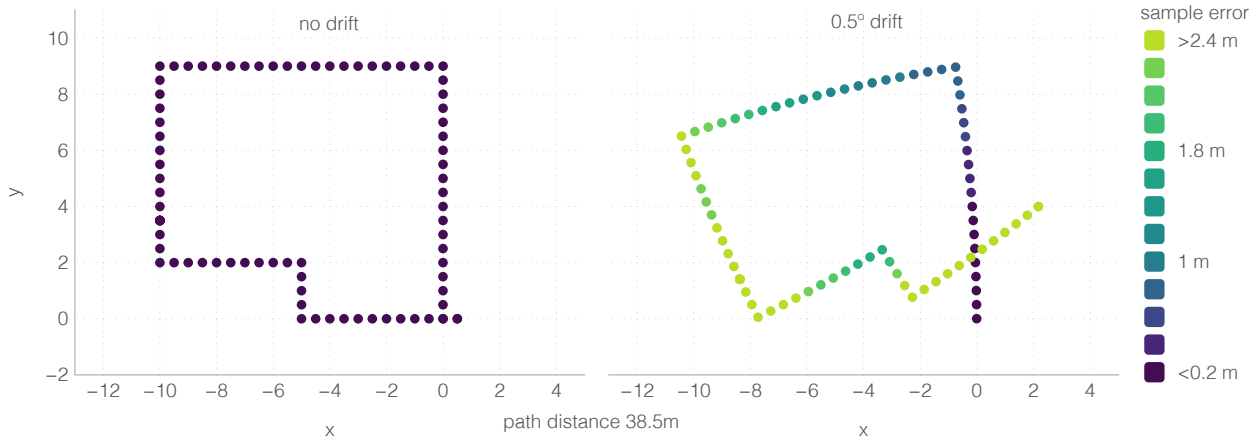

Figure 4.17: Path with and without drift (using path adjustment method)

The path affected by 0.5° of drift results in a progressive increasing position error along the travelled distance (38.5 meters). Increasing the travelled distance, increases the effect of the drift in the position estimation, and by consequence the positioning error. In this research domain, the error is usually defined as being the percentage between the Euclidean error and the traveled distance, therefore in this case: $2.4m/38.5m = 6.2\%$.

Solutions for Drift: In the previous experiment we saw how a sensor with a 0.5º of drift can affect a pure dead reckoning solution.

Some works have addressed this problem, and developed methods to reduce the effect of the drift, in order to improve the orientation, using for example wearable sensors [90]. However, this type of approach used to remove the drift effect, often depends on the type of sensors, the type of movement, the travelled distance, or requires previous calibration.

Each sensor can be differently affected by drift. For that reason one of the parameters of commercial IMU units is in fact the amount of drift over time. Advanced IMUs use techniques such as sensor fusion to improve estimations and correct the drift, combining the gyroscope and Active Heading Stabilization (AHS) that uses the magnetometers to estimate the gyro bias. This way, drift in heading can be as low as 1 deg after 60 minutes, providing accurate orientation measurement even in long time operation. In smartphones the sensor fusion is also applied to improve the orientation readings over time.

Despite of addressing the drift problem being out of the scope of this thesis, and although

there are IMU units on the market that provide accurate orientation readings with low drift, an additional experiment was performed to evaluate how the FastGraph method handle data affected by drift.

The Impact of Drift on the Proposed Solution: With the combination of Wi-Fi and dead reckoning it is expect that the developed approach will be able to reduce the impact of drift by taking advantage of the Wi-Fi readings to compensate the position error introduced by the drift.

As explained before, the Group Adjustment method adjusts the nodes of a group based on the group combined force $g\vec{F}$ and the node individual force $c\vec{F}$ (Eq. 4.25). This method takes advantage of the information provided by each sample of the Group, and combines that information to adjust the samples position based on the group force.

If only the group force was used to move the nodes, the drift would also be preserved. However the individual force of the node also contributes to the node position allowing individual corrections, which may allow to compensate at least part of the drift in each node.

In fact, in ideal conditions, where the Wi-Fi RSSS readings are not affected by noise, the drift would be completely compensated. Figure 4.18 shows the result where the FastGraph algorithm corrects the drift using the Wi-Fi information, and the path shape is preserved.

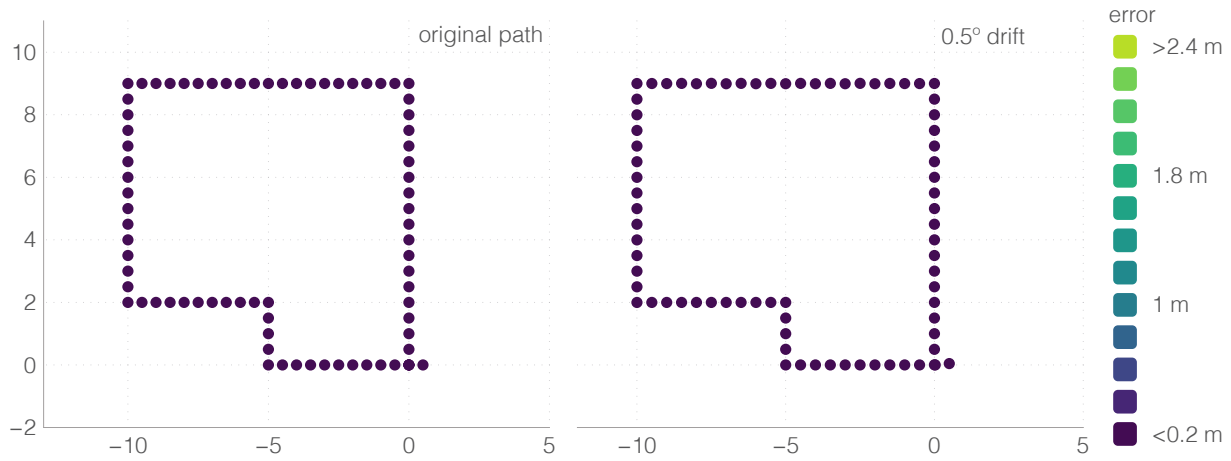

Figure 4.18: Drift correction with Wi-Fi (no RSS noise)

However, in real world, where we have noise in the Wi-Fi RSS measurements, each sample will be affected by a different amount of error, and the path shape deteriorates when we give more weight to the individual $c\vec{F}$ of each sample node. Figure 4.19 shows the same path adjusted with different group force ($g\vec{F}$) and individual force ($c\vec{F}$) percentages, using the linear combination (Eq. 4.26) with different β_2 values. For this experiment the synthetic dataset used was generated adding Gaussian noise with zero mean and $\gamma = 3$ to each Wi-Fi RSS

measurement.

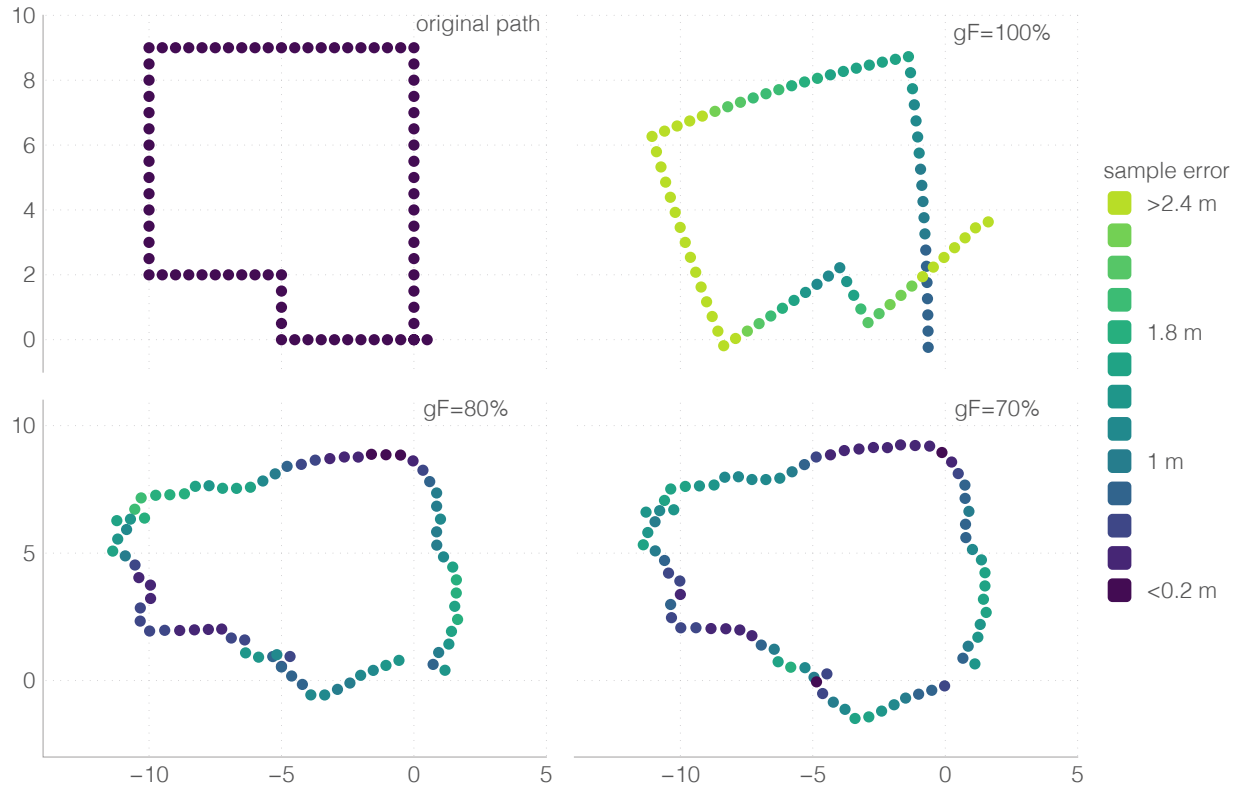

Figure 4.19: Paths affected by drift and Wi-Fi noise: Adjustment with different group force percentage (β_2)

The results show that when the group force is the only force applied to the nodes, as expected the drift is fully preserved (not corrected). In the cases where the group force is 80% and 70%, the drift is compensated. Less percentage of group force reduces the effect of the drift, but in neither of the cases it is completely corrected, and the path shape is destroyed by the noise in the Wi-Fi RSS readings.

Reducing the Drift Impact and Preserving the Path:

In order to improve the drift correction and avoid the path deformation, due to the noise in the Wi-Fi RSS measurements, the algorithm was updated to create additional edges between the sample nodes of a Group. With more edges, the path shape will be less affected by RSS noise, while keeping the effect of the APs' forces on the sample nodes to correct the drift. Therefore, instead of connecting only the current sample node to the previous sample node, the algorithm connects the current node to the previous n nodes, that belong to the same sequence. The length of this additional edges are estimated by applying dead reckoning techniques from the present to the past (Figure 4.20).

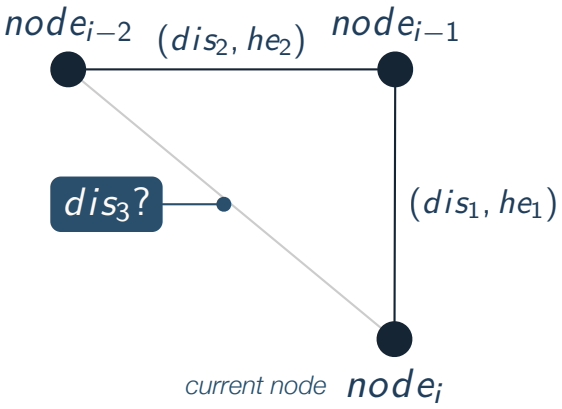

Figure 4.20: Edges length estimation between the current and the previous nodes

To establish an edge between the current node ($node_i$) and a previous node ($node_{i-2}$), the edge length (dis_3) as to be calculated. However, the current position of the previous nodes on the Graph can't be considered, since it may have been changed by the adjustment process and the influence of the RSS edges. For this reason, a dead reckoning process is used to estimate the position where the previous nodes should be considering the most recent node current position. For this orientation (he) and displacement (dis) information of each sample node is taken into account.

Therefore, in relation to the current position of most recent node ($node_i$) the position of the $node_{i-1}$ will be given by:

$$\begin{aligned} node_{i-1}.x &= node_i.x + dis_1 \times cos(he_1) \\ node_{i-1}.y &= node_i.y + dis_1 \times sin(he_1) \\ \vec{node}_{i-1} &= (node_{i-1}.x, node_{i-1}.y) \end{aligned} \tag{4.30}$$

Following the node sequence the position of the node ($node_{i-2}$) in relation to the calculated position of the node $node_{i-1}$ is then given by:

$$\begin{aligned} node_{i-2}.x &= node_{i-1}.x + dis_2 \times cos(he_2) \\ node_{i-2}.y &= node_{i-1}.y + dis_2 \times sin(he_2) \\ \vec{node}_{i-2} &= (node_{i-2}.x, node_{i-2}.y) \end{aligned} \tag{4.31}$$

Then the edge length (dis_3) is given by:

$$dis_3 = |\vec{node}_i - \vec{node}_{i-2}| \tag{4.32}$$

With this we can formulate that the natural length (e_{CL}) of an edge between the current sample node ($node_i$) and a previous sample node ($node_{i-j}$) of the same Group with n nodes is given by:

$$\begin{aligned} node_{i-j}.x &= node_i.x + \sum_{i=i-1}^{n} \vec{dis}_{i-j} \times cos(he_{i-j}) \\ node_{i-j}.y &= node_i.y + \sum_{j=i-1}^{n} \vec{dis}_{i-j} \times sin(he_{i-j}) \\ e_{CL} &= \sqrt{(node_i.x - node_{i-j}.x)^2 + (node_i.y - node_{i-j}.y)^2} \end{aligned} \tag{4.33}$$

With more edges, the shape of the trajectory at tight turns, as the ones depicted in Figure 4.20, is more preserved by forming triangles. Figure 4.21 shows the result using the solution with this improvement to correct the drift. The dataset used was the same from the previous experiment (see Figure 4.19), with the same noise in the Wi-Fi RSS measurements. The additional edges between samples help to further correct the drift without completely destroying the path shape. In this experiment edges between the current node and the 15 previous nodes were added.

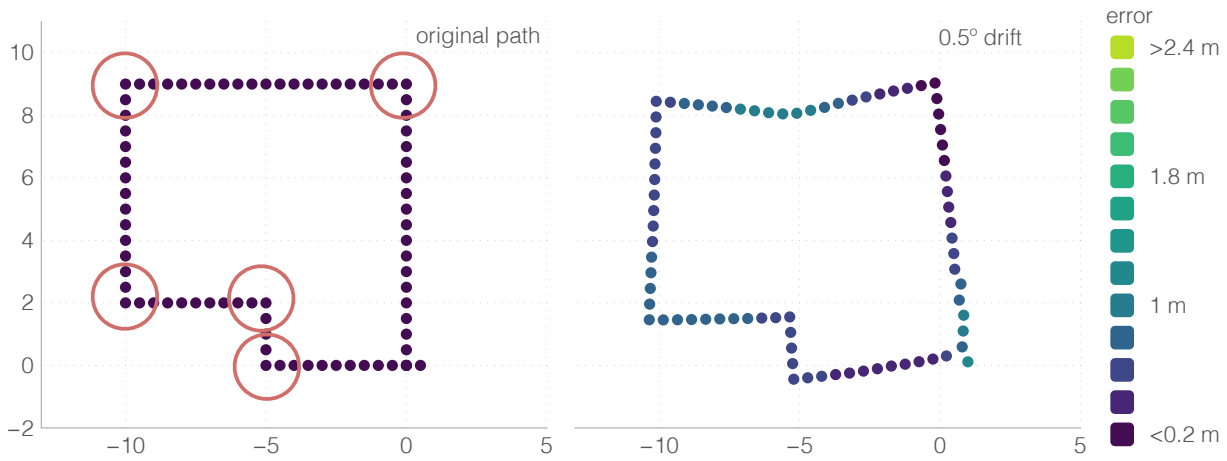

Figure 4.21: Improved solution with drift correction

As mentioned before, the drift correction is out of the scope of this thesis, however with a relative simple approach it was possible to minimize the impact of the drift in the proposed solution. This improvement enables this solution to work better with low grade sensors and to be more robust against possible errors in the dead reckoning information.

4 Environment Dynamics and Scalability

Many Wi-Fi positioning solutions are limited in terms of scalability. For example a Fingerprinting solution requires an extensive radio map to be available, which is difficult to create. In addition, in the online phase, the number of fingerprints in that radio map will influence the matching time necessary to estimate a position. In this context, and as explained previously, different matching process can be used.

In the FastGraph approach it is easy to understand that, without any scalability mechanism, increasing the Graph size will increase the adjusting time, also affecting the position estimation time. Therefore several scalability mechanisms ensure that the FastGraph solution can be used in a large environments with multiple users.

- **Anchor Samples Fusion:** As described before, the fusion or merging of the Anchors samples improves the length estimation of the Anchors edges, by averaging multiple RSS measurements. However, this feature is also important in terms of scalability. By fusion the samples from an Anchor in a single node, the number of nodes on the Graph is significantly reduced, with less nodes and edges to be handled by the adjustment process.

- **Partial Adjustment:** It was also explained before that when the Graph has a large number of nodes, adjusting the entire Graph when a new node is added to Graph is not efficient. Therefore, the partial adjustment is also an important feature in term of scalability. This way the Graph is adjusted by sections, where only the nodes that may be affected by a new node are adjusted. These sections can be seen as sub Graphs. In this process a different number of nodes can be adjusted, depending on the node type or the node source.

- **Graph Pruning:** In contrast to the two previous mechanisms, the Graph Pruning process was specially designed and implemented to further improve the FastGraph scalability. This process ensures that the solution can maintain the performance despite the operation time or number of users. The pruning process eliminates old nodes from the Graph, while ensuring that a necessary minimum number of nodes is maintained. The nodes removed from the Graph can be maintained in a separated database and used in other applications as a radio map.

Results regarding the computational time, specifically the time required to provide a position estimation, will be presented and discussed in the results chapter.

4.1 Graph Pruning Process

The Graph size can increase with time when more samples are added. A partial adjustment process was designed to be efficient with short adjustment times. However, if using the full adjustment process, more nodes increase the adjustment time.

In addition, the fingerprints or samples collected as explained before age up and become outdated due to radio environment changes, this means that old fingerprints may not be good at characterizing the environment.

Therefore, the pruning process serves two purposes.

- First, it ensures that the computational performance is maintained.
- Second, it helps the Graph to dynamically adjust to the variations of the radio environment, including radio infrastructure changes.

The main factors that lead to radio environment dynamics were already described before, being the radio infrastructures changes one of the most severe alterations. For example, APs being removed, added or changed from one position to another in maintenance processes have high impact on the radio signatures.

For FastGraph, the new APs are not a problem since they are handled and added to the Graph by the normal process described before. The pruning process helps the FastGraph to eliminate old APs that were not observed by any samples for a specific time, which can configured to be hours or days. If an AP is moved from one position to another, the algorithm will start to reposition the AP in the Graph based on the new samples, then the pruning process removes the old samples and edges related to the AP previous position.

A set of rules are followed to maintain the Graph consistency, such as ensure a minimum number of edges for each AP or ensure that the sample node are not left without edges.

The pruning process can be activated by the algorithm if the number of nodes is over a configurable number or if the average time to process a sample increases over a specific threshold.

5 Estimating the Path Loss Exponent (η)

The importance of the Path Loss Exponent (η) was frequently mentioned along the previous sections of this thesis, and the complexity of estimating the value of this parameter was described in Chapter 2 (Page 23). In this section are described the methods that were designed to estimate this parameter and how they perform. During this research three different approaches were considered: Analytic Calculation, Optimization, and Learning.

In the FastGraph approach, the η of each edge is used to compensate not only the path loss, but also other factors that affect the received signal at a specific area. In addition, a fixed RSS0 is kept for all APs and the difference in the RSS0 for each individual AP is also compensated by this value. With preliminary experiments using synthetic data it was confirmed that, as expected, the nodes would not converge to the correct positions without estimating the path loss exponent for each edge of the Graph. This confirms that the path loss is a complex parameter with significant impact, since the FastGraph in previous experiments was able to handle significant noise levels added to the RSS values, and even compensate drift added to the motion information.

5.1 Minimum Energy Optimization

The η can be analytically calculated using the LDPL model:

$$n = -\frac{\frac{(RSS - RSS0)}{log_{10}(d/d0)}}{10} \tag{4.34}$$

However, to calculate the η at a given position based on the LDPL model, in addition to have a RSS measurement and the RSS0, it is also necessary to know the AP position and the device position, in order to calculate the distance (d). In most of the cases of interest, neither are known.

As described in Chapter 2, previous works suggest that a system of non-linear LDPL equations has no analytical closed form solution [63], therefore in that work the problem is addressed using optimization algorithms to find the solution that minimizes the least mean absolute error. However, as reported by the authors, that approach has high computational requirements where the algorithm can take several hours to find a solution. In addition, only 2D positions for the APs are considered, which simplifies the problem.

On the other hand, the FastGraph approach solves a system of LDPL equations, where each equation is represented by an edge between a device and an AP. Considering this, in order to

reach the minimum energy state, each edge force must be zero. This means that the natural distance, estimated by the LDPL equation, and the current distance of the edge are the same. Hence, in FastGraph the sum of all edges differences of a Node can be used as an objective function, and therefore the solution for the system of LDPL equations is where the objective function is zero (when the forces of all edges are zero).

In order to reach the minimum energy state, the η of all edges must be corrected, otherwise the difference between the natural and current distance of an edge will generate a force. Following this idea, minimum search algorithms can be designed to find the values of η that minimize the objective function, where the edges energy sum is zero.

Therefore, the energy of a node j can be given by:

$$energy_j = \sum_{i=0}^{n} \|NL_i - CL_i\| \tag{4.35}$$

where, NL is the natural length given by the LDPL model using the η value, the CL is the current length of the edge, and n is the number of edges connected to the node.

5.1.1 Finding the best common Path Loss (η) for an AP

As explained before, in a real world environment one cannot assume a single value of η for an AP. However, in order to understand how complex it is to estimate the path loss exponent a simple approach was initially designed. This approach tries to find the best common η value for an AP.

The best common η is the value that gives the minimum energy for the AP when applied to all of its edges. With this common η, the algorithm can estimate an initial position for the AP. After applying the common η to all edges of the AP, and adjusting the Graph, the AP Energy (ap_e) will be given by the equation defined before (Eq. 4.35).

This process tries values of η between 1.5 and 5 with steps of 0.01, and calculates the AP Energy for each value. The η that generates the minimum energy will be selected the as the common path loss exponent for the AP.

To experiment this process a synthetic dataset was generated in a space with an obstacle (Figure 4.22). The obstacle produces variations in the path loss by introducing 6 dB of attenuation to the edges that intersect it. A $\eta = 2.5$ was used as standard free space path loss exponent.

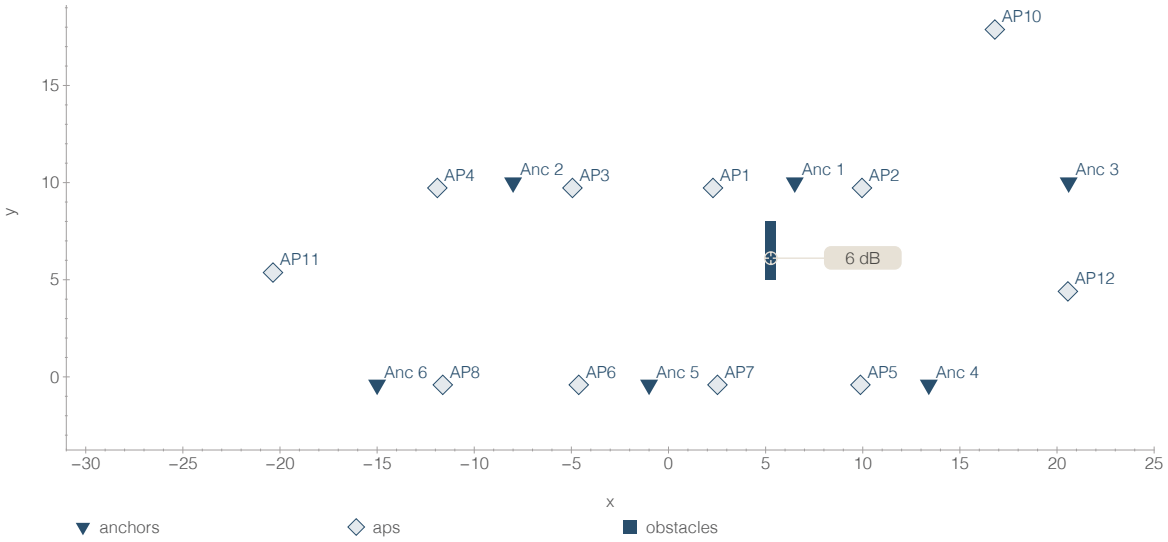

Figure 4.22: Virtual space with obstacle (n variation)

The evolution of the energy as different η values are applied to the edges between Anchor nodes and AP nodes is shown in Figure 4.23.

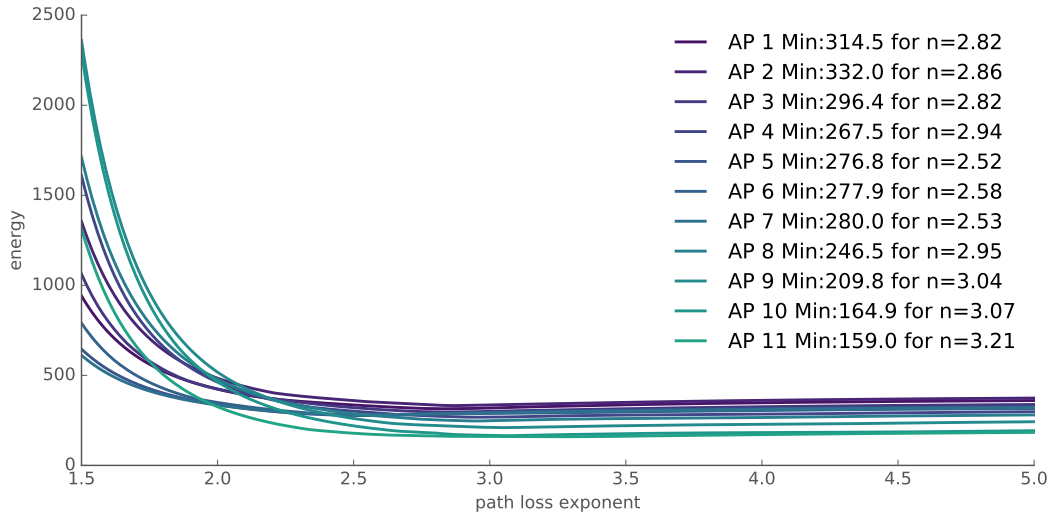

Figure 4.23: Energy using the same Path Loss Exponent for all edges

As expected the η values estimated for the APs 5,6 and 7 are close to the standard free space η used in the simulator (2.5), since they have line of sight to almost all Anchors (see Figure 4.22). The energy for all APs decrease rapidly for lower values of η, and stabilizes for higher values. This is related to the LDPL logarithmic base that is used to estimate the natural length for the edges, as the following plot shows (Figure 4.24).

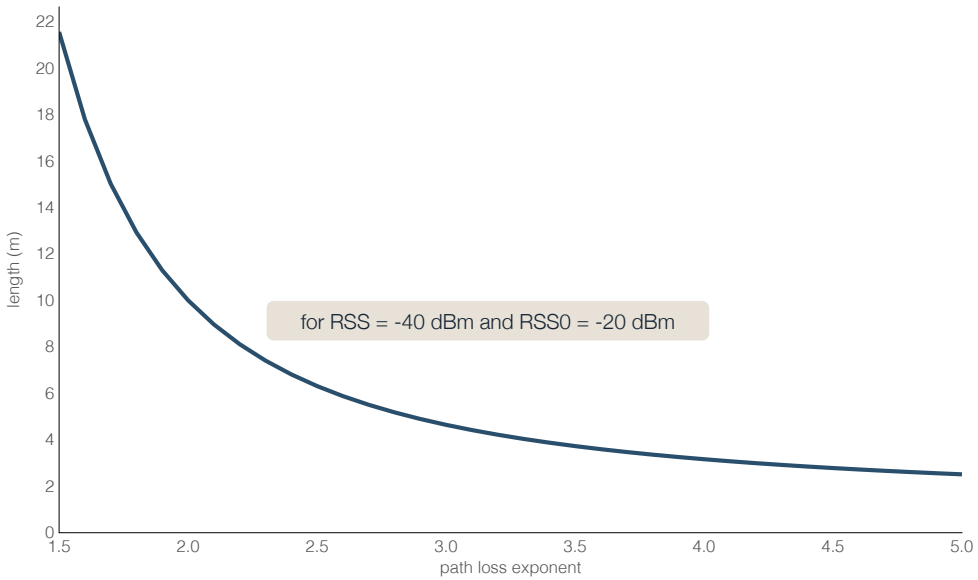

Figure 4.24: Distance vs Path Loss Exponent

Although the minimum energy curve (Figure 4.23) becomes almost flat for large values of η, the value that produces the minimum energy for an AP when applied to all edges can be easily found. However as expected, with this method the zero energy state can't be reached, which confirms that each edge must have a different η value.

5.1.2 Finding the best Path Loss (η) for each Edge

After finding the best common η for all edges of a node that minimizes the energy, the η of each edge can be individually adjusted.

The base principle is the same, the η for each edge will be selected based on the global minimum energy of the node (Eq. 4.35), and by testing values between 1.5 and 5 with steps of 0.01. However, in this case this values are tested to each edge individually, and the node energy is measured after each value applied. The best η for an edge is the value that generates the minimum energy for the node.

The process follows the node edges sequentially, and as the η of the last edge is adjusted, the process can be repeated to improve the η values based on the new state of the other edges. This iterative process ends when there is no variation in the AP energy from one iteration to another.

The same dataset was used to experiment this process. The following plot (Figure 4.25) shows the η adjustment for an edge, after three iterations, when the energy variation was zero.

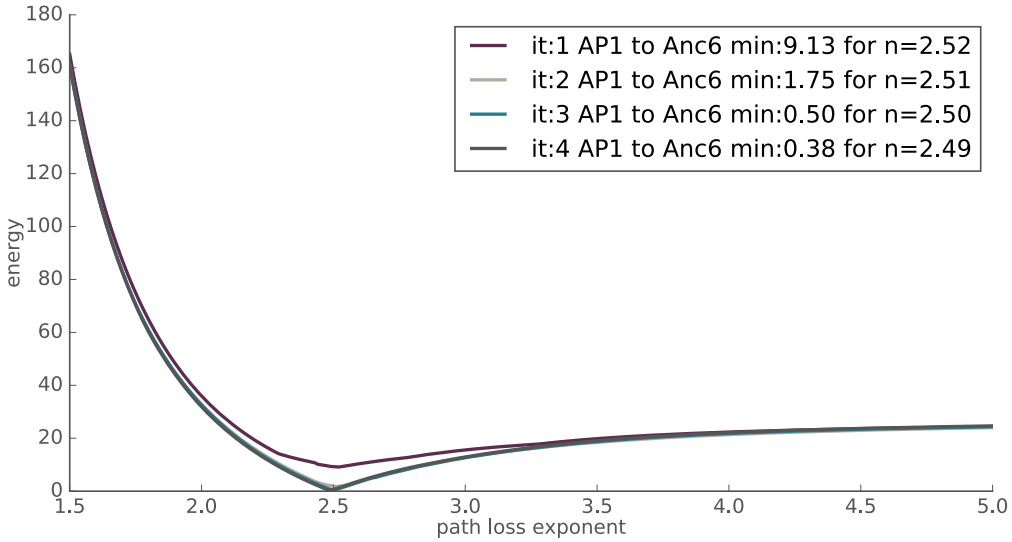

Figure 4.25: Distance variation in relation to η (AP1 to Anchor 6)

From the first iteration to the second, the AP energy dropped more significantly. However as we can see the η for this edge had a low variation from iteration to iteration. The drop in the first iteration was because initially all edges had the same η, and for some reason that value was far from the correct η. This is an example of an edge with line of sight.

The next plot (Figure 4.26) shows an edge that intersects the obstacle, and therefore is affected by attenuation. This means that the correct η has to be superior to the standard free space η. Also the minimum is superior to the minimum obtain in the line of sight example (Figure 4.25), due to the obstacle.

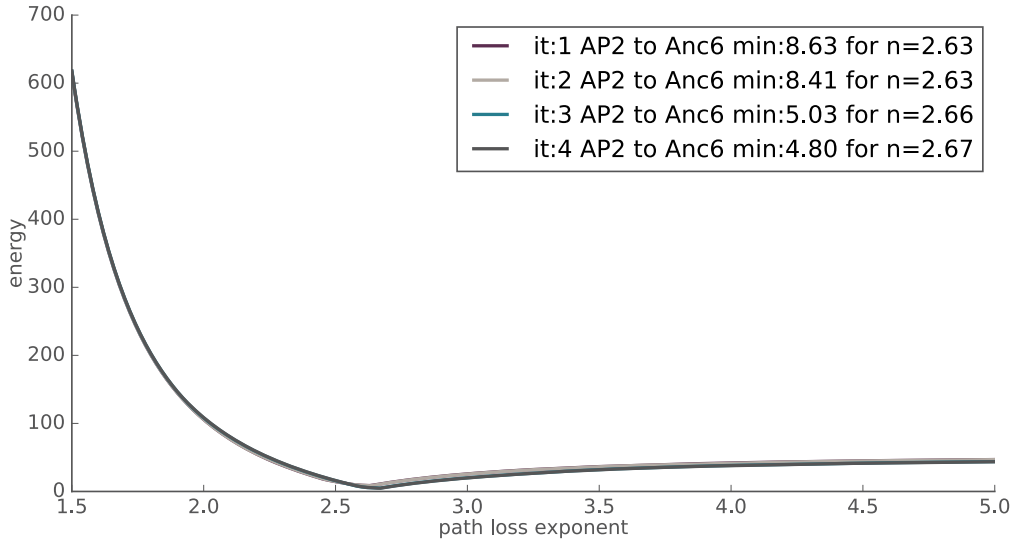

Figure 4.26: Distance variation in relation to η (AP2 to anchor 6)

With this method it is possible to estimate different values of the η for each edge of a node, resulting in a lower energy, and a lower error in the length of the edges. However, also in this

case it was not possible to reach the zero energy state, even in this simple example where we have only an obstacle and a low number of edges. Each node is always adjusted based on the state of the other edges, which makes it difficult to find the global minimum, even if keeping adjusting for several iterations.

5.1.3 Finding the Correct η Values (The Needle in a Haystack)

After several experiments the results have confirmed that the correct values of η are very difficult to find, even when using optimization algorithms, because, in contrast to other solutions, the FastGraph handles 3D spaces.

In order to further demonstrate this, another synthetic dataset was generated using the simulator. In the virtual 3D space were placed three Anchors at different positions. Then a different η was defined for each one of the Anchors areas. With this configurations the RSS measured by each Anchor was affected by a different path loss exponent. The η values used to generate the dataset were $\eta_1 = 2.8$, $\eta_2 = 1.9$, $\eta_3 = 3.2$ (Figure 4.27).

Then, the possible permutations of values of η between 1.5 and 3.4 with 0.1 interval (20 values), for each one of the three edges of the AP were tested. For an AP with only three edges the number of possible solutions are:

$$P^R(n, r) = P^R(20, 3) = n^r = 20^3 = 8000 \tag{4.36}$$

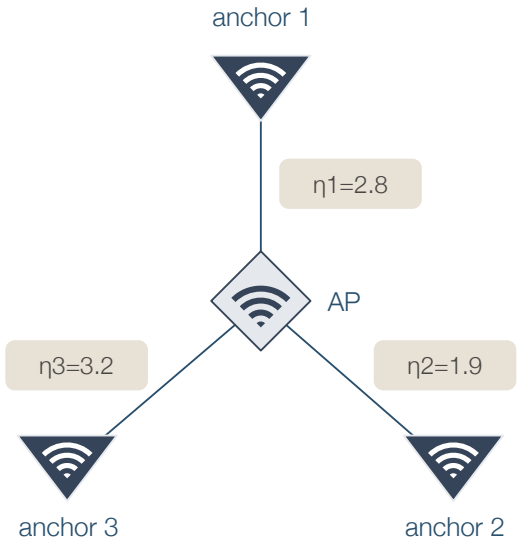

Figure 4.27: Search for the η values

In this experiment, each one of the possible solutions was tested, and for each solution the sum of the difference between the natural and current length ($|NL - CL|$) of the three edges

was computed. The correct solution will have sum of zero (zero energy state). Only three Anchors were used to be possible to plot a surface and identify the minimums.

The zero energy was naturally obtained for the correct values of η. Figure 4.28 shows a 3D and 2D surface slices. In the first slice is represented the energy minimums for different values of η_2 and η_3 for $\eta_1 = 1.5$. In the plots was used a logarithmic scale, therefore the edges energy can assume negative values.

As can be observed in these plots, there are several local energy minimums with a very low magnitude, and we are considering only a single slice of the space of solutions in $\eta_1 = 1.5$. For this slice the absolute minimum was at $\eta_1 = 1.5$, $\eta_2 = 3$, and $\eta_3 = 3.2$ with an energy magnitude of 0.12. This value is very close to zero however we have wrong values for the η_1 and η_2, that should be $\eta_1 = 2.8$ and $\eta_2 = 1.9$.

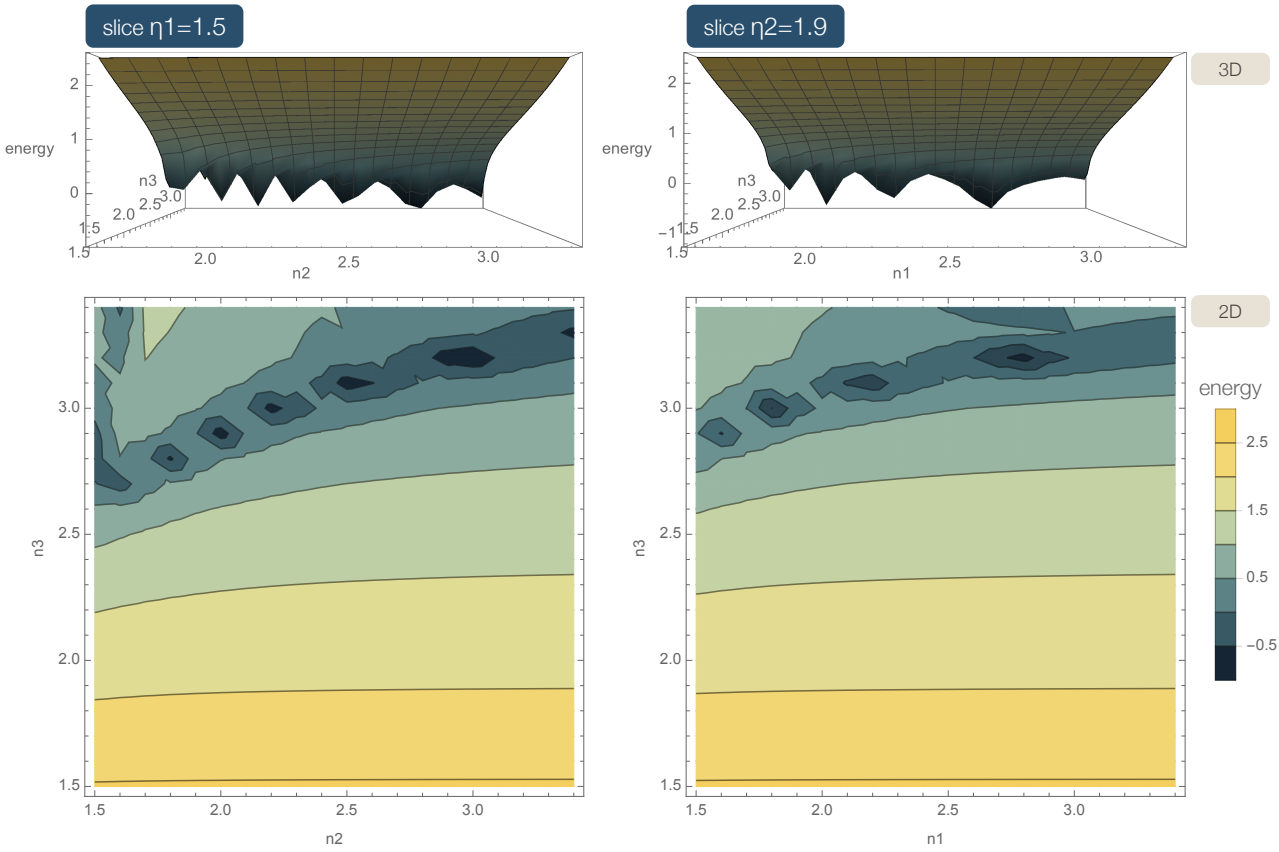

Figure 4.28: Slice of η solutions surface for $\eta_1 = 1.5$

Figure 4.28 also shows another slice of the solutions space, this time for $\eta_2 = 1.9$, which is the correct value for the η_2. Therefore in this slice is the absolute minimum, with zero energy.

The absolute minimum appears at $\eta_1 = 2.8$ and $\eta_3 = 3.2$, which are in fact the correct values. However, as in the previous slice, also here there are local minimums with very low energy.

These results allow to clearly understand that finding the correct values of η for a node

with several edges, without information about the APs positions and in a 3D space, is a really complex problem. Even for this simple example, where an AP has only three edges, and the possible values that η can assume are limited, several low energy local minimums exist, even for significantly wrong η values.

This is a complex multidimensional minimum search problem, since there are several local minimums with very low magnitude where a minimum search algorithm can fall. Simple optimization and minimum search algorithms were tested to solve this problem, such as Gradient Descent (GD) and a Genetic Algorithm (GA). However, as these results had already suggested, these algorithms also failed to find the absolute minimum. Perhaps more advanced optimization algorithms be able to provide better results, however that is a topic that is far out of the scope of this thesis.

Nevertheless, these experiments and results were important and useful to understand the problem. Based on this, was designed a new approach that uses the Gradient Descent, which provide a decent initial estimation for the values of the η, and then that initial estimation is improved based on a progressive learning process, that will be explained later.

5.2 Estimating the η values for Independent Sample Nodes

The η has higher impact in the initial position of independent samples, such as the first sample node of a Group, or samples without motion information, because the sample initial position is defined relying only on Wi-Fi.

Without knowing the correct η for the sample, the algorithm applies the default η, which may need to be adjusted. To reduce the impact of using the default η in the sample node initial position, a new η is estimated for each edge of the sample node, using a Gradient Descent (GD) algorithm.

The GD algorithm search the values of η for all edges of the sample node that lead each edge to a relax state or minimum energy state, minimizing the $c\vec{F}$ (Figure 4.29). The possible values for the η for each edge can be limited if the resulting distance is an impossible value, considering the typical Wi-Fi signals range.

As mentioned before, the global minimum is very difficult to find and the GD algorithm does not ensures that the global minimum is reached. However, a local minimum provides better initial η values than the default and common η.

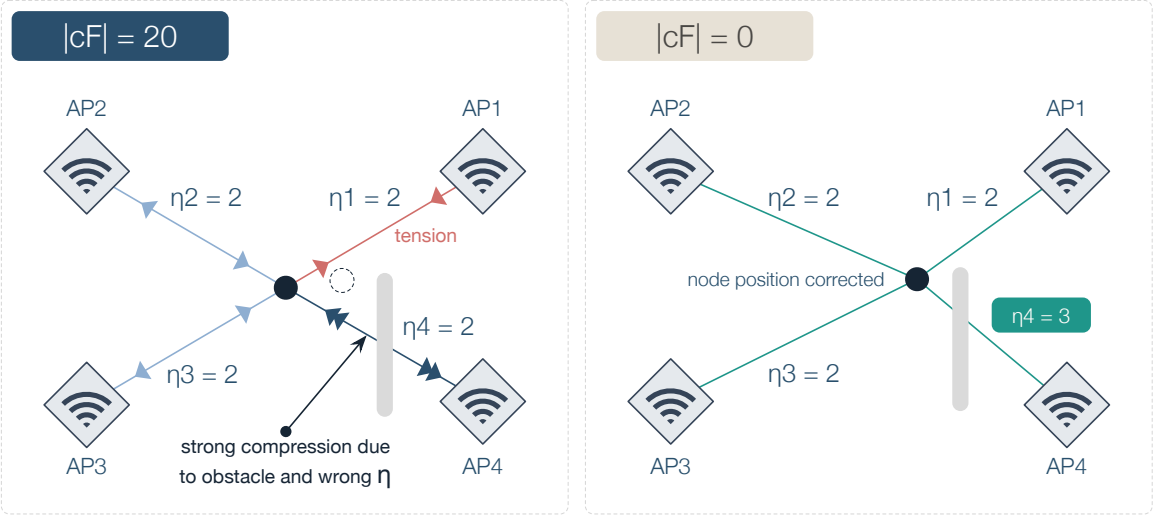

Figure 4.29: Adjusting the η values to reach a minimum energy for the node

5.3 Progressive Path Loss Exponent (η) Learning

The difficulty of modeling the path loss in a complex 3D indoor environment using analytical and optimization approaches is well known and was verified during this research work.

It was demonstrated that when the algorithm tries to estimate the path loss for each edge of an AP the existence of multiple minimums is a problem. Advanced optimization algorithms to search the global minimum may be able to obtain good estimations, but even in 2D problems the process requires high processing power and time, as some researchers have already reported [63].

Considering all of these aspects a different approach was designed. The solution proposed to deal with this problem takes advantage of the unique features of the FastGraph algorithm, and is based on one of the initial hypothesis for this research: with more data processed the Graph becomes a better representation of the radio environment. Considering this, some modifications were made in the FastGraph algorithm, which is now able to improve the estimations of the path loss exponents for each edge of the Graph, when more information is added.

This can be seen as a progressive learning process, in the sense that, as the Graph evolves, the algorithm is able to improve the values of η estimated initially. Then the learning process progressively adjusts the η using more and more information from the samples processed. The η is adjusted progressively by a linear combination to minimize the tension or compression of each edge. The current length and natural length of the edge will progressively match.

It may seem odd to adjust the η value in order to the natural length match the current length, since the current length may not be correct. However, as more and more samples are processed, the several edges will create a group averaging effect, that forces lead the node to converge to the correct position. As the node approaches that position the η values are

improved. When reached the correct position the natural length of the edges is in fact correct.

In Equation 4.37, $e_{ZE\eta}$ is the η value that produces zero energy to an edge e. To obtain the zero energy for the edge the Current Length (CL) is considered as the target length:

$$e_{ZE\eta} = -\frac{\frac{RSS - RSS0}{\log_{10}(CL/d0)}}{10} \tag{4.37}$$

Then the η value is updated by a linear combination between the edge current η and the $e_{ZE\eta}$, which allows progressive convergence (learning) to the correct value:

$$e_\eta = (\beta_3 - 1) \times e_\eta + \beta_3 \times e_{ZE\eta} \tag{4.38}$$

The learning rate is defined by β_3. A high learning rate can be used to converge fast while keeping part of the edges energy, in order to corrected the nodes and the AP's positions. The new forces introduced by the edges of new nodes will increase the energy of the nodes, that help improving the nodes positions, and by consequence the η values.

Figure 4.30 shows a synthetic dataset with eleven APs and 6 Anchors. Three obstacles were added to the scenario, two of them introducing 6 dB of attenuation, and another introducing 10 dB. With these obstacles the RSS measurements are affected by different combinations of attenuation, depending on the obstacles between the sample and an AP.

The smooth walk motion model was used to collect 1500 samples. Each one of the RSS measurements in each sample were also affected by Gaussian noise with zero mean and $\gamma = 3$.

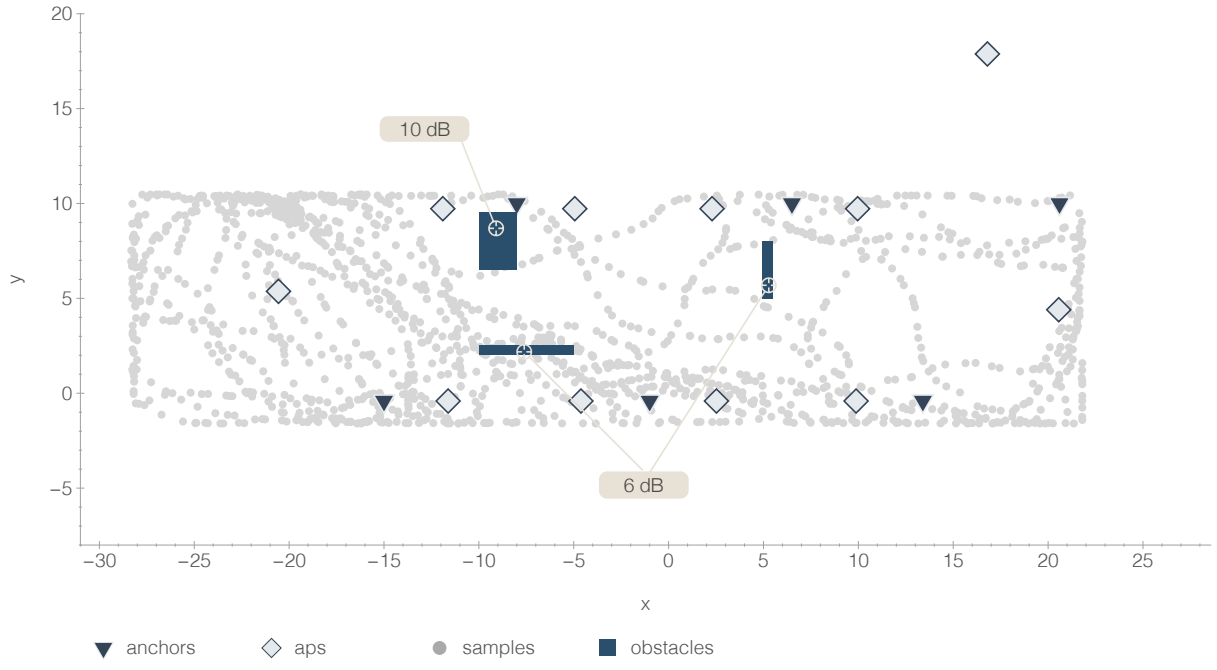

Figure 4.30: Experimenting progressive η learning (Simulation Setup)

With the progressive η estimation, the FastGraph was able to estimate the η value for 25503 edges with an average error of 0.06. More results regarding the this topic will be presented in the results chapter.

6 Summary and Discussion

This chapter described the fundamental principles and the approach behind the FastGraph.

The 3D Force-Directed Graph is the foundation that distinguish the FastGraph from other solutions. The organic nature of this approach allows to solve complex multidimensional scaling problems, without trying to analytically solve multiple sets of equations.

The Graph is adjusted based on a minimum energy approach, and evolves to be a representation of the physical and radio environment, with nodes representing APs, Anchors, or Samples from Moving Devices. The reference samples, usually collected by Anchors, are used to initialize the Graph, estimating the initial positions of the APs and matching the Graph to the physical space.

The nodes are connected by edges, that can be RSS based, representing communications channels, or Motion-Edges that characterize the device motion. The natural length of the RSS edges is given by the LDPL model, and each edge has its own propagation characteristics, with a different path loss exponent. The natural length of motion edges is given by displacement information, when available in the moving devices samples, or based on time between consecutive samples. Each edge type has different elastic constants, given different level of influence on the Graph, depending on the accuracy of the data used to establish the edge length.

The FastGraph solution includes mechanisms to address problems such as oscillatory nodes and orientation drift. Moreover, to deal with the environment dynamics and the solution scalability, some mechanisms were designed and implemented such as the Anchors sample fusion, the partial adjustment, and the Graph pruning process. These mechanisms ensure that the computational performance is maintained, and helps the Graph to follow the variations of the radio environment, including radio infrastructure changes.

The complexity in estimating a path loss exponent for each edge, in a 3D indoor environment, was documented with experimental examples. Considering this, a strategy based on minimum energy optimizations and progressive learning was conceived, which leverages from the Graph evolution, with more and more information from new samples, to progressively improve the path loss estimations.

In order to validate the proposed hypothesis, and evaluate the solution in real word experiments, the FastGraph algorithm was implemented in Java and can be executed in a common machine with regular hardware specifications. With minor adjustments, it can even run directly in Android Smartphones, which may be interesting in specific scenarios.

Chapter 5

Real World Setup

The virtual environment from the simulator provided valuable synthetic data, allowing controlled experiments to tackle a specific problem at a time.

After encouraging results obtained with synthetic data, the natural step was the evaluation in real world scenarios. The real world introduces additional levels of complexity and variability difficult to simulate.

In this chapter is described how the proposed solution was deployed in two very distinct real world spaces, allowing to test the solution in two different operation modes.

Section I and II describe the spaces and experimental setups where FastGraph was evaluated.

Section III and IV describe the devices and additional Software modules developed as part of the FastGraph prototype, to allow the deployment of the solution.

1 PIEP - Industrial Enviroment

The first deployment space is a polymers engineering building PIEP (Pólo de Inovação em Engenharia de Polímeros) at University of Minho (Figure 5.1). The PIEP is very similar to a factory building, with an area of ≈1000 m^2.

Figure 5.1: Real World Experiment Space A (PIEP)

In this space there are several metallic elements, industrial machinery and exposed beams, that result in several reflections, multi-path effects and non line of sight. This is a challenging space for radio signal propagation, and is also very dynamic, because machines and cargo are moved from place to place, and people are also frequently present in the space. The space was mapped, and a XYZ referential defined. A ground truth grid, based on floor tags spaced 1 meter from each other, was added to the space, to define paths with known positions. Six anchors were installed to cover the area.

1.1 Space Setup

The PIEP had very low Wi-Fi coverage, with almost no access points at the testing site. For this reason, eleven access points were installed to provide the necessary support for the experiments.

Two different types of APs were installed, in order to have distinct radiation models and power transmissions. Eight celling type Access Points were installed on the existing technical cable trays, and 3 APs were places in walls (Figure 5.2). The APs were placed at different heights ranging from 3.95 meters to 5.46 meters, and placed near the walls, because due to the PIEP daily operations it was impossible to place them in central positions (Figure 5.3).

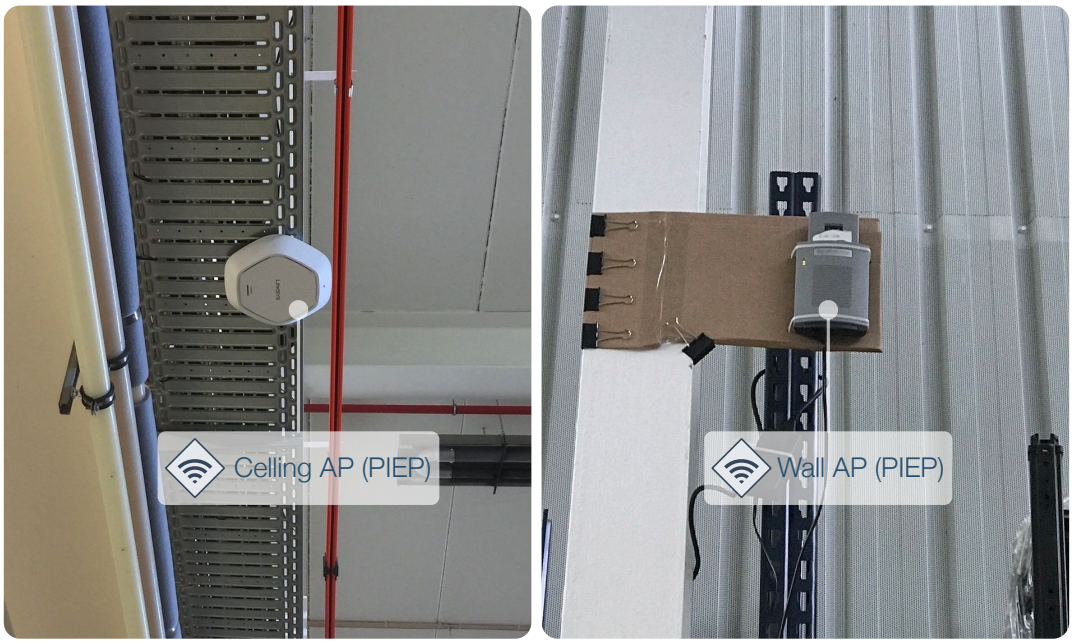

Figure 5.2: Photo of APs at PIEP

In order to better evaluate the positioning solution, in addition to mapping the space and define a referential, a ground truth grid was also added (Figure 5.3).

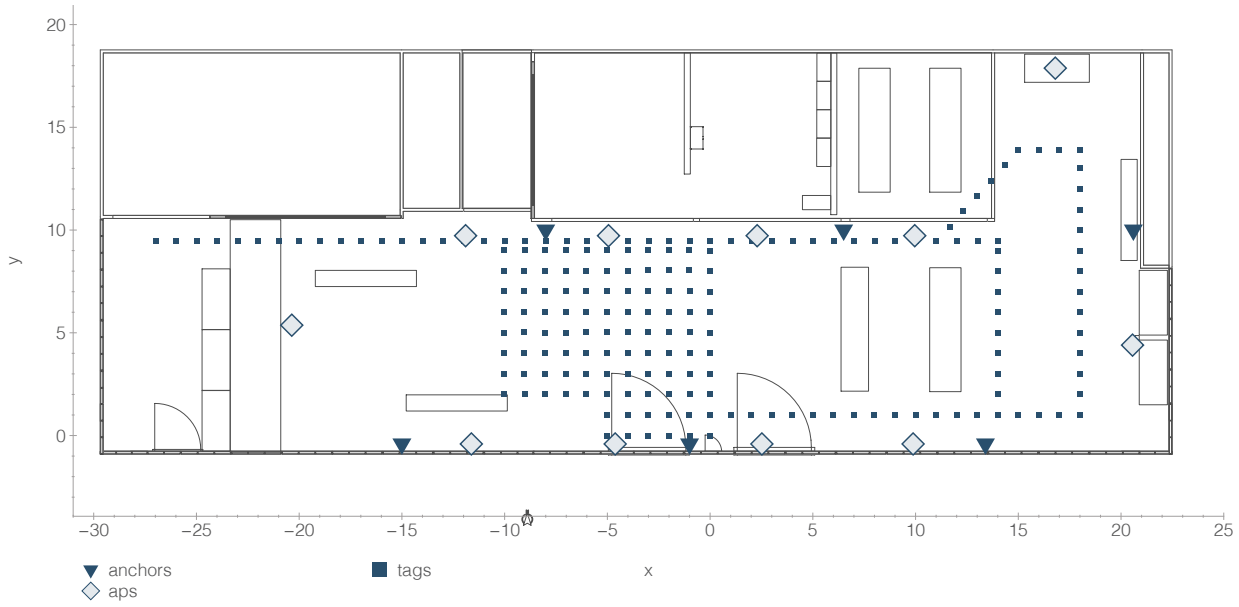

Figure 5.3: Access Points and Tags Positions PIEP

The ground truth grid is based on near 100 numbered floor tags (Figure 5.4) with known positions on the defined referential. The tags were spaced 1 meter from each other.

Figure 5.4: Photo of tags at PIEP

With the ground truth grid, different paths can be followed and the position tracked, using a camera to record the tags. The ground truth for a specific sample can be obtained by comparing the sample timestamp with the timestamp of each tag in the video. The clock of the camera and the device that collects the samples were synchronized by a clock synchronization algorithm.

The ground truth for the samples collected between tags was obtained by interpolation, since the testing device used in the experiments moves at a uniform speed.

1.2 Solution Deployment

The installation of the APs and the ground truth grid were steps necessary to perform the experiments at PIEP, but in a normal deployment are unnecessary. The ground truth referential is only needed for evaluation, and it is assumed that in most spaces the APs are already installed. For these reasons, it makes sense to conceptually separate these tasks from the actual solution deployment phase.

The actual solution deployment involves the anchors installation, and for the PIEP application scenario, the setup of the MTU (Moving Testing Unit), which features will be detailed in the experiment devices section.

Six Anchors (Figure 5.5) were installed at PIEP, being distributed in the space and placed in the walls at 2 meters of hight (Figure 5.6). The number of Anchors used corresponds to a density of ≈ 1 Anchor for each 166 m^2.

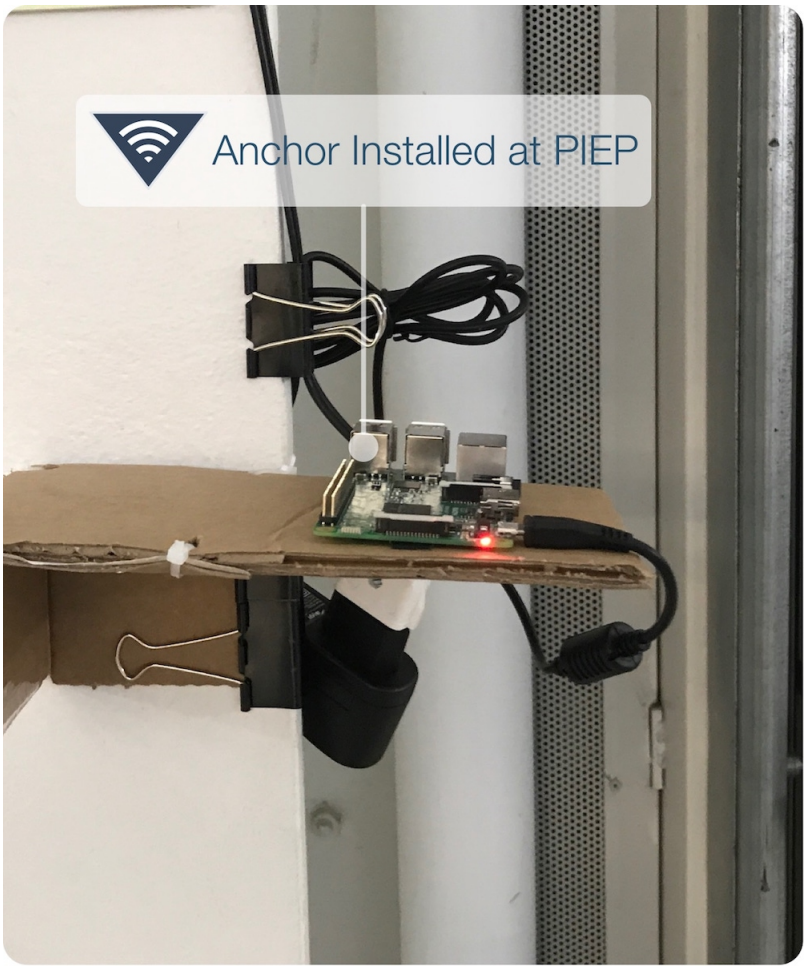

Figure 5.5: Photo of an Anchor at PIEP

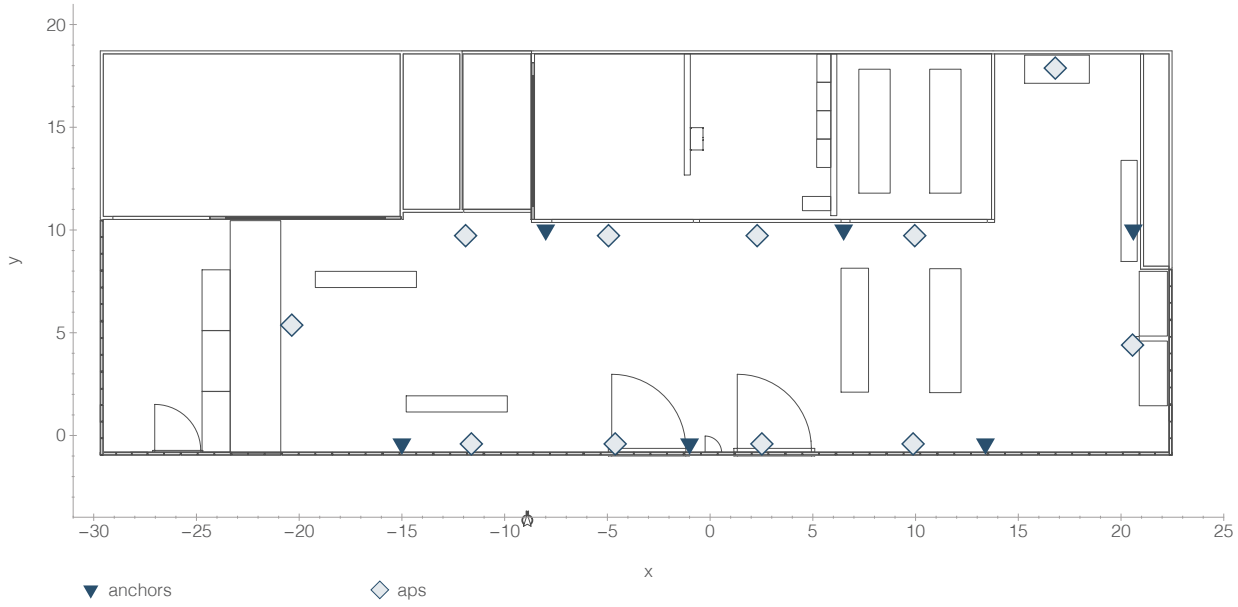

Figure 5.6: Anchors and APs positions at PIEP

2 DSI-DEP: University Building

The second deployment space was another building at the University of Minho. A floor of this building has a combined square footage of around 4638 m^2. The DSI-DEP, occupying the first floor of building 11, is a space with characteristics very distinct from PIEP. The layout of the DSI-DEP (Figure 5.7) is complex with several offices, rooms and corridors, which result in different propagation characteristics in relation to PIEP, where the main obstacles are large industrial metallic machines.

Figure 5.7: Real World Experiment Space B (DSI-DEP)

In indoor positioning, some solutions have different performances depending on the space layout, with systems obtaining better results in narrow layouts, such as corridors, and others in open areas. Therefore this space, allows to evaluate how FastGraph Wi-Fi positioning performs in a typical offices building.

2.1 Space Setup and Solution Deployment

The DSI-DEP is a space with a large number of Access Points, therefore it was not necessary to install additional APs. The only necessary step to prepare the experimental setup to evaluate the solution, was to define a spatial referential and map the position of some APs. Fifhteen APs in accessible areas, such as corridors and public rooms, at the first and second floor were mapped (Figure 5.8).

At the DSI-DEP the strategy for the solution deployment was different from that used at PIEP. The DSI-DEP is a very large space, being necessary more Anchors to cover the space. Considering that the experiment setup was temporary, finding power sources and places to install them was a problem. Therefore, instead of having several Anchors installed at fixed

positions, a unit that can be used as moveable Anchor (Tripod Testing Unit - TTU) was designed. With this approach, data was collected at 13 different locations in the first floor of DSI-DEP, one location at a time, simulating the use of 13 Anchors. The number of "virtual" Anchors used corresponds to a density of ≈ 1 Anchor for each 357 m^2, a lower density compared to PIEP.

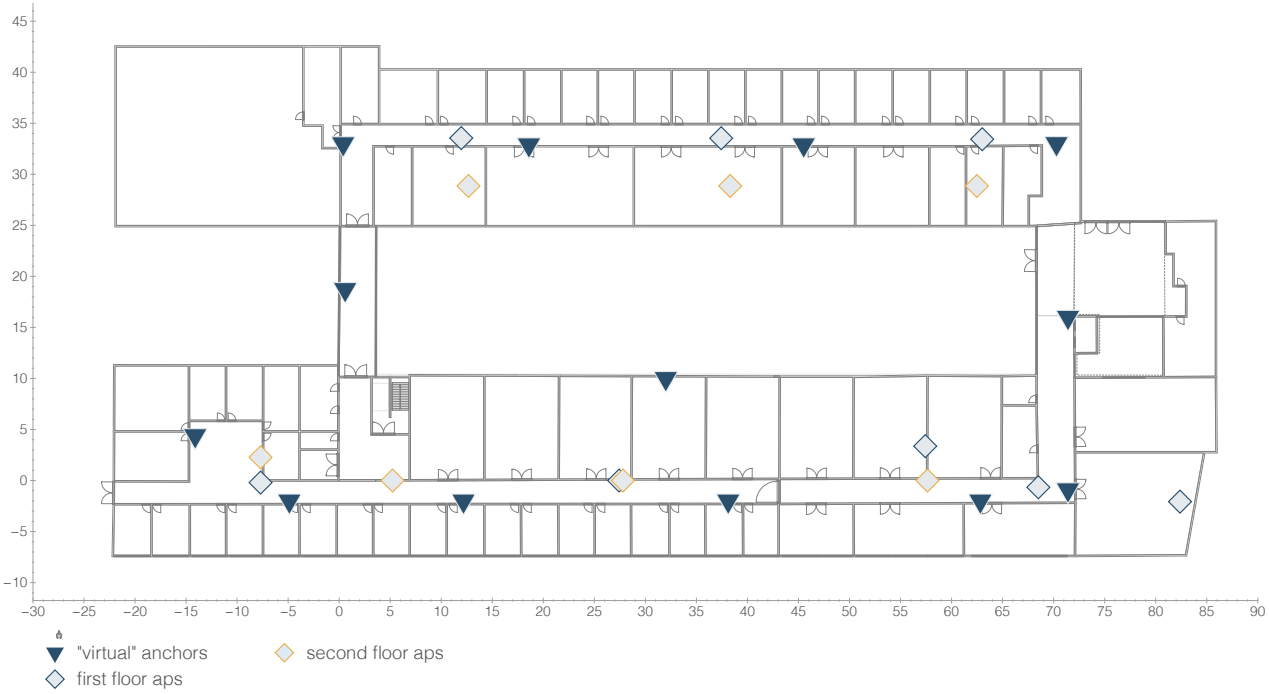

Figure 5.8: Access Points (Mapped) and Anchors

3 Experiment Devices

To deploy the FastGraph system and perform the evaluation was necessary to design prototype solutions to be used as Anchors and as Moving Devices.

After defining the requirements, adequate hardware and sensors were selected, and then software modules were developed to perform the necessary tasks in the selected hardware.

The Raspberry Pi 3 (RPi3) was choosed as the core for the solution implementation. The low cost and small single board computer has the necessary resources with a low power consumption. The internal Wi-Fi interface allows to collect Wi-Fi data, being therefore perfect to work as Anchors. The several IO interfaces allow to extend the board enabling the use of external sensors to track displacement and direction.

3.1 Anchors

Each Anchor is based on a single RPi3 (Figure 5.9). The low cost and low power consumption are advantages in the deployment scalability. Even a Raspberry Pi Zero W v1.1 (Figure 5.10) with integrated Wi-Fi could be used, reducing the deployment cost from 29€ (RPi3)[1] to 11€ (RPi Zero) per unit[2].

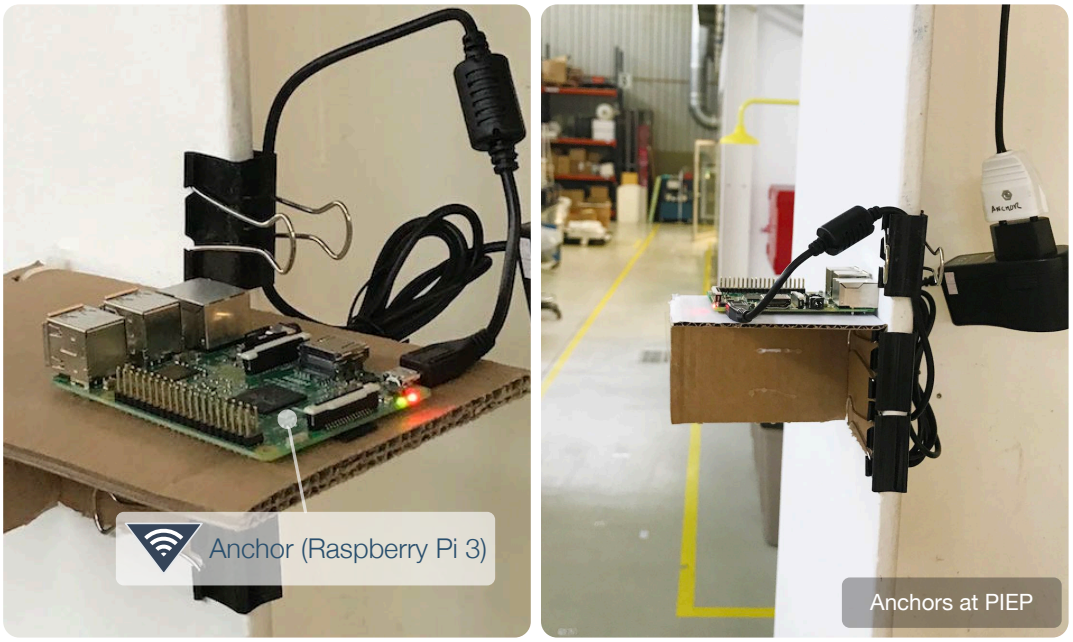

Figure 5.9: Photo of Anchors (PIEP)

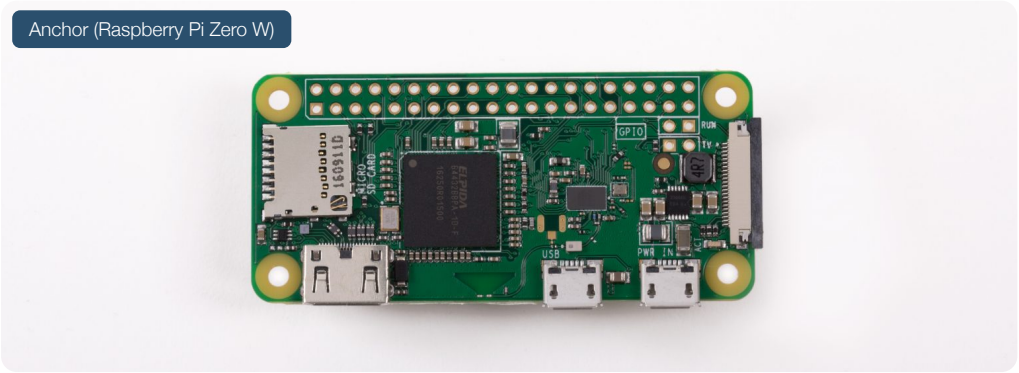

Figure 5.10: Raspberry Pi Zero W[3] (11€ board) that can be used to implement the Anchors

In these experiments the RPi3 was used for the anchors since it can be used on other experiments. Using the RPi Zero W the PIEP setup would cost 66€ (instead 174€ when using RPi3) and the DSI-DEP deployment would cost 143€ (instead of 377€), which is very inexpensive.

[1]Price in 06.06.2018 from https://pt.rs-online.com
[2]Price in 06.06.2018 from www.kubii.fr
[3]From: https://www.raspberrypi.org/products/raspberry-pi-zero-w/

3.2 Moving Testing Unit (MTU)

To evaluate the FastGraph as an industrial positioning application, a Moving Testing Unit (MTU) was developed (Figure 5.11). The MTU allows to emulate an autonomous robot or machine moving inside a factory.

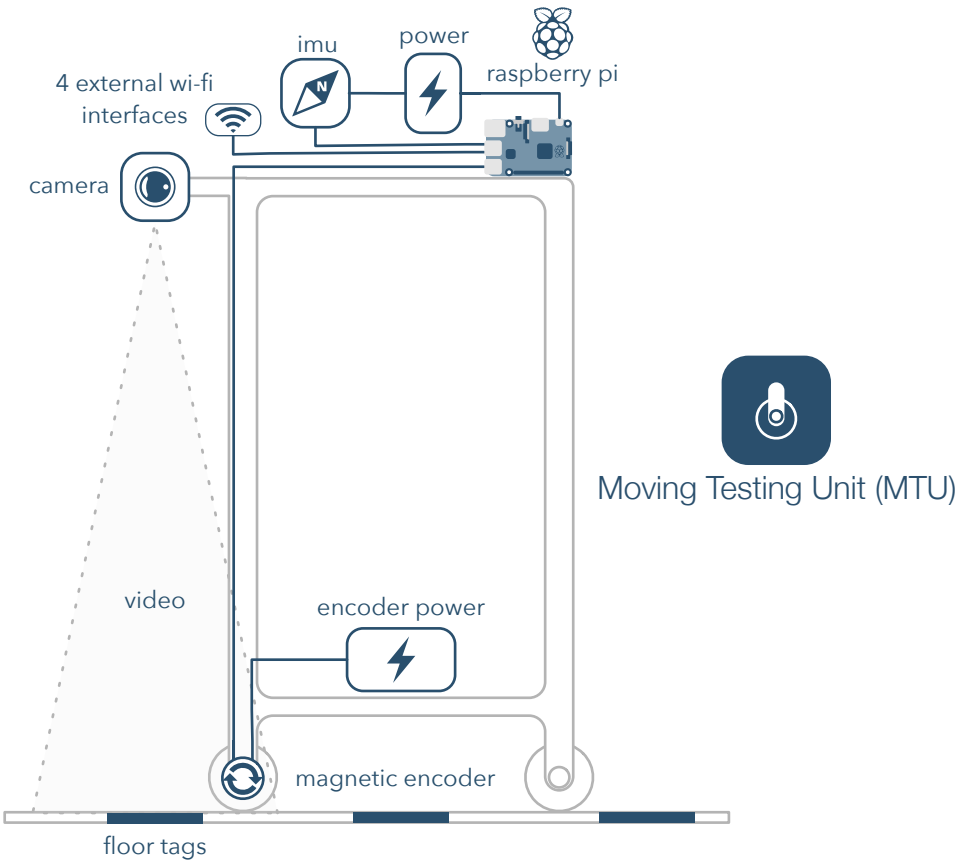

Figure 5.11: Moving Testing Unit (MTU) Diagram

The MTU core is also based on a RPi3. The USB interfaces allow the integration of external sensors. To measure displacement a Magnetic Encoder was used. The Magnetic Encoder measures the angular rotation of the wheel, that is then converted into distance. To track the direction of movement, an Internal Measurement Unit (IMU) was used, that measures several parameters such as Roll, Pitch, and Yaw. For this specific solution only the Yaw was used, giving the orientation of the device.

A camera (smartphone) was also installed in the MTU to keep track of the ground truth tags. In the MTU were also added four additional 2.4 GHz Wi-Fi interfaces to collect additional data. Figure 5.12 shows photos of the described MTU unit. The IMU used was the Xsens MTi-300-2A5G4-DK[4] and the encoder was the US Digital A2 Absolute Encoder[5].

[4]https://www.xsens.com/products/mti-100-series/
[5]https://www.e-motionsupply.com/product_p/a2.htm

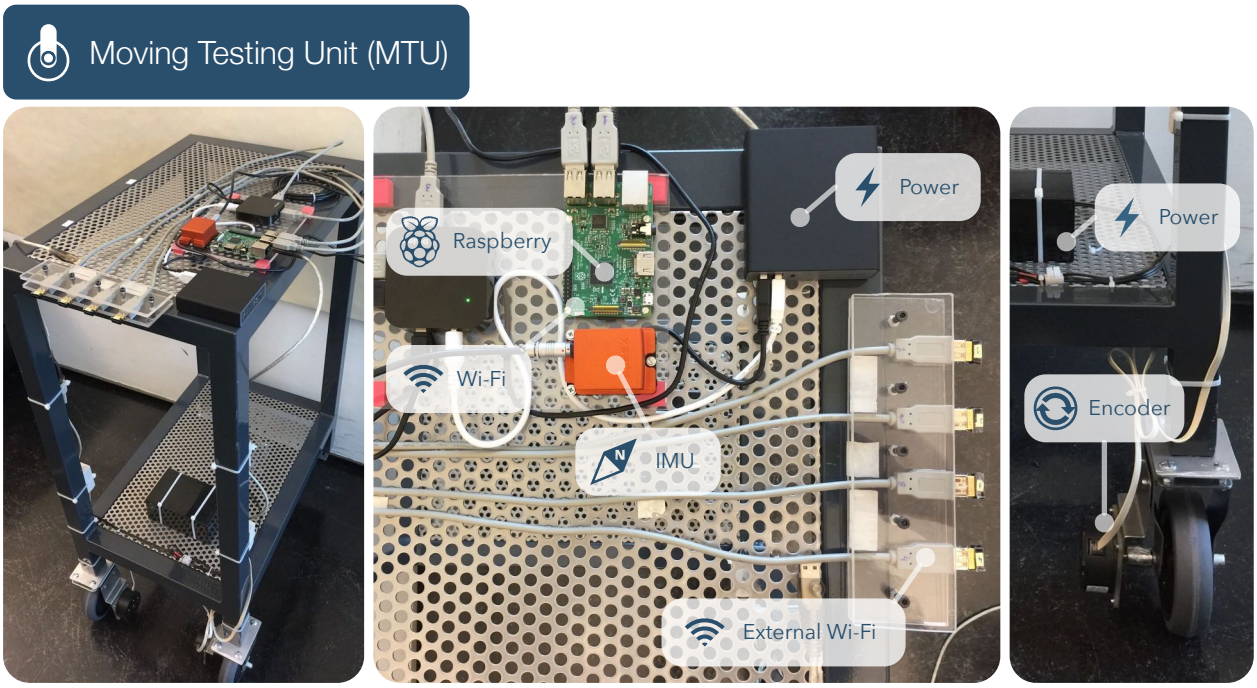

Figure 5.12: Moving Testing Unit (MTU) Photos

3.3 Tripod Testing Unit (TTU)

In the DSI-DEP experiment the Anchors were based in the same hardware and software used at PIEP. However, at DSI-DEP, in order to simplify the experiment preparation, instead of having several Anchors installed in fixed positions, it was decided to use an Anchor that could be easily moved.

With this objective a tripod base unit was designed to collect data. The Tripod Testing Unit (TTU) diagram can be seen in Figure 5.13. This unit has a Raspberry Pi, four additional external Wi-Fi interfaces, a power bank, and a Wi-Fi AP. The Wi-Fi AP installed in the TTU provide direct communication with the Raspberry Pi in order to control and monitor the operation. A smartphone can also be attached to collect several other types of data, such as cellular, acceleration and orientation. The TTU is light and can be easily moved, therefore can be used not only to work as an Anchor, but also as a Moving Device in applications where the magnetic encoder and the IMU is not necessary.

This simple approach, based on a tripod, can be easily placed at any location, and the height and direction of the testing unit can be precisely set. The photos in Figure 5.14 shows the TTU used as an Anchor to collect data at DSI-DEP.

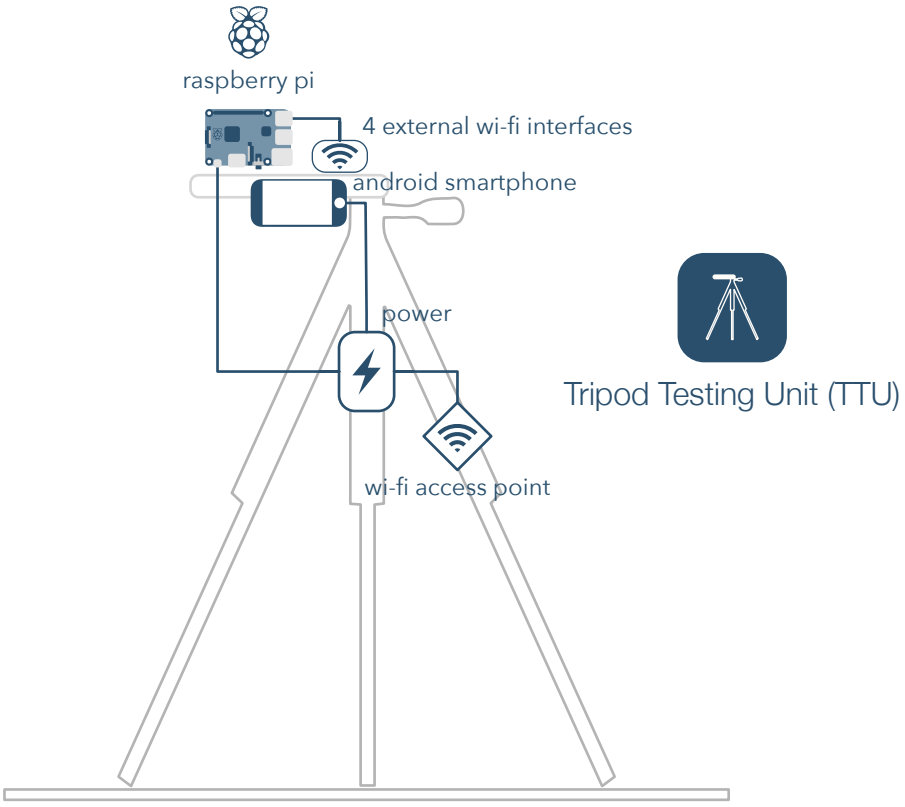

Figure 5.13: Tripod Testing Unit (TTU) Diagram

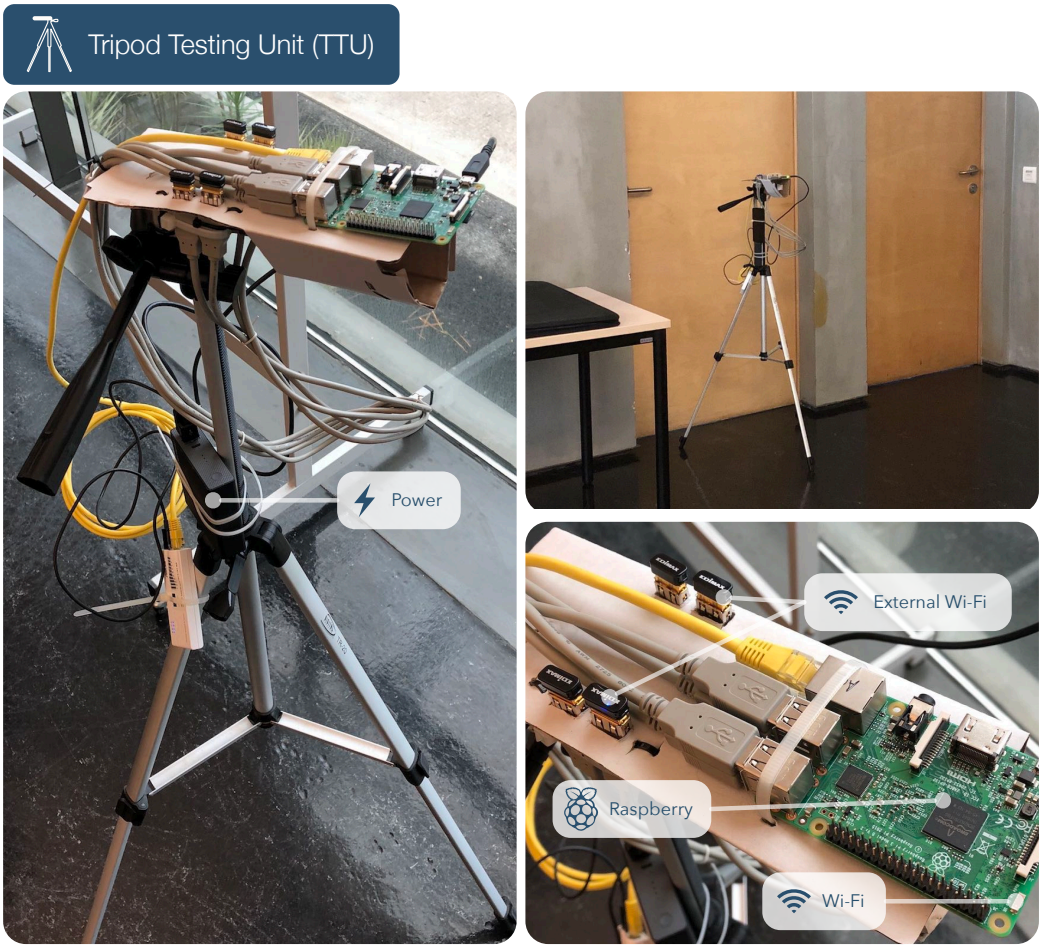

Figure 5.14: Tripod Testing Unit (TTU) Photos

4 Software Modules

The FastGraph deployment prototype also required the development of software modules.

4.1 Data Collection Modules

In the Anchors and Moving Device, the Wi-Fi samples are collected periodically by a python script that automatically runs when the device starts. This script collects several information about the nearby APs, such as BSSID, RSS, Channel and Frequency.

In the MTU, another python script reads the angle given by the magnetic encoder mounted in the wheel. The IMU information, specifically the Yaw or direction, is obtained by a program written in C.

All of this information is stored in SQLite databases, one for each sensor or interface in order to avoid concurrency problems.

4.2 Support Modules

The time synchronization is not important for the solution operation, but for the evaluation it is convenient that all devices are in the same time frame. A Raspberry Pi is not able to keep the clock synched if disconnected from power, it corrects the time when connected to the Internet. To adjust the time in all devices without relying that they have an Internet connection, a clock sync script and an Android application were developed. The script communicates with the mobile application in the Android allowing all devices to sync their time by the time provided by the Android smartphone.

4.3 Android Data Collection Application

An Android mobile application was also developed to collect several types of data (Figure 5.15). In this application was integrated an algorithm developed to detect motion in order to save energy when the device is in stationary state [59]. This algorithm can be configured or completely disabled. When the motion algorithm is disabled, data is collected even when the application is closed, with all the operations performed in a background service.

Several parameters can be configured in the main interface, such as the sampling periods, and the data collection from each sensor or interface can be individually enabled or disabled. All the data collected is stored in a local SQLite database and also sent to a server. The number of samples stored in the device is displayed in the App and can be managed by source. The App interface displays also information regarding the running time of the service, the time

collecting data and the time suspended.

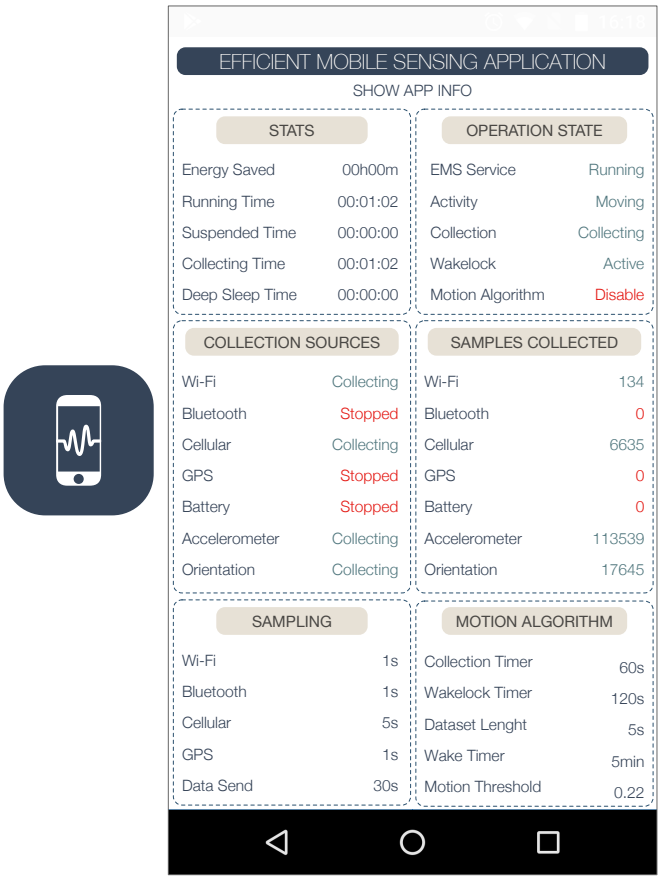

Figure 5.15: EMSA data collection application for Android

This application is an useful tool to obtain data from several interfaces and sensors on Android devices, data that can be used to test different solutions of positioning, maintaining the energy efficiency. Cellular data was also collected with this app, and will be useful to test the possible application of the proposed solution with cellular networks.

Data collected by the app:

- **Wi-Fi:** SSID (Service Set IDentifier); BSSID (MAC Address); RSSI (Received Signal Strength Indication); Channel; Capabilities
- **Accelerometer:** X-Axis Acceleration; Y-Axis Acceleration; Z-Axis Acceleration
- **Gyroscope/Magnetometer Fusion:** Yaw; Pitch; Roll
- **Celular:** MCC (Mobile Country Code); MNC (Mobile Network Code); LAC (Location Area Code); Cell ID; RSS (Received Signal Strength)
- **Bluetooth:** MAC Address; Device Name; Class; RSS (Received Signal Strength); Services

- **GPS:** Latitude; Longitude; Speed; Altitude; Accuracy
- **Battery:** Percentage

4.4 Experiment Monitoring Application

The devices can be controlled by an SSH (Secure Shell) connection. In addition, an iOS App for iPad was developed to monitor the operation process, where the most recent information collected by each device can be seen.

Figure 5.16: Monitoring iOS Application (iPad)

5 Summary and Discussion

In this Chapter were described two real world spaces, with very distinct characteristics, were FastGraph was tested. In addition, were also presented the prototype devices and software modules developed to allow the FastGraph deployed and evaluation.

The PIEP is an industrial environment, similar to a factory, with around ≈1000 m^2 of area. In this space several metallic elements and large industrial machines lead to severe multi-path and reflection effects and there are multiple Non Line of Sight (NLoS) areas. In PIEP were installed eleven APs, and to deploy the FastGraph 6 fixed Anchors were deployed, which corresponds to a density of ≈ 1 Anchor for each 166 m^2 square meters. The positions of the APs were mapped, and a ground truth grid with floor tags was installed.

The FastGraph was also deployed at DSI-DEP, an office building at University of Minho. A floor of this building has a combined square footage of around 4638 m^2. This space has characteristics very distinct from PIEP, with a more complex and larger layout, and is highly dynamic in terms of density of people. In this space can be detected a large number of APs, therefore the installation of additional APs was not necessary.

Since was not possible to install fixed Anchors at DSI-DEP the developed Tripod Testing Unit (TTU) was used as a "virtual" Anchor to collect reference samples in 13 different positions. The Anchor density at DSI-DEP was ≈ 1 Anchor for each 357 m^2, resulting in a significantly lower Anchor density in relation to PIEP. Fifteen APs installed in corridors and public access rooms, at the first and second floor were mapped.

Different prototype devices were developed to allow the FastGraph deployment. The Anchors were based on Raspberry Pi 3, but can also be used a Raspberry Pi Zero, which reduces the deployment costs. A Moving Testing Unit (MTU) with multiple sensors and Wi-Fi interfaces was also developed and can be used for example to simulate an autonomous machine. To the MTU was also attached a camera to record the ground truth tags. Was also developed the Tripod Testing Unit (TTU) that can be also used as moving device or as a movable "virtual" Anchor. This unit can be easily moved and placed at any location.

Different software modules (Python Scripts, Java Applications, Mobile Applications) to collected the data from the different interfaces and sensors and also to support the experimentation process were developed.

Chapter 6

Results

The previous chapter described the experimental setups and the deployment of FastGraph prototype in two spaces. The DSI-DEP, which is an office type building, and the PIEP, an industrial type building.

In this chapter, the experiments performed in these two spaces are described, and the results obtained are presented and discussed. These real world results are important to support and validate the initial proposed hypothesis, as well as the FastGraph performance as a unsupervised positioning system.

The first section presents the results obtained at DSI-DEP, where the solution was deployed and tested using only Wi-Fi data.

Section II presents the results obtain at PIEP, where the accurate ground truth grid allowed to extend the experiments. With the PIEP's setup it was possible to test the FastGraph in scenarios where Wi-Fi can be combined with data from other sensors to improve the accuracy.

This chapter ends in Section III, comparing the obtained positioning results with other state of the art positioning solutions.

1 Experiments at DSI-DEP

Wi-Fi only positioning solutions are the most suitable to provide positioning in large spaces, such as Airports, Shopping Malls, Stadiums or Office Buildings, since the other types of sensors are difficult to explore in these types of environments.

The DSI-DEP office type building characteristics are suitable to evaluate the FastGraph as a Wi-Fi only positioning solution, where techniques such as Fingerprinting are normally used.

As mentioned before, a large number of APs can be detected at DSI-DEP (Figure 6.2), but many of them are inside private offices or installed in nearby buildings. To evaluate the AP location feature of FastGraph, the real position of APs was mapped at the corridors and some inside rooms with public access, in the first and second floors (Figure 6.1).

In these experiments data was collected from the 13 “virtual” Anchors and from a moving device (TTU). In each device, the Wi-Fi data was collected by a single 2.4 GHz Wi-Fi interface.

The first set of results show the algorithm APs’ position estimation for the APs at two different floors. The second set of results show the positioning error with the algorithm operating using only Wi-Fi data.

The Graph initialization phase, where only samples from Anchors were processed, took only around 6 minutes worth of data, after which the position estimation of the moving device was initiated.

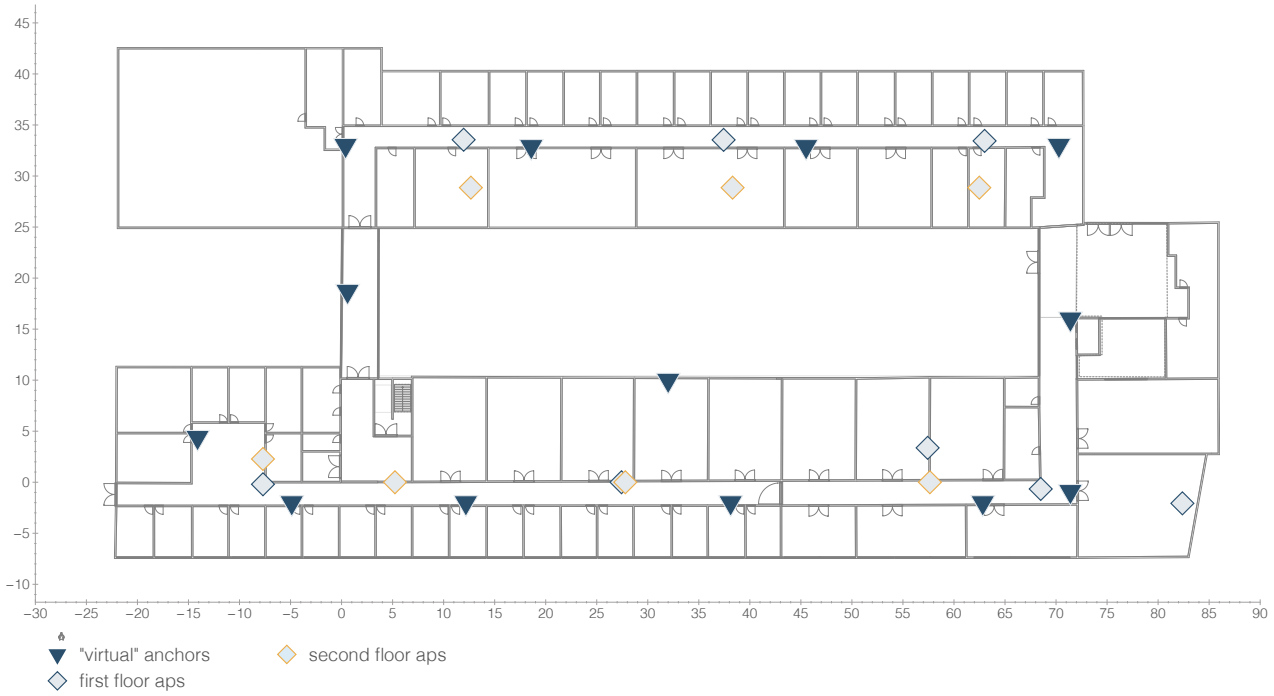

Figure 6.1: DSI: Anchors and Mapped APs

Data collected:

- Anchors Wi-Fi samples: 4066 ($\approx$ 313 per Anchor)
- TTU Wi-Fi samples: 455 (10 positions)

1.1 Access Points Position Estimation Error

Figure 6.2 shows all the APs detected at DSI-DEP. The position of each AP in the figure was estimated by FastGraph. As shown, the algorithm placed some APs outside of the DSI-DEP building, since there are other buildings with APs nearby.

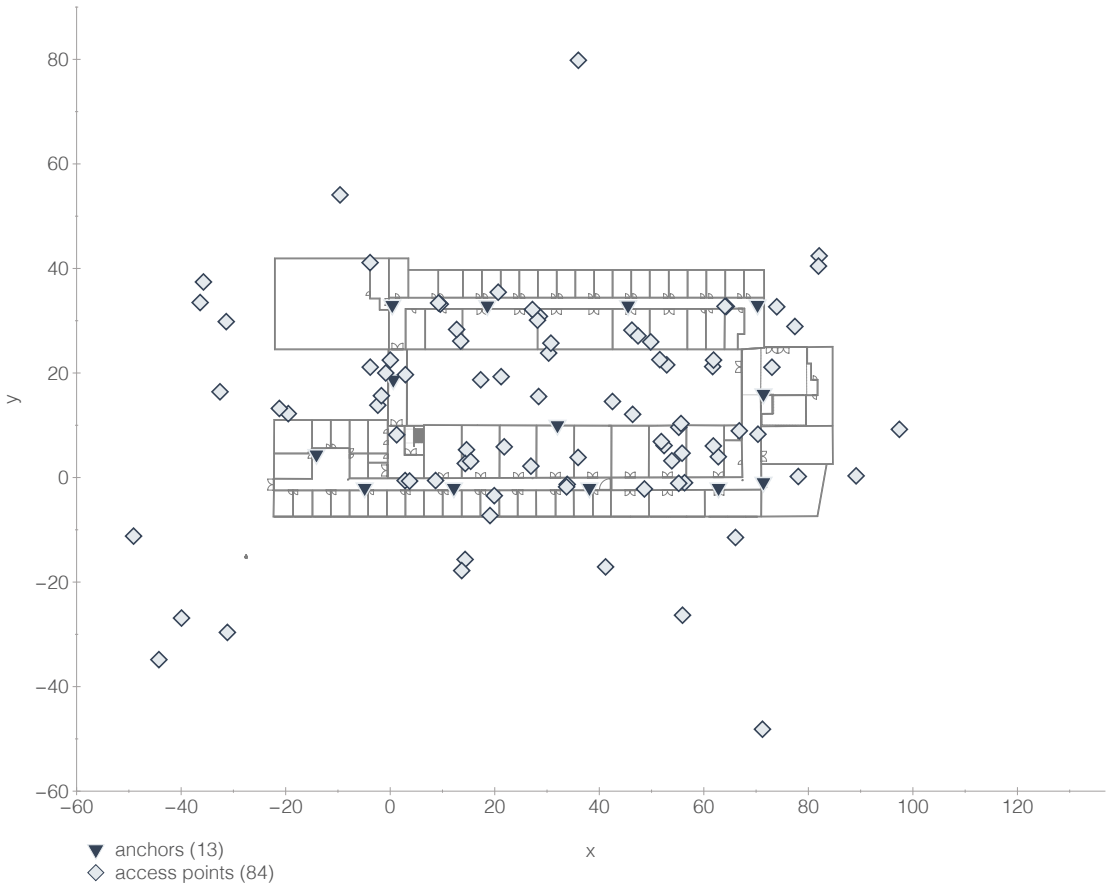

Figure 6.2: APs observed at DSI

For the APs mapped at the first floor (see Figure 6.1), the average XYZ error was 8.21 meters with a max error of 15.88 meters (Figure 6.3 and Table 6.1).

AP	XY Error	Z Error	XYZ Error
AP1	6.56	0.35	6.57
AP2	5.39	0.20	5.39
AP3	10.30	0.35	10.31
AP4	2.41	0.35	2.44
AP5	1.48	0.37	1.52
AP6	11.42	0.48	11.43
AP7	12.19	0.43	12.20
AP8	15.87	0.49	15.88
Average	**8.20**	**0.38**	**8.21**

Table 6.1: APs Position Error DSI Floor 1 in Meters

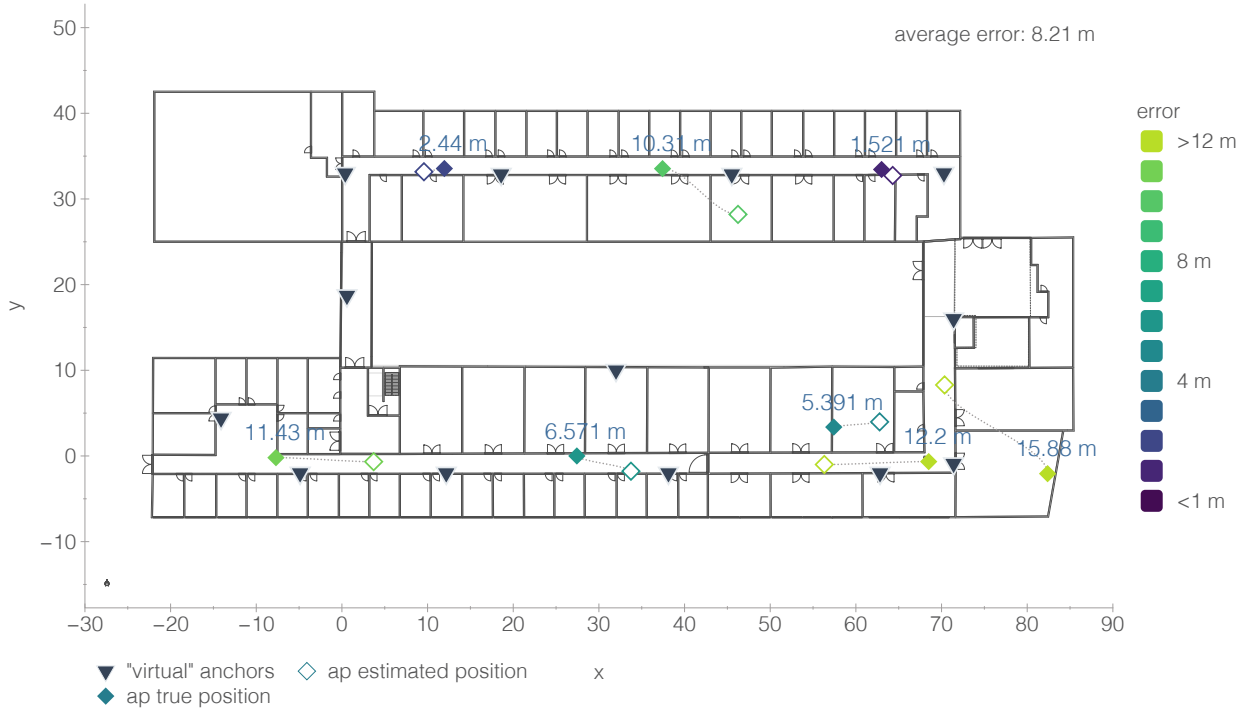

Figure 6.3: APs Error DSI: Floor 1

For the APs at the second floor the average XYZ error was 15.58 meters with a max error of 23.26 meters (Figure 6.4 and Table 6.2). It is important to note that all the Anchors were at the first floor, and the position of all APs was estimated based only on these Anchors. The larger error in the position estimation for the APs at second floor is therefore expected. For optimal results Anchors should be added to the second floor.

AP	XY Error	Z Error	XYZ Error
AP9	9.45	2.62	9.81
AP10	11.04	1.67	11.17
AP11	0.54	0.89	1.04
AP12	23.25	0.89	23.26
AP13	20.91	1.67	20.97
AP14	19.24	3.71	19.59
AP15	22.87	4.22	23.25
Average	**15.32**	**2.24**	**15.58**

Table 6.2: APs Position Error DSI Floor 2 in Meters

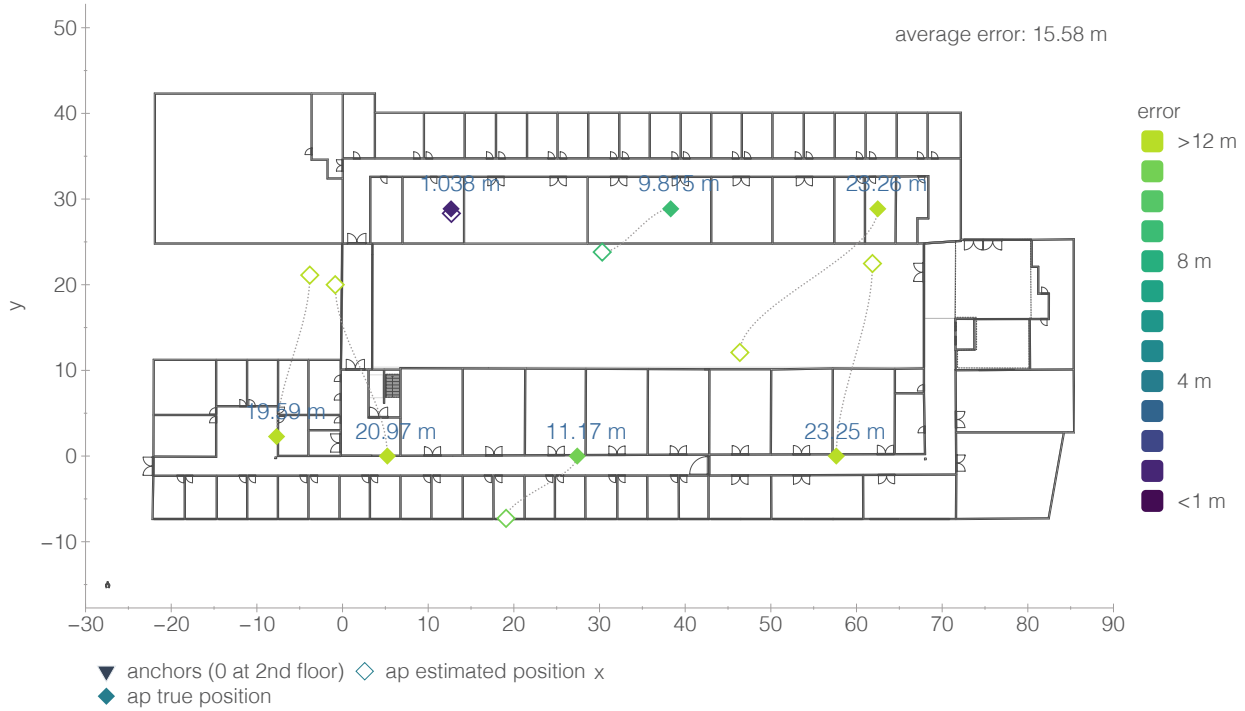

Figure 6.4: APs Error DSI: Floor 2

These results were compared with Serendipity [66], that locates APs using reference scans at known positions, and the dissimilarities between pairs of APs. Serendipity requires a dense AP coverage, with the overlapping of at least two APs. Therefore an AP cannot be located individually. Also scans in all floors are required. The number of floors has to be known in order to cluster the APs in different floors, a 2D position is then estimated for each AP. An error between 3.5 meters and 6.7 meters is reported, for two buildings with floor area of 1000 m^2 and 1750 m^2.

FastGraph estimates a 3D position for each AP. The 3D error of 6.73 meters at PIEP (in an area of 1000 m^2), and the 8.2 meters error in the first floor of the DSI-DEP (with area of 4638 m^2), are interesting results. The results at the second floor of the DSI-DEP are not

comparable, since those APs were located without any scan in that floor.

1.1.1 Access Points Floor Estimation

Figure 6.5 shows the XZ plan, with the estimated position of the APs at the first and second floors.

Is evident that even without Anchors at the second floor, the FastGraph placed almost all APs in the correct floor. The exception are two APs of the second floor, which are estimated to be in the floor limit.

This suggests that this results can be improved with Anchors also in the second floor. A very small error in the Z position, similar to the APs in the first floor, is also expected.

Discussing multi-floor applications is out of the scope of this research, but these results strongly suggest that FastGraph can easily support multi-floor positioning. Moreover, in some scenarios may even not require Anchors in all floors.

Figure 6.5: APs Z Error DSI 1st and 2nd Floors

1.2 Wi-Fi Only Positioning

Figure 6.7 shows the positioning results at DSI-DEP, at ten testing positions across the space (Figure 6.6). The samples were collected in testing positions at different distance from each other in order to verify the error variation between close and far positions. In each position a different number of samples was collected with an average of ≈ 46 samples per position. All the APs detected were added to the Graph, and used in the positioning process, including the APs at the second floor, which have higher error. An average error of 5.08 meters with a max error of 14.86 meters was obtained.

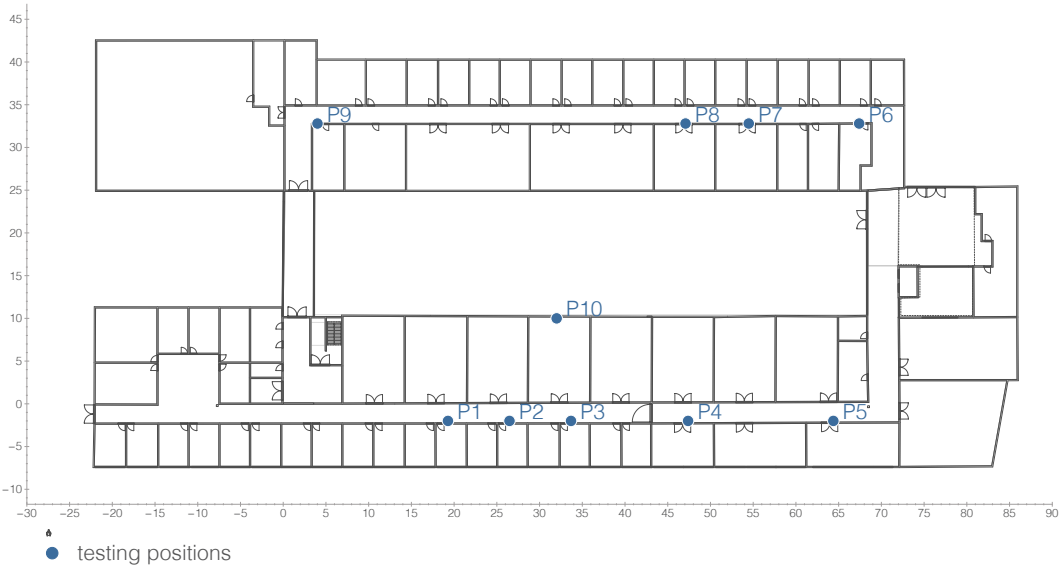

Figure 6.6: DSI-DEP Testing Positions

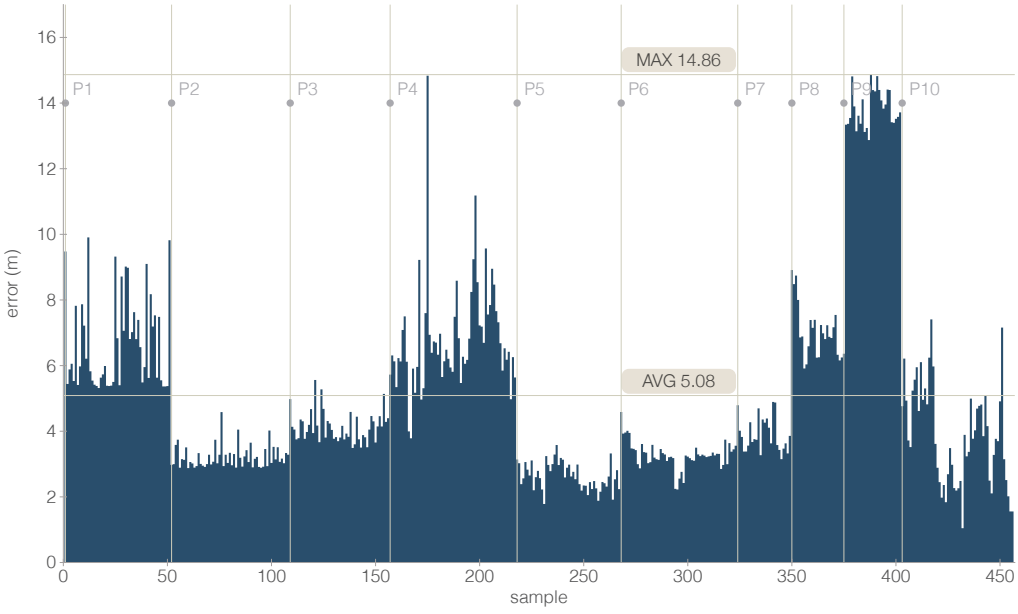

Figure 6.7: RT Position Error: WiFi Only at DSI

Figure 6.8 shows the CDF of the error, with 6.21 meters for the third quartile.

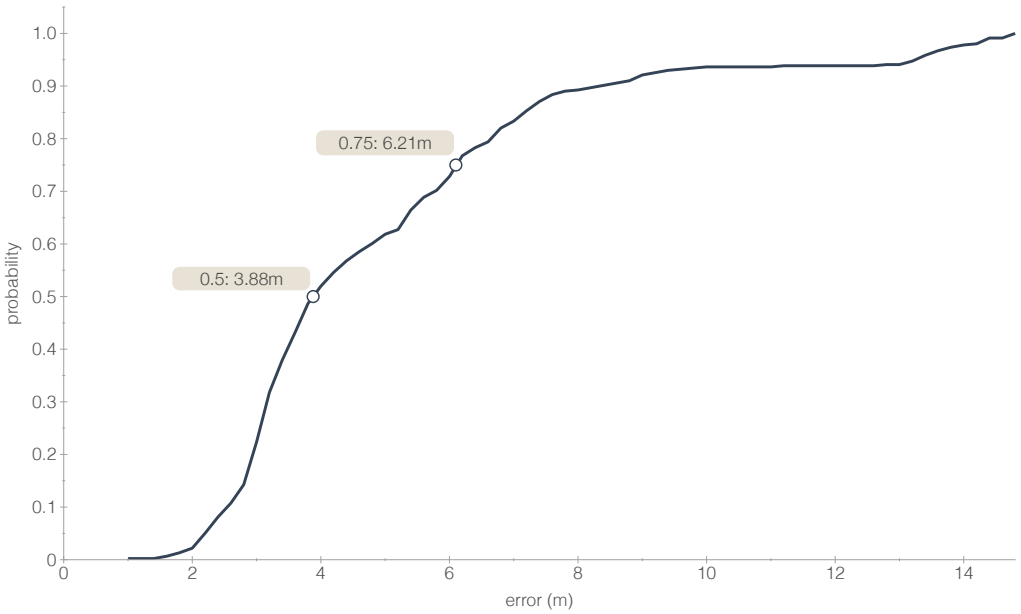

Figure 6.8: Real Time Positioning Error CDF

Adding Anchors to the second floor, the APs' position estimation is expected to improve, also improving the positioning results. Moreover, a method to select a set of APs to be used may also be useful. Previous works suggest that using all the APs detected can in fact produce worse positioning results [66].

The histograms of the error in X and Y (Figure 6.9) shows that the error distribution is in general around zero, with the tendency more evident in the X.

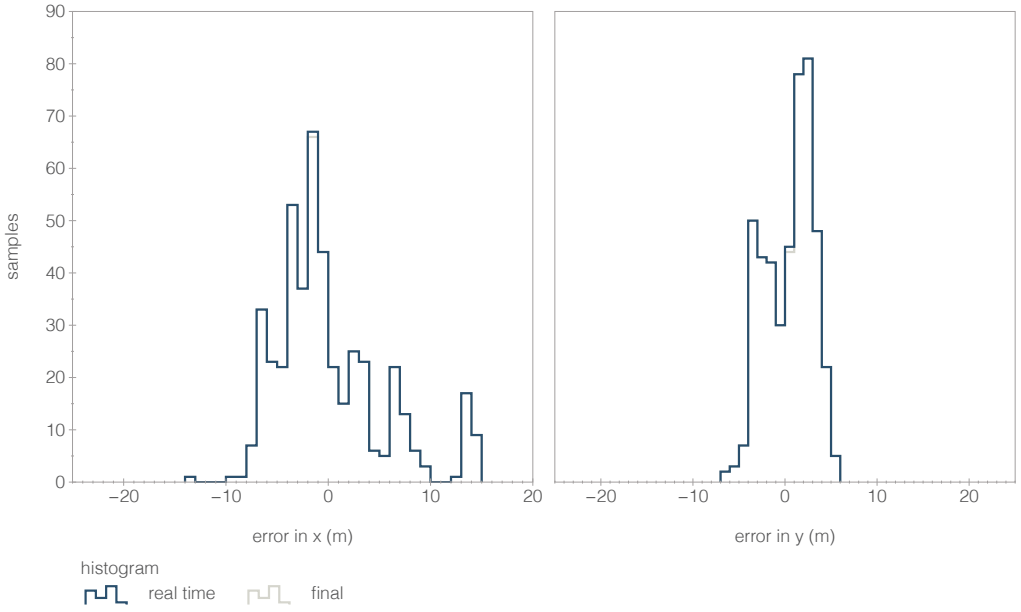

Figure 6.9: Wi-Fi Only DSI XY Error Histograms

When the same data is processed several times the results that are obtained may vary

slightly, due to the process of creating the Graph that estimates the initial position for the APs (see Figure 4.3), and the progressive path loss exponent learning. After processing the data 10 times, the average error in real time was 5.20 meters and the overall max error was 14.86 meters (Table 6.3). These results show that the algorithm operation is consistent.

Run	**Average Error (m)**	**Max Error (m)**
1	5.23	14.84
2	5.24	14.85
3	5.37	14.86
4	5.06	14.85
5	4.94	14.85
6	5.53	14.85
7	5.09	14.85
8	5.23	14.83
9	5.22	14.85
10	5.07	14.84
Average	**5.20**	**14.85**

Table 6.3: Experiment Repetition

The results at DSI-DEP, where only Wi-Fi data was used, show that the error magnitude is similar for the samples collected in each test position. Some near testing positions have similar error (e.g. P2 and P3), but other close positions have different error (e.g. P1 and P2 or P7 and P8), and this is understandable considering that even in close positions the propagation effects can be different. The position P9 has the worst results and this can be explained by the fact that the samples were collected exactly in a corner (see Figure 6.6) where the multi-path and scattering effects, as well as the level of interference (as will be discussed in Chapter 7) are more significant and lead to higher errors.

The results (see Figure 6.7) also show higher variation of the error in position P4. These variations may be due to the proximity to a door being opened and closed several times by people passing through the corridor. Also on multiple occasions, especially in the initial position P1, people with curiosity leaned over the TTU to see the device, severely attenuating the radio signals and resulting in variations in the error.

2 Experiments at PIEP

The PIEP space provided the ideal conditions to deploy and experiment the FastGraph in an industrial environment. These experiments were important to validate the solution in applications such as indoor navigation of autonomous machines, where orientation and displacement data can be combined with Wi-Fi. Moreover, PIEP has distinct propagation characteristics, which allowed to further evaluate the performance consistency of FastGraph in Wi-Fi only operation.

The experimental setup at PIEP, with an accurate ground truth grid and all APs mapped, allowed to extend the experiments for a more comprehensive evaluation of the FastGraph. The ground truth grid was essential to evaluate the algorithm performance when using displacement and orientation, for high accuracy positioning.

Two sets of experiments were performed to evaluate the FastGraph as:

- Wi-Fi Only Positioning Application
- Enhanced Accuracy Positioning Application (where the Wi-Fi is combined with orientation and motion data).

Additional experiments also evaluate other features of the FastGraph solution:

- The evolution of all APs' position estimation, using only the Anchors (including the initialization phase).
- The APs' position error after samples containing motion information.
- The progressive η learning process.
- The influence of the number of Anchors in the positioning performance.
- The time required to obtain a position estimation.

In these experiments, Wi-Fi data was collected by six Anchors and by the MTU (Moving Testing Unit). The MTU followed four different and individual paths, with a total travelled distance of 302 meters (Figure 6.10), using the ground truth tags as reference. Between each path the MTU stopped for a few minutes, and the data collection process shutdown, simulating a device becoming offline and online. The tags of each experiment path were recorded by the

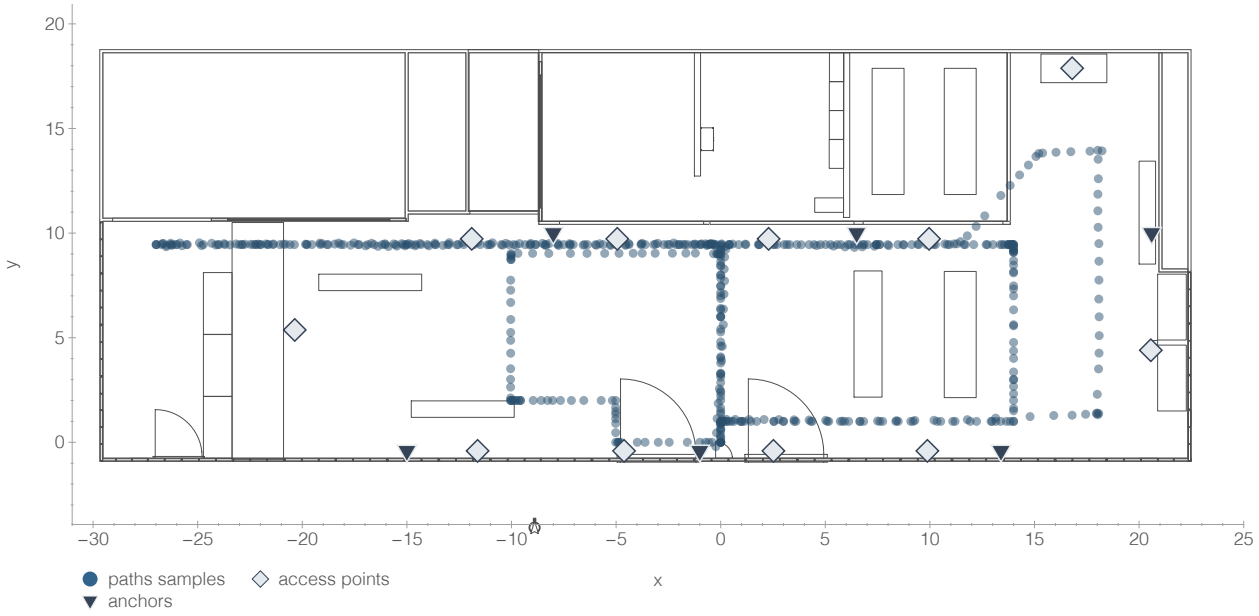

Figure 6.10: Experiment Paths

smartphones' camera installed in the MTU, and the video was used in post processing to extract the ground truth information for the paths.

The sampling periods for the Wi-Fi, IMU and Encoder are shown in the Table 6.4.

Wi-Fi	**IMU (orientation)**	**Encoder (distance)**
1 s	50 ms	20 ms

Table 6.4: Sampling Periods

Data collected:

- Anchors Wi-Fi samples: 40059 ($\approx$ 6677 per Anchor)
- MTU Wi-Fi samples: 743
- IMU readings (orientation): 61652
- Encoder readings (distance): 202550

The positioning results are presented as:

- **Real Time Error:** The error in the position estimation, obtained right after a sample is processed and the Graph is adjusted.
- **Final Error:** As the graph evolves, with more samples processed, the position of the nodes already on the Graph can change. Therefore, the final error is the error in the

position of the samples at the end of the experiment. This can be used as a metric of the resulting radio map quality.

2.1 APs' Position Estimation in the Initialization Phase

The objective of this experiment was to understand how the position estimation for all APs evolve with the number of samples processed from the Anchors.

As explained before, the Graph algorithm uses the known and fixed position of the Anchors to obtain improved distance constrains to the APs, by averaging the radio signal readings for the same AP. This process helps to reduce the noise that is related to the signal level fluctuations.

Figure (Figure 6.11) shows the sum of the position change (top plot), for all APs, at every 100 samples processed. The plot shows around 2h of sampling, with around 6677 samples per Anchor. The bottom plot shows the APs' average positioning error, in the same conditions.

These results show that, as expected, the APs positions' average error decreases when more samples from the anchors are processed. The APs positions start stabilizing as more samples are processed. Despite the APs moving less with more samples, their position error keeps dropping, as smaller corrections are made based on the new information provided by the new samples. These results suggest that, with time, the anchors samples allow to perform small corrections in the APs positions, improving the their position estimation.

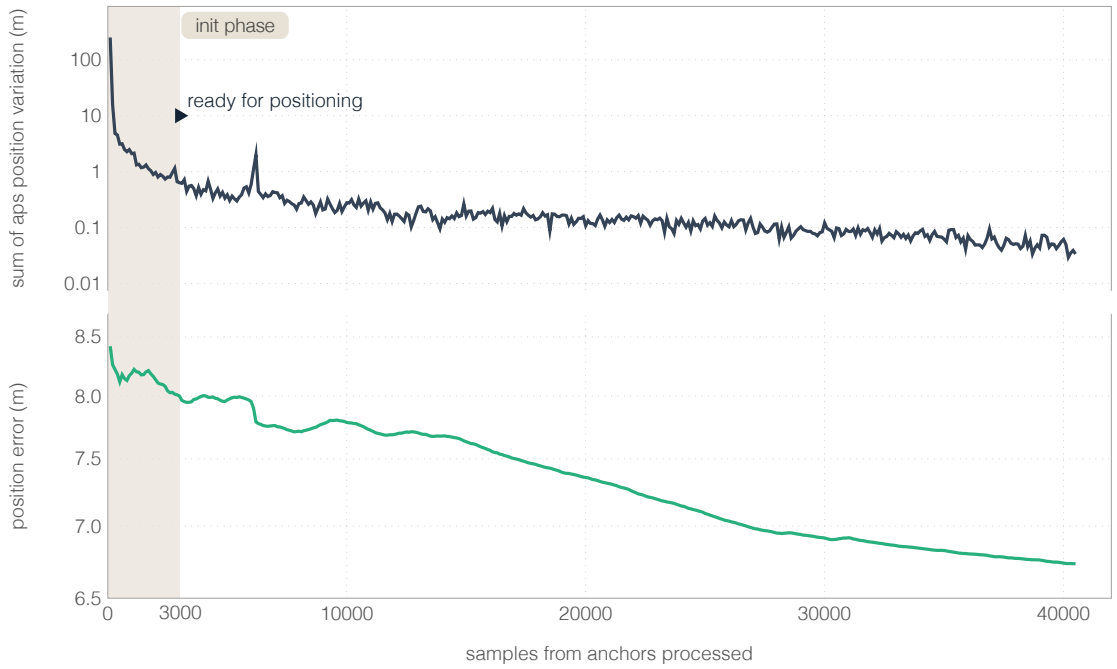

Figure 6.11: APs positions variations over samples processed (40500 samples)

The Graph initialization phase took only around 8 minutes of sampling data from the Anchors (around 500 samples from each Anchor), after which the system was ready to provide

positioning.

The APs' average XYZ position error, using only the Anchors' data, was 6.73 meters. The errors for each AP are presented in the Table 6.5 and Figure 6.12. As the results show, the average error is considerable affected by the AP11, which has a error much larger than the other ten APs. In addition, the error in Z is consistent, and small for most of the APs.

AP	XY Error	Z Error	XYZ Error
AP1	4.79	1.54	5.04
AP2	7.20	1.54	7.36
AP3	3.73	1.24	3.93
AP4	4.33	0.44	4.36
AP5	4.17	0.0	4.17
AP6	5.0	2.87	5.78
AP7	8.82	0.0	8.82
AP8	3.67	2.96	4.71
AP9	5.26	1.33	5.43
AP10	6.90	0.5	6.92
AP11	17.47	1.33	17.52
Average	**6.49**	**1.26**	**6.73**

Table 6.5: Anchors Only: APs Position Error in Meters

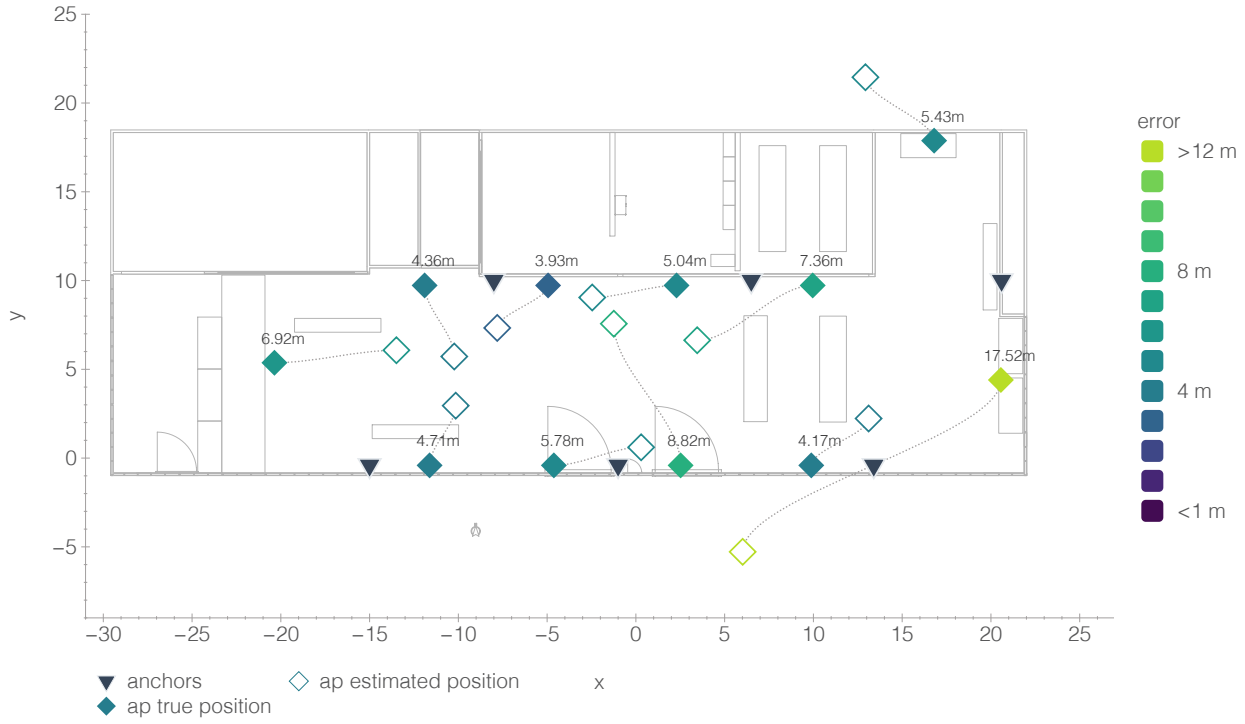

Figure 6.12: Anchors Only: APs Position Error Map

2.2 Wi-Fi Only Positioning

The results that will be presented next, were obtained in an experiment to further evaluate the FastGraph in a Wi-Fi only positioning application.

It is important to note that, these results are using only Wi-Fi information, from a single 2.4 GHz interface, and without any information regarding the MTU orientation or displacement.

After the initialization phase (around 8 minutes after the system boot), which created the initial 3D Graph, with an initial estimation of the APs positions, the algorithm started estimating the MTU position, by processing its samples.

The real time positioning error obtained for each processed sample of the MTU, along the four different paths, is shown in Figure 6.13.

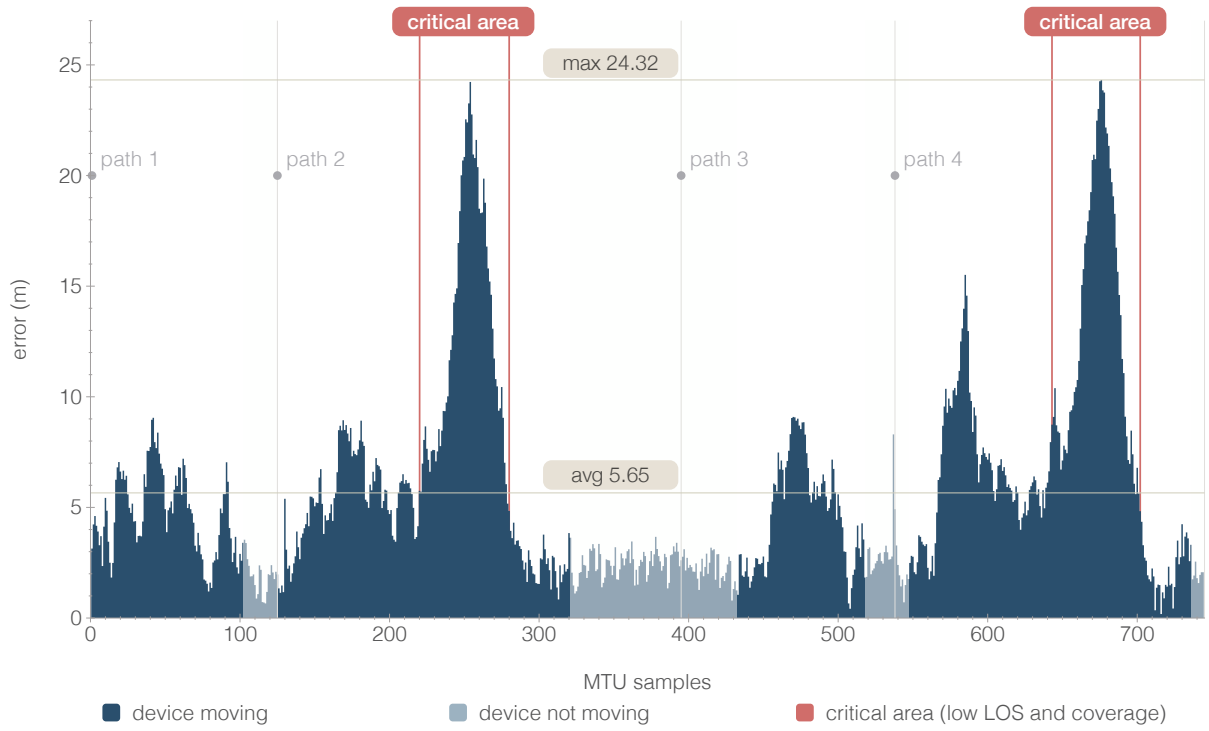

Figure 6.13: Real Time Position Error: WiFi Only

The most noticeable aspect in the positioning error plot are the two error peaks, with almost identical magnitude (around 24 meters), one during the path two and the other during the path four. In fact, these two peaks occurred when visiting the same area of the PIEP, below a metal bridge and behind shelfs. This was identified as a critical area, due to low coverage and low line of sight (LoS) with the APs.

The map in Figure 6.14 shows the position where each sample was collected. As the map also shows, there is no APs at the left side of that area, and the bridge, at a lower level than the APs, heavily obstructed the signals. The color of each sample is related to the error obtained

for that sample. The critical area is identified by the red rectangle, but can also be clearly identified by a large group of samples with the light green color.

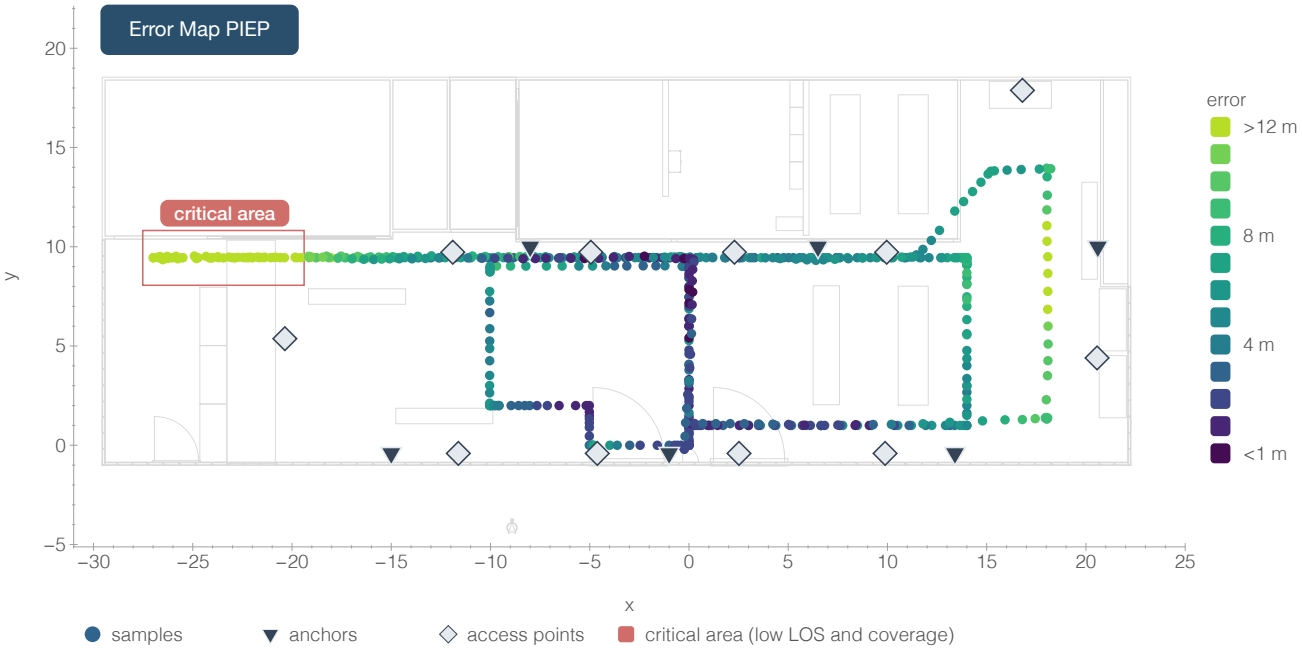

Figure 6.14: PIEP Critical Area (Error Wi-Fi Only)

In Figure 6.13 it is also noticeable that the error increases and decreases almost the same way around the two peaks, because every time that the MTU travels to the critical area, it has to make the inverse path, as there are no other possible way. From the map plot, it is also perceptible that the more central areas are less affected by error, since that areas have better coverage from the APs. In addition, the APs at the outer positions are more affected by error due to the Anchors positions, that are more centered in the space.

As shown in Figure 6.13, the average error (5.65 meters) is highly affected by the two large peaks. However, even so, this is a very interesting result, considering how fast the system becomes ready to provide positioning, and when compared to the positioning results of traditional fingerprinting approaches, which require extensive calibration. Moreover, PIEP is a complex environment, with the factory-like characteristics, full of metallic elements and machines that result in non line of sight, reflections, multi-path effects and attenuation.

Figure 6.15 shows the CDF of the position error, in real time and at the final of the process. Both CDF are very similar, but the final error CDF is slightly better, because as new samples are added, small corrections are made on the old samples of the Graph.

The critical area problem can be addressed with installation of new access points and Anchors. Just as reference, if not considering the samples collected at that area, the average error drops to 4.26 meters (1.39 meters less) with the max error dropping to 15.51 meters. Figure

6.16 shows the position error CDF with and without considering the samples collected at the critical area, where the improvement is evident.

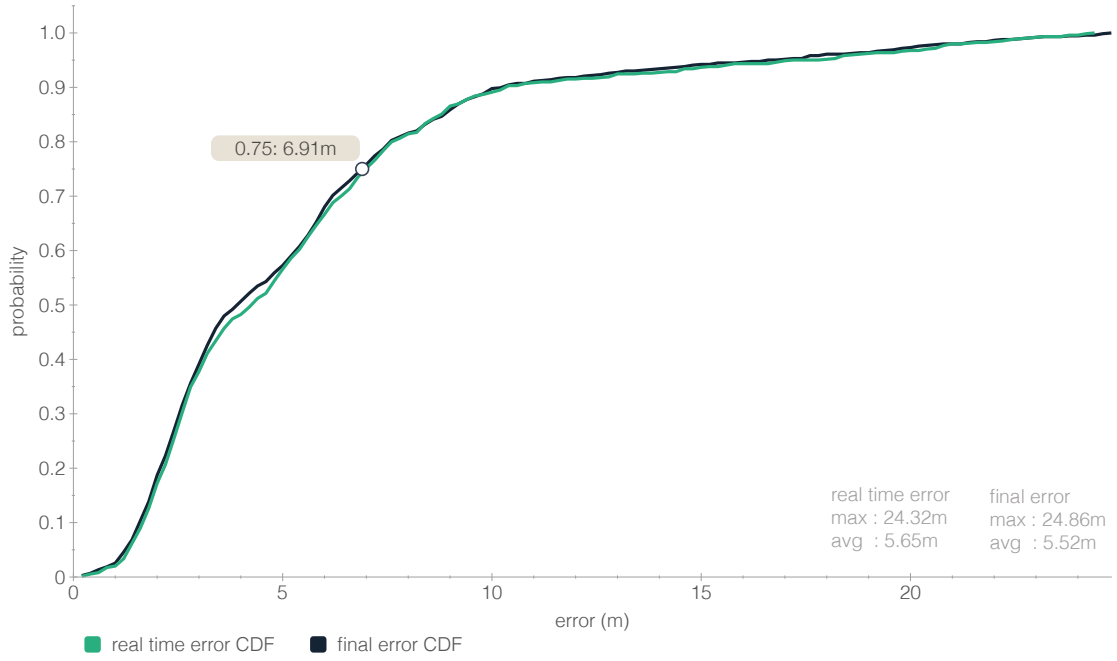

Figure 6.15: Wi-Fi Only Position Error CDF

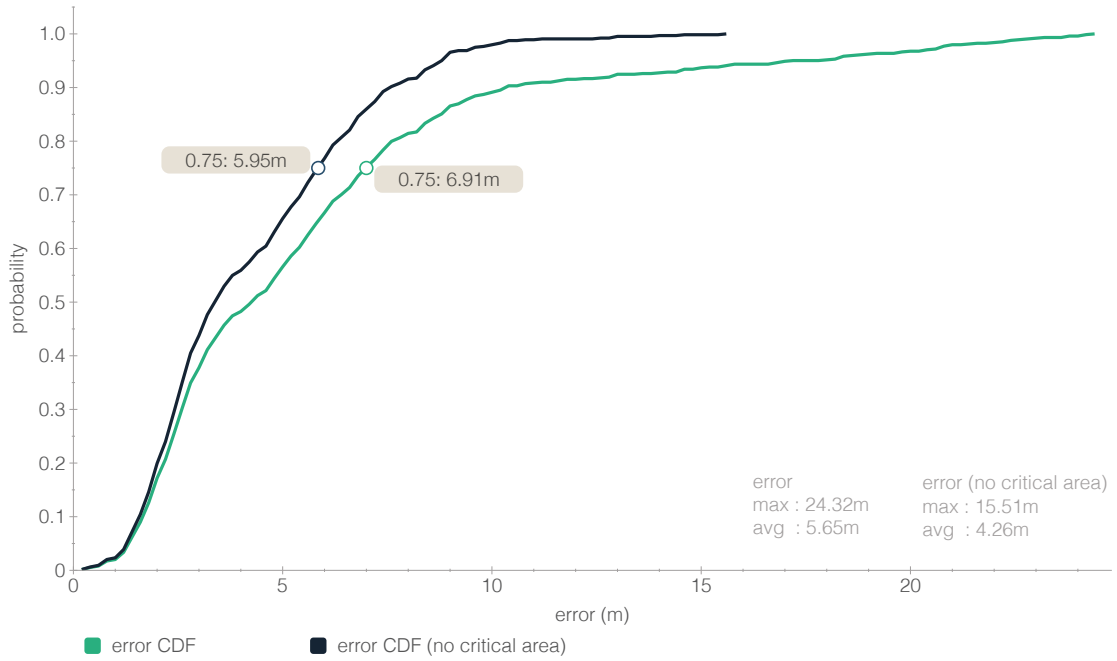

Figure 6.16: Wi-Fi Only Position Error CDF: With and Without Critical Area

The histograms of the error in X and Y (Figure 6.17) follow a normal distribution with null mean, while slightly skewed due to the critical area.

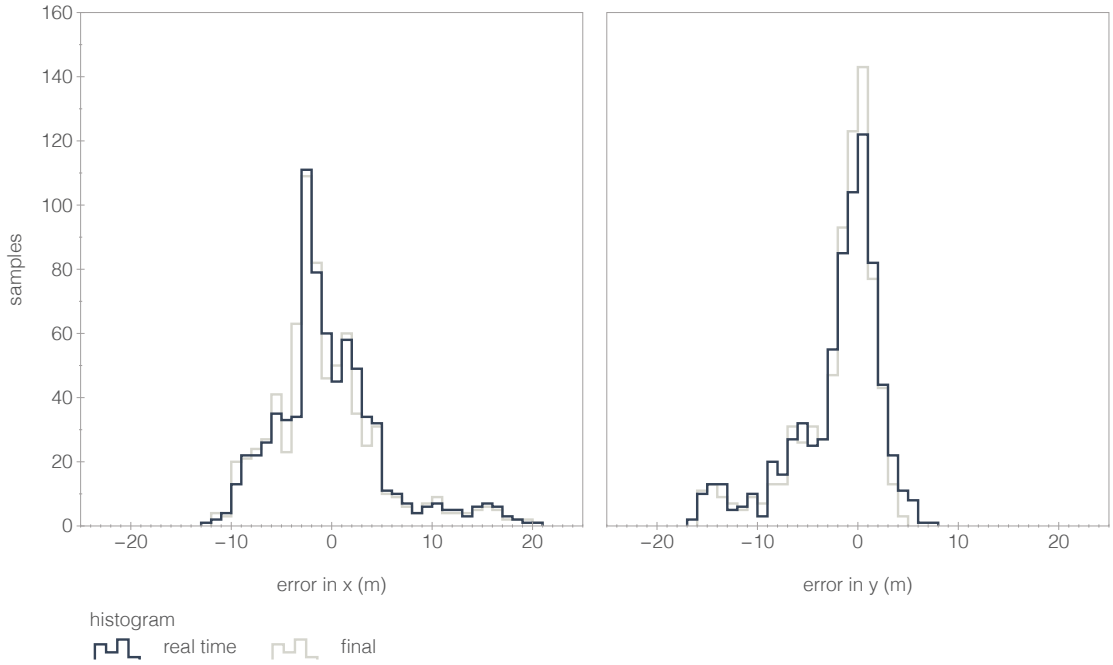

Figure 6.17: Wi-Fi Only XY Histograms

As explained before, the obtained results can vary slightly when processing the same data multiple times. After 10 runs the average error in real time was 5.65 meters and the overall max error was 27.27 meters (Table 6.6). These results show that the algorithm operation is consistent.

Run	Average Error (m)	Max Error (m)
1	5.69	24.32
2	5.67	26.11
3	5.55	26.33
4	5.58	25.55
5	5.66	26.85
6	5.71	25.48
7	5.67	26.36
8	5.68	27.27
9	5.66	26.28
10	5.67	26.50
Average	**5.65**	**26.10**

Table 6.6: Experiment Repetition

2.3 Wi-Fi Only Positioning With Known APs Positions

The data collected by the Anchors allow the algorithm to extract spatial constrains that define the positions for the APs. In Wi-Fi only applications, the estimated APs positions provide the constrains to estimate the position of a moving device. Therefore, more accurate estimation for the APs' positions means more accurate positioning for the moving devices.

Considering this hypothesis, FastGraph was further tested in a Wi-Fi only scenario, but where the algorithm can have previous knowledge about the APs positions. This experiment is relevant because, in specific applications, this information may be already available, or can be easily obtained, and can be used to improve the system performance.

Moreover, with knowledge about the APs' positions, the Anchors may even be discarded. However, if the Anchors are maintained, they can be used to better estimate the propagation parameters in specific areas, since the position of the Anchors and the APs are known. This information also helps in the estimation of the path loss exponent for the edges associated to samples collected by the target devices.

As expected, a significant drop in the average error was observed, from 5.65 meters to 3.98 meters (Figure 6.18), suggesting, that in fact, the positioning performance can be improved in such conditions.

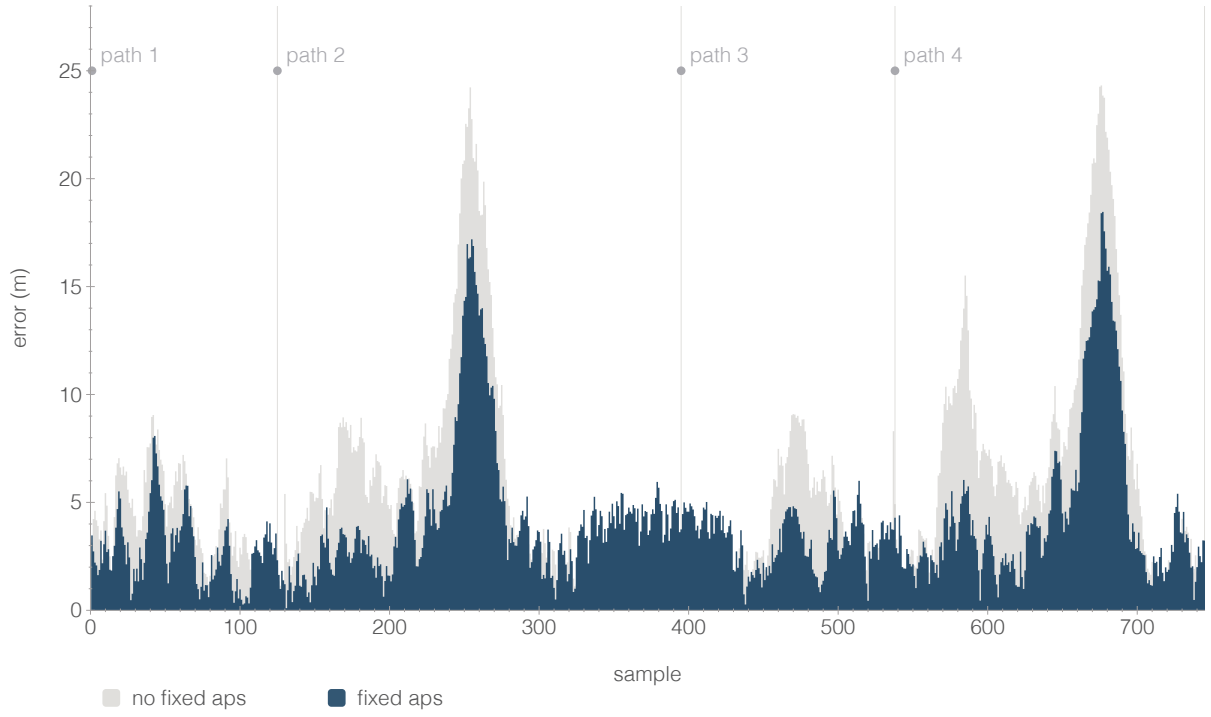

Figure 6.18: RT Position Error WiFi Only: Fixed APs

The two peaks, resulting from the critical area, are still evident, but have lower magnitude. The other smaller peak, near the sample 600, disappears. This may be related to the APs at the outer positions, which were more affected by error when their positions had to be estimated. Figure 6.19 shows the CDF of the positioning error, in real time and at the final of the process. As shown, the CDFs are very similar, as the APs position are known from the beginning, and no corrections to their positions are necessary along the process.

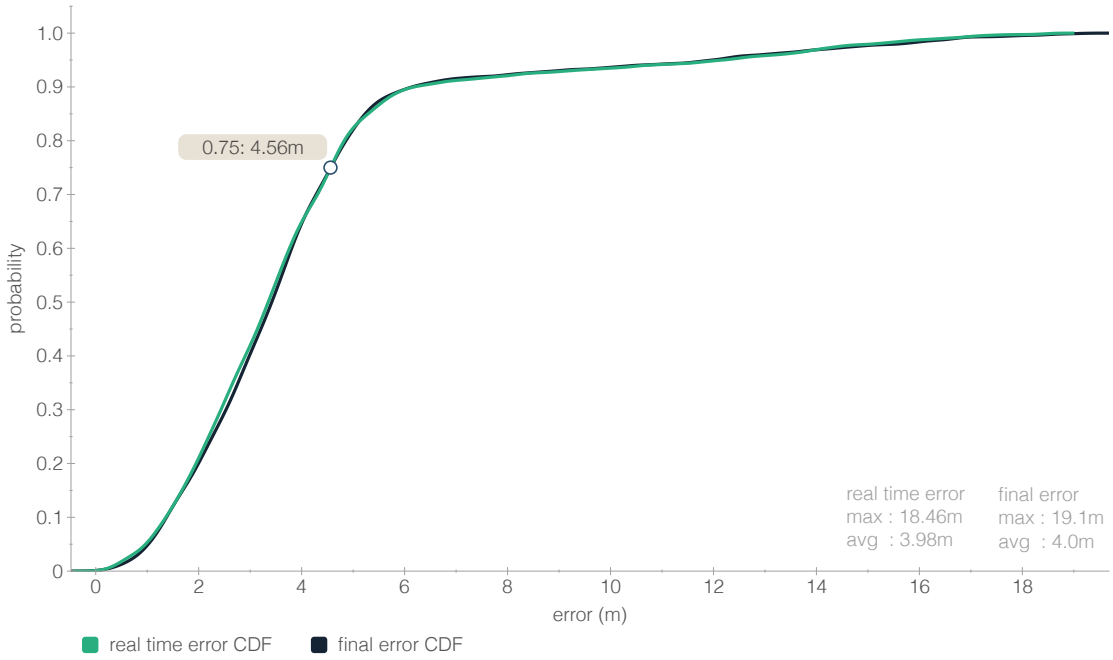

Figure 6.19: Wi-Fi Only and Known APs Position Error CDF

When compared with the experiment without the fixed APs (Table 6.7), an improvement of 1.67 meters for the average error, and 5.86 meters for the max error were observed. Moreover, for the third quartile, there is an improvement of 2.35 meters.

Data	APs Positions	Max Error	Avg Error	CDF 0.75
Wi-Fi only	Unknown	24.32m	5.65m	6.91m
Wi-Fi only	Known	18.46m	3.98m	4.56m

Table 6.7: Position Results Comparation (Wi-Fi Only)

In an analysis similar to the previous experiment, where the samples collected in the critical area are not considered, the results improve to around 8 meters of max error and 3 meters of average error.

The distributions of the error in X and Y (Figure 6.20) are similar to the previous experiment, following a normal distribution around zero, but with lower magnitude of error. The histograms of X, follow a right-skew distribution, related with the samples collected in the critical area, which have higher error.

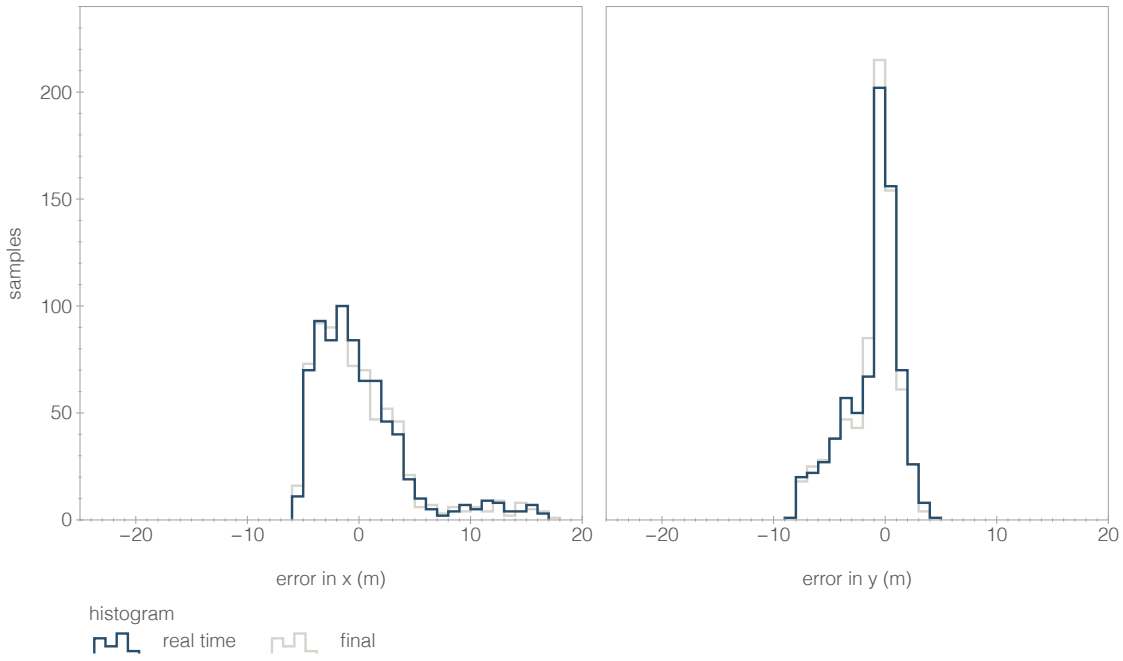

Figure 6.20: Wi-Fi Only Known APs Positions XY Histograms

2.4 Enhanced Positioning: Wi-Fi and Motion Data

In the results presented before, the FastGraph was evaluated when using only Wi-Fi data from the moving device.

The next experiments evaluate the FastGraph in applications where, in addition to Wi-Fi, it is also possible to obtain displacement and direction information. The orientation and displacement data allow the Graph algorithm to implement dead reckoning techniques, in order to establish additional constrains.

There is a wide range of positioning applications where displacement and direction information are available. In smartphones, this information can be obtained directly from the smartphone built in sensors, such as accelerometer, gyroscope and magnetometer. With sensor fusion techniques the measurements from these sensors can be improved. In industrial scenarios, IMUs and Encoders installed in robots and machines can provide accurate measurements regarding the moving displacement and direction.

Therefore, these experiments evaluate the FastGraph positioning capabilities when combining the Wi-Fi with orientation and displacement data.

Each MTU position estimation is compared to the ground truth information, resulting in a more precise evaluation of the positioning error. The positioning errors are presented for real time and at the final of the process.

2.4.1 Real Time Positioning Error

Figure 6.21 shows the error evolution in real time, right after each sample being processed. This plot includes the first path, which works as warm up or learning for the algorithm. After the warm up path, three additional paths are followed, with different areas visited and with different traveling distances. As explained before, between each path the MTU was stopped, and after some seconds the data collection process was terminated, simulating in this case an autonomous machine turning off and on. In Figure 6.21, are also represented a few samples collected while the MTU was stopped, before and after the shutdown between paths.

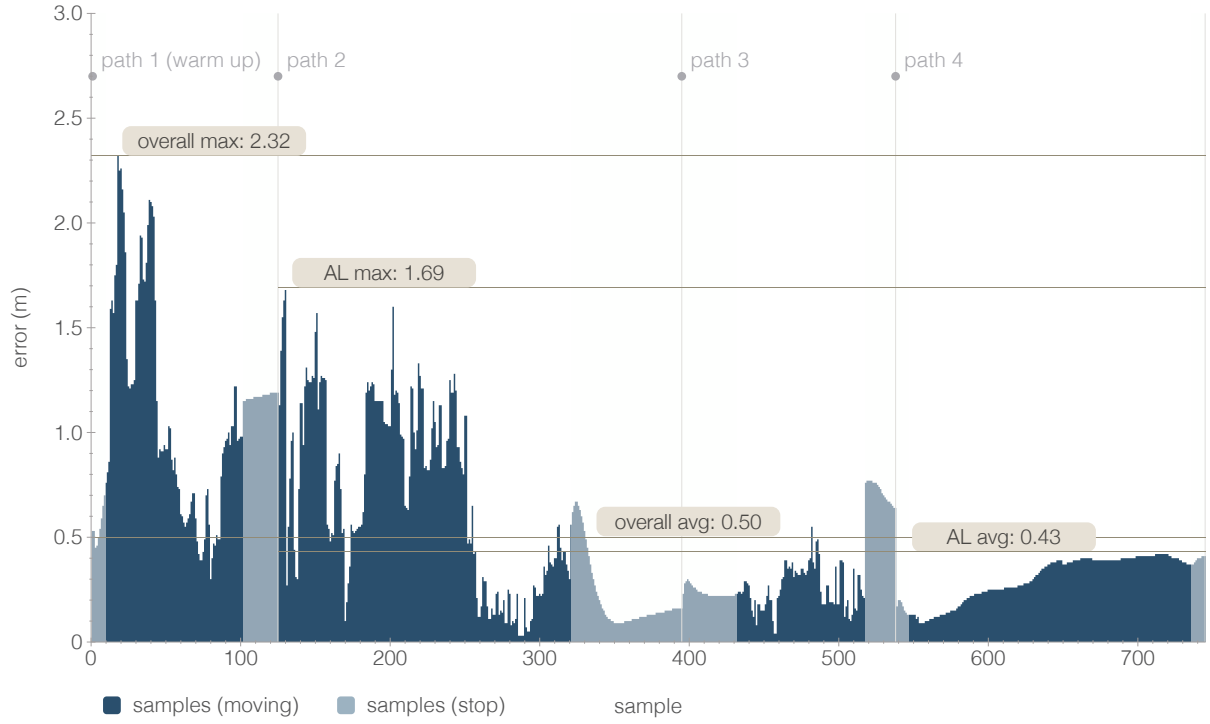

Figure 6.21: Real Time Position Error Evolution

The positioning error shows a drop tendency as the device moves and collects more samples, with the higher error registered in the first path during the warm up period. The average error in the warm up path is also not significantly high, but the max errors are much lower after the warm up showing a drop tendency. Moreover, as more samples are processed less peaks occur, and with lower magnitude. This is also expected, since with more information, more spatial constrains are created, leading to better initial estimations. This is the expected behavior of the FastGraph algorithm, where the Graph becomes more consistent with more information

added, therefore these results support the algorithm convergence principle.

Figure 6.22 shows the MTU sample nodes at the estimated positions in real time, with the node color representing the real time error. This figure shows that the nodes most affected by error in real time belong to the warm path, and that the following paths have lower real time error.

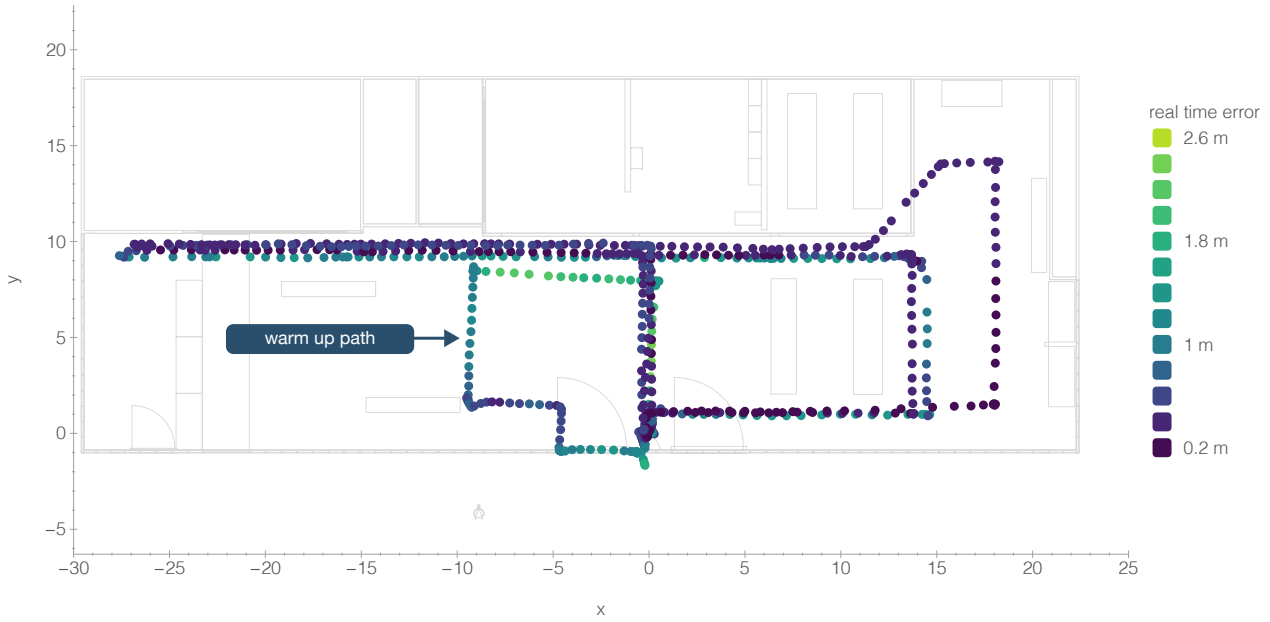

Figure 6.22: Real Time Error : Paths Plot

Figure 6.23 shows the CDF of the positioning error. The top plot shows the CDF for all paths, including the warm up path. The bottom plot shows the CDF after the warm up path. The green line is the real time positioning error and the blue line the final error.

Considering also the warm up path in the error statistics, the average error in real time was 0.53 meters, with 0.76 meters for the third quartile.

The error of the samples at the end of the process (after all samples processed) is lower, with 0.54 meters for the third quartile and 0.48 meters of average. In this case, the average error is lower at the final of the process than it is in real time. This results from the samples with higher error from the warm up path, which have their position estimation improved with the Graph evolution, showing a lower error at the end.

When the warm up path is excluded from the error statistics, the result can be considered as the system performance after warm up. In this case the average error was 0.43 meters with 0.49 meters for the third quartile. The final average error was the same (0.43 meters) and the third quartile error was 0.46 meters.

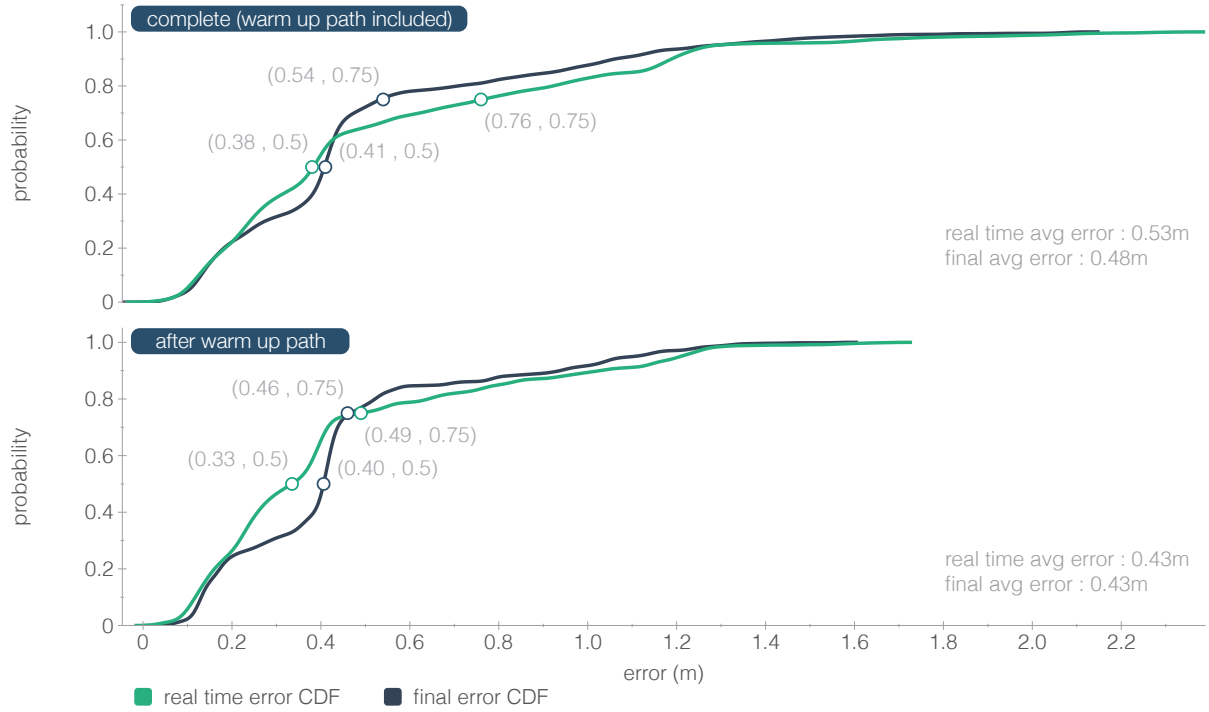

Figure 6.23: Real Time Positioning Error CDF

Table 6.8 shows the results obtained when running the experiment ten times. The real time average position error obtained, after warm up, was 0.506 meters (see Table 6.8) with a max overall error of 2.025 meters. This results also show the algorithm positioning estimation consistency. Although the variations are small, at this level of accuracy are more noticeable.

Run	Average Error (m)	Max Error (m)
1	0.427	1.688
2	0.571	2.025
3	0.565	2.017
4	0.556	2.019
5	0.426	1.697
6	0.556	2.020
7	0.552	2.011
8	0.556	2.020
9	0.425	1.697
10	0.425	1.694
Average	**0.506**	**1.889**

Table 6.8: Experiment Repetition

Figure 6.24 shows the Histograms of the real time and final positioning error in X and Y. As can be observed, the error in X and Y in real time are more distributed, which means that the nodes of one or more paths were shifted from the original position. This is mainly related to the nodes of the warm up path. As can also be observed, in the final of the process, when the

samples from the warm up path were already improved, the histograms become more centered in zero, reducing the shift.

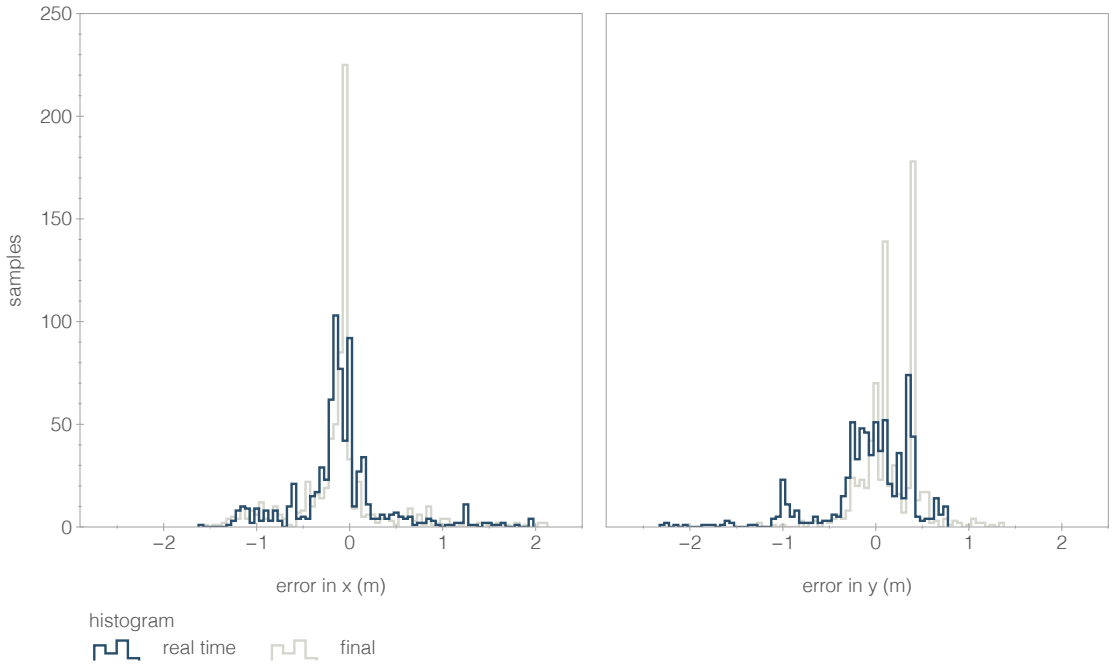

Figure 6.24: Position Error Histograms X and Y

2.4.2 APs' Position Error After Samples with Motion Data

When processing samples with orientation and displacement information from moving devices, the Motion-Edges between the sample nodes create additional constrains, which are also used to improve the APs' positions. Table 6.9 and Figure 6.25 show the error in the APs' positions after processing the samples containing motion information from the previous experiment.

AP	XY Error	Z Error	XYZ Error
AP1	1.16	1.45	1.85
AP2	2.58	1.45	2.96
AP3	2.26	0.97	2.46
AP4	1.84	1.45	2.34
AP5	3.10	1.74	3.55
AP6	4.67	2.96	5.53
AP7	2.62	2.91	3.92
AP8	5.51	0.55	5.54
AP9	10.19	1.24	10.26
AP10	4.70	0.14	4.71
AP11	10.08	0.74	10.11
Average	**4.43**	**1.42**	**4.84**

Table 6.9: APs Position Error in meters (After Sample with Motion Data)

As expected, the average error in the APs' position estimation is lower after the samples with motion data (4.84 meters), when compared to the APs' error based only on the Anchors (6.73 meters).

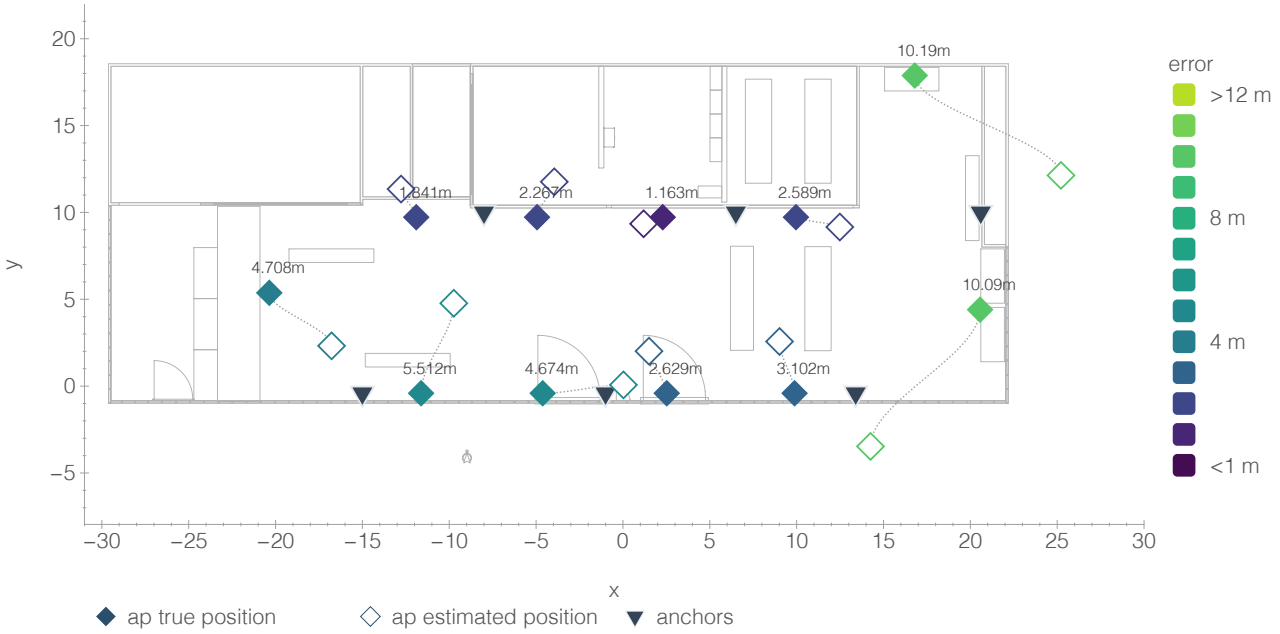

Figure 6.25: APs Final Position After MTU samples

2.5 Progressive η learning

The complexity in estimating the Path Loss Exponent (η) lead to a progressive learning strategy to improve the initial estimations, which was explained in Chapter 4.

In order to evaluate the error in the estimation of each edge path loss exponent (η), it is necessary to known the correct value of η for that specific edge. However, the exact value is not easy to obtain since as explained before, the path loss exponent is dynamic and can be affected by several effects.

Despite of that, in the experimental setup, the correct positions of the APs and the correct positions of the samples is known. Therefore, this information can be use to obtain a fair approximation of the correct η value for each edge. Therefore, to enable an evaluation, the correct η values were calculated using the LDPL model, with the correct distance between an AP and a sampling point, as well as the RSS measurement as input.

Figure 6.26 shows, for three APs at PIEP, η estimations evolve with samples being processed. The PL error is the average error of the path loss exponent (η) estimation for all edges connected to the AP. The three APs show a significant η error drop in the beginning, then become more stable.

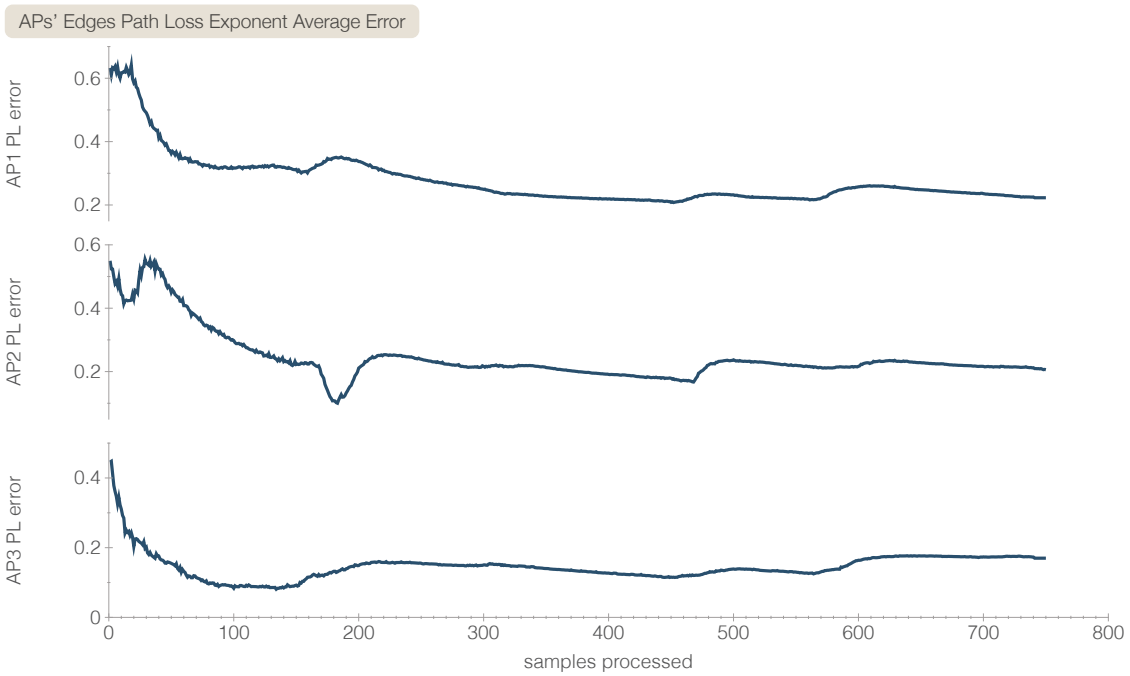

Figure 6.26: Progressive Path Loss (PL or η) learning

It is important to note that, with more samples, the number of edges for which the η has to be adjusted increases. This means that in addition to improve the η values of the existing edges, the algorithm has also to estimate new η values for the edges of each new sample. This is reflected in the variations that can be observed along the process. The variations are also related to the new edges that are created with the default η value, which can be further way or closer from the correct value. Nevertheless, the algorithm is able to maintain a low average error for the path loss exponents, without requiring computational or time demanding algorithms.

In this experiment, for 12630 edges of the Graph, the algorithm was able to estimate the value of the η with an average error of 0.337. This value is low, but given the logarithmic nature of the LDPL model, can represent a different error in an edge length, depending on the RSS magnitude and the η magnitude. Therefore, edges with lower RSS values or lower η values will be more affect by errors in the estimation of the η parameter. However, this is minimized by the way that the elastic constants are applied depending on the natural length of the edge, as was explained before.

The difficulty in estimating the path loss exponent in indoor environments is well known and was clear during this research. In contrast to many other approach, the FastGraph estimates a different path loss exponent for a specific communication channel (edge), between an AP and a sampling position. This strategy, although correct, increases the complexity of the problem even more. Considering this, the obtain results are interesting, but this topic certainly needs to be further explored.

2.6 Using Different Number of Anchors

The FastGraph algorithm relies on reference samples, normally given by the Anchors, to estimate the position of the APs, which are then used to estimate the position of a target device. Therefore, it is relevant to understand how the number of Anchors impact the positioning performance of the algorithm.

At PIEP, FastGraph was evaluated operating with a different number of Anchors. Figure 6.27 shows that when using four Anchors instead of six, the impact in the positioning performance is limited, and more noticeable on the average error, increasing around 0.36 meters[1]. When less Anchors are used the positioning error increases significantly. Considering the number of Anchors for the space dimensions, these results are acceptable and expected.

It is understandable that a space with 1000 square meters can't be correctly covered by only three Anchors. In fact three is the lowest number of Anchors to be considered, which is the standard requirement in the trilateration principle. As the results show, with only two Anchors the error is incomparable higher.

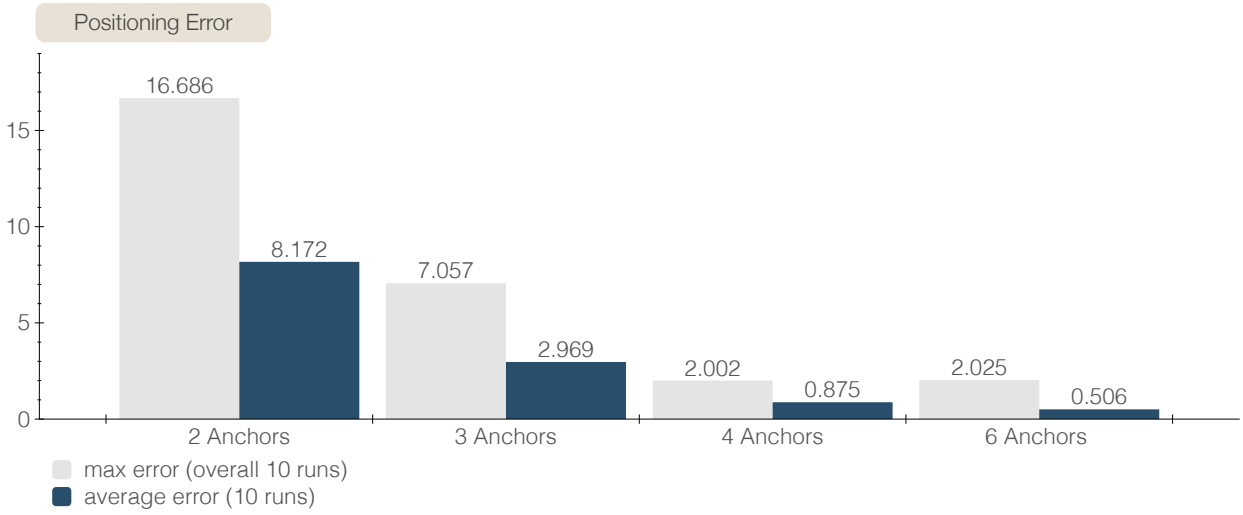

Figure 6.27: Real time Positioning Error with different number of Anchors

The estimation of the APs' positions is also affected by using a lower density of Anchors (Figure 6.28). However the impact is more noticeable in the max error than in the average error, which is a positive result. Therefore, the increase in the real time positioning error, when using less Anchors, is probably related with some of the APs with higher error.

In these experiments only eleven APs were used, therefore each AP is expected to have more impact in the positioning performance. With more APs the device's position estimation

[1]The average is error is the average of ten runs for each Anchor configuration. The max error is the overall max of all runs.

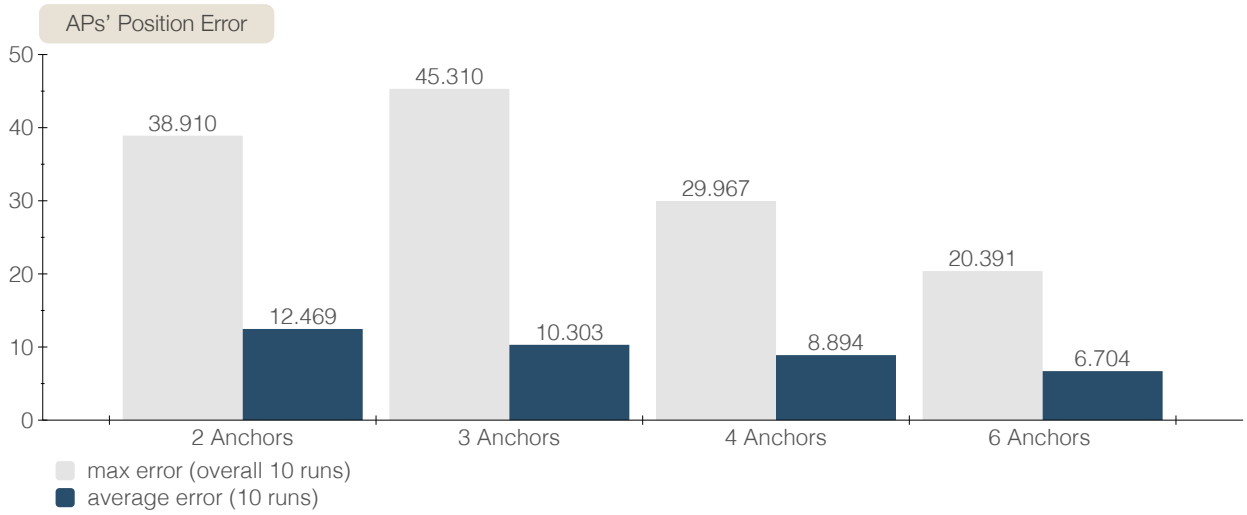

Figure 6.28: APs' position Error with different number of Anchors

will be probably less affected by individual APs with higher error.

With more Anchors the positioning error may improve. However, to further evaluate this topic, a more accurate method to obtain the ground truth is required. At this level of accuracy, the benefits from using more Anchors may not be reflected on the positioning performance, due to small errors in the ground truth data.

These results also provide an idea about the robustness of FastGraph, showing that the solution can operate with acceptable positioning performance if an Anchors fails, giving time to solve the problem without significant impact in the positioning service. In addition, at DSI-DEP, where a lower density of Anchors was used, FastGraph was able to achieve similar positioning results in WI-Fi only mode, suggesting that the system can provide interesting performance with a low density of Anchors.

2.7 Position Estimation Time

As explained before, some positioning solutions proposed by other authors can have demanding computational requirements or high processing times. However, in most scenarios, a fast position estimation is required, otherwise that position is no longer valid or useful. The computational performance of FastGraph is maintained by a set of scalability mechanisms described in Chapter 4.

The time required by the FastGraph to obtain a position estimation was evaluated. The results consider the time from the beginning of processing of a sample, until a position is estimated. To take into account the max possible processing time, only samples with orientation and displacement information, collected at PIEP were considered.

Figure 6.29 shows the processing time by sample, after being already processed around

3000 samples (warm up period). An average time of 37 ms and a maximum of 280 ms were observed. The processing time has variations along the process, this is related to each sample having reference to a different number of APs, and the RSS measurements that can be affected by different levels of noise. In these cases the adjusting process can take more time, but as the results show the max processing time is still low.

In this experiment, the average positioning error for all samples was around 0.43 meters. The results were obtained running the FastGraph solution in a Core i5-2400 at 3.10 GHz desktop machine, with 4 GB of RAM and a 64-bit operating system. The machine specifications are fairly low. In a more powerful machine, and with multithreading optimizations, the processing time is expected to be lower. Moreover, as explained before, the Graph pruning process can be configured to keep a maximum number of nodes on the Graph, or to maintain a specific processing time. This feature can be used to adapt the solution to different scenarios, adapting for example to specific positioning time requirements. Moreover, mechanisms to filter and remove APs from the Graph can be implemented. In specific scenarios, where for example the FastGraph is deployed for a single building, the APs at nearby buildings can be removed from the Graph. The building area can be defined by the Anchors, and the algorithm can identify the APs with positions estimated outside the building. This can improve both the computational and positioning performance, as previous works suggest [35, 63].

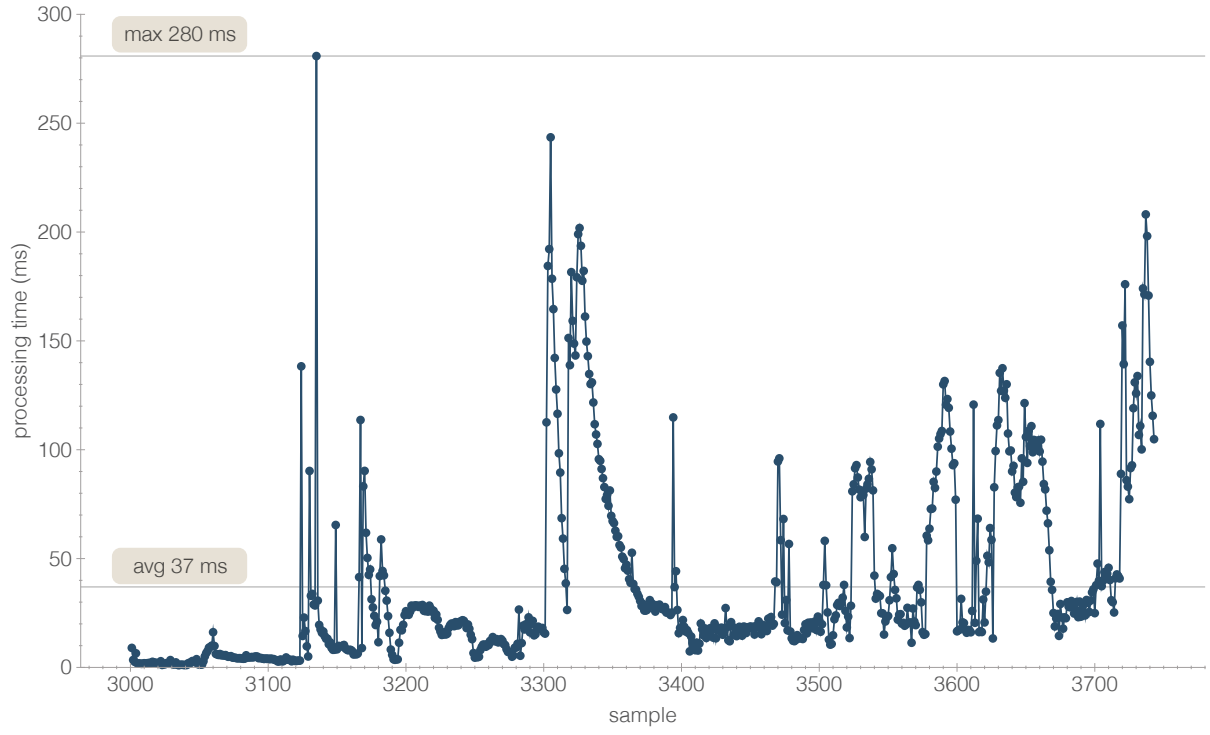

Figure 6.29: Processing Time by Sample

3 The Radio Maps

The radio maps created by FastGraph are expected to be useful in different contexts and applications. The accuracy of the resulting radio map will be dependent on the data used by the FastGraph. The radio maps obtained when processing samples with orientation and displacement information will have higher accuracy, when compared to the radio maps obtained in Wi-Fi only applications.

However, each type of radio map has its own benefits. The radio maps based only on Wi-Fi are easier to obtain, an can be obtained in any space, independent of the space dimensions or the moving devices used. The radio maps created with Wi-Fi only, even though less accurate, may provide interesting results when used in Fingerprinting, since most of the Fingerprinting solutions estimate a position based not only on a single fingerprint, but in a set of fingerprints, using for example weighted methods.

The radio maps created based on Wi-Fi combined with motion data can be used in applications where more accurate radio maps are required. Both types of these radio maps can be created using smartphones as moving devices, not being required special hardware. However, with smartphones, there are some practical challenges to address, including the inertial sensors' noise, the orientation drift, or the step detection. It is out of the scope of this research to further address this topic, but this is a well known and frequently studied subject. Several solutions were already proposed and even integrated in the mobile operating systems. Techniques such as sensor fusion are already integrated on the mobile OSs, such as the Android, in order to provide accurate orientation data and more accurate step counting.

FastGraph' radio maps are not common radio maps, such the ones created for fingerprinting. First of all, these radio maps are automatically created, which is their main advantage. Secondly, the additional information on these radio maps expand the possible application scenarios. Figure 6.30 shows a Graph representation of an example 3D radio map, that was obtained with FastGraph for PIEP. In addition to the traditional position referenced fingerprints, these radio maps have information about the APs positions, and the propagation characteristics of the communication channel (edges), between an AP and a fingerprint's position.

The additional data in these radio maps can be used in different fields, such as for advanced interference mapping and analysis, or for automatic physical space mapping. These two possible applications will be discussed, and an initial validation will be presented, in the next chapter.

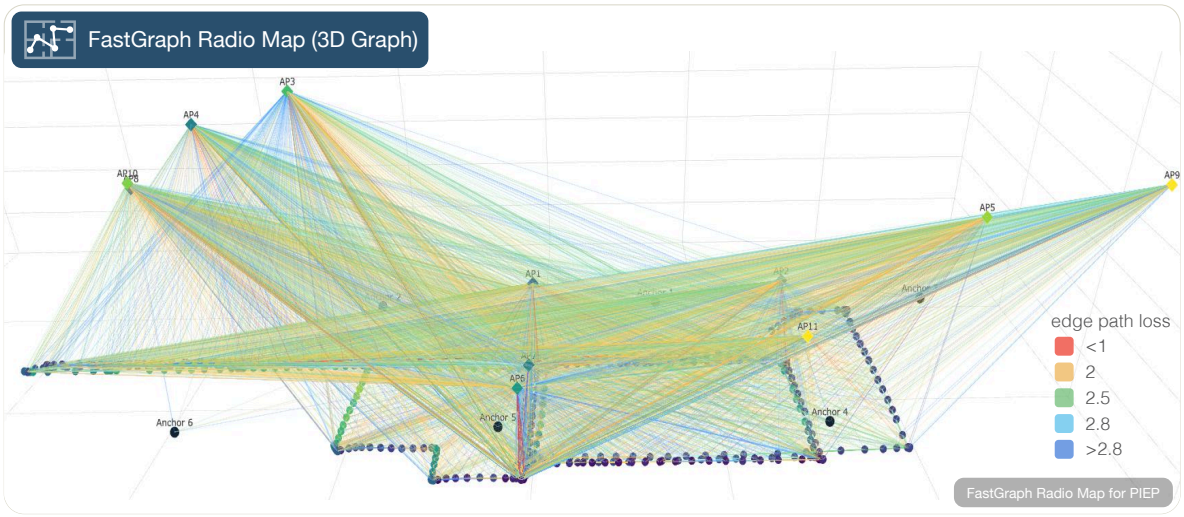

Figure 6.30: Example of a FastGraph's 3D Radio Map for PIEP

4 Comparing with Other Positioning Solutions

The FastGraph positioning results were compared with the results obtained in two Tracks of the 2017 IPIN competition [1]:

- Track 1: Smartphone-based (on-site)
- Track 3: Smartphone-based (off-site)

The IPIN competition has been attracting the attention of many teams working on indoor positioning solutions from around the world. In the 2018 edition, which was recently held in Nates, France, the number of competing teams increased including also teams from large companies, including teams from Google and Sony[2]. Since the results from the 2018 IPIN competition were unavailable at the time of this thesis writing, the results from the 2017 edition were considered.

The objective of Track 1 is to evaluate the competing system on-site, and in real-time. The competing teams are not allowed to install any instrumentation in the competition area. The FastGraph uses the Anchors to collect reference samples. This data is essential to initialize the Graph, but not required in the real-time positioning phase. Since the competing teams have at least a full day before the competition to survey the area, obtaining the reference samples would not be a problem. The TPU device could be used to collect the necessary data to initialize the Graph, instead of using Anchors.

[2]http://ipin2018.ifsttar.fr/competition/competition-teams/

In this track, competing teams are given the coordinates of a starting point, from where they must start walking over a reference path. Any sensor available in the smartphone can be used to track the user trajectory. In the experiments with FastGraph, reported in the previous sections, the starting point was unknown to the algorithm.

In Track 3 the teams compete off-site, being provided all the data for calibration, which includes Wi-Fi and data from all sensors available in a smartphone. In addition, the teams can calibrate their solutions with several ground-truth databases.

Table 6.10 compares the results obtained for FastGraph with the results obtained in these two tracks.

Track 1 (on-site)	**Data**	**Mean (m)**	**3/4 (m)**
SNU-NESL PDR Team	Wi-Fi + Sensors	6.2	8.8
MCL Team	Wi-Fi + Sensors	12.6	16.8
XMC Team	Wi-Fi + Sensors	23.0	30.8
Track 3 (off-site)	**Data**	**Mean (m)**	**3/4 (m)**
UMinho Team	Wi-Fi + Sensors	3.00	3.48
AraraDS Team	Wi-Fi + Sensors	3.74	3.53
Yai Team	Wi-Fi + Sensors	3.51	4.41
HFTS Team	Wi-Fi + Sensors	3.52	4.45
FastGraph (DSI-DEP)	Wi-Fi	**5.08**	**6.21**
FastGraph (PIEP)	Wi-Fi	**5.65**	**6.91**
FastGraph (PIEP)	Wi-Fi + Sensors	**0.50**	**0.55**

Table 6.10: Comparing with IPIN'17 Track 1 and Track3 [1]

When using only Wi-Fi, FastGraph performs worse than Track 3 (off-site) and better than the Track 1 (on-site), however in both tracks, additional sensors are used and previous calibration is allowed. The results of FastGraph when using also data from the additional sensors are significantly better. The comparison of results when using sensors may be limited by differences in the sensors quality.

Moreover, it is important to note that FastGraph' results, when using only Wi-Fi, are consistent in two completely distinct environments, despite different layout, dimensions, build materials, propagation characteristics and AP density.

All aspects considered, these are certainly interesting and encouraging results, even when using only Wi-Fi, since FastGraph is full unsupervised and automatic, and is compared with state of the art solutions.

5 Summary and Discussion

In this chapter were presented the results obtained when deploying the FastGraph in two real word spaces. The experiments evaluated the FastGraph when operating only with Wi-Fi data, and when Wi-Fi data was combined with orientation and motion data.

The DSI-DEP, with office-type building characteristics, provided the ideal environment to deploy and test the FastGraph in a Wi-Fi only scenario.

The PIEP's industrial characteristics provided the ideal conditions to test the enhanced FastGraph which combines Wi-Fi with motion data. The controlled experimental setup with the ground truth grid, allowed to test the FastGraph in higher accuracy positioning applications, such as autonomous machines indoor navigation, where orientation and displacement data can be combined with Wi-Fi. The developed Moving Testing Unit (MTU) was used to simulate an autonomous machine.

The results showed that the FastGraph initialization phase, where the reference samples from the Anchors are used to locate the APs and start the Graph, takes less than 10 minutes. After this the system is ready to provide positioning.

The solution showed positioning results consistent in two very distinct spaces, with around 5 meters of average error in Wi-Fi only mode. When combining Wi-Fi with orientation and displacement data the positioning performance changes to a completely different level of accuracy, with an average error around 0.5 meters. These positioning results compare well against the results obtained by state of the art solutions, such as those in the IPIN'17 competition.

FastGraph was able to estimate the 3D position of the APs with interesting results in both spaces, and the results also showed the Graph convergence principle, since with more samples the APs' positions becomes more stable, and the estimations are improved. Moreover, with motion enable samples, the solution is able to significantly improve the estimated APs' positions. The results at DSI-DEP also suggest that the algorithm may even operate without Anchors in all floors.

The computational performance was also evaluated, specifically regarding the time necessary to obtain a position estimation. The results showed that the algorithm takes around 37 ms (with a max of 280 ms) to process a sample and estimate a position. These results were obtained in a low grade machine and without optimizing the algorithm for multithreading or to improved processing times.

The results using different number of Anchors are within what was expected, since a low

density of Anchors is already used. However, these results are interesting as indicator of the expected positioning performance in relation to the Anchors density. In specific applications a low density of Anchors may be enough to provide the necessary positioning performance.

Chapter 7

FastGraph: Extended Features

In the previous chapter were presented the results of several experiments where the FastGraph was deployed and evaluated in two different spaces.

This chapter describes how the FastGraph approach can be used and extended to other applications, especially for additional automatic mapping features.

The first additional application proposed for the Radio Maps created by FastGraph is their use as interference maps for analysis of the radio environment. The interference analysis process normally requires a manual site survey, similar to the site survey required by Fingerprinting, which, as previously discussed, in large buildings is unpractical.

The proposed solution can automatically provide interference maps based on two distinct strategies: AP-Based Interference Maps or Sample-Based Interference Maps, each one with its own characteristics.

Another proposed application for the FastGraph' Radio Maps is to automatically create a map of the physical space. This can be accomplished by detecting the obstacles positions using the path loss values of the several communication channels (edges) between an AP and a sample position. By combining the information of the attenuation introduced in multiple RF signals at multiple locations, it may be possible to automatically obtain the floor plant or the location of the obstacles, in an approach somewhat similar to tomography.

An initial exploration of these two applications is presented in this chapter. The results suggest that the FastGraph may be a versatile solution that can be in fact used in different contexts.

The first section describes the Automatic RF Interference Mapping (ASIR-Mapping) feature, where the metric used to measure the interference is described. Moreover, the two methods to create the interference maps are explained. Examples for real world spaces are provided.

Section two presents the Automatic Space Mapping solution, describing the approach and presenting results based on synthetic data from the virtual space simulator.

1 Automatic RF Interference Mapping

The interference severely affects the communication quality in Wi-Fi networks. Access Points operating in channels with overlapping frequencies interfere with each other, and due to the uncoordinated deployment of Wi-Fi networks this problem is frequent. Tools to survey the radio environment are useful to analyse how the frequencies are being used by the Access Points, and to define strategies to reduce the interference. The available tools to perform this type of analysis require manual survey of the space, by manually collecting data at several positions, in a process similar to create a radio map for fingerprinting. The quality of the interference map increases with more positions surveyed. The interference affecting the signals of a specific AP, at a specific position, can be described by the Signal-to-Interference Ratio (SIR).

1.1 Signal-to-Interference Ratio (SIR)

The interference can be due to Co-Channel Interference (CCI) and Adjacent Channel Interference. Channels operating in the same frequency are denominated by Co-Channels, and lead to Co-Channel Interference (CCI). Adjacent Channel interference is related to channels operating in partial overlapping frequencies (Figure 7.1).

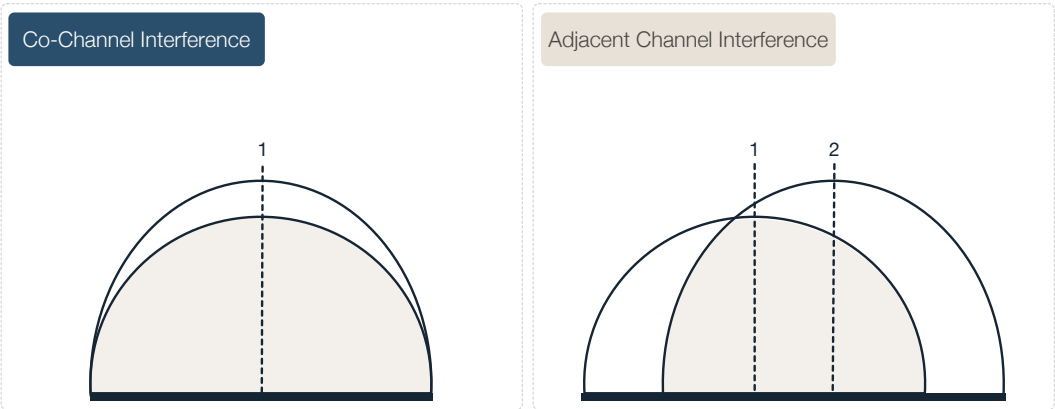

Figure 7.1: Channel Interference

The Signal-to-Noise Ratio (SNR) is the ratio of the power of a signal, to the power of the background noise:

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{signal}(w)}{P_{noise}(w)} \right) \tag{7.1}$$

A SNR higher than 0 dB means that the power of the signal is higher than the power of the noise. Therefore, the higher the values of SNR ratio, the better the system performance in presence of noise. However, increasing the SNR don't solve the Co-Channel Interference, and

in fact can make it worse. To reduce the CCI it is necessary to increasing the distance between Co-transmitters using the same channel (frequency band).

Channels that are adjacent in frequency are supposed to not interfere with each other, but in practice, adjacent channels may have sidebands that interfere.

The interference is therefore related to the power and frequencies of the signals in the environment. In Wi-Fi, the frequency is related to the channel in which each Access Point is operating.

The Signal-to-Noise-Plus-Interference Ratio (SNIR) [91, 92] considering two APs can be defined as follows:

$$SINR = \frac{P_i}{P_j(1-\alpha) + N} \tag{7.2}$$

where P_i is the received power of the main AP, P_j is the interfering power received from the adjacent AP_j, N is the power of background noise or noise floor, α is a constant that accounts for interference between adjacent channels. The noise floor term can be seen as constant, and when considered as zero, the Signal-to-Noise-Plus-Interference Ratio (SNIR) can be reduced to Signal-to-Interference Ratio (SIR), in the same way as when the interference is zero we have the Signal-to-Noise Ratio (SNR). For the interference mapping, the noise floor is not relevant and therefore we can use the SIR as a metric of performance.

To calculate the total interference affecting an AP_i, in a specific location, all the signals received from the n other APs in range are considered as possible interference. Therefore the interference (I_i) affecting the AP_i is the sum of the interference from each in range AP_j:

$$I_i = \sum_{j=1}^{n} P_j(w) \times CO_{ij} \tag{7.3}$$

where, CO_{ij} is the level of overlapping between the channel used by AP_i and that of AP_j, that can range between 0 (when there is no overlapping) and 1 (for Co-Channels). P_j is the power (in watts) of the signal of the interfering AP, and can be obtained from the received power (in dBm) by:

$$P(w) = 10^{-3} \times 10^{\left(\frac{P(dBm)}{10}\right)} \tag{7.4}$$

Then the Signal-to-Interference Ratio in dB, for an AP_i in a given location, can be given by:

$$SIR_i(dB) = 10 \log_{10} \left(\frac{P_i(w)}{\sum_{j=1}^{j=n} P_j(w) \times CO_{ij}} \right) \tag{7.5}$$

This metric relates the level of power from the interfering signal with the level of overlapping with the channel of the target AP. This means that larger overlapping and higher signal power from the interfering signal results in higher contribution to the overall interference. The higher the SIR_{dB}, the lower the impact of the other APs in the target AP.

1.2 Available Site Survey Tools

There are several site survey tools available, both free and commercial, that can be used to perform interference analysis.

TamoGraph[1] is an application to perform Wi-Fi site surveys. They suggest that a site survey should be performed as:

- **Pre-deployment survey:** In order to test the network plan in the real-world environment, using temporary APs to fine-tune the APs and antennae placement, to avoid poor coverage zones.

- **Post-deployment survey:** To verify if the WLAN performance and coverage meet the design requirements after deployment.

- **Regular, ongoing surveys:** Regular surveys are necessary to maintain high performance and coverage, due to several factors such as new users, site expansion and neighboring WLANs.

TamoGraph can perform passive surveys, where the application collects data on the RF environment without connecting to WLANs. In active surveys, a connection to the chosen wireless network is done to measure actual throughput rates and a few other metrics. TamoGraph also provide predictive modeling which is performed off-site, and is based on computer simulations.

Regarding to interference analysis, TamoGraph suggest that one AP should be selected at a time for SIR analysis in order to obtain a better representation. This way the AP-specific problem zones can be isolated by selecting APs one by one. Analysis based on a cumulative representation, displayed when multiple APs are selected, is more difficult. In their website TamoGraph suggest a set of solutions to deal with interference such as, change the channel selection and work with the classical “honeycomb” AP placement, as well as using non overlapping channels.

[1] https://www.tamos.com/products/wifi-site-survey/

NetSpot[2] is another commercial tool for Wi-Fi site surveying, with features similar to TamoGraph, allowing to perform interference analysis. They also suggest some strategies to minimize co-channel interference, such as adjusting the AP transmitting capacity. They also suggest that poor SIR levels are considered to be below 30 dB and critical levels are below 10 dB[3].

Ekahau[4] has another commercial survey tool which can be used with Ekahau Sidekick (Figure 7.2), which is defined by Ekahau as *All-in-One Wi-Fi Site Diagnostics and Measurement Device*, and described as a professional measurement device for Wi-Fi site surveys and spectrum analysis. Ekahau Sidekick allows simultaneous active and passive site surveys, with dual-band Wi-Fi adapters, in order to minimize the site survey and troubleshooting time.

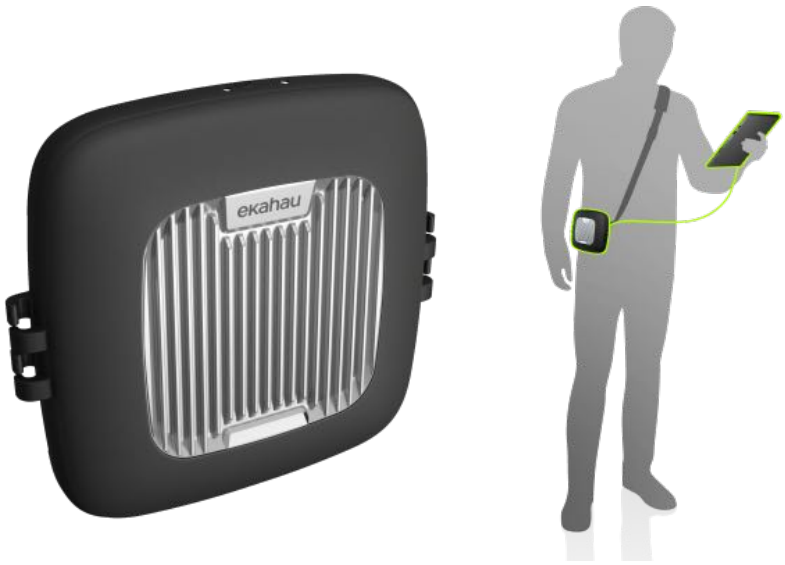

Figure 7.2: Ekahau Sidekick[5]

The available commercial Wi-Fi survey applications already pack the necessary tools to analyse Wi-Fi environments. However, the survey process can take a long time, since the areas of the space must be manually surveyed, requiring collecting data at several locations. This is a problem in large spaces, leading to solutions such as the Ekahau Sidekick, to improve the required manual survey. In addition, as referred by TamoGraph, regular surveys are essential to maintain high performance and coverage. For these reasons, an automatic survey tool, able to provide constant analysis of the radio environment can be a very interesting solution.

[2]https://www.netspotapp.com

[3]On the NetSpot webpage the SIR values are in dBm; since SIR is a ratio between two powers, was considered that NetSpot meant dB instead of dBm.

[4]https://www.ekahau.com

[5]https://www.ekahau.com/products/sidekick/overview/

1.3 Automatic Interference Maps

The FastGraph algorithm allows automatic site survey, and to perform the interference analysis based on the Automatic Interference Maps (ASIR-Maps) created using two distinct strategies:

- **Sample-Based Interference Maps:** This method uses the data samples collected by the target devices and the estimated positions from FastGraph, to create the interference map. This strategy uses the actual RSS measurements in each sample to provide accurate and position-based interference analysis.
- **AP-Based Interference Maps:** The APs positions, estimated by FastGraph, are used to easily create an interference map for the entire space by estimating the power levels distributions using the LDPL model. This strategy estimates the RSS measurements at each position.

Interference Maps can be created for:

- **A single target AP:** An AP is selected an the interference map represents how the selected AP is affected by the radio environment is different areas of the space.
- **A target Network:** An network is selected, by the Service Set Identifier (SSID), and the created interference map reflects how the APs with the selected Service Set Identifier (SSID) are affected by the radio environment. For each location of the interference map, the target AP is the AP of the selected network with the strongest RSS.
- **The strongest AP (of any network):** This mode is similar to the Network mode, but in this case there are no target Network. Therefore, when selecting the target AP, based on the strongest RSS, all APs from all available networks are considered.

1.3.1 Sample-Based Interference Mapping

This method uses the actual RSS measurements in the sample nodes, which combined to the position estimate by the FastGraph, allows to create a more accurate representation of the interference at each sample position. Figure 7.3 shows a Sample-Based Interference Map created for PIEP where the different levels of interference can be observed. The level of accuracy required for interference analysis is related to the APs coverage area, which in relation with the samples position error is much higher.

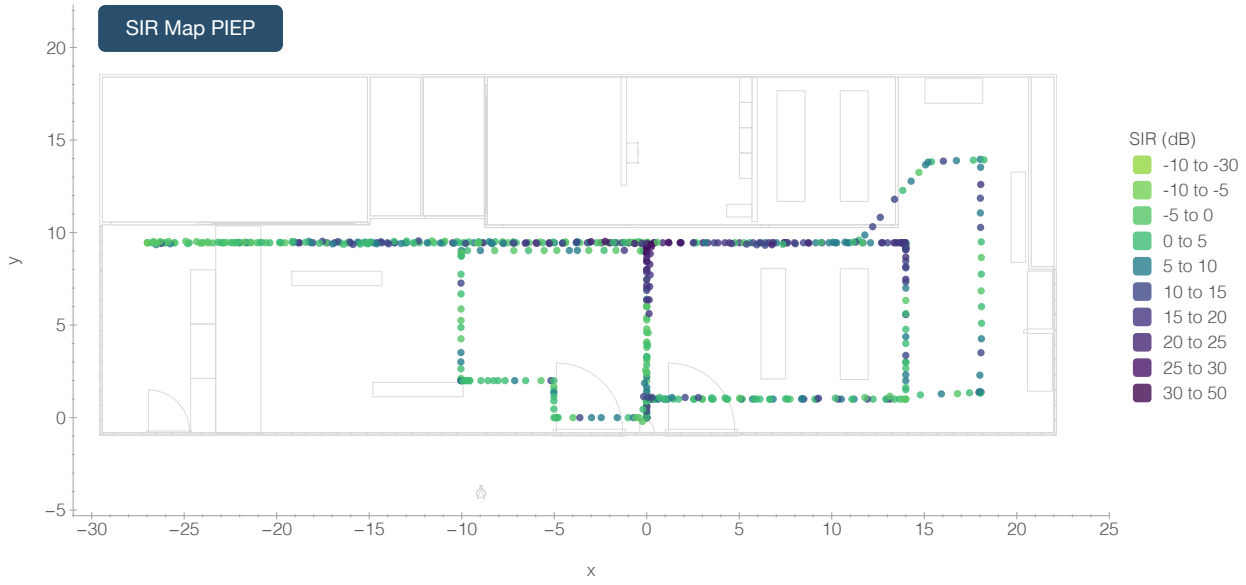

Figure 7.3: Example of Sample-Based SIR Map (PIEP space)

However, with this method, in order to obtain an interference map for the entire space, it is necessary to wait until samples are collected in different areas of the space. The time required will of course depend on the number of users and their motion patterns. Therefore, this method is more suitable to create accurate interference maps for the most important areas of a space.

1.3.2 AP-Based Interference Mapping

The APs positions, automatically estimated by the FastGraph algorithm, can be used to easily compute and maintain an interference map for the entire space. The only requirement is the deployment of the Anchors or to collect reference samples. After provided the space dimensions (X,Y,Z), the system creates a grid of mapping positions for example at every 1 meter (configurable parameter). Then the interference level in each position is estimated by considering the positions of the APs in range, and their channels. The RSS for each AP in a given position is approximated by using the LDPL propagation model (Equation 4.8). For the positions of the Anchors, the interference calculation is based on the actual measured

RSS values. Figure 7.4 shows an AP-Based Interference Map where the positions of the APs estimated by FastGraph were used to automatically estimate an interference map for the entire DSI-DEP space.

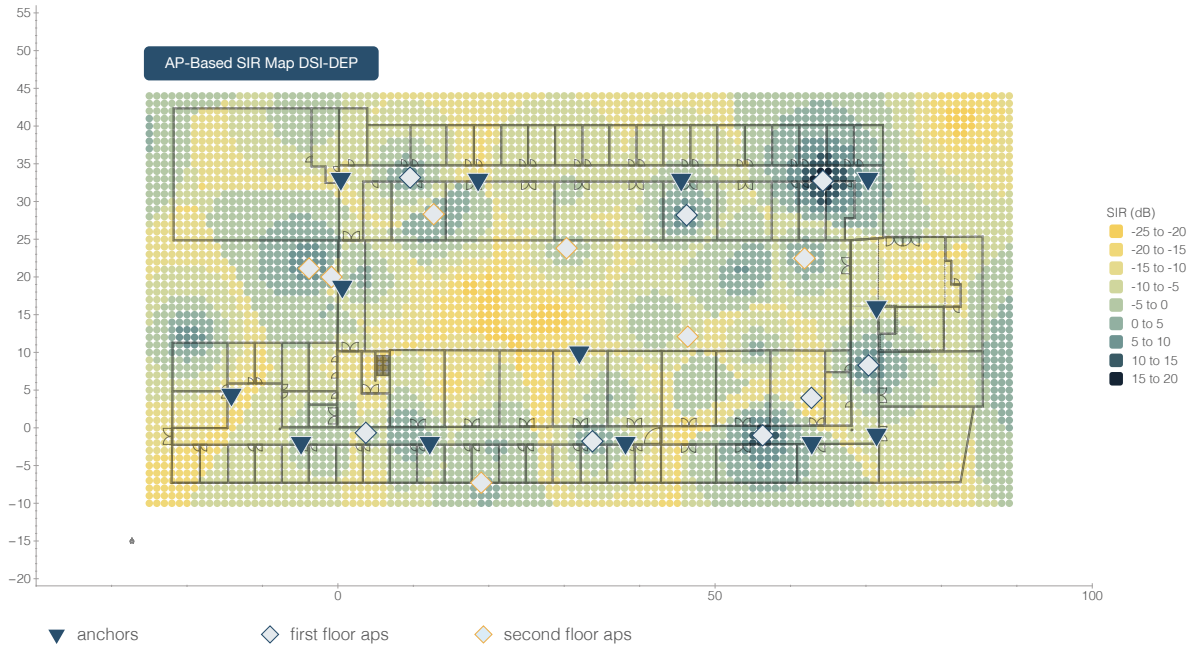

Figure 7.4: Example of AP-Based SIR Map (DSI-DEP space)

The accuracy of these interference maps, based on RSS estimations, is lower than the interference maps created based on sample nodes with actual RSS measurements. However, these interference maps can provide a rapid and easy overview of the radio environment for the entire space. In very large spaces or when an overview of the channels usage and distribution is required, these maps can be more useful.

1.4 Interference versus Positioning Error

During the analysis of the interference maps created for PIEP, a relationship between the interference levels and the positioning error was observed. This relationship is apparently new, and not previously reported by the research community.

As the results presented before revealed (see Section 2.2), the error was higher in areas with low coverage and Non Line of Sight (NLoS) (critical areas). The interference maps show that in the areas with higher position error, such as the critical area, the interference level is also significantly higher. Figure 7.5 shows the error map (Wi-Fi only results) and Figure 7.6 shows the interference map for the same space.

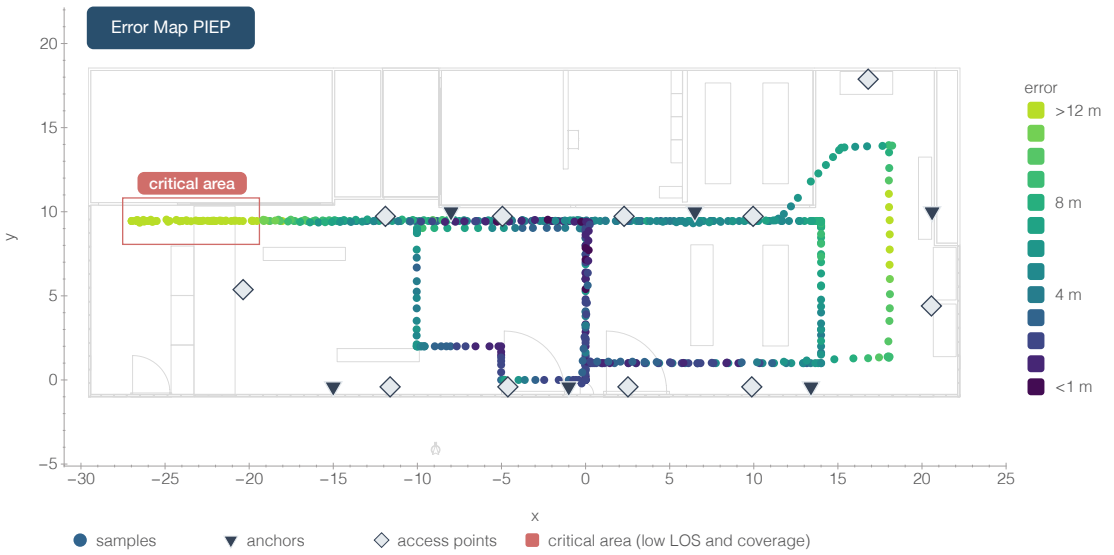

Figure 7.5: Error map (Wi-Fi only)

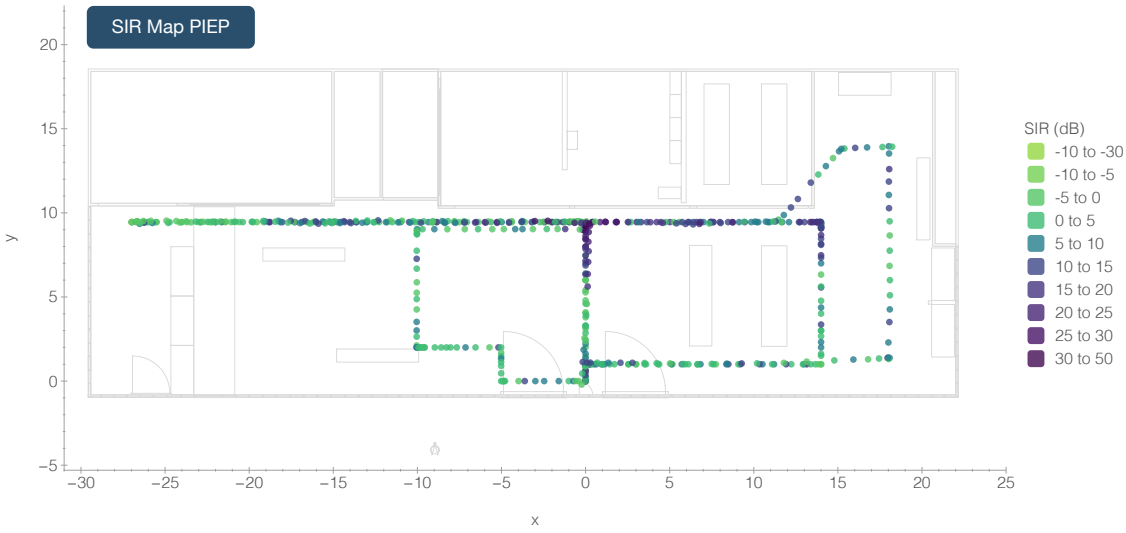

Figure 7.6: SIR map (Wi-Fi only)

It is perceptible that the areas with larger error in Figure 7.5 also have high interference in Figure 7.6. In the critical area (samples with error >12 meters at the left) it is clear a relationship between the high interference (low SIR) and the large positioning error.

Some areas with high interference have low positioning error. This is expected since the FastGraph algorithm is able to compensate in some situations, and obtain a low error even when the radio signals are affected by interference. However, in low interference areas, the positioning error is consistently low. This behavior can be observed in Figures 7.7 and 7.8, where the positioning error is plotted against the SIR for each sample processed by the FastGraph algorithm.

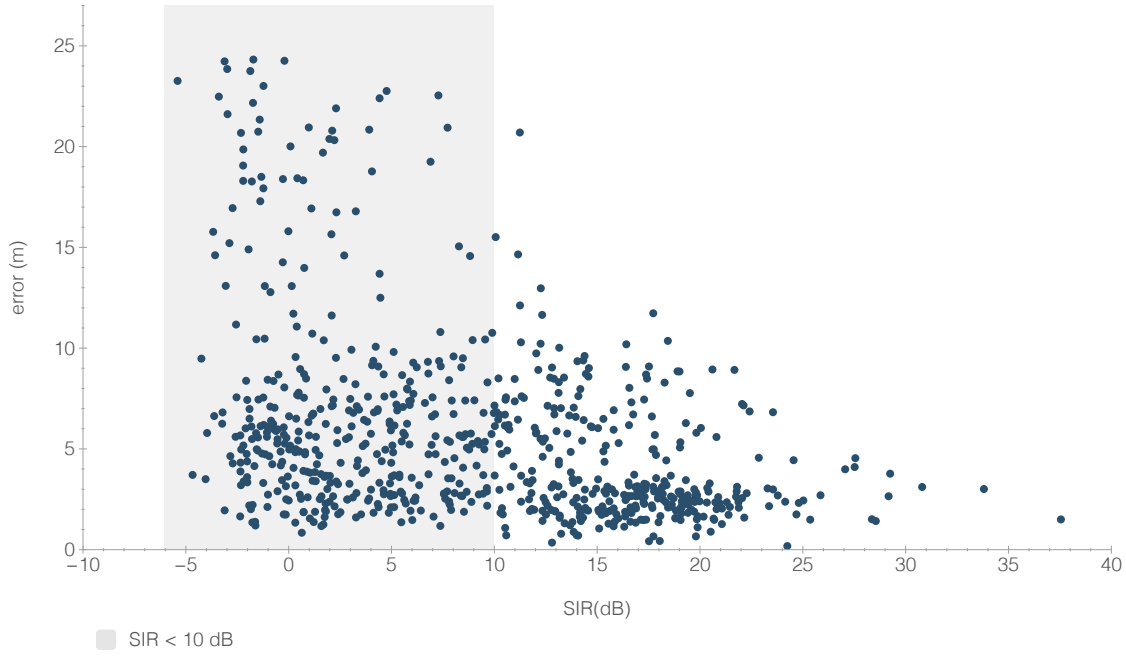

Figure 7.7: Positioning error vs SIR (Wi-Fi Only at PIEP)

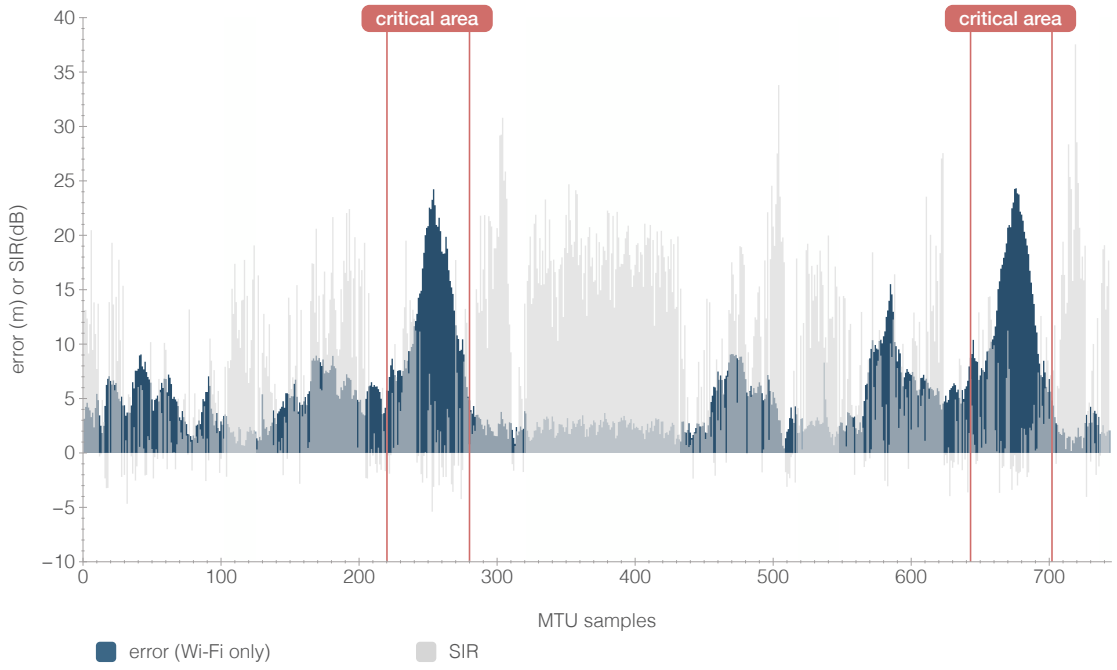

Figure 7.8: Positioning error vs SIR (Wi-Fi Only at PIEP) by Sample

In these plots the SIR and the positioning error relationship is clear, and suggests that lower SIR (hight interference) increases the probability of larger positioning error, and with high SIR (over 25 dB) the probability of large error is very low. Moreover, for values of SIR below 10 dB, the probability of larger error is far superior. This is in agreement with the NetSpot's[6] suggestion that SIR values below 10 dB are considered critical levels. Figure 7.8 also clearly

[6]https://www.netspotapp.com

shows that in the two peaks of error, that resulted from the critical area, the SIR is very low which means high interference.

1.4.1 Improving Positioning with SIR Information

Considering the observed relationship between the SIR and the positioning error, it was hypothesized that a SIR value can be computed for each sample processed by the algorithm and based on the SIR value increase or decrease the elasticity (ke value, see Eq. 4.16) of the RSS edges. With this approach the position of the samples with higher SIR can be more influenced by the RSS edges since in theory this edges are less affected by interference. A method to compute the samples SIR was implemented in the FastGraph solution and the PIEP dataset from the previous experiments was used to compare the positioning results (Table 7.1).

Run	**Avg. (m)**	**SIR Avg. (m)**	**Max (m)**	**SIR Max (m)**
1	0.427	0.537	1.688	1.915
2	0.571	0.540	2.025	1.916
3	0.565	0.440	2.017	1.590
4	0.556	0.440	2.019	1.584
5	0.426	0.439	1.697	1.596
6	0.556	0.504	2.020	1.887
7	0.552	0.537	2.011	1.911
8	0.556	0.542	2.020	1.922
9	0.425	0.441	1.697	1.579
10	0.425	0.439	1.694	1.585
Average	**0.506**	**0.486**	**-**	**-**
Overall Max	**-**	**-**	**2.025**	**1.922**

Table 7.1: Comparing Positioning Error using SIR

The results show a slight improvement in the positioning errors when using the SIR information from each sample to control the influence of the RSS edges. In addition, with the SIR the results between runs are more consistent. These results indicate that this subject needs further research and evaluation, which was not possible in this thesis time frame. With a more advanced approach to use SIR information it might be possible to obtain more improvements in the positioning performance. Moreover, if this relationship can be further proved it may also be useful to provide a confidence or accuracy level with every position estimation.

2 Automatic Space Mapping

This section describes how the FastGraph's Radio Maps can be used to automatically map obstacles.

With further research it may be possible to automatically create the floor plant of a space. For now, the hypothesis and the base approach will be discussed. A simple initial validation will be presented, using synthetic data from the 3D space simulator.

The base idea is to explore the path loss information of each edge. As explained before, an edge represents a unique communication channel between an AP and a sample position. Therefore the path loss exponent of each edge, characterize how the signal power decays with the distance from the AP. The obstacles introduce attenuation to the signal, decreasing the signal power faster. Hence, an edge with high path loss exponent means that the signal was attenuated by obstacles.

Following this idea, using the path loss information of multiple edges, it may be possible to identify the areas that introduce attenuation to the signals, and that areas can be defined as a possible obstacle. Moreover, each material introduces a different level of attenuation, therefore depending on the global path loss factor for that area, it may even be possible to infer the obstacle building material.

This approach is somewhat comparable with tomography, where penetrating waves, such as X-Rays, are used to produce images or 2D slices of 3D objects.

Figure 7.9 shows the edges, samples and APs, one of the Radio Maps that was created with FastGraph. This Radio Map was the result of a synthetic dataset, generated by the virtual space simulator, with a configuration of APs similar to PIEP. A single 3D obstacle[7] was added to the space. This obstacle introduces an attenuation of 6 db to the signals intersecting it. In addition, in order to simulate the noise introduced by other propagation effects, all Wi-Fi signals are affected by noise following a normal distribution with zero mean and standard deviation of 3.

In Figure 7.9 all edges between the samples and the APs can be observed. The standard path loss exponent (η) used in the simulation was 2.5. It is clear in this plot that the edges crossing the obstacle have an higher path loss exponent. To detect obstacles only the edges with high η are considered, therefore the edges with η below 2.8 were filtered out. The remaining edges are shown in Figure 7.10, where it is more clear the obstacle relation with the edges path

[7]Obstacle dimensions in meters: (0.5 width x 3 length x 10 height)

loss.

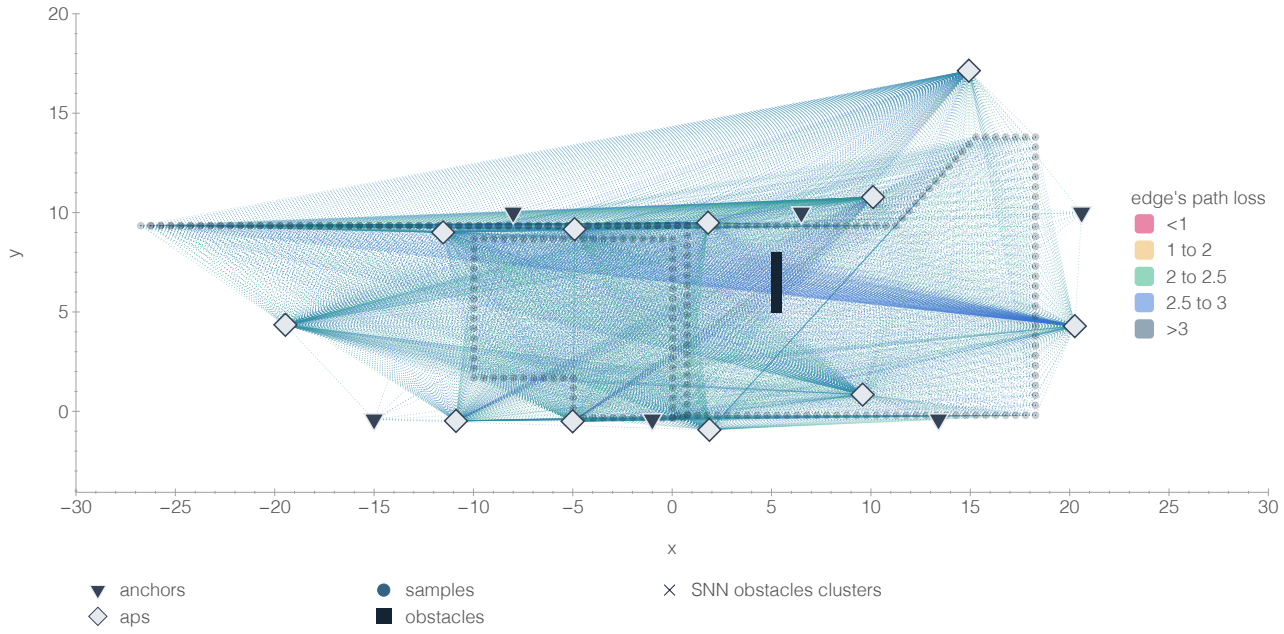

Figure 7.9: FastGraph's Radio Map: Edges Path Loss

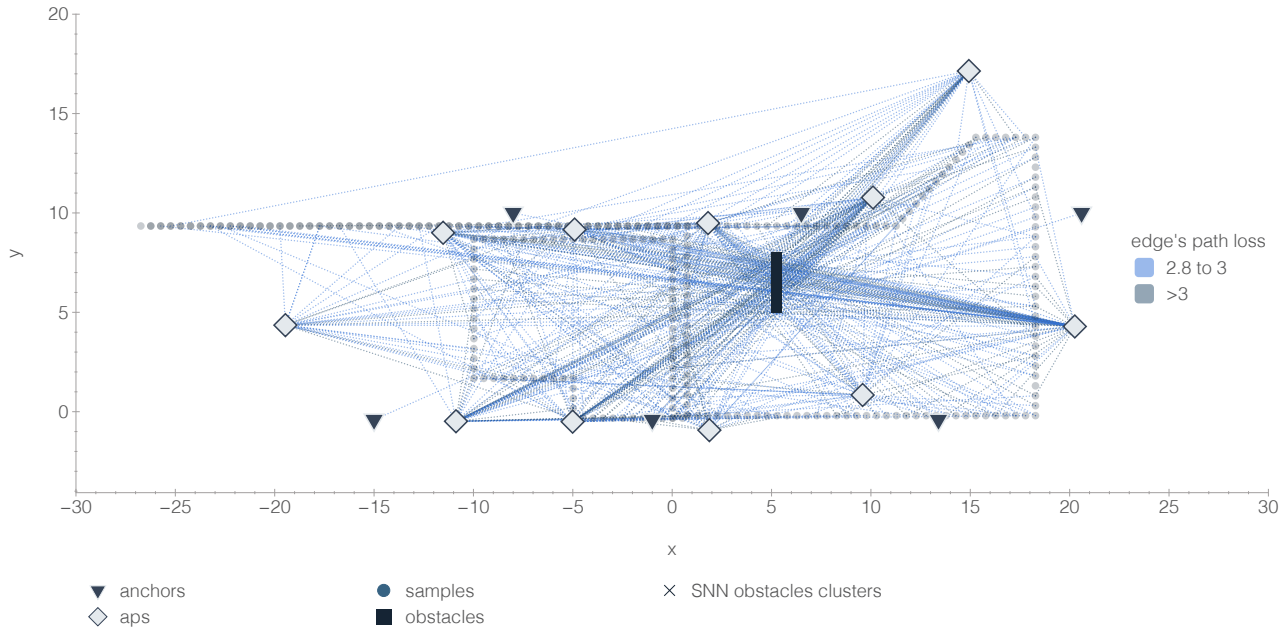

Figure 7.10: Edges with path loss exponent (η) over 2.8

It can clearly be observed that in the obstacle area there is a high density of edges intersecting in the XY plan. There are other edges with high η but unrelated with the obstacle. This is due to the noise introduced in the virtual space.

By observation it is possible to identify the obstacle position (Figure 7.11), but a computational method has to be developed to perform this identification automatically.

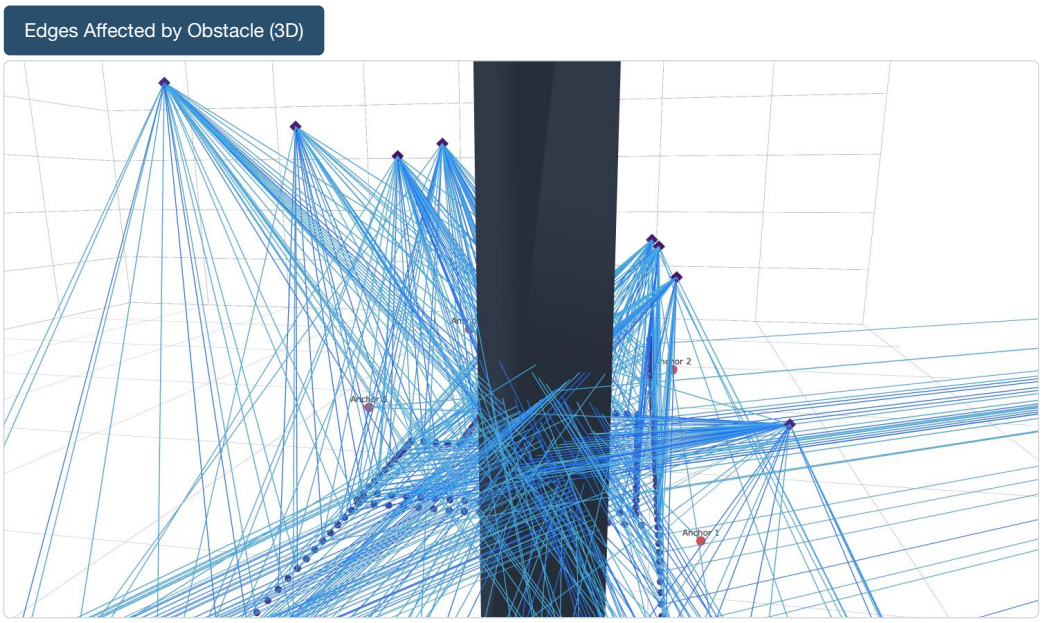

Figure 7.11: Edges in a 3D plan

In a 2D plan the intersection points of multiple edges can be used. However, when considering a 3D space, the edges intersection will not work to find the obstacle position. In 3D, the edges can intersect the obstacle at different Z coordinates, and most of them will in fact not intersect.

3 Identifying Obstacles Positions

The approach proposed to identify the obstacles in a 3D space, is to calculate the shortest distance between a pair of edges, obtaining this way a line segment that connects each pair of edges at the shortest distance.

There are, however, two problems with this approach. The first is when the two edges have the same origin or destination node. In this case the shortest distance will be at the origin or destination, and not at the obstacle position (Figure 7.12). This segments will not be useful to estimate the obstacles positions, therefore are eliminated.

The second problem appears when considering two edges from samples in positions too close. In this case, the shortest distance segment may be out of the obstacle. Figure 7.13 shows an example where the segments between the edges of consecutive samples are outside of the obstacle (out target), but the segments between the edges of intercalated samples are on the obstacle (on target).

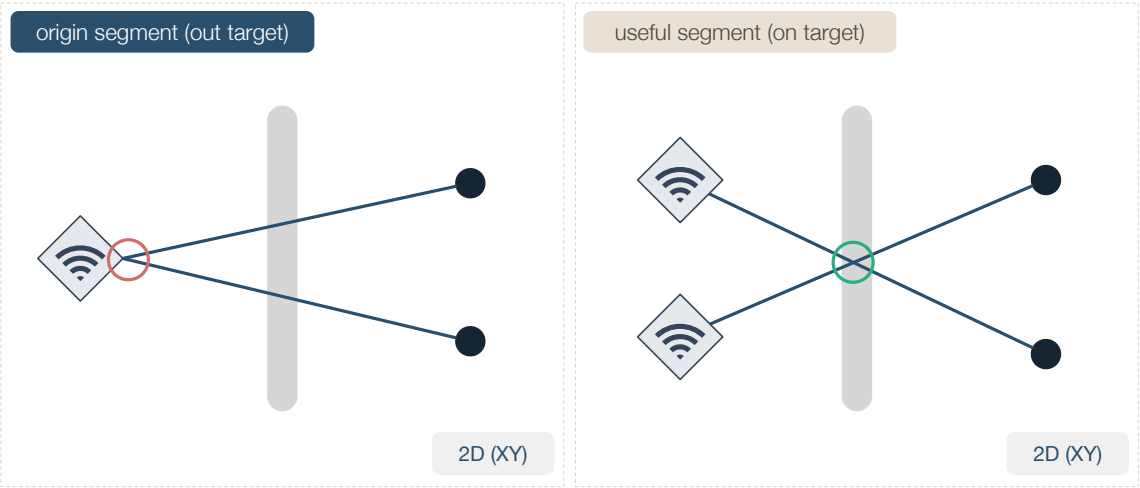

Figure 7.12: Shortest Distance Segments (Origin Segments)

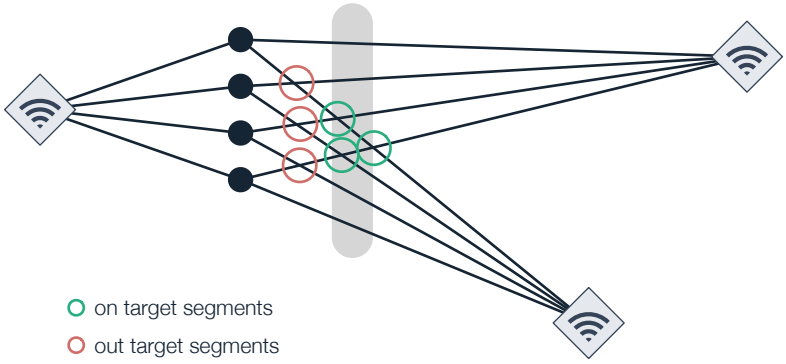

Figure 7.13: Shortest Distance Segments (Near Samples)

This problem is more complex than the origin segments, but can be minimized by considering only pair of edges from samples at a specific minimum distance.

To each segment, a weight is given based on the combined path loss factor of the two edges. For the time being, the weight will not be used, but in future experiments may be important to eliminate segments or to help identifying the obstacle material.

After obtained the line segments for all pairs of edges (Figure 7.14) are eliminated, the segments with length over 1 meter are eliminated. This removes the segments from the edges that are far from each other. Then the mid point of each segment is calculated, and the segments are converted into a cloud of points (Figure 7.15).

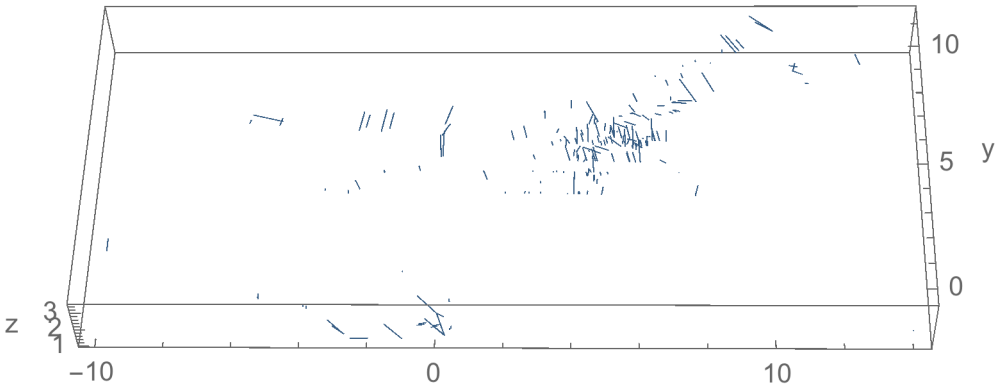

Figure 7.14: Segments Between Pairs of Edges

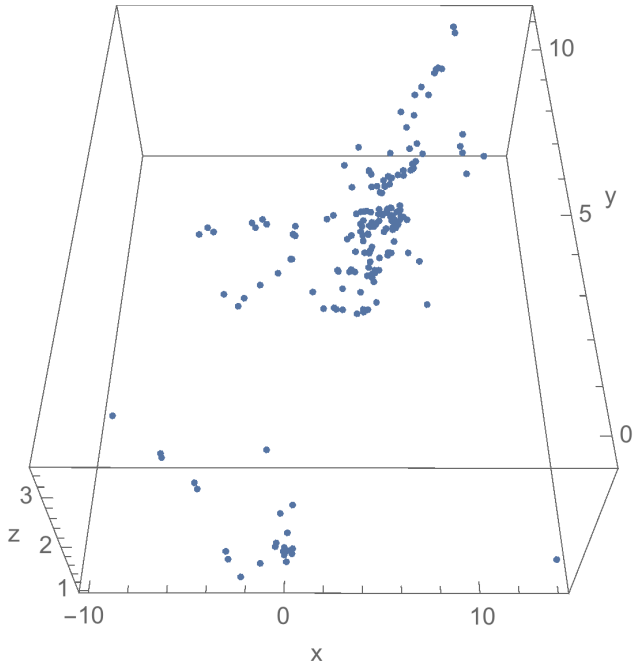

Figure 7.15: Cloud of Points

In the two previous plots, a main cluster of segments and points can be identified, at the obstacle position area. Therefore, a clustering algorithm is used to identify different possible clusters in the cloud of points, in order to identify one or more obstacles, and eliminate noise points.

Figure 7.16 shows the point cloud after applying the clustering algorithm. In this case the Shared Nearest Neighbor (SNN) algorithm was used. The SNN is a density based clustering algorithm, which uses the similarity distance within the shared nearest neighbors as metric.

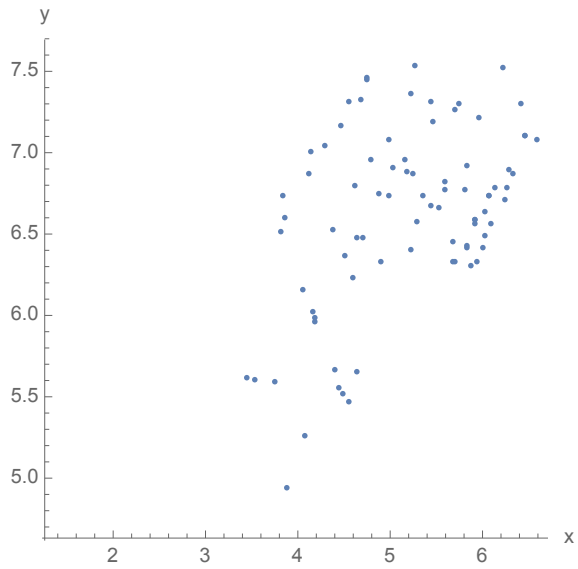

Figure 7.16: Point Clusters (SNN Algorithm)

When plotting the resulting cluster in the Map (Figure 7.17), it is possible to see that the cluster position is on the obstacle area. Although some points are out target, the obstacle is corrected defined in Y by the cluster points. In X, the points spread outside the obstacle area. This may be due to the small dimension of the obstacle in X (0.5 meters). In addition, the strategy to avoid the out target segments, resulting from the near (consecutive) samples was not implemented.

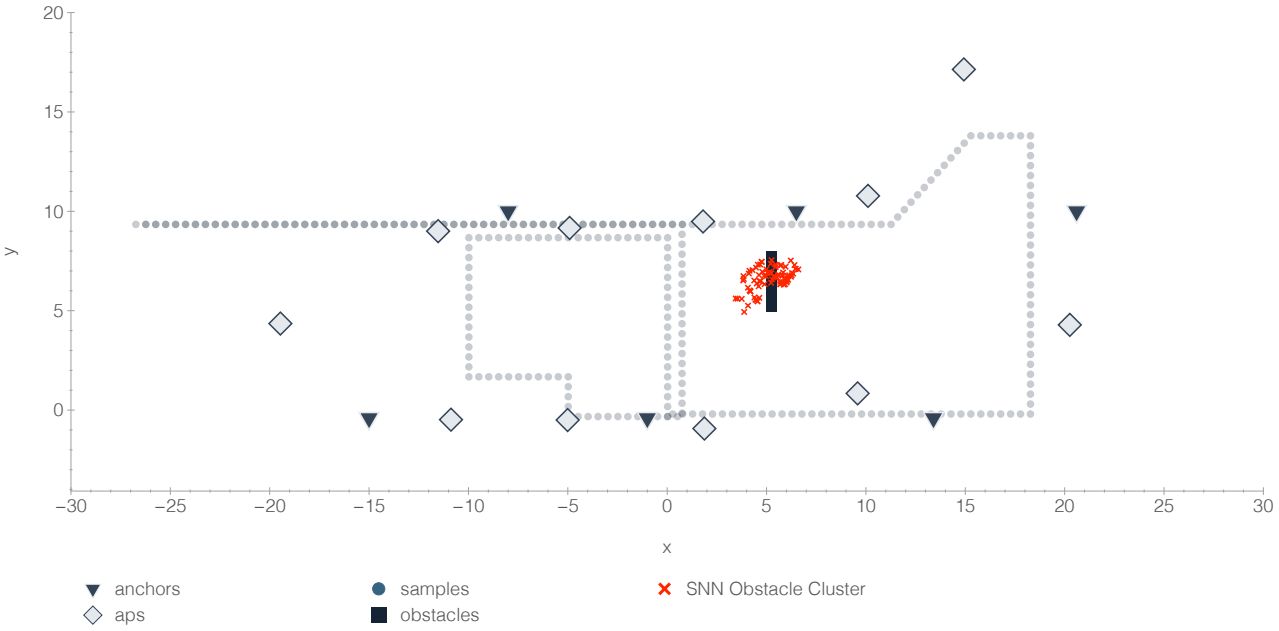

Figure 7.17: Obstacle Position Identification by Point Clustering

To complete the initial validation of this possible application, an additional simulation was done, this time with three obstacles with different dimensions and attenuation of 10 dB. Figure

7.18 shows the obstacles positions and the edges with path loss exponent over 2.8.

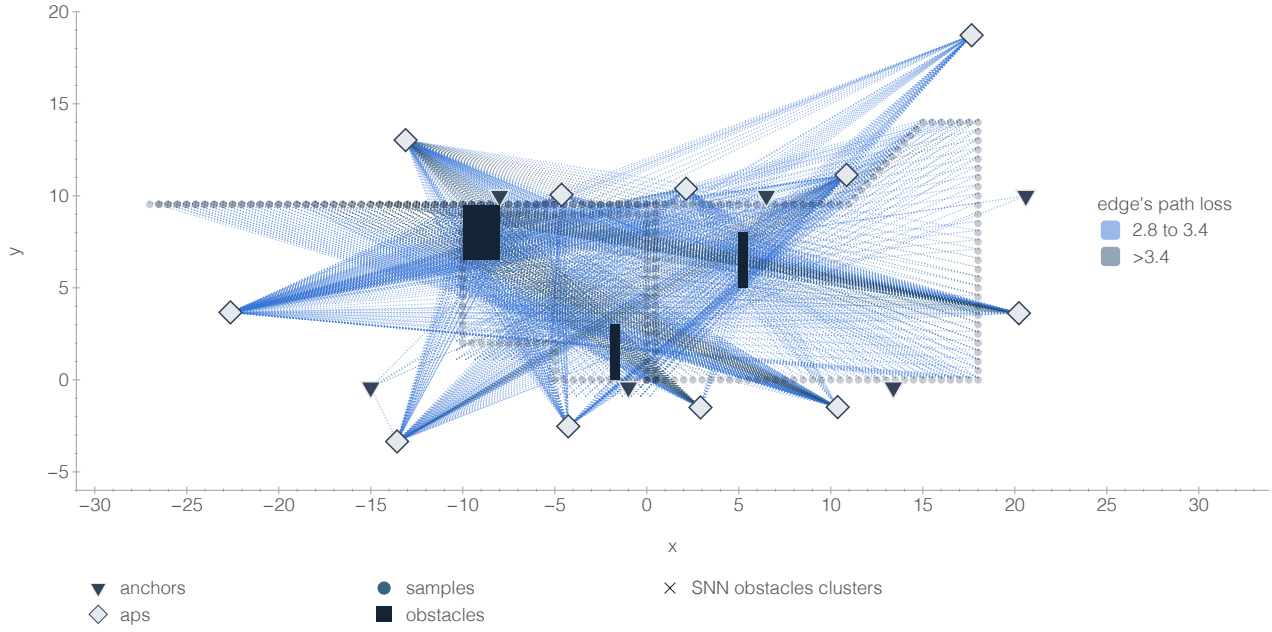

Figure 7.18: Edges with path loss exponent (η) over 2.8 (three obstacles)

When comparing to the same plot of the previous experiment (see Figure 7.10), the most noticeable aspects are the larger number of edges with η over 2.8. Moreover, the max path loss values are superior. This is related with having more obstacles, and the higher obstacles' attenuation, which is combined for the edges intersecting more than one obstacle. After the same steps described in the first experiment, three clusters were obtained, as shown in Figure 7.19.

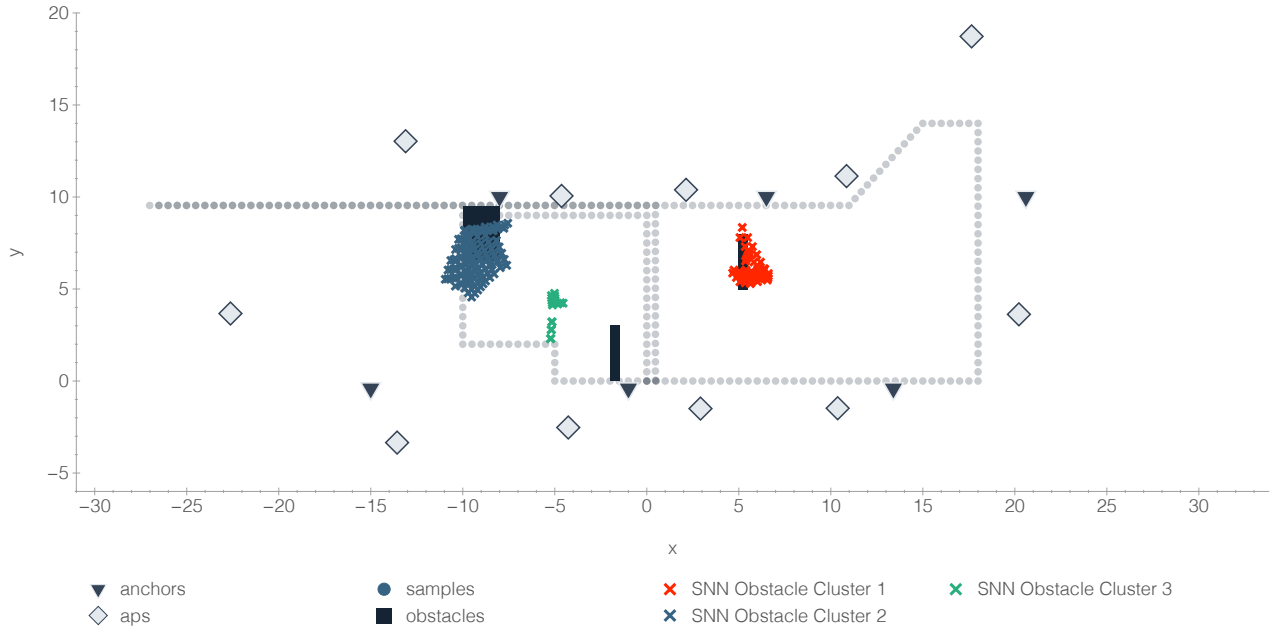

Figure 7.19: Obstacle Position Identification by Point Clustering (three obstacles)

The number of clusters obtained is correct, however the result shows that the green cluster is not correctly placed at the obstacle position. The reason for this is not clear, but may be due to a shadowing effect caused by the other two obstacles, especially the large obstacle at left. Or the result of segments from near consecutive samples, as described before.

It was not possible to further investigate this case. However, implementing the strategy to avoid segments from near consecutive samples, and reducing the shadowing areas with more samples in the Radio Map, may contribute to solve this problem.

In general, the other two clusters are well placed in the obstacles areas, which is an encouraging result and suggest that in fact, with further research and more elaborated methods, it may be possible to automatically create a floor plant, or at least map large obstacles.

4 Summary and Discussion

In this chapter were discussed two additional potential applications for the FastGraph solution. It was proposed that the additional information in the automatically created Radio Maps, especially the APs positions and propagation information, can be explored to automatically create interference maps and to map the physical elements of a space.

It was demonstrated that two types of interference maps can be created: Accurate Sample-based Interference Maps, or Fast Interference Maps for the entire space, created based on the APs positions estimated by FastGraph.

With the Anchors, both of these interference maps can be automatically created and maintained. Considering that the available tools for interference mapping and analysis rely on manual site survey, the FastGraph automatic interference mapping features are interesting and useful.

While researching the interference maps topic, an interesting and potential important discovery was made. A relationship between the interference level and the obtained positioning errors was observed. This relation, as far as we known, was never reported in the research community, and can be significant for the indoor positioning field. At this point it is not clear if this relation is a FastGraph's unique characteristic, related to the distinct aspects of the Force-Directed Graph approach, or if can be explored in other Wi-Fi positioning techniques. Preliminary analysis suggest that this relation can be found in other positioning methods, such as Fingerprinting. Therefore, in the future this subject has to be further developed and validated.

Regarding the physical space mapping hypothesis, the base idea to automatically map obstacles, using the path loss exponent information in the FastGraph' Radio Maps was described. The edges with high path loss exponent are combined in pairs to obtain a cloud of points. The position of the obstacles is then discovered by applying a clustering algorithm in the cloud of points. Two experiments were presented as an initial validation. The RF 3D Environment Simulator was used to generate two synthetic datasets. A different number of obstacles, with different attenuations, were used in each dataset. In addition, noise was added to the RSS measurements in order to simulate other propagation effects in real world environments. The results are preliminary and further research and experimentation is required to validate the proposed hypothesis in more complex scenarios, which in the time frame of this PhD was not possible to accomplish. Nevertheless, these results are encouraging and suggest that, at least in specific scenarios, with FastGraph it is possible to automatically discover the position of obstacles, with acceptable accuracy.

Chapter 8

Conclusions and Future Work

This chapter concludes this PhD thesis, summarizing the fundamental discoveries and new contributions, as well as the main prospects for future work.

The first section begins with a brief recap regarding the initial hypothesis and objectives, as well as an general overview of the proposed approach to achieve them. Moreover, some relevant and practical aspects related to the solution deployment are also discussed.

Section two introduces some ideas to be addressed in a near future and presents long-term prospects for future research, discussing the FastGraph solution in a broader perspective.

1 Conclusions

In the beginning of this PhD, the main research objective was defined as the creation of a new method for unsupervised radio mapping and positioning, that may be applied to different wireless technologies. In this research, the Wi-Fi was considered for being an ubiquitous and well establish technology in the indoor environments and in the indoor positioning field.

The main contribution from this research are a new method to automatically create radio maps and provide positioning, as alternative to the exhaustive manual calibration required by the main indoor positioning techniques using Wi-Fi, such as Fingerprinting.

To address the problem stated, a new hypothesis was proposed: **A dynamic 3D Graph that is built iteratively and adjusted by a Force-Directed approach, that can be used to model radio environments and to automatically create and maintain Radio Maps, and that can be used in different contexts.**

The research of this problem and the validation of the proposed hypothesis resulted in a new approach named FastGraph, which can be described as a new positioning and mapping solution, that as shown can be used in different applications.

The FastGraph algorithm implements the Force-Directed 3D Graph approach, which is used as alternative to trying to analytically solve multiple sets of equations, in a multidimensional scaling problem. In fact, along the research process, the FastGraph solution surpassed the initial objective of automatic creation of radio maps, becoming a SLAM solution, able to provide positioning while creating the radio map.

Following the initial objectives set for this research, an in contrast with other approaches, the proposed method does not rely on previous information or manual calibration to provide positioning, such as radio maps or knowledge about the APs' positions. In addition to create and maintain Radio Maps, FastGraph is in fact also able to automatic estimate the APs' positions. Moreover, the proposed solution is able to keep all of this information updated by adjusting dynamically to the radio environment changes. Also, no assumptions are made about the propagation parameters being uniform for a given space, or that propagation characteristics for an AP are equal in all areas of the space. This makes the solution more versatile to be used in different environments.

It was also theorized that the additional information in the FastGraph's Radio Maps (APs positions and propagation information) can be used in other applications. An initial validation suggest that the radio maps from FastGraph can be used to automatically create interference

maps and map the physical elements of a space. The current tools for interference mapping and analysis rely on manual site survey. Therefore, the FastGraph approach, with automatic interference mapping features, can be an important contribution in this area. Moreover, during the research process, an interesting relation between the Signal to Interference Ratio (SIR) and the positioning error was found. As far as we known this relation has never been reported before, and can lead to improvements in the indoor positioning field, deserving further research.

An initial and simple validation of the hypothesis of using the radio maps to automatically map the layout and obstacles of a space was also performed, using synthetic data. The results suggest that the hypothesis is valid, but the research of this topic is only in an initial stage not allowing to draw further conclusions.

In the positioning field it was demonstrated that FastGraph can provide real-time unsupervised positioning, within only a few minutes after deploying the Anchors. It was also shown that the solution can operate in Wi-Fi only applications, as well as in enhanced accuracy positioning applications, by combining the Wi-Fi with data from sensors such as IMUs and encoders.

The FastGraph positioning performance was validated in two different environments, showing promising results an prospects in this field. When using only Wi-Fi, the FastGraph was able to provide positioning with an average error in the 5 meters range, which compare well with state of the art solutions that require calibration.

When combining the Wi-Fi with data from other sensors (orientation and displacement) the results obtained are especially interesting, with average error around 0.5 meters. This enhanced operation mode can be used in applications with more strict accuracy requirements, such as autonomous navigation inside industrial environments.

1.1 Practical Aspects: Deployment, Maintenance and Operation

The effort and costs related to the deployment and maintenance of a positioning solution are two of the most important practical aspects, and can compromise the solution regardless the achieved positioning performance. In this context, a brief analysis of these aspects regarding FastGraph is an interesting complement.

1.1.1 Easy deployment, configuration and self-maintenance

As explained before, the only requirement to deploy the FastGraph is the installation of the Anchors, after that FastGraph is a full unsupervised positioning and mapping solution. In specific scenarios, where the Anchors can't be installed, referenced fingerprints can be used instead.

The advantage of using the Anchors is that after being installed no further effort or calibration is required to maintain the solution performance. In contrast, in most positioning solutions such as fingerprinting, the Radio Maps for the entire space have to be manually created and then updated from time to time. Moreover, the robustness is one of the foundations in the FastGraph approach. FastGraph was designed to dynamically adjust to the radio environment changes, due to for example obstacles being moved, different density of people, and even due to the radio infrastructure being updated, with APs' being removed or added. In addition, it was shown that FastGraph can maintain an acceptable positioning performance if an Anchor fails.

1.1.2 Fast Initialization After Deployment

One of the advantages of the FastGraph is the ability of providing positioning only a few minutes after the deployment. In the experiments in two very distinct spaces, the FastGraph initialization took only between 6 to 8 minutes of sampling from the Anchors.

1.1.3 Scalability and Positioning Time

The FastGraph can be deployed in different spaces, with the scalability mechanisms ensuring the algorithm processing performance. The experimental results show that the solution can provide a position estimation quickly, while maintaining the positioning performance, even when the algorithm is running in a low grade machine. This also suggests that the FastGraph may be able to run directly for example in a Smartphone or Vehicle, which can be useful in specific conditions.

1.1.4 Deployment Costs

The experimental results also suggest that interesting positioning performance can be obtained with a low density of Anchors, which results in very low deployment costs. Based on these results, some deployment costs indicators[1] were estimated and are presented next, when using the Raspberry Pi Zero W (11 EUR) as Anchors and a density of 1 Anchor for 357 m^2 (same density used at DSI-DEP):

- South China Mall (≈892,000 m^2) one of the world's largest malls: 2499 Anchors (27 489 EUR).
- King Fahd International Airport - Dammam, Saudi Arabia (≈327,018 m^2): 916 Anchors (10 076 EUR).

[1]These values are a estimative based on the space gross area, and can vary depending on the space layout.

These are very low deployment costs, even more considering that after the deployment the solution is full automatic not requiring calibration or maintenance.

2 Future Work

The limited time frame defined for a PhD means that in general many ideas are left unexplored. The results suggest that this solution can be extended to different fields and has potential to grow and to be improved, therefore the scope for future research is wide.

The short-term objectives include testing the FastGraph solution with 5 GHz Wi-Fi enable devices, as it is well known that the 5 GHz channels are less affected by noise and signal fluctuations. With this, the positioning performance is expected to improve. This was not already done because the Wi-Fi interface in the Raspberry Pi used (RPi3) has only support for 2.4 GHz. The new version (RPi 3B+) already has support for 5 GHz bands. Testing the solution, and the positioning performance, with Anchors in multiple floors is also planned.

Another idea to be explored in the near future is to add an additional Wi-Fi interface to the Anchors. With another interface, the Anchors can also work as Wi-Fi Beacons, and can be explored by FastGraph as APs with known positions, creating additional constrains, and possible improving the positioning performance without any extra deployment effort. Moreover, when Anchors can detect each other, more accurate path loss values can be calculated for the communication channel between the two Anchors. As explained before, the air pressure can be used as reference to identify a floor, however it varies over time. For this reason, reference barometer sensors deployed in different floors are frequently used to collect air pressure readings to be used as floor reference [85,86]. Therefore, another idea to be explored is the integration of a barometer sensor in the Anchors, to provide air pressure references continuously in multiple floors, also without any extra deployment effort.

As shown, the radio maps created and maintained by the FastGraph can be explored in different applications. The automatic network maintenance and planing, and the physical mapping are only two examples covered in this thesis, and the initial validation provided promising indicators. The relation between the SIR and the positioning error is specially promising, since is easy to obtain and can contribute to improve the positioning performance and provide a confidence metric for the position estimations. These two application require further research and broader evolution, and may be an important contribution to improve positioning solutions and communications quality in Wi-Fi environments.

The easy and inexpensive deployment, the fast positioning, and the automatic adaptation to the environment changes, place the FastGraph as a versatile solution, that can be used in different applications and environments, and specially promising in the positioning field.

2.1 FastGraph's Possible Future Application Scenarios

During this thesis were discusses different contexts where the FastGraph can be used for radio mapping and positioning. The experiments and results presented in this thesis provided a validation for the proposed solution, and covered two of the possible scenarios.

The next step is to deploy the FastGraph in more spaces, as a full operational and long-term positioning and radio mapping solution. The deployment in more buildings in the University campus, operating as regular indoor positioning applications, is the logical next step, being only necessary to install the Anchors (Figure 8.1).

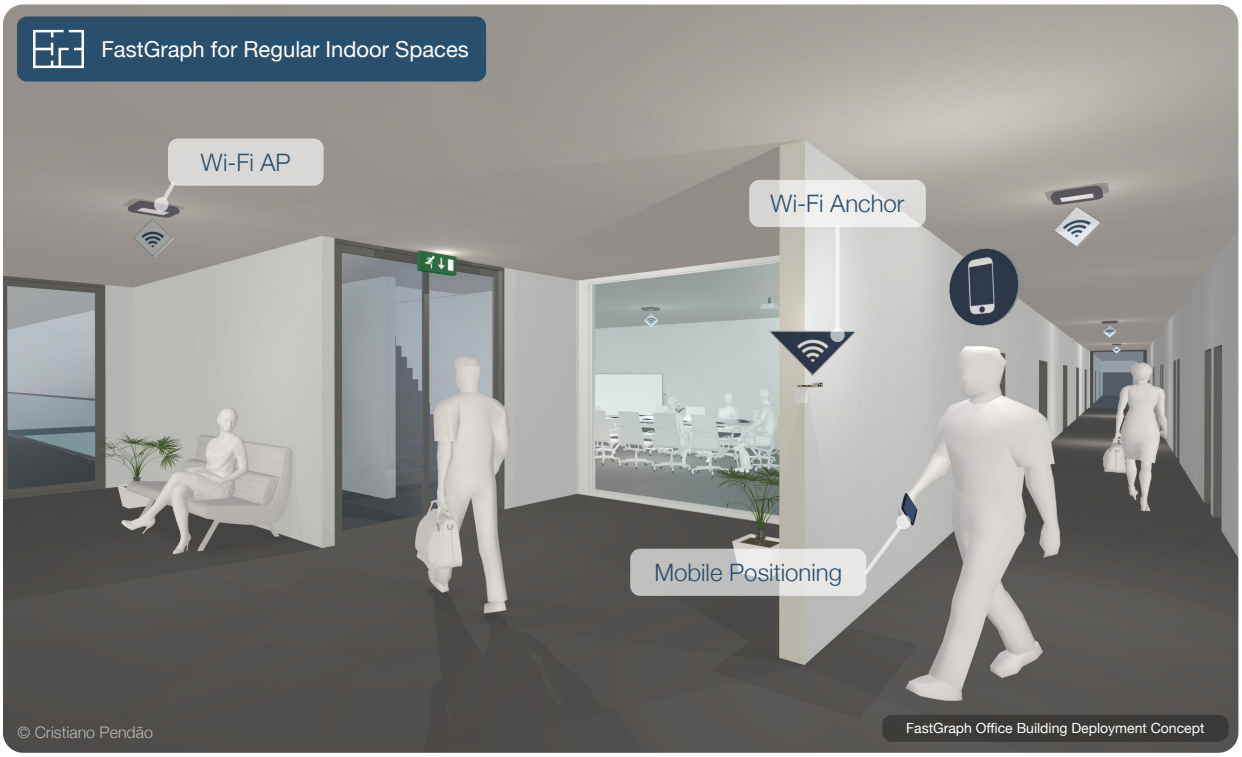

Figure 8.1: Smart Office Building Deployment Concept

The positioning performance in this type of scenario can be enhanced by combining the Wi-Fi data with the data from smartphone's sensors. Moreover, it may also worth to experiment the algorithm running directly in Smartphones. With the Anchors working also as Wi-Fi beacons, this approach may be specially interesting in specific conditions.

It is also important to deploy FastGraph in more industrial environments, and taking the experiments further by integrating with autonomous machines, such as fork lifters and cargo robots, and perhaps in the future even with warehouse package carrying drones (Figure 8.2).

The results from the experiments at PIEP suggest that this is possibly one of the most promising application scenarios for the FastGraph's positioning features. In this context, the next step is the integration of FastGraph with autonomous machines navigation systems and sensors.

Figure 8.2: Smart Factory Deployment Concept

Another promising application scenario mentioned before is the positioning in large spaces, such as Airports or Shopping Malls. Figure 8.3 shows a deployment concept for an Airport, and in the depicted concept scenario, the FastGraph provides positioning for the passengers using smartphones and for autonomous bag carrier vehicles.

This is a more ambitious scenario, however with the Anchors installed the FastGraph can provide unsupervised positioning, not requiring further intervention or calibration. In large spaces, such as an Airport, this is one essential aspect.

A long-term objective is the extension of FastGraph for outdoor environments, and in this context the upcoming technologies and future applications have to be discusses and considered.

Cellular Networks will have an essential role in outdoors positioning, for this reason cellular data was already collected in the performed experiments. The next step is to prepare the FastGraph to handle this data and combine it with Wi-Fi. The cellular base stations can be treated in the same way as the Wi-Fi APs (Figure 8.4), however it is necessary to model the propagation of the cellular radio signals. The cellular networks operate in different frequency bands and the propagation is different from Wi-Fi and Bluetooth.

Extending FastGraph to outdoor environments can open a different new set of opportunities,

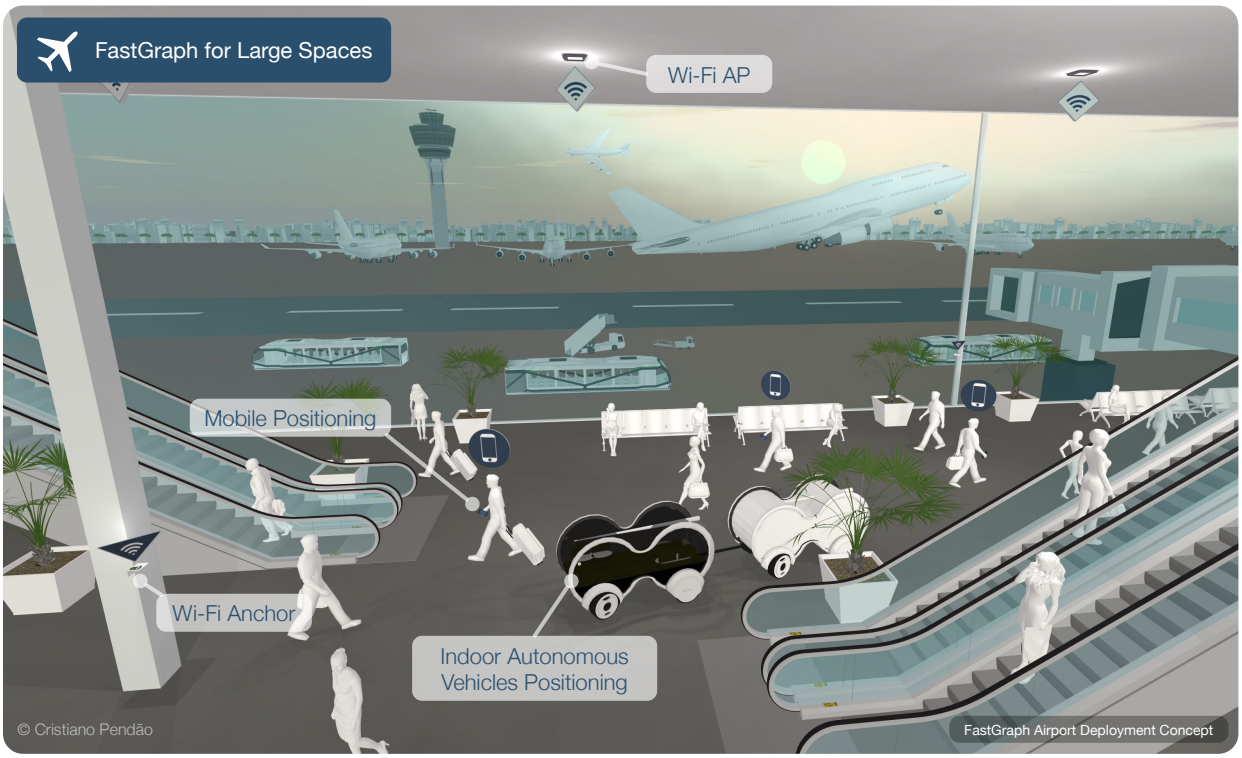

Figure 8.3: Airport Deployment Concept

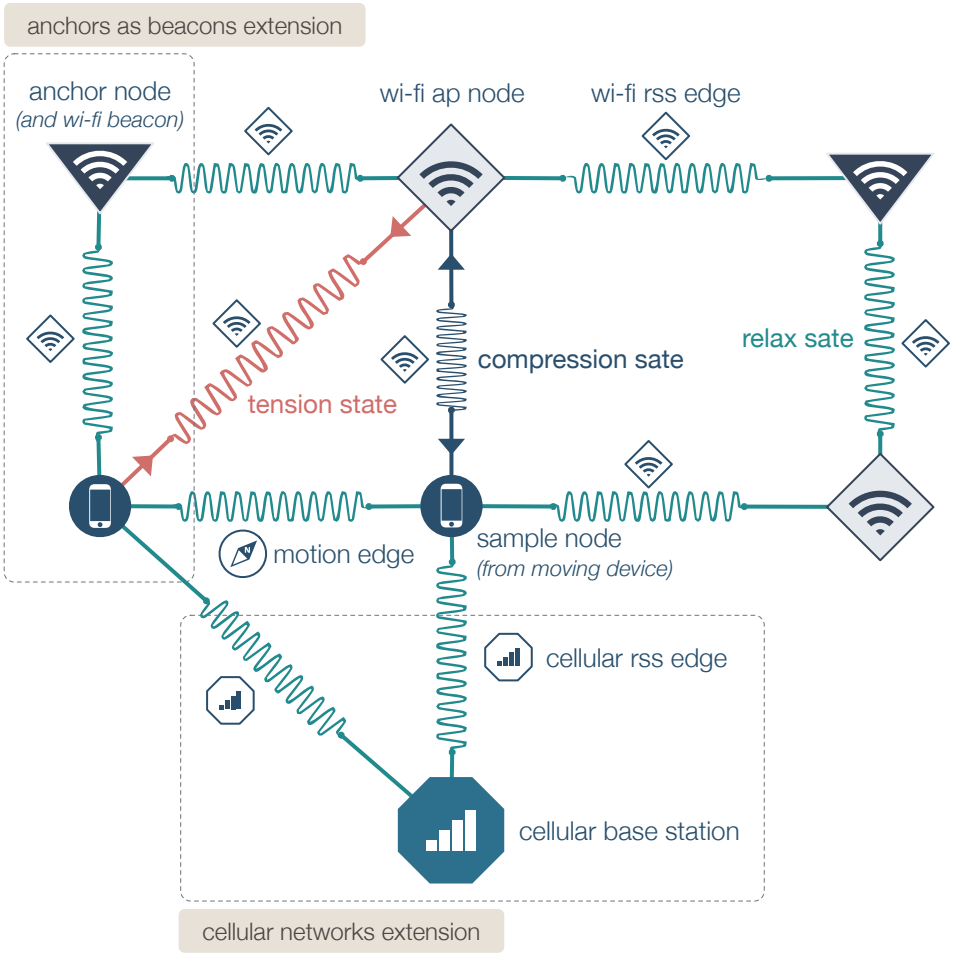

Figure 8.4: FastGraph: Anchors as Beacons and Cellular Networks Extensions

specially considering the 5G specifications. Small cells are expected to have a key role in 5G, being deployed for example in city lamp posts (Figure 8.5), with the necessary power and with no visual footprint. This smart light posts can also include Wi-Fi, environmental sensors or security cameras. In addition, 5G will probably require the deployment of small cells indoors, for example in Shopping Malls or Hospitals. This can benefit solutions targeting seamless positioning between outdoors and indoors.

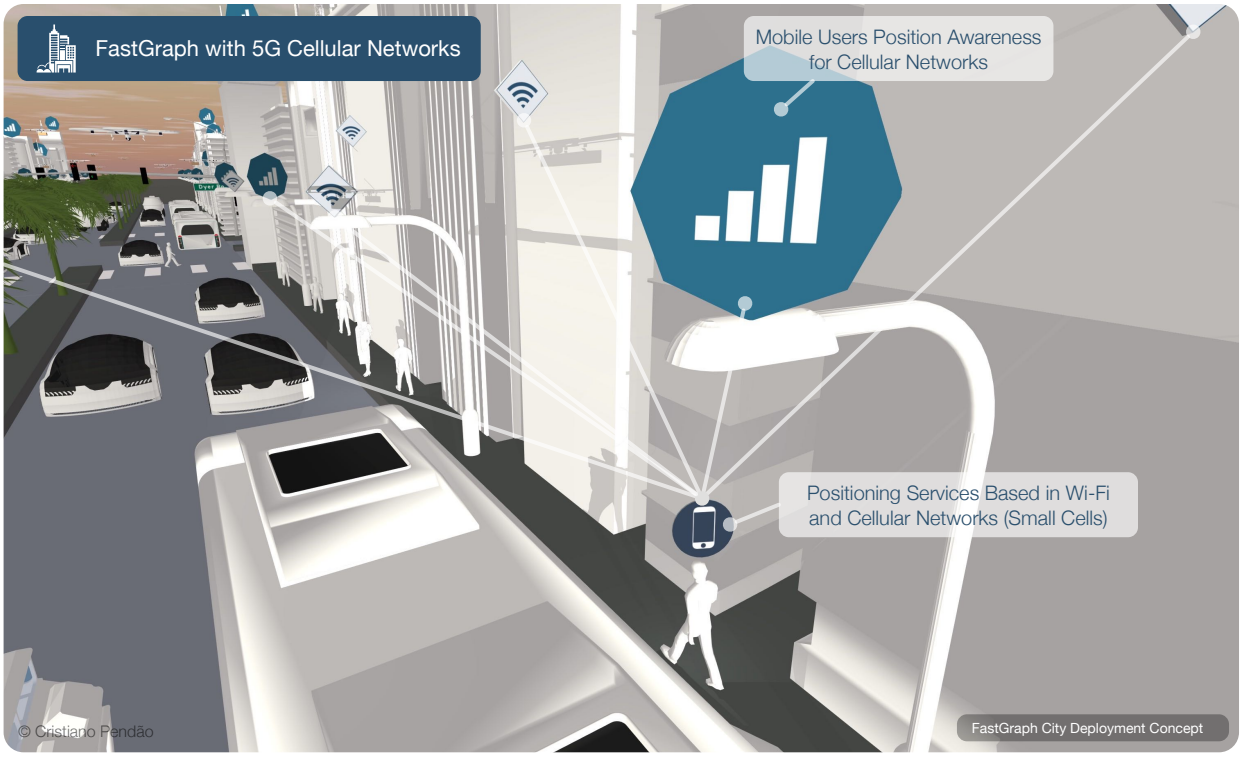

Figure 8.5: Cellular and Wi-Fi Positioning

The Millimeter Wave (MM-Wave) is a promising technology for the next generation of cellular communications. Operating in the high frequency bands enables improved speeds and capacity [93]. However, the transmission with millimeter waves usually need line of sight to work, leading to dense deployments, which can benefit the positioning accuracy.

Considering all of these prospects, the FastGraph approach seams a interesting base to support future positioning systems, combining Wi-Fi, Cellular and Sensors data, and perhaps providing seamless positioning between indoors and outdoors.

In addition to provide positioning for moving devices, a solution such as FastGraph may also be used by the cellular networks to obtain accurate positioning information about the mobile terminals, even indoors. This positioning information can be used to improve spectrum usage, increase data transmission rates, and characterize and map how the mobile network is used in terms of services, users density and motion patterns.

The current and future applications for a scalable and calibration-free positioning solutions

such as FastGraph are therefore endless. And will be required to operate with same performance both in indoors and outdoors.

The Internet of Things (IoT) and Augmented Reality (AR) are two examples of applications that will require accurate and ubiquitous positioning solutions (Figure 8.6).

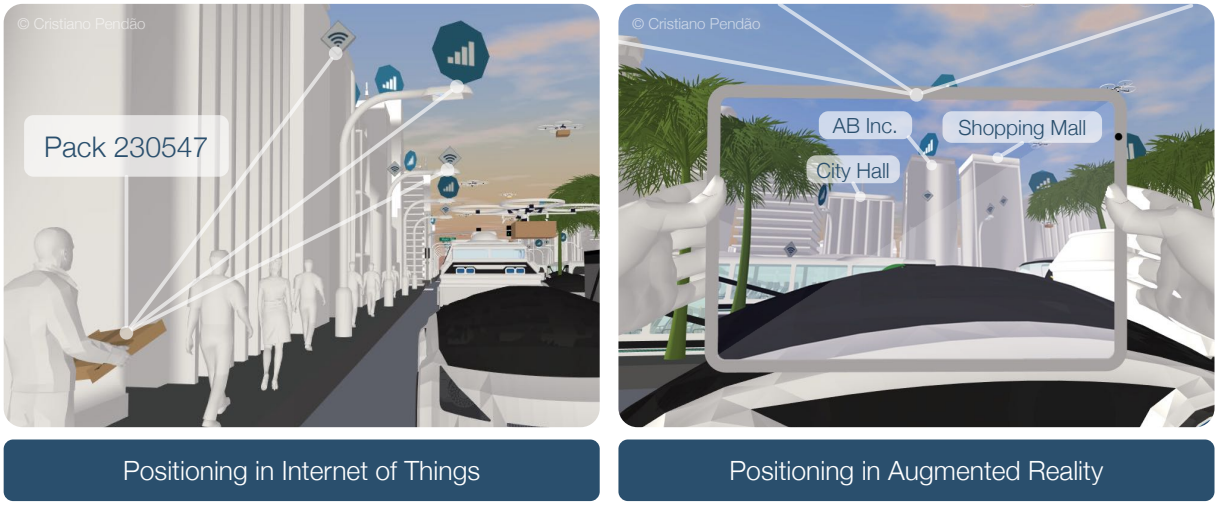

Figure 8.6: Positioning in IoT and AR

The current and most popular AR applications are powered by GNSS services. However, the normal accuracy of for example GPS (5 to 20 meters), is not enough for advanced AR, such as display 3D content over a store front door. In addition, even in outdoors the GNSS services can operate poorly, since in large cities the signals can be affected by the buildings (urban canyons), with a reported accuracy of 38 meters in dense urban environments, even when combining GPS+GLONASS+Galileu[2]. For indoor, the same source reports 67 meters of accuracy in the best scenario and also using the three GNSS services.

Since Wi-Fi can be used as an alternative to GNSS for indoors, solutions based for example in fingerprinting have been already proposed for AR [3]. However, as mentioned before, this type of solutions have low accuracy or require constant calibration, being expensive and difficult to deploy and maintain. With accurate positioning Augmented Reality can enable intuitive visual navigation, being possible to easily find and identify a product inside a large store, such as IKEA, or to create a position-based experience in natural parks, museums or sport events.

The Internet of Things (IoT) is a upcoming reality, and is already being considered in the new generation of cellular technologies, with the 3GPP working in positioning support for IoT [2]. The examples of IoT applications requiring positioning in indoors and outdoors are numerous, such as package and cargo tracking in warehouses or in the street (massive IoT), as well as inventory tracking in hospitals. This is one of the reasons why the GNSS services

[2]https://www.gsa.europa.eu/newsroom/news/results-are-galileo-increases-accuracy-location-based-services

is not considered suitable for supporting IoT in most scenarios. Wi-Fi and cellular networks, specially the upcoming generations, are therefore promising alternatives in this field.

Future applications such as a drone delivering a package or medical supplies inside a building, or even directly on a car in traffic, are a reality not so far, and will most certainly use innovative positioning approaches and communications technologies being developed today (Figure 8.7).

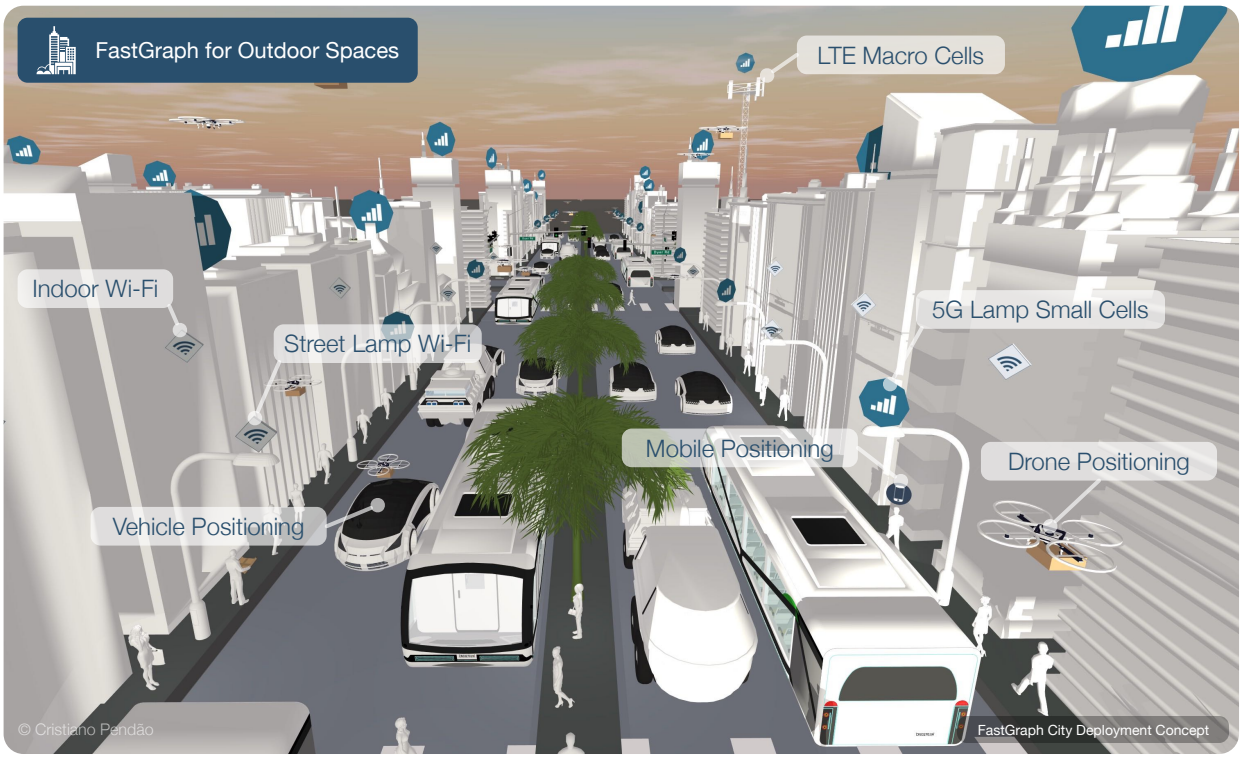

Figure 8.7: Future City Deployment: A Not so Far Concept

Bibliography

[1] J. Torres-Sospedra, A. Jiménez, A. Moreira, T. Lungenstrass, W.-C. Lu, S. Knauth, G. Mendoza-Silva, F. Seco, A. Pérez-Navarro, M. Nicolau, A. Costa, F. Meneses, J. Farina, J. Morales, W.-C. Lu, H.-T. Cheng, S.-S. Yang, S.-H. Fang, Y.-R. Chien, and Y. Tsao, "Off-Line Evaluation of Mobile-Centric Indoor Positioning Systems: The Experiences from the 2017 IPIN Competition," *Sensors*, vol. 18, no. 2, p. 487, 2018.

[2] X. Lin, J. Bergman, F. Gunnarsson, O. Liberg, S. M. Razavi, H. S. Razaghi, H. Rydn, and Y. Sui, "Positioning for the Internet of Things: A 3GPP Perspective," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 179–185, 2017.

[3] D. Shi, F. Liu, Q. Yutian, and Y. Ji, "A WLAN-based positioning system for indoor augmented reality services," *Proc. - 2014 Int. Conf. Inf. Sci. Electron. Electr. Eng. ISEEE 2014*, vol. 1, no. 0800219162, pp. 420–424, 2014.

[4] K. Trawiński, J. M. Alonso, and N. Hernández, "A multiclassifier approach for topology-based WiFi indoor localization," *Soft Comput.*, vol. 17, no. 10, pp. 1817–1831, 2013.

[5] B. R. Stojkoska, J. Palikrushev, K. Trivodaliev, and S. Kalajdziski, "Indoor localization of unmanned aerial vehicles based on RSSI," *IEEE EUROCON 2017 -17th Int. Conf. Smart Technol.*, pp. 120–125, 2017.

[6] Z. Lin, T. Lv, and P. T. Mathiopoulos, "3-D Indoor Positioning for Millimeter-Wave Massive MIMO Systems," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2472–2486, 2018.

[7] A. Dammann, R. Raulefs, and S. Zhang, "On prospects of positioning in 5G," in *2015 IEEE Int. Conf. Commun. Work. ICCW 2015*, pp. 1207–1213, 2015.

[8] V. Savic and E. G. Larsson, "Fingerprinting-based positioning in distributed massive MIMO systems," *2015 IEEE 82nd Veh. Technol. Conf. VTC Fall 2015 - Proc.*, 2016.

[9] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, "When cellular meets WiFi in wireless small cell networks," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 44–50, 2013.

[10] A. Kashyap, U. Paul, and S. R. Das, "Deconstructing interference relations in WiFi networks," *SECON 2010 - 2010 7th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Networks*, no. July, 2010.

[11] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding congestion in IEEE 802.11b wireless networks," *Proc. 5th ACM SIGCOMM Conf. Internet Meas. - IMC '05*, p. 1, 2005.

[12] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based characterization of 802.11 in a hotspot setting," *Proceeding 2005 ACM SIGCOMM Work. Exp. approaches to Wirel. Netw. Des. Anal. - E-WIND '05*, no. July 2015, p. 5, 2005.

[13] H. Rocha, T. Cacoilo, P. Rodrigues, S. Kandasamy, and R. Campos, "Wi-Green: Optimization of the power consumption of Wi-Fi networks sensitive to traffic patterns invited paper," *2017 15th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wirel. Networks, WiOpt 2017*, pp. 1–5, 2017.

[14] I. Cisco Systems, "Enterprise Mobility 4.1 design guide: Cisco Systems, Inc.," no. 6387, 2008.

[15] J. Hightower, J. Hightower, G. Borriello, and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Comput.*, vol. 34, pp. 57–66, 2001.

[16] L. Ni, "LANDMARC: Indoor Location Sensing Using Active RFID," *PerCom 2003*, pp. 701–710, 2003.

[17] J. Ledlie, J.-g. Park, D. Curtis, A. Cavalcante, L. Camara, A. Costa, and R. Vieira, "Mole: A scalable, user-generated WiFi positioning engine," in *Indoor Position. Indoor Navig. 2011 Int. Conf.*, pp. 1–10, sep 2011.

[18] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-Effort Crowdsourcing for Indoor Localization," *Proc. 18th Annu. Int. Conf. Mob. Comput. Netw. - Mobicom '12*, p. 293, 2012.

[19] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche, "A comparative survey of WLAN location fingerprinting methods," in *Positioning, Navig. Commun. 2009. WPNC 2009. 6th Work.*, pp. 243–251, mar 2009.

[20] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91–102, 1992.

[21] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Pers. Commun.*, vol. 4, no. 5, pp. 42–47, 1997.

[22] A. Williams, D. Ganesan, and A. Hanson, "Aging in place: fall detection and localization in a distributed smart camera network," *Proc. 15th Int. Conf. Multimed.*, pp. 892–901, 2007.

[23] a. Kotanen, M. Hannikainen, H. Leppakoski, and T. Hamalainen, "Experiments on local positioning with Bluetooth," *Proc. ITCC 2003. Int. Conf. Inf. Technol. Coding Comput.*, 2003.

[24] J. Hallberg, M. Nilsson, and K. Synnes, "Positioning with Bluetooth," *10th Int. Conf. Telecommun. 2003. ICT 2003.*, vol. 2, 2003.

[25] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, 2007.

[26] P. Muller, M. Raitoharju, and R. Piche, "A field test of parametric WLAN-fingerprint-positioning methods," in *Inf. Fusion (FUSION), 2014 17th Int. Conf.*, pp. 1–8, jul 2014.

[27] T. Pulkkinen, T. Roos, and P. Myllymäki, "Semi-supervised learning for WLAN positioning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6791 LNCS, no. PART 1, pp. 355–362, 2011.

[28] C. Wu, Z. Yang, and Y. Liu, "Smartphones based Crowdsourcing for Indoor Localization," *IEEE Trans. Mob. Comput.*, vol. 14, no. 2, pp. 444–457, 2015.

[29] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor Fingerprint Positioning Based on Wi-Fi: An Overview," *ISPRS Int. J. Geo-Information*, vol. 6, no. 5, p. 135, 2017.

[30] A. Moreira, I. Silva, F. Meneses, M. J. Nicolau, C. Pendão, and J. Torres-Sospedra, "Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning," in *2017 Int. Conf. Indoor Position. Indoor Navig. IPIN 2017*, vol. 2017-Janua, pp. 1–8, 2017.

[31] K. Kaemarungsi and P. Krishnamurthy, "Analysis of WLAN's received signal strength indication for indoor location fingerprinting," *Pervasive Mob. Comput.*, 2012.

[32] C. Feng, W. S. Anthea Au, S. Valaee, and Z. Tan, "Compressive sensing based positioning using RSS of WLAN access points," in *Proc. - IEEE INFOCOM*, 2010.

[33] S. Knauth, M. Storz, H. Dastageeri, A. Koukofikis, and N. A. Mahser-Hipp, "Fingerprint calibrated centroid and scalar product correlation RSSI positioning in large environments," in *2015 Int. Conf. Indoor Position. Indoor Navig. IPIN 2015*, 2015.

[34] G. Caso, L. D. Nardis, F. Lemic, V. Handziski, A. Wolisz, and S. Member, "ViFi : Virtual Fingerprinting WiFi-based Indoor Positioning via Multi-Wall Multi-Floor Propagation Model," vol. 14, no. 8, 2015.

[35] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," *Proc. IEEE INFOCOM 2000. Conf. Comput. Commun. Ninet. Annu. Jt. Conf. IEEE Comput. Commun. Soc. (Cat. No.00CH37064)*, vol. 2, 2000.

[36] S. Conceição, C. Pendão, A. Moreira, and M. Ricardo, "Evaluation of medium access and a positioning system in wireless underground sensor networks," in *IFIP Wirel. Days*, vol. 2016-April, 2016.

[37] M. Youssef, a. Agrawala, and a. U. Shankar, "WLAN location determination via clustering and probability distributions," *Proc. First IEEE Int. Conf. Pervasive Comput. Commun. 2003. (PerCom 2003).*, 2003.

[38] M. Azizyan, R. R. Choudhury, and I. Constandache, "SurroundSense : Mobile Phone Localization via Ambience Fingerprinting," *MobiCom '09*, pp. 261–272, 2009.

[39] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," *Proc. 10th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '04*, p. 70, 2004.

[40] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale Wi-Fi localization," *MobiSys '05 Proc. 3rd Int. Conf. Mob. Syst. Appl. Serv.*, pp. 233–245, 2005.

[41] P. C. Paul Castro, "A Probabilistic Room Location Service for Wireless Networked Environments," in *UbiComp '01 3rd Int. Conf. Ubiquitous Comput.*, pp. 18–34, 2001.

[42] R. Battiti, T. Le Nhat, and A. Villani, "Location-Aware Computing: a Neural Network Model for Determining Location in Wireless Lans," *Povo – Trento*, no. February, 2002.

[43] P. Kontkanen, P. Myllymaki, T. Roos, H. Tirri, K. Valtonen, and H. Wettig, "Topics in probabilistic location estimation in wireless networks," in *15th IEEE Int. Symp. Pers. Indoor Mob. Radio Commun.*, vol. 2, pp. 1052–1056, 2004.

[44] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A Probabilistic Approach to WLAN User Location Estimation," *Int. J. Wirel. Inf. Networks*, vol. 9, no. 3, pp. 155–164, 2002.

[45] S. Siddiqi, G. Sukhatme, and A. Howard, "Experiments in monte-carlo localization using wifi signal strength," *11th Int. Conf. Adv. Robot. Coimbra, Port.*, no. July, 2003.

[46] a.M. Ladd, K. Bekris, G. Marceau, a. Rudys, D. Wallach, and L. Kavraki, "Using wireless Ethernet for localization," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 1, 2002.

[47] Y. Luo, O. Hoeber, and Y. Chen, "Enhancing Wi-Fi fingerprinting for indoor positioning using human-centric collaborative feedback," *Human-centric Comput. Inf. Sci.*, vol. 3, no. 1, p. 2, 2013.

[48] R. Mautz, *Indoor Positioning Technologies Habilitation Thesis*. PhD thesis, 2012.

[49] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges," 2017.

[50] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, "Locating user equipments and access points using RSSI fingerprints: A Gaussian process approach," in *2016 IEEE Int. Conf. Commun. ICC 2016*, 2016.

[51] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," *Proc. 3rd Int. Conf. Mob. Syst. Appl. Serv. - MobiSys '05*, p. 205, 2005.

[52] A. K. M. M. Hossain and W.-s. S. Soh, "A Survey of Calibration-free Indoor Positioning Systems," *Comput. Commun.*, pp. 1–18, 2015.

[53] S. He and S.-H. H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[54] M. Paciga and H. Lutfiyya, "Herecast : An Open Infrastructure for Location- Based Services Using WiFi," *Access*, vol. 9, 2004.

[55] P. Bolliger, "Redpin-adaptive, zero-configuration indoor localization through user collaboration," *… Work. Mob. entity localization Track. …*, p. 55, 2008.

[56] a. K. M. Mahtab Hossain, H. Nguyen Van, and W. S. Soh, "Utilization of user feedback in indoor positioning system," *Pervasive Mob. Comput.*, vol. 6, no. 4, pp. 467–481, 2010.

[57] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," *Proc. - IEEE INFOCOM*, pp. 2481–2489, 2013.

[58] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based Data Fusion of Pedometer and WiFi Measurements for Mobile Indoor Positioning," *Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. - UbiComp '14 Adjun.*, pp. 147–158, 2014.

[59] C. G. Pendao, A. C. Moreira, and H. Rodrigues, "Energy consumption in personal mobile devices sensing applications," *2014 7th IFIP Wirel. Mob. Netw. Conf.*, pp. 1–8, 2014.

[60] M. F. Cátedra and J. Perez, *Cell planning for wireless communications*. Artech House, Inc., 1999.

[61] D. J. Cichon and T. Kürner, "Digital mobile radio towards future generation systems," tech. rep., Technical report, COST 231, 1999.

[62] S. Zvanovec, "Wireless LAN networks design: site survey or propagation modeling?," *Radioengineering*, pp. 42–49, 2003.

[63] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," *16th Annu. Int. Conf. Mob. Comput. Netw. - (MobiCom '10)*, p. 173, 2010.

[64] H. Lim, L. C. Kung, J. C. Hou, and H. Luo, "Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure," *Wirel. Networks*, vol. 16, no. 2, pp. 405–420, 2010.

[65] A. Goswami, L. E. Ortiz, and S. R. Das, "WiGEM : A Learning-Based Approach for Indoor Localization," *CoNEXT*, 2011.

[66] J. Koo and H. Cha, "Unsupervised locating of WiFi access points using smartphones," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 6, pp. 1341–1353, 2012.

[67] M. Youssef, A. Youssef, C. Park, and C. Rieger, "PinPoint : An Asynchronous Time-Based Location," *MobiSys '06 Proc. 4th Int. Conf. Mob. Syst. Appl. Serv.*, pp. 165–176, 2006.

[68] H. Balakrishnan, T. Supervisor, and A. C. Smith, "The Cricket Indoor Location System," *Architecture*, vol. 16, no. 2001, p. 199, 2005.

[69] J. Krumm and K. Hinckley, "The NearMe Wireless Proximity Server," *Proc. Int. Conf. Ubiquitous Comput. (UbiComp '04)*, pp. 283–300, 2004.

[70] A. Günther and C. Hoene, "Measuring Round Trip Times to Determine the Distance Between WLAN Nodes," in *Proc. 4th IFIP-TC6 Int. Conf. Netw. Technol. Serv. Protoc. Perform. Comput. Commun. Networks; Mob. Wirel. Commun. Syst.*, no. May in NETWORKING'05, (Berlin, Heidelberg), pp. 768–779, Springer-Verlag, 2005.

[71] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng, "I am the antenna: Accurate Outdoor AP Location using Smartphones," in *Proc. 17th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '11*, p. 109, 2011.

[72] R. Peng and M. L. Sichitiu, "Ad Hoc Positioning System (APS) using AoA," in *INFOCOM' 03*, no. November 2015, 2003.

[73] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," *Proc. 20th Int. Jt. Conf. Artifical Intell.*, pp. 2480–2485, 2007.

[74] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor WiFi GraphSLAM," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1038–1043, 2011.

[75] L. Bruno and P. Robertson, "WiSLAM: Improving FootSLAM with WiFi," *2011 Int. Conf. Indoor Position. Indoor Navig. IPIN 2011*, no. September, pp. 21–23, 2011.

[76] J. Koo and H. Cha, "Autonomous construction of a WiFi access point map using multidimensional scaling," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6696 LNCS, pp. 115–132, 2011.

[77] Y. Shang and W. Ruml, "Improved MDS-based localization," in *IEEE INFOCOM 2004*, vol. 4, pp. 2640–2651 vol.4, mar 2004.

[78] J. a. Costa, N. Patwari, and A. O. Hero, "Achieving high-accuracy distributed localization in sensor networks," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. III, no. June, pp. 1–26, 2005.

[79] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction.," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[80] Z. Sun, Z. Zhang, and H. Wang, "Manifold regularization based semisupervised semiparametric regression," *Neurocomputing*, vol. 73, no. 10-12, pp. 2203–2216, 2010.

[81] J. J. Pan and Q. Yang, "Co-Localization from Labeled and Unlabeled Data Using Graph Laplacian," *Network*, pp. 2166–2171, 2005.

[82] J. Pan, Q. Yang, and H. Chang, "A manifold regularization approach to calibration reduction for sensor-network based tracking," *Proc. 21st Nat. Conf. Artif. intell.*, vol. 1, no. c, pp. 988–993, 2006.

[83] A. W. Tsui, Y. H. Chuang, and H. H. Chu, "Unsupervised learning for solving RSS hardware variance problem in WiFi localization," *Mob. Networks Appl.*, vol. 14, no. 5, pp. 677–691, 2009.

[84] J. G. Park, D. Curtis, S. Teller, and J. Ledlie, "Implications of device diversity for organic localization," *Proc. - IEEE INFOCOM*, pp. 3182–3190, 2011.

[85] S. Kim, J. Kim, and D. Han, "Floor Detection Using a Barometer Sensor in a Smartphone," in *2017 Indoor Position. Indoor Navig.*, (Sapporo, Japan), pp. 1–4, 2017.

[86] H. Xia, X. Wang, Y. Qiao, J. Jian, and Y. Chang, "Using multiple barometers to detect the floor location of smart phones with built-in barometric sensors for indoor positioning," *Sensors (Switzerland)*, vol. 15, no. 4, pp. 7857–7877, 2015.

[87] A. R. Sandeep, Y. Shreyas, S. Seth, R. Agarwal, and G. Sadashivappa, "Wireless Network Visualization and Indoor Empirical Propagation Model for a Campus WI-FI Network," in *World Acad. Sci. Eng. Technol.*, pp. 730–734, 2009.

[88] B. Li, a.G. Dempster, C. Rizos, and J. Barnes, "Hybrid method for localization using WLAN," *Spat. Sci. Conf.*, no. August, pp. 12–16, 2005.

[89] Y. Hu, "Efficient, High-Quality Force-Directed Graph Drawing," *Math. J.*, vol. 10, no. 1, pp. 37–71, 2005.

[90] R. Takeda, G. Lisco, T. Fujisawa, L. Gastaldi, H. Tohyama, and S. Tadano, "Drift Removal for Improving the Accuracy of Gait Parameters Using Wearable Sensor Systems," *Sensors*, vol. 14, no. 12, pp. 23230–23247, 2014.

[91] F. F. Baccelli and B. Blaszczyszyn, "Stochastic Geometry and Wireless Networks: Volume I Theory," *Found. Trends Netw.*, vol. 3, pp. 249–449, mar 2009.

[92] F. D. Angelis, I. Habib, P. Altieri, and G. Giambene, "Differentiated QoS provision for multimedia traffic in WiFi systems," in *IEEE Int. Conf. Commun. 2005. ICC 2005. 2005*, vol. 2, pp. 1251–1255 Vol. 2, may 2005.

[93] J. S. Kim, J. S. Shin, S. M. Oh, A. S. Park, and M. Y. Chung, "System coverage and capacity analysis on millimeter-wave band for 5G mobile communication systems with massive antenna structure," *Int. J. Antennas Propag.*, vol. 2014, 2014.