Ricardo Bruno Barbeiro dos Santos

Bachelor Degree in Biomedical Engineering Sciences

Human Crowdsourcing Data for Indoor Location Applied to Ambient Assisted Living Scenarios

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Biomedical Engineering

Adviser: Hugo Filipe Silveira Gamboa, Professor Auxiliar, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa

Examination Committee

Chairperson:	Prof. Dr. Carla Maria Quintão Pereira
Raporteur:	Prof. Dr. Ricardo Nuno Pereira Verga e Afonso Vigário
Member:	Prof. Dr. Hugo Filipe Silveira Gamboa

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

November, 2018

Human Crowdsourcing Data for Indoor Location Applied to Ambient Assisted Living Scenarios

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

The success of my academic journey, especially in the past months with the development of this thesis, would not be possible without the immense support of several people, who always had my back and surely never let me down.

First of all, I would like to express my gratitude to my academic supervisor, Professor Hugo Gamboa, for his advice and guidance through the all project, and especially, for welcoming me at Fraunhofer AICOS. Thank you for introducing me to the amazing world of data science.

I would also like to thank *Associação Fraunhofer Portugal Research*, for giving me the opportunity to work and grow as a professional in an amazing environment, with all the necessary conditions to produce "remarkable technology, easy to use". Among the excelent people I have met, I want to manifest my sincere appreciation to Marília Barandas, one of the persons that most contributed to my success, for her guidance and constant support. I am sure it was not easy to putting up with me in all the endless questions. Thank you! I cannot not leave Ricardo Leonardo out of this list. His help and suggestions during the entire process were also fundamental to the accomplishment of this project. For that, I want to express my gratitude!

I want to give a very special word to the colleagues that accompanied me during this months, in this "thesis life". Mariana, Patrícia, Rui and Helena, thank you for all the moments of fun and focus we have passed, you certainly made this process easier.

The last five years were, with no doubt, the most enjoyable time of my life. It was surely due to all the incredible people that have crossed my path, whom I am lucky enough to call friends. To the people I have met at FCT, and especially to the members of the "sueca" team, thank you for all the great moments, all the laughs and cries, all the parties and study sessions. To the friends that I have made during my journey at Primark, thank you for all the lessons you taught me, I have grown so much while I had so much fun! To the friends that are standing by my side for a number of years that I cannot count anymore, thank you for everything, you mean so much to me. The stories and adventures we have been through are countless, and so much more are yet to come!

Por fim, o meu maior agradecimento é dirigido aos responsáveis pela minha existência e pelo homem que sou hoje. Um muito obrigado à minha família por tudo o que me proporcionou e por todos os ensinamentos que me fizeram chegar a este momento, o culminar de toda a minha jornada académica. Obrigado aos meus pais e à minha irmã por terem estado sempre a

meu lado, por me terem apoiado, dado força e acreditado de que eu era capaz de aqui chegar. Nem sempre foi fácil, mas tudo é possível quando se acredita. Obrigado às minhas avós pela sabedoria que me transmitiram, pela inspiração que são, e pelo carinho que me deram. Ao meu avô, que não teve oportunidade de me ver chegar ao fim deste capítulo, obrigado por tudo, as saudades já apertam. Um muito obrigado final a todos, por nunca terem duvidado de mim e das minhas capacidades, tendo sempre mostrado um enorme orgulho em mim. Sem vocês tudo o que alcancei não seria possível. Espero poder sempre retribuir o vosso amor. Muito obrigado por tudo!

Abstract

In the last decades, the rise of life expectancy has accelerated the demand for new technological solutions to provide a longer life with improved quality. One of the major areas of the Ambient Assisted Living aims to monitor the elderly location indoors. For this purpose, indoor positioning systems are valuable tools and can be classified depending on the need of a supporting infrastructure. Infrastructure-based systems require the investment on expensive equipment and existing infrastructure-free systems, although rely on the pervasively available characteristics of the buildings, present some limitations regarding the extensive process of acquiring and maintaining fingerprints, the maps that store the environmental characteristics to be used in the localisation phase. These problems hinder indoor positioning systems to be deployed in most scenarios.

To overcome these limitations, an algorithm for the automatic construction of indoor floor plans and environmental fingerprints is proposed. With the use of crowdsourcing techniques, where the extensiveness of a task is reduced with the help of a large undefined group of users, the algorithm relies on the combination of multiple sources of information, collected in a non-annotated way by common smartphones. The crowdsourced data is composed by inertial sensors, responsible for estimating the users' trajectories, Wi-Fi radio and magnetic field signals. Wi-Fi radio data is used to cluster the trajectories into smaller groups, each corresponding to specific areas of the building. Distance metrics applied to magnetic field signals are used to identify geomagnetic similarities between different users' trajectories. The building's floor plan is then automatically created, which results in fingerprints labelled with physical locations.

Experimental results show that the proposed algorithm achieved comparable floor plan and fingerprints to those acquired manually, allowing the conclusion that is possible to automate the setup process of infrastructure-free systems. With these results, this solution can be applied in any fingerprinting-based indoor positioning system.

Keywords: Ambient Assisted Living, Crowdsourcing, Indoor Location, Fingerprinting, Time Series Similarities, Machine Learning

Resumo

Nas últimas décadas, o aumento da esperança média de vida tem acelerado a procura por novas soluções tecnológicas capazes de fornecer, durante mais tempo, uma vida com qualidade. Uma das maiores áreas da Assistência à Autonomia no Domicílio procura monitorizar a localização dos idosos dentro de edifícios. Para tal, os sistemas de posicionamento *indoor* são ferramentas valiosas, podendo ser classificados conforme a necessidade de uma infraestrutura de suporte. Os sistemas dependentes de infraestrutura requerem um investimento em equipamento dispendioso. Já os sistemas que não necessitam de uma infraestrutura de suporte, ainda que utilizem as características omnipresentes dos edifícios, apresentam algumas limitações relacionadas com o processo de aquisição e manutenção de *fingerprints*, os mapas que guardam as características ambientais a ser usadas no processo de localização. Estes problemas impedem que os sistemas de posicionamento *indoor* sejam aplicados na maioria dos cenários.

De modo a ultrapassar estas limitações, um algoritmo para a construção automática de plantas de edifícios e *fingerprints* ambientais é proposto. Com o uso de técnicas de *crowdsourcing*, onde a extensividade de uma tarefa é reduzida com o apoio de um amplo grupo indefinido de utilizadores, o algoritmo combina múltiplas fontes de informação, adquiridas de forma não anotada por *smartphones* comuns. Os dados recolhidos por *crowdsourcing* são compostos por sensores inerciais, responsáveis por estimar as trajetórias dos utilizadores, leituras de Wi-Fi e sinais de campo magnético. Os dados de Wi-Fi são utilizados para agrupar as trajetórias em pequenos grupos, em que cada um corresponde a uma zona especifica do edifício. Métricas de distância aplicadas aos sinais de campo magnético permitem identificar similaridades geomagnéticas entre as trajetórias de diferentes utilizadores. Posteriormente, a planta do edifício é automaticamente construída, resultando em *fingerprints* rotuladas com localizações físicas.

Resultados experimentais mostram que o algoritmo proposto alcançou, para o mesmo edifício, a planta e as *fingerprints* comparáveis com as adquiridas manualmente, permitindo concluir que é possível automatizar o processo de configuração dos sistemas livres de infraestrutura. Com estes resultados, esta solução pode ser aplicada a qualquer sistema de posicionamento *indoor* baseado em *fingerprinting*.

Palavras-chave: Assistência à Autonomia no Domicílio, *Crowdsourcing*, Localização *Indoor*, *Fingerprinting*, Similaridade de Séries Temporais, *Machine Learning*

Contents

List of Figures

List of Tables

Acronyms

AAL Ambient Assisted Living.

AP Access Point.

BLE Bluetooth Low Energy.

BSSID Basic Service Set Identifier.

DBSCAN Density-Based Spatial Clustering of Applications with Noise.

DR Dead Reckoning.

DTW Dynamic Time Warping.

GPS Global Positioning System.

HAR Human Activity Recognition.

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise.

IMU Inertial Measurement Unit.

IPS Indoor Positioning Systems.

K-NN K-Nearest Neighbours.

MAC address Media Access Control address.

MDS Multidimensional Scaling.

PDR Pedestrian Dead Reckoning.

PIL Precise Indoor Location.

PoC Proof of Concept.

RSS Received Signal Strength.

RSSI Received Signal Strength Indicator.

SLAM Simultaneous Localisation and Mapping.

WAP Wireless Access Point.

WLAN Wireless Local Area Network.

WPA Windowed P-norm Alignment.

Chapter 1

Introduction

1.1 Motivation

The scientific development that has taken place in the last decades has caused enormous changes in human being's quality of life. Improvements, especially in healthcare, have led to a rise of life expectancy. According to Eurostat[1], over the period from 1980 to 2016, the average life expectancy of a Portuguese citizen increased almost 10 years, and is expected to increase further (see Figure 1.1).

Given this, the age structure of the population is changing. While the proportion of the working age population is decreasing, the proportion relative to the elderly is taking the opposite course (see Figure 1.2). Considering this, it is imperative to build a sustainable society, where health and social care costs are supportable, while elderly people quality of life is enhanced.

Ambient Assisted Living (AAL) is a concept created in this line of thought. It comprises all the research done to create innovative technological solutions to provide better life conditions to the older adults, allowing them to live longer in their preferred environment, independently and safely [1, 2].

One of the key tasks of AAL is to monitor the elderly at their homes, either by keeping track on their daily activities, or by knowing their location. An intelligent system installed in every elderly's home would, for example, identify if the medicines were taken or forgotten, giving an alert if necessary. The monitoring of the number of hours that an elderly slept or the number of times that went to the bathroom, by knowing their positions in their house through the day, could help on understanding if everything is fine.

Positioning systems are thus valuable tools, not only for the elderly, but also for their families and healthcare professionals. These systems can help on ensuring the safety of

[1] Eurostat is the statistical office of the European Union.

the elderly in their homes, providing a fast response in an emergency situation, or even to help them reach some destination, as in a hospital or a supermarket, large buildings where an elderly could easily get lost and disoriented.

While Global Positioning System (GPS) is the standard solution for outdoor positioning and navigation, its precision is compromised indoors by the presence of walls and ceilings, which attenuate the received signal. For this reason, Indoor Positioning Systems (IPS) rely on alternative sources of information to provide location-dependent services indoors, such as the buildings' Wi-Fi radio signal distribution or its magnetic field behaviour.

The ubiquity of the smartphones enhances the dissemination of IPS. However, the need for some of these systems to have the updated building data hinders IPS to be deployed in most buildings, since the data collection process brings intensive costs on manpower and time. Therefore, it is imperative to improve this process. In the last years, crowdsourcing is being used by organisations to transfer the execution of a task or a process to an undefined group of anonymous users, lowering its costs and increasing the quickness of its conclusion.

With the application of this model, it is expected that the use of crowdsourced data will help on diminishing the time-consuming process of implementation, improving the scalability of current IPS systems.

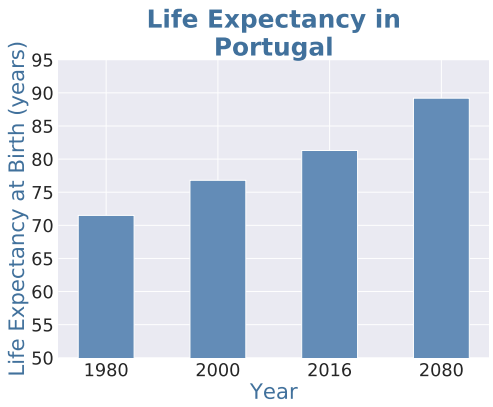


Figure 1.1: Life expectancy in Portugal, with data from 1980 to 2016, and projections for 2080. Data source: [3, 4].

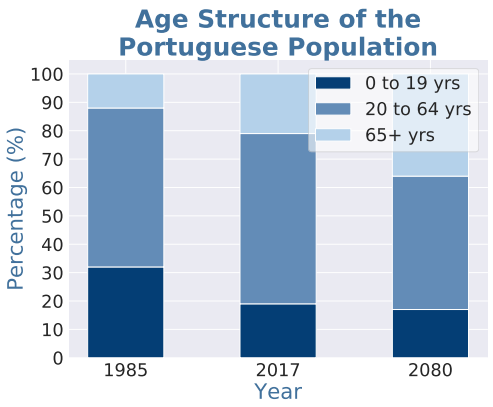


Figure 1.2: Age structure of the Portuguese population, with data of 1985, 2017, and projections for 2080. Data source: [5, 6].

1.2 Objectives

The main purpose of this thesis is to potentiate existing indoor location solutions with the use of crowdsourcing, by diminishing the expensive process of the setup phase, a prerequisite of typical infrastructure-free solutions. Currently, the first step of the setup

phase consists on the upload of the buildings' floor plan, which might need to be constructed, if not available. The second phase includes the process of collecting fingerprints, the maps of the characteristics of the buildings, which are used as reference to provide the localisation. The setup phase is commonly pointed as a great disadvantage of typical infrastructure-free solutions, for being very extensive and expensive, since it requires an expert to walk through the whole building to collect the necessary data [7].

To overcome these limitations, the main goal of this work is to develop an algorithm that automatically constructs indoor floor plans and collects environmental fingerprints with crowdsourced data. This process will be done without any explicit effort from the users, since data will be collected during their natural moving inside buildings.

1.3 Applications

The developed algorithm will extend the range of applications for IPS. Besides the aforementioned applications for AAL, there are several services that can be improved with indoor location. If applied in a hospital, these systems will be useful not only to locate medical equipment when lost, but also to call the nearest doctors in emergency situations [8]. Another area that will thrive with IPS is the retail. In a mall or a supermarket, for example, indoor location services can help consumers optimise their time, if the ideal route for their shopping list is automatically prepared. For retailers, the possibility to know the behaviour of the consumers inside their stores allows the development of new sales techniques, improving their results [9]. Indoor location will also be helpful in smart homes, where energy savings can be achieved if the location of their inhabitants is known [10]. In large buildings, as conference centres, IPS can help a visitor to navigate to a desired area, or simply to meet a friend in an unknown location. Beyond these examples, there are several other areas that will benefit with indoor location, thus inspiring the development of easily deployable solutions.

1.4 Literature Review

In the last two decades, the effort from the scientific community to develop a robust and precise indoor location system resulted in an large number of solutions. However, the limitations that these solutions present leave open the search for a system reliable enough to be widely marketed.

Although GPS signal is not available indoors, several other sources can be used, by taking advantage of the wide range of sensors commonly present in modern smartphones [11] or smartwatches [12], to help in the indoor localisation processes.

Current IPS can be divided in two types, depending on the need of a supporting infrastructure. Infrastructure-based systems normally rely on beacons, which are small devices that transmit a specific signal to the device to be located. The most common source of information used with beacons is Bluetooth [13–15]. However, other sources

can be used, namely ultrasounds [16] or light, either in infrared spectrum [17, 18] or in visible light [19], among others. In contrast, infrastructure-free systems use opportunistic readings from signals that are pervasively available in the majority of environments, like magnetic field [20], atmospheric pressure [7, 21], ambient light [22] or sound [23, 24]. Wi-Fi radio signal readings are also used in this type of system [7, 25–27], even though they are emitted by an Access Point (AP), which can be considered a type of beacon. However, APs are widely present in most of nowadays' buildings. Both systems can also rely on inertial data collected from the user's device's sensors [28, 29], as the accelerometer, the magnetometer and the gyroscope, to characterise the human movement by dead reckoning techniques.

While infrastructure-based systems usually do the localisation process through methods such as trilateralation [13] or triangulation [25], infrastructure-free systems are commonly based on fingerprinting [7, 30], which consist on maps with the characteristics measured on the buildings.

Precise Indoor Location (PIL) is an IPS developed by researchers at Fraunhofer AICOS [7], that runs on a common smartphone, as it can be seen in Figure 1.3. This infrastructure-free system relies on smartphone's inertial sensors to track users' movement in a building. Since these sensors accumulate error, PIL system uses fingerprints of Wi-Fi radio and magnetic field to provide more accurate results. With a particle filter that expands while a user walks, the most probable positions are chosen with great certainty. The fingerprints used in PIL are previously acquired, where a user walks through defined routes with a smartphone that collects the required data. Besides this, the floor plan of the building needs to be uploaded to the server, so the particle filter can be constrained and the fingerprints can be matched to absolute positions. Even though this system can use Google Maps' indoor floor plans automatically, the number of available buildings is very limited, making the construction of the floor plans an indispensable process.

Figure 1.3: PIL solution running on a smartphone. © Fraunhofer Portugal Research

Thus, both types of IPS have limitations. On the one hand, infrastructure-based systems require the installation and maintenance of beacons, which make these solutions more expensive and less appealing to be applied in large scale. On the other hand, infrastructure-free systems based on fingerprinting require an extensive process to collect and update the fingerprints, mainly in large buildings, which increases the cost of these systems too. To overcome current IPS limitations, crowdsourcing is suggested as a way to create self-sustaining systems, which do not require almost any specific human

intervention. There are already some solutions that use crowdsourced data to collect fingerprints, but also to build the floor plans of the buildings, increasing the spectrum of buildings where indoor location can be applied, to any unknown building.

1.4.1 Crowdsourced Fingerprints

The process of building fingerprints is known as the training or offline phase of any infrastructure-free indoor location system. It consists on the collection of the desired data to be used in the operating or online phase, to provide the location of the user. The larger the building, the more labour-intensive and time-consuming this process becomes, greatly increasing the costs of these systems. Thus, the use of crowdsourcing to automatically build fingerprints stands as a viable solution to solve this extensive and expensive process, where some solutions have been already developed.

One of the most well known solution belongs to Rai et al. [31], who developed *Zee*, an algorithm that collects data from smartphones' inertial sensors to track users' paths in indoor environments, while simultaneously performs Wi-Fi scans. The algorithm then combines the inferred trajectories with the constrains of the floor plan given as input, such as walls and other barriers, to obtain the most probable last position of each trajectory. Finally, the remaining positions of each trajectory are obtained with a backward propagation algorithm and the Wi-Fi fingerprints are obtained.

Wu et al. [32] designed *LiFS*, a complete indoor location system that relies only on the crowdsourced collection of Wi-Fi radio data and step detection to build fingerprints. Instead of using the traditional floor plan in two dimensions, the authors apply Multidimensional Scaling (MDS) to obtain a multidimensional floor plan that stores the distances between every pair of interest locations. Then, with the application of the same principle to obtain multidimensional fingerprints, each reading is then attributed to a real physical location by corresponding both the new floor plan and the multidimensional fingerprint.

In the work of Niu et al. [33], WicLoc was created, an indoor location system that uses Wi-Fi radio data and inertial tracking to detect steps and turns. They identify similarities between collected signals by computing a distance matrix of fingerprints. It consists on the differences of the Received Signal Strength Indicator (RSSI) between every pair of APs. Then, the distance matrix is converted to the multidimensional space by a modified MDS algorithm. Finally, the absolute positions of the fingerprints on the floor plans are obtained by comparing the new distance matrix to a set of previously defined anchor points.

Recently, Chen et al. [34] proposed *UILoc*, an unsupervised IPS that builds Wi-Fi fingerprints. It uses dead reckoning techniques to track users trajectories, and has a particle filter to correct the steps length and heading bias. The system requires the installation of Bluetooth Low Energy (BLE) beacons to serve as landmarks, so errors can be fixed. With the obtained position, the fingerprints are built. Although *UILoc* avoids the setup phase, the need of an infrastructure might hinder this system to be widely

deployed.

1.4.2 Crowdsourced Floor Plans

The floor plan of a building is required to most of infrastructure-free indoor location systems, not only to obtain the fingerprints over an absolute reference, in the case of infrastructure-free systems. A floor plan is also needed to allow the navigation process, so the system can drive the user to a destination, considering the constrains of the building. However, floor plans might not be always accessible, or may not even exist. Thus, the process of constructing floor plans is often necessary. However, it is also an extensive and time-consuming process, especially for large buildings, which may be unaffordable. As a result, new techniques for automatic floor plans construction have been developed exploiting the use of crowdsourcing.

CrowdInside was designed by Alzantot and Youssef [35]. It is an automatic floor plan construction system that leverages smartphones' inertial sensors to reconstruct users' trajectories. To avoid the error accumulation of noisy sensors, this system identifies anchor points, which correspond to unique sensor signatures, as when a user passes by an elevator or by stairs. With the information of the initial position, which corresponds to the last known GPS coordinate, all trajectories are matched and corrected with the anchors, and at last the floor plan is obtained.

Shen et al. [36] developed *Walkie-Markie*, an algorithm that uses dead-reckoning to recognise users' movement and relies on buildings' Wi-Fi infrastructure. The algorithm identifies *WiFi-Marks*, locations at which the trend of the Received Signal Strength (RSS) of an AP reverses. Given the initial position of each reconstructed trajectory, the system assigns absolute positions to every *WiFi-Mark*. Finally, by identifying similarities between them, trajectory errors are corrected and the final floor plan is obtained by merging all trajectories.

1.4.3 Crowdsourced Fingerprints and Floor Plans

To be deployed in large scale, such as in every elderly's residence or healthcare facility, an IPS needs to be cheap to install and to maintain, without compromising its results. In order to achieve this, the scientific community has developed, in the last years, new autonomous systems that do not require any specific human intervention, allowing its application in any unknown building. New solutions have been published regarding the automatic construction of floor plans and the simultaneous mapping of fingerprints, relying only on crowdsourced data.

In the work developed by Shin et al. [37], *SmartSLAM* is a new solution to construct automatic indoor floor plans and Wi-Fi radio fingerprints. They use a variant of Simultaneous Localisation and Mapping (SLAM), a technique commonly used in robotics, where the location is provided by dead reckoning techniques and the mapping of unknown areas is done at the same time. Since smartphones' noisy sensors accumulate too much

error, the authors implemented a particle filter to weight the possible positions. With the Wi-Fi readings obtained through the users' trajectories, the floor plans are built, along with the radio fingerprints.

Leveraging the use of the magnetic field interferences, instead of the buildings' radio data, H. Luo et al. [38] designed a solution that uses crowdsourcing to build automatic floor plans and magnetic field fingerprints. They apply hierarchical clustering to the process, where the first step consists on separating the routes by straight segments. Then, these segments are clustered by their length, the average of the absolute heading and finally their similarity on the magnetic field behaviour. Finally, the floor plan is constructed by merging routes with identified similarities and then the magnetic fingerprint is obtained.

PiLoc is an IPS developed by C. Lou et al. [39]. It also uses inertial data to track the users and Wi-Fi radio data to construct the floor plans and fingerprints. Their system consists on clustering all trajectories by their AP coherence, that is, joining in the same cluster all the trajectories that have APs in common. With this, the authors separate the crowdsourced data by floor, since normally an AP can't be read in different floors, due to construction materials attenuation. Then, the algorithm segments the trajectories by curves and straight lines with a minimum length and compares their inertial behaviour and Wi-Fi data trends to identify similarities. To obtain the final floor plans and fingerprints, all trajectories are merged. This systems also allows the automatic update of the obtained floor plans and fingerprints.

Lastly, Wang et al. [40] created *UnLoc*, a complete IPS that has the same principle as *CrowdInside*, since it relies on the landmarks of a building, locations with specific signatures. These landmarks can be previously defined (seeded) if the floor plan is available, or can be organic if identified by the algorithm. Inertial Wi-Fi radio and magnetic field data is compared to identify patterns. At the same time, users' trajectories are reconstructed using dead reckoning techniques, with the definition of the last known location, which corresponds to the last known GPS coordinate. If the patterns are coherent in terms of location, even with some accepted error, they become a landmark. In the localisation phase, those landmarks are used to correct dead reckoning errors to provide an accurate absolute position.

1.4.4 Discussion

Even with this large number of solutions that apply crowdsourcing to indoor location, the market is still waiting for the announcement of a solution that is robust enough to satisfy all the needs.

While some solutions require the input of the floor plan to automatically build fingerprints, others only are able to construct floor plans, leaving the problem of the extensive process of mapping fingerprints unsolved.

Regarding the solutions that can build fingerprints while constructing floor plans,

some limitations still discourage the investment on their application worldwide. In *SmartSLAM* [37], the way that the particle filter is developed to reconstruct the movement, only allows the system to build maps with the layout of the buildings' main corridors, requiring a further process to explore unmapped areas. The solution presented by H. Luo et al. [38] will have problems if applied in large buildings, since every segmented signal is compared to each other, rising the processing costs. Contrarily, *PiLoc* [39] only works for large buildings due to the fact that the system only segments curves with more than 10 steps and straight line paths with more than 30 steps, a number too high for an elderly's home, for example. Finally, *UnLoc* [40] needs at least one seeded landmark to trigger the system, which can be for example the entrance of a building, with a GPS coordinate. The system will start working only when a user passes by this landmark, a requisite that might not be feasible in buildings with multiple entrances, as a hospital.

Thus, the system proposed in this thesis aims to solve these problems, creating an autonomous crowdsourcing solution that relies on inertial, magnetic field and Wi-Fi radio data, without any specific effort from the users. Using information pervasively available in every building, it will be possible to offer a solution with conditions to be widely deployed. Without any restrains concerning the buildings' dimensions, the applicability of this system will go from small environments, such as homes, to big environments, such as hospitals, malls or airports.

1.5 Work Summary

The goal proposed on this thesis will be achieved with the use of diverse techniques. The data acquisition is done by leveraging the ubiquity of smartphones in our daily life. Common smartphones are now equipped with a wide range of sensors that can be used to understand the human motion and behaviour. Crowdsourcing will be applied by collecting huge amounts of non-annotated data, by different users on different smartphones. The automatic construction of floor plans and environmental fingerprints will be accomplished by identifying similarities between the reconstructed routes.

The algorithm will apply different techniques to process different types of information. Inertial sensors provide information about the linear and angular velocity of the users, that is processed to obtain the characterisation of the movement, as the detection of steps and their length and direction computation. Wi-Fi radio data will be processed with unsupervised machine learning techniques, to restrain the area to search for similarities, that will be further discovered by applying time series analysis methods to the magnetic field data. Finally, the algorithm will match all the identified similarities to obtain the final map and the necessary fingerprints.

All the algorithms created in this work will be tested in Fraunhofer AICOS's PIL solution, a fully developed indoor localisation system that has some features to allow the verification of this algorithm's quality.

1.6 Thesis Overview

This thesis is organised in 4 main elements, divided into six chapters and two appendices, as it is outlined in Figure 1.4.

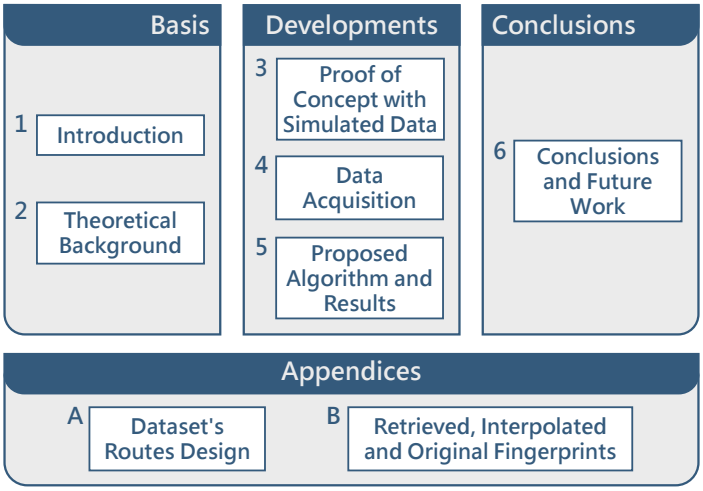


Figure 1.4: Thesis overview scheme.

The basis element is composed by the first two chapters. The current chapter introduces the problem that motivated this research, as well as the literature review about the automatic construction of floor plans and environmental fingerprints topic and the brief explanation of the developed work. Chapter 2 aims to contextualise the reader to the concepts and methods used in this thesis.

The second element of the current work includes three chapters that comprise all the achieved developments. A first approach to test the feasibility of the proposed objectives is explained in the Proof of Concept (PoC) of Chapter 3. After the evaluation of the preliminary results, Chapter 4 describes the process of data collection, that was used in the development of the proposed algorithm of Chapter 5. This chapter not only explains the functioning of the final solution, but also presents the intermediary results, obtained through all the steps of the development, as well as the final results.

Chapter 6 constitutes the third element and makes the final balance of this thesis, where the taken conclusions are discussed and some future guidelines are suggested.

Finally, Appendix A shows the designed routes for the dataset built in this project. Appendix B displays the fingerprints obtained with the developed algorithm, along with their respective interpolations and the originals, collected with the traditional methods, to provide a tool for the results evaluation.

Chapter 2

Theoretical Background

To properly understand this work, the applied concepts are presented in this chapter. Firstly, the Ambient Assisted Living (AAL) concept is introduced, followed by the concepts involving the Human Activity Recognition (HAR). Next, an explanation about fingerprinting-based indoor location is given, including all the types and information sources commonly used. Finally, the signal processing techniques used in this project are described and concepts about machine learning are presented.

2.1 Ambient Assisted Living

AAL encompasses all the technological products and services developed to improve the quality of life of older adults and people with special needs, in all aspects [41]. The main objective of AAL is to increase the safety and autonomy of these people, allowing them to stay at their preferred environment longer, usually their homes [8].

The demand for these technologies emerged due to the changes in world's demography. The rise of the life expectancy in developed countries is changing the age structure of the population. While the number of the elderly population is increasing, the working age population, responsible for sustaining the costs of the healthcare systems, is comparatively declining [1]. This unbalance, along with the fact that an ageing population is more prone to age-related diseases, such as Alzheimer's disease or Parkinson's disease, rises even more the healthcare costs [42].

To overcome these problems, several research support programmes have been created. The Active and Assisted Living programme[1] funds projects in the field of information and communication technologies. This programme is co-founded by the European Commission and 19 other countries of the European Union, with the main goal of encouraging

[1] More information available in http://www.aal-europe.eu/ (visited on 09/17/2018).

researchers to develop new technologies and services for ageing well [2].

AAL technologies are supported by several techniques, that have been developed in the last years. Activity recognition is one of the main tasks in AAL, where several sensors and cameras are combined with intelligent programming, such as machine learning, to monitor elderly people through their day. With this, it is possible to create anomaly detection algorithms, useful to identify problems, as a fall or even a heart attack. Context modelling is another task, which consists on structuring the available information of the users, such as their profile, medical history and data collected from sensors, for example, to be used in several applications. Automatic planning is another important mission, where applications and devices can help older adults to not forget to take their medicines, among other uses. Finally, location services are very useful to track and monitor elderly people outside, where GPS is used, and inside, with the rising of IPS [41, 43].

2.2 Human Activity Recognition

HAR is one of the most important tasks in AAL. It is defined as the capacity of a system to automatically interpret a body gesture or a movement, thus classifying it as an action or activity [44]. This process aims to monitor people in their daily life, while they move naturally. HAR is very challenging since human activities are very diverse and complex, where most of the times different activities have similar movements [45].

There are numerous applications where HAR is useful, especially in AAL. Monitoring the activities performed by an elderly through the day makes it possible to know if everything is fine, without the permanent presence of a healthcare professional or a relative. A system that identifies if the monitored elderly has forgotten to take their medication can be very effective on avoiding further health problems, for example.

In the scope of indoor location, some types of IPS are based on the reconstruction of the trajectory described by the users, so it is important to recognise not only their movement, but also its characteristics. To achieve this, HAR techniques are employed and some specific sensors are used to collect the necessary data.

2.2.1 Sensors

The development that happened in the last decades in microelectronics and computer systems allowed the creation of a wide range of sensors. They are now small, low-cost and come with a high computational power. With all these characteristics, sensors are being used to collect pervasive data that is used to recognise human activities [46].

To address this issue, sensors are mainly deployed in two ways. External sensors are fixed in interest locations, where the user is supposed to interact with. Smart homes are based in this type of sensors, where actions are triggered when the user passes by, in the case of a motion detector, or touches it, in the case of a faucet. Cameras, normally used in

surveillance systems, are also considered a type of external sensor, since the recognition of gestures and actions can also be identified by the image processing [46].

Contrarily, wearable sensors are attached to the user, measuring continuously the interest data [46]. These sensors are usually available in devices that we use everyday, as in smartphones, smartwatches or smart clothes. This last type is commonly used to monitor physiological conditions in elderly people and athletes [47]. Wearable sensors can measure various types of data, normally related to the user's movement, as the acceleration, physiological signals, as the heart rate, or user's environmental conditions, as the temperature or air pressure [46].

Inertial sensors are the ones that allow the reconstruction of users' trajectories, being for this reason very important for this work. These sensors are available in almost every smartphone in an Inertial Measurement Unit (IMU). IMUs are electronic devices composed by an accelerometer, a gyroscope and usually a magnetometer, the necessary sensors to reconstruct the users' movement.

- **Accelerometer**: This sensor aims to measure the device's linear acceleration relative to a fixed referential, the Earth's. IMUs accelerometers usually sense data along three axes: x (lateral), y (vertical) and z (longitudinal), as it can be seen in Figure 2.1. The measured acceleration is frequently represented using SI units, meters per second squared (m/s^2), but some devices measure in g-force units (g). The measured signal has two components, one is static, and is caused by the Earth's gravitational acceleration, and the other is dynamic, caused by the device's movement. This last component allows the identification if a user is standing or moving, as well as his/her linear velocity.

- **Gyroscope**: This sensor is used to measure the angular velocity of a device, and can be used to compute its relative orientation to a previous instant. The SI units for the angular velocity are radians per second (rad/s). IMUs gyroscopes are normally three dimensional too, with the same axes as the accelerometer. After the integration of gyroscope data, (pitch, roll, yaw) are obtained, which represent the rotation angles around the (x, y, z) axes, respectively (Figure 2.1). Thus, it is possible to identify if a user is moving forward or changing direction.

- **Magnetometer**: This last inertial sensor measures the intensity and the direction of the Earth's local magnetic field. With this data it is possible to obtain the absolute orientation of the device relatively to the North Pole. They also collect data in three dimensions, being usually represented in microtesla (μT) units. However, magnetometers are also used as environmental sensors, since the read magnetic field is often influenced by buildings' ferromagnetic construction materials and electrical equipment. The interference might be a singular characteristic of a specific building, as it is explained in Section 2.3.2.2.

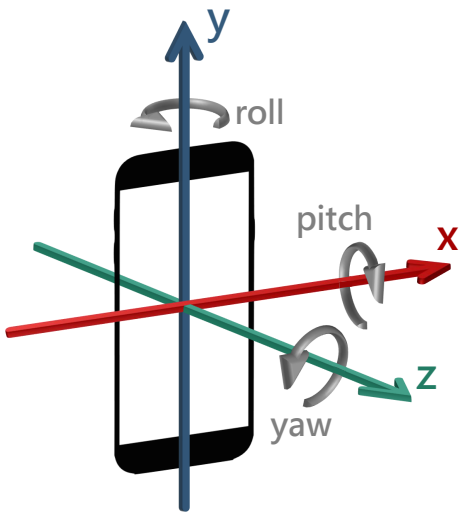


Figure 2.1: Representation of the smartphone IMU axes and rotation angles.

Leveraging the ubiquity of smartphones in our daily life, IPS are normally deployed in these devices. The smartphone is thus responsible for collecting the necessary data to reconstruct the movement, through the built-in IMU. Then, the information is processed locally or remotely, and the user's position is displayed on an interface, where he/she can interact. Considering the role of smartphones for indoor location, they will be used in this work to collect data from the IMU, to reconstruct the trajectories developed by the users.

2.2.2 Pedestrian Dead Reckoning

Dead Reckoning (DR) is a technique used mainly in navigation to infer a trajectory or a path described by a person or a vehicle, but also to predict their next positions. DR has many applications, as in marine or air navigation. The goal is to determine the positions by which some ship or aircraft will pass over time, considering their current speed and direction [48].

Pedestrian Dead Reckoning (PDR) is one of the most recent applications of DR, where this technique estimates the successive positions of a person, from a known initial point, by estimating his/her travelled distance and the direction of the movement [49]. To reconstruct the movement, data collected from the IMU sensors is processed.

The first phase consists on detecting the moments where the user took a step, as well as their corresponding length. To do this, the linear acceleration retrieved from the accelerometer is used. Theoretically, the integration of the linear acceleration should be enough to obtain the user's velocity and travelled distance. However, the error of the low-cost accelerometers in IMUs prevents an accurate estimation [50]. Thus, the analysis of the human gait movement allowed the understanding that this cyclic pattern is well reproduced in the magnitude of the accelerometer signal [7]. A step happens in the moment when a person touches the floor, which causes a vertical peak on the acceleration.

Thus, the identification of this peak allows the step moments detection.

After the step detection, the step length can determined by several techniques. The constant model [39] considers that the step length is approximately always the same, considering or not an error for the calculations. The linear model [51] determines the step length by the following equation:

$$Step\ Length = A + B \cdot Freq + C \cdot Var + w \tag{2.1}$$

where $Freq$ is the step frequency and Var the step acceleration variance. A, B and C are linearly regressive parameters, previously determined in the training phase, and w the added Gaussian noise. The non-linear model [7] consists on an empirical relation between the peaks of the vertical acceleration and is described by the following equation:

$$Step\ Length = K \cdot \sqrt[4]{A_{max} - A_{min}} \tag{2.2}$$

where A_{max} and A_{min} are the maximum and minimum vertical accelerations of a step, respectively. K is a calibration constant, recursively adjusted by the least-squares method. The non-linear model is the most used since it only requires one estimated parameter, facilitating its use.

The second phase of PDR consists on determining the direction of every step. Two approaches can be considered. The first uses the magnetometer signal to identify the absolute orientation of the device to the North Pole [50]. However, this method might fail indoors, since the interferences that affect the magnetic field deviate the North estimation, as it is explained in Section 2.3.2.2.

The second approach computes the changes of direction between steps, using the vertical axis of the gyroscope in the Earth frame. The conversion of the device to the Earth frame is known as sensor fusion [7]. It consists on identifying the long-term directions of the gravitational acceleration and the North Pole, provided by the accelerometer and the magnetometer, respectively. With this information, the coordinates of the device are transposed, to align the device to the Earth, which is independent from the device orientation on the body. This allows the identification of the variations on the trajectory's direction, although with some error. The short-term accuracy of the gyroscope causes the inferred trajectory to drift, producing a cumulative error as it can be seen in Figure 2.2. To solve this problem, Guimarães et al. [7] eliminate small direction variations, based on the principle that a user tends to walk straight in the corridors of a building.

In summary, PDR techniques rely on the data retrieved from the IMU's sensors to provide the length and the direction of each step. It allows the reconstruction of the trajectories described by the users, although with some faults, mainly due to the smartphones' noisy sensors.

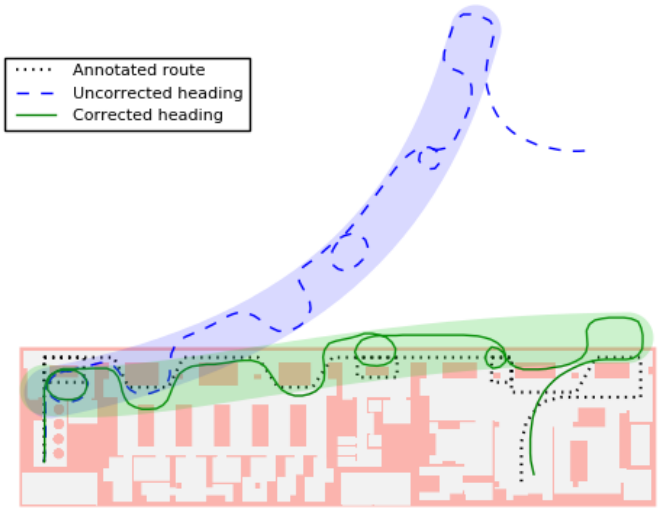


Figure 2.2: Illustration of the gyroscope's drift problem, where the inferred direction is traced in blue. The algorithm of Guimarães et al. [7] corrects the drift, obtaining a more accurate heading, traced in green. The pointed curve represents the original route. Image retrieved from [7].

2.3 Indoor Location

Indoor location is an active research area that comprises all the systems created to provide localisation-based services indoors, either to locate people or objects, where GPS signal is unavailable. To face the complexity of this area, numerous solutions that use different approaches have been unveiled. Indoor location systems can be classified in two categories, according to the type and the source of the information required to be processed.

Infrastructure-based indoor location systems depend on previously installed beacons through the buildings, that function as a source of signal that will be interpreted by the devices to be located or vice-versa. Contrarily, infrastructure-free systems rely on information pervasively available in the buildings that is collected and processed by the same devices. This thesis aims to make improvements in fingerprinting-based solutions, a type of systems of infrastructure-free Indoor Positioning Systems (IPS), as it will be further explained.

2.3.1 Fingerprinting-Based Solutions

Fingerprinting is one of the most applied methods to provide infrastructure-free indoor localisation. These solutions rely on fingerprints, which are floor plans that store the characteristics of the buildings. These characteristics depend on the formulation of the solution and consist on the pervasive signals available in each building. Figure 2.3 shows a fingerprint of the vertical component of the magnetic field for an office building. A fingerprint is relative to a specific location, where each coordinate of the two dimensional

Cartesian plane refers to the same coordinates of the respective building, being meaningless if applied in another. For this reason, fingerprinting-based solutions must identify the building *a priori*, so the correspondent fingerprints can be loaded [30].

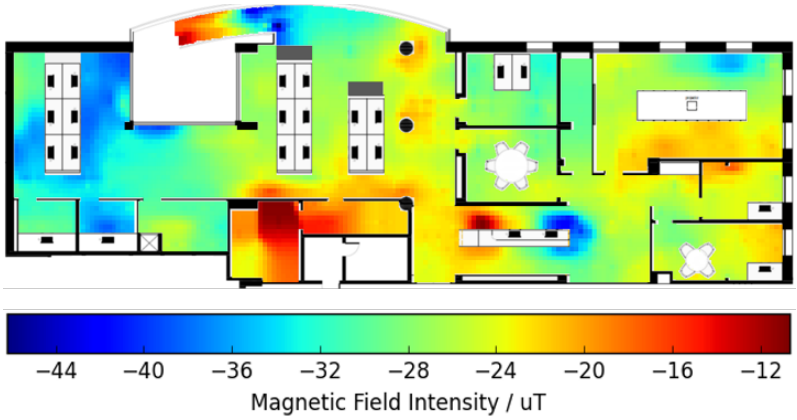


Figure 2.3: Fingerprint of the vertical component of magnetic field for an office building. © Fraunhofer Portugal Research

The fingerprints of each building must be collected before the launch of the system. The process of collecting fingerprints consists on acquiring data from the necessary sources of information with a smartphone, for example. Passing by every location of the building, the fingerprints are progressively mapped. In large buildings, this process becomes very extensive, rising the costs of IPS. In an attempt to overcome this problem, Guimarães et al. [7] developed a system that is capable of compute the fingerprints of unmapped positions, according to the distance of the nearby mapped positions. This way, the necessary coverage of the map is diminished, easing the collection process.

The location is usually obtained by the following process. At an unknown location, when the application running on the device to be located sends a localisation request, the system will record the fingerprint at that location, by registering the data coming from the desired sensors. If the system runs online, the data will then be sent to the server to be compared to every position of the complete fingerprints in the database. The difference of an offline system is that it has the building's fingerprints stored in the device's memory. Then, the most similar position in the fingerprint is the most likely location of the device. However, the variability of the data, even in different times of the day, often affects the localisation process, returning a wrong location. Thus, improvements have been made to the existing solutions to overcome these problems.

To improve the location estimation, Viel and Asplund [30] applied an unsupervised machine learning technique (Section 2.6), namely the K-Nearest Neighbours algorithm, to help on deciding the most probable location, among the K smallest distances. Some more advanced solutions employ PDR techniques to improve the classic fingerprinting-based systems [7, 52]. These improved systems are able to track users through their trajectories, with the identification of the movement's characteristics. However, as it was explained in Subsection 2.2.2, the reconstruction of the trajectories is subject to error

accumulation, making solely dead reckoning solutions very inaccurate. When PDR and fingerprint-based systems are employed together, it is possible to improve the reliability of infrastructure-free IPS. This fused mechanism often works resorting to a particle filter.

A particle filter is a technique that implements a Bayesian filter using a Sequential Monte Carlo method [53], on which a set of random particles are distributed through the building. A particle is defined by a probability of a certain position to be the real position of the device. The probability of every particle is initially the same, but each one is updated and propagated to its neighbouring positions over time, depending on the readings of the environment and the movement of the user. Weighting every available information, the probabilities of some particles increase, meaning that they are possibly close to the real location, while the probabilities of others decrease to the point that they do not make sense anymore. Some solutions, like the one developed by Guimarães et al. [7], are able to estimate the area where the user initially is, on which the particles are distributed. Then, their propagation considers the progression of the user's movement and their susceptibility to error, as well as the readings from the environmental data. Solutions that use particle filtering allow the navigation in a building, instead of providing a singular response to a localisation request.

2.3.2 Fingerprinting Sources

Several information sources can be employed to IPS. Although the use of many information sources heightens the complexity of these systems, it also improves their accuracy. Thus, a trade-off between the complexity and the accuracy must be carefully planed. The information of the environmental magnetic field and Wi-Fi radio data, available on almost every building, are believed to be very useful [7, 26, 28]. Therefore, these two sources will be used in this work.

2.3.2.1 Wi-Fi Radio

The use of Wi-Fi radio as a pervasive signal is done due to the great availability of Wi-Fi networks in most of nowadays buildings, along with the fact that almost every mobile device has a built-in Wi-Fi receiving module, especially in smartphones.

The basis of every Wi-Fi network is a Wireless Access Point (WAP), or commonly named just as Access Point (AP). APs are the devices that allow the connection of a Wi-Fi device to a wired network, typically the Ethernet, by projecting its signal wirelessly. Fingerprinting often uses two main features of Wi-Fi networks: the name that characterises every AP and its perceived signal strength.

Every AP is characterised by a unique name, the Basic Service Set Identifier (BSSID). This unequivocal designation is given by the Media Access Control address (MAC address) of the equipment, a 48-bit sequence with 12 hexadecimal numbers. The first six numbers identify the manufacturer and are established by the Institute of Electrical and Electronics Engineers (IEEE). The last six are attributed by the manufacturer to every equipment,

under the constrains of uniqueness. Besides this, an AP often is capable to transmit in two radio bands, the 2.4 GHz and the 5 GHz. For each radio band, depending on the AP, several Wireless Local Area Network (WLAN) can be configured. This happens very frequently when a network provider, as an university, wants to offer different network services to different users, as for students and teachers. Every WLAN is identified by a also unique name, commonly given by a variation of the last character of the AP's original BSSID. Thus, for localisation purposes, since an AP can be characterised without any ambiguity, it is possible to identify the specific area where a device is by identifying the MAC address of the AP that is transmitting the collected Wi-Fi radio signal.

The characteristics and the behaviour of the Wi-Fi radio signal are also useful for indoor localisation. This signal is transmitted by the APs' antennas and is expressed in milliwatt (mW) units. However, since the Wi-Fi signal has a low transmit power, the Received Signal Strength (RSS) values in mW would be too difficult to interpret, due to the high number of decimal places. Thus, the signal values are converted, by the Wi-Fi devices, to more intuitive forms, expressed by the Received Signal Strength Indicator (RSSI) values. The RSSI can be represented as a dimensionless measure that translates the relation between the strength power and a value, commonly from 0 to 60, 100 or 255, depending on the manufacturer [54]. However, most systems express the Wi-Fi RSS in decibel milliwatt (dBm) units, as the Android operative system [55]. dBm is the absolute signal power over a logarithmic relation, being expressed in decibels (dB) with reference to 1 mW. This relation is obtained by the following equation:

$$P_{dBm} = 10 \log_{10}\left(\frac{P}{1mW}\right) \tag{2.3}$$

The values of dBm tipically vary between 0 and approximately -100. The higher the value, the better the quality of the signal. Since this representation comes in a logarithmic scale, the quality does not vary linearly. Instead, it follows the rule of the 3s and 10s. This means that a gain of 3 dBm doubles the signal strength in mW, while a loss of 3 dBm halves it. Besides this, if the strength gains 10 dBm, the strength in mW is multiplied by 10, as well as the opposite. For example, if a transition from -20 to -10 dBm is registered, then the strength rises from 0.01 to 0.1 mW. Regarding the decay pattern of the Wi-Fi signal strength, it follows a Gaussian distribution, where its signal strength decreases with the increasing of the distance between the AP and the receiving device [56].

The use of Wi-Fi radio signal for fingerprinting requires the collection of many fingerprints, one for each detected AP, in each radio band. However, this source of information has some problems. The main challenges of using Wi-Fi radio signal for fingerprinting are related to the fact that its decay pattern can be very affected by the involving environment. This signal often suffers of problems such has diffraction, reflection, scattering or absorption during its propagation. The human body is one of the main causers of this interference. Ma et al. [56] made readings from the Wi-Fi signal by placing a user in a

distance of only one meter from an AP. The variability of the readings when the user was facing front and back the AP reached 5 dBm, which is enough to affect the positioning.

2.3.2.2 Geomagnetic Field

The earth's magnetic field has great potential for indoor location, due to its omnipresence. Although the intensity of the geomagnetic field changes around the world, varying from about 24 to 66 μT, in the same area, this value remains stable. For example, accordingly to IPMA[2] [57], in continental Portugal the intensity of the magnetic field varies around 44 μT.

The stable pattern of the geomagnetic field is affected indoors. The presence of metallic construction materials and electrical equipment causes the local magnetic field pattern to change. However, these anomalies are stable over time, as long as the layout of the building remains the same [28]. Thus, the uniqueness of these disturbance patterns can be used to identify a physical location. Fingerprinting-based IPS can easily rely on the pervasive magnetic field of every building, through the mapping of its magnetic fingerprints. Commonly, a fingerprint is mapped for every axis of the magnetometer, as well as for its magnitude.

Nevertheless, some problems can be pointed out as impediments to the solely use of this source for infrastructure-free indoor location. The fact that the geomagnetic field is very weak lows the discernibility of this signal, due to the high amount of electromagnetic noise sources, making it difficult to process. Furthermore, the variability that the readings of magnetic signal suffer between different devices also becomes a problem if the necessary processing is not properly considered [28]. Thus, the use of several sources of information to support indoor location systems, although it represents an increase to the complexity of the system, also increases the quality of its results.

2.4 Crowdsourcing

Crowdsourcing is a term that defines a contribution model employed by individuals, organisations and companies to solve a problem, develop a task or to reach a goal, relying on the contribution of a group of anonymous people, the crowd [58, 59]. With an open call, where the participation of the users is voluntary, the crowdsourcers obtain help from a heterogeneous group of individuals with different knowledge and experience.

Two techniques can be defined, depending on the contribution of the crowd. In participatory crowdsourcing, users contribute actively to reach the final goal, by performing some computations or generating data. When the users participate in an passive way, crowdsourcing is said opportunistic. In this type, devices are used to automatically collect data from sensors, or use their processing power to perform some computations,

[2] IPMA (*Instituto Português do Mar e da Atmosfera*) is the Portuguese Institute for Sea and Atmosphere.

generally in device's background [58]. In this work, since the objective is to develop a solution that does not require any explicit effort from the users, opportunistic crowdsourcing is the ideal technique to be used.

Smartphones are excellent devices for opportunistic crowdsourcing, since they not only are equipped with a wide range of sensors, but also are almost permanently connected to the internet. Besides the use of crowdsourcing for indoor location, there are several other uses of opportunistic crowdsourcing. A common use of this type of crowdsourcing in our daily lives is the traffic information provided in real time by mobile navigation apps, as in Google Maps app. These apps rely on data collected from the users' devices, that is automatically sent to the company's servers to be processed. Finally, the traffic on the desired route is provided to the user [58]. Participatory crowdsourcing can also be applied in outdoor navigation apps, as in Waze, where users are called to report events, such as accidents or police operations.

Several other applications use crowdsourcing to reach their goals. Collaborative translation is one of the examples, where a group of people can be called to help on translating documents, thus lowering its costs and fastening its completion. Crowdfunding is seen as a variation of crowdsourcing, where projects are funded by people that share the same interests. In this case, with a monetary contribution, instead of working for some task, the crowd supports the development of a product, for example, and often receives a prototype before it reaches the market. Among several others, Wikipedia is another example of participatory crowdsourcing, on which a "collaborative encyclopedia" is constructed with the expertise and experience of users worldwide.

2.5 Signal Processing

A signal is defined as a function that carries information about the behaviour of a system or the characteristics of some phenomenon, and can either exist in nature or can be synthesised [60]. Thus, useful information can be extracted from signals to help developing new systems, by the means of signal analysis, one of the areas of signal processing. Besides this, signal processing also includes the techniques that are use to modify signals so they can be improved.

In this work, two main types of signal will be handled. Time series are the first type of signal to be processed, in order to obtain useful information to be used in the algorithm that will be created. Images are the second type of signal that will be processed, with the purpose of improving the final solution.

2.5.1 Time Series Analysis

Time series are defined as a set of observations x_t, measured over time t. Time series either can be discrete, when observations are taken, for example, in fixed time intervals, or can

be continuous, when observations are measured continuously over some time interval [61].

Time series analysis is the process of extracting useful information from time series, to be used in many purposes. Generally, the methods used in time series analysis can be divided in two categories, depending on the domain where the methods are applied. When the analysis is applied in the frequency-domain, the studied time series have to be converted to the right domain, by several techniques, where one of the most popular is the Fourier transform. However, in this work, time-domain techniques will be applied, where the time series are directly analysed [62].

Time series analysis can be divided in several areas. The area that will be applied in this work attempts to measure the similarity between two time series. This measure is very useful for several classification and clustering problems, since the main goal is to identify connections among all the available data, as further explained in Section 2.6.

Depending on the data where similarities are sought, there are various families of measures which compare the data differently. In this work, methods based on these two families are used [63, 64]:

- **Lock-step measures**: Comparison between two time series happens directly, where the i^{th} point of the first is compared to the the i^{th} point of the second.
- **Elastic measures**: This type allows the comparison of one point of the first time series to many points of the second, thus handling series that are not temporally aligned.

2.5.1.1 Euclidean Distance

Euclidean distance is a lock-step measure that compares time series at the same temporal location. It is one of the most used similarity measures due to its computational simplicity and indexing capabilities. This measure requires that both time series have the same length. In the cases that this does not happen, usually a resampling is performed [63, 65]. The formula for Euclidean distance is:

$$distance_{Euclidean}(P,Q) = \sqrt{\sum_{i=1}^{d}\left(P_i - Q_i\right)^2} \tag{2.4}$$

where P and Q refer to the two time series to be compared. d represents the length of both time series. Figure 2.4 has represented an example of the application of the Euclidean distance between two distorted signals.

2.5.1.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is used to measure two time series that are characterised by having different velocities. DTW is an elastic measure that aims to find the optimal

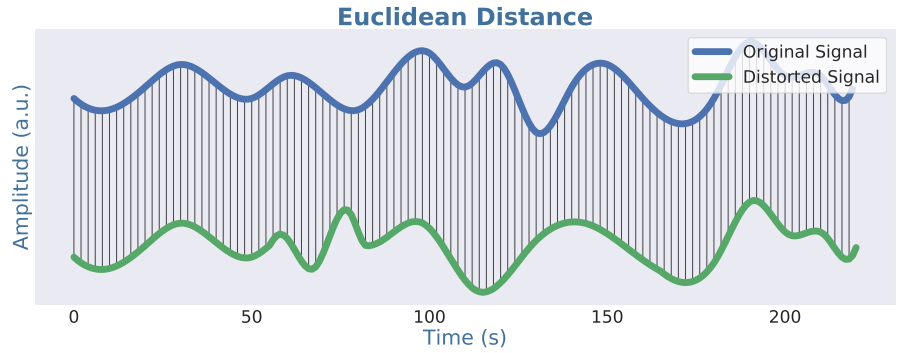


Figure 2.4: Representation of the Euclidean distance between two distorted signals.

alignment between two time series, in a way that minimises the accumulated cost function. With this, it is possible to "stretch" or "compress" a time series to test their similarity to other [63, 64]. In Figure 2.5 it is possible two see the results of the application of DTW to two distorted signals with different lengths.

In a extensive comparison of several methods, Wang et al. [64] came to the conclusion that the DTW is one of the most effective methods, although it has a higher processing time when compared to more recent measures. This processing time difference is diminished when the length of the time series increases.

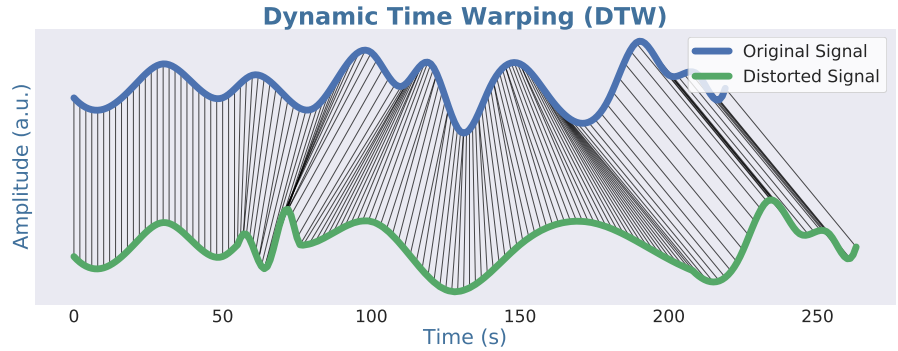


Figure 2.5: Representation of the DTW distance between two distorted signals.

2.5.1.3 Windowed P-norm Alignment

Windowed P-norm Alignment (WPA) [66, 67] is a measure that aims to find the best alignment between a previous computed window, $window$, with length N, and a second time series. The window slides through the second time series, one sample at a time, $sig_{[i:i+N]}$, which has the same length as the window. The distance between the two series is computed for each alignement i, being represented by the following equation:

$$distance_i = \frac{\sum_{j=1}^{N} \left| sig_{[i:i+N]_j} - window_j \right|}{N} \tag{2.5}$$

The distance measured between the window and every sample of the second series generates a function of similarity. The lowest values of this function are the most similar points. Even though this specific formula applies the absolute value of the differences between the two signals, other measures can be applied, as the Euclidean distance. Therefore, with this method it is possible to obtain a measure that identifies similarities between two signal with different lengths. Figure 2.6 has represented an example of the application of WPA to a signal, where a predefined window slides through it and computes the difference between them. The plotted alignment identifies the higher similarity.

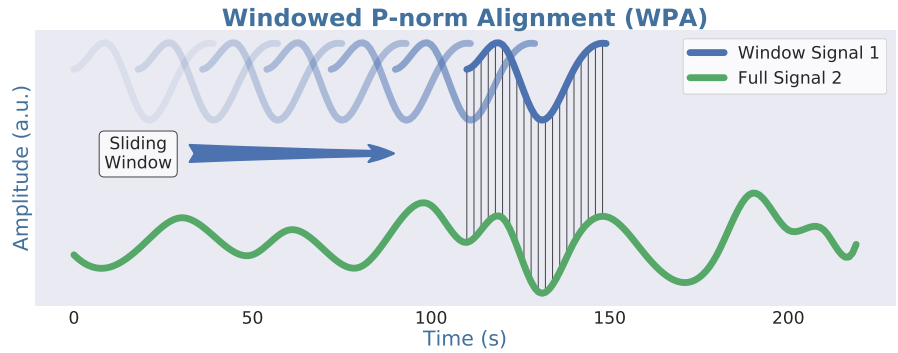


Figure 2.6: Representation of the WPA distance between a defined window and a signal. The window slides through the signal and computes the difference between them at each position. The point with the minimum distance is the optimal alignment, being represented in this Figure by the vertical lines.

2.5.2 Digital Image Processing

An image can be seen as a two-dimensional function, $f(x,y)$, defined by its spatial coordinates x and y. The intensity or grey level of an image is given by the corresponding amplitude of f at any pair of coordinates (x,y). When the image's coordinates and their corresponding intensities are finite and discrete quantities, the image is called digital [68]. Each element of a digital image is named pixel, which consists on the respective intensity of a particular location (x,y).

Digital image processing comprises all the operations performed on a digital image, either to extract some useful information, or to enhance it. In this type of signal processing, the inputs are digital images, and the outputs can either be new images, or the characteristics extracted from them.

One of the most extensive classes of image processing operations, applied to binary images, are the morphological operations [69]. Morphological operations on images are tools developed to extract information about their shape, as their boundaries or skeletons, being useful for filtering purposes. Operations consist on the adding or removal of pixels from an image, depending on the pattern of a pixel's neighbouring pixels. Morphological operations include dilation and erosion, and modifications and combinations of them, depending on the objective of the operation.

- **Dilation**: Dilation process adds pixels to a binary image, turning their value from 0 to 1. At each pixel, an evaluation of its original neighbourhood is made, and if any of those values is set to 1, the value of the evaluated pixel is also set to the same value. In practice, this operation enlarges the shape of the image, since it adds a layer of pixels around features and regions, possible causing them to merge. Dilation also fills small holes within features [69].

- **Erosion**: Contrarily, erosion removes pixels of an image that should not be there, turning them to 0. By evaluating the neighbourhood of each pixel, its value will only be 1 if every neighbour also has the same value. This process removes a layer of pixels around the limits of all features and regions, causing the image to shrink. Erosion is used to remove erroneous pixels, often caused by noise [69].

- **Closing**: Closing operation combines the two previous methods. It consists in the application of a dilation, followed by an erosion. This operation is done mainly to close breaks in features, eliminating 0 valued isolated pixels that are within features or gaps between portion of a feature [69].

- **Opening**: Contrarily, opening is the inverse combination of morphological operations of closing and it is used to separate touching features in an image [69].

The aforementioned image processing techniques will be used in the final part of this work, where the resulting floor plan of map matching process (Section 5.4), which can be considered as an image, has some gross errors that must be resolved. Thus, in Section 5.5, these methods are studied and applied.

2.6 Machine Learning

Machine learning is the subfield of computer science that is dedicated to the development of computational techniques that are able to automatically learn from large amounts of training data, by recognising their patterns. Then, the algorithm is able to apply the learned patterns to future inputs of unknown data, even at a bigger scale [70, 71].

There are several techniques that can be used, depending on the type of information available. Thus, four types of machine learning can be defined:

- **Supervised learning**: In this type, also known as classification, the algorithm is developed with the objective of identifying connections between a set of inputs and a set of outputs, previously trained with a set of labelled data [72].

- **Semi-supervised learning**: Being a variation of the supervised learning process, the training phase of this type of machine learning relies on a small set of labelled data and a large set of unlabelled data [24].

- **Unsupervised learning**: Also called clustering, this type aims to divide a set of unlabelled data into clusters, considering their similarity [72].

- **Reinforcement learning**: This algorithm aims to achieve an objective, such as winning a game, by learning in a dynamic environment, instead of learning with a set of discrete training data [24].

For this work, both supervised and unsupervised techniques have been applied in different steps. Classification algorithms were used in Chapter 3, while clustering algorithms were applied in Chapter 5.

2.6.1 Supervised Machine Learning or Classification

Supervised machine learning algorithms aim to generate, from externally supplied instances, hypothesis capable of making predictions about future instances [72]. The results are classifiers, since they are able to classify a set of features from objects with unknown labels, based on the previous evaluation of labelled features. Features are the characteristics of the dataset's objects where the algorithm will look for similarities to generate the classifier.

Depending on the available dataset and the objectives of the classifiers, several classification algorithms can be selected. For this work some algorithms were tested and the one that gave the best results was chosen.

2.6.1.1 K-Nearest Neighbours

K-Nearest Neighbours (K-NN) algorithm classifies a new unlabelled object by the most frequent class of the K nearest objects. The K value is chosen by the user. K normally is small comparing to the size of the training dataset, which is useful for datasets with poorly defined decision boundaries. However, higher Ks are more robust to noisy datasets [24, 71].

2.6.1.2 Decision Trees

This algorithm attributes labels to unknown objects by a set of decisions, depending on the training data. A decision tree consists on a set of nodes connected by branches, like in a real tree. At each decision node, the value of a single feature is tested, and depending on the result, the following branch is selected. A set of features from an object travels through the tree until a terminal node is found, where the label for that object is attributed [72].

Decision trees have the advantage of being very intuitive for the users, as it can be seen in Figure 2.7, where four features are progressively tested to classify the data objects into one of five labels. Besides this, decision trees can handle both numerical and categorical data, which amplifies the number and type of features that can be extracted from the

dataset. However, decision trees can easily suffer from overfitting if the maximum number of nodes is not well defined.

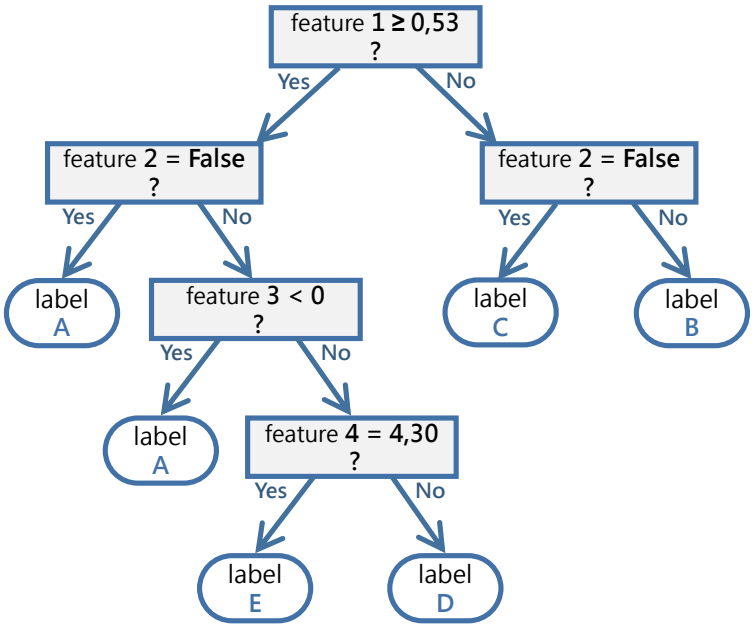


Figure 2.7: Illustration of a decision tree for a specific dataset. Each object of the dataset is characterised by four features (1 to 4), that will be evaluated so the object can be classified in one of five labels (A to E).

2.6.1.3 Random Forest

As the name suggests, this algorithm creates "forests", a set of decision trees. The higher the number of decision trees, the higher the robustness of the algorithm. The classification is then based on the most obtained label from all the decision trees in the "forest". Each tree is created using a subset of the dataset [24].

2.6.2 Unsupervised Machine Learning or Clustering

Clustering algorithms have the objective of separating a given set of unlabelled data into small similar groups, named clusters, without any prior knowledge of the clusters' definitions [73]. Each object of the dataset is analysed so their features can be extracted. After the evaluation of the features of each object, all objects are separated into clusters with similar features. At the end, all the objects within the same cluster have the maximum similarity, and minimum similarity between clusters [74, 75].

This unsupervised method is defined by not using labelled objects beforehand. Instead, it tries to find a structure within the given dataset. Thus, clustering is considered to be of exploratory nature [74, 75].

Clustering algorithms can be separated by several techniques, depending on the characteristic that is being studied [76]. When considering the hardness of the cluster assignment, two categories can be considered:

- **Hard clustering**: In this deterministic type of clustering, an object either belongs to a specific cluster or not.
- **Soft or Fuzzy clustering**: Instead of assigning a cluster to each object, this type attributes to an object the degree of belonging to each cluster. This way, an object belongs to more than one cluster. Soft clustering can be transformed into hard clustering if the object is attributed to the cluster with the highest probability.

For the purposes of the application of clustering in this work, hard clustering algorithms will be preferred. Regarding this type, algorithms can be further divided int new categories, depending on the relation between the created clusters and the process of achieving them:

- **Hierarchical clustering**: This type of clustering aims to create a hierarchy of clusters. The process consists on assigning a cluster to every object and consecutively merging clusters until a sopping criteria is met. The opposite process can also happen, where at the beginning all objects belong to the same cluster, being separated successively [73].
- **Partitioning clustering**: This type consists on initially partitioning all objects into a defined number of clusters. Then, the clusters are iteratively improved until a stable division is reached [73].
- **Density-based clustering**: This last type is characterised for discovering clusters with arbitrary shapes, depending on the number of neighbouring objects of each cluster's object. Contrarily, the other methods search for objects where a minimum distance from the centre of the cluster is met. This is useful to protect the clusters against outliers on the dataset [73].

There are several algorithms for each type of clustering. However, there is no algorithm that works properly under all the situations. Thus, the choice of the best algorithm for a specific dataset is often done empirically. The following algorithms that belong to the aforementioned types were tested in this work.

2.6.2.1 Agglomerative Clustering

This is the most common hierarchical clustering algorithm [77]. It uses a "bottom up" approach, where every object starts with is own cluster. The process of agglomerative clustering algorithm is [73]:

1. Initially, every object is in its own cluster.
2. From all clusters, selection of the two nearest clusters.
3. Merging of these two clusters into a single one.
4. Repetition of steps 2 and 3 until the stopping criteria is met, which can usually be a predefined final number of clusters.

The results of hierarchical clustering are often represented as a dendogram, a graphic that shows the relationships between the old clusters and the new merged clusters, as well as the order in which the clusters were merged. A dendogram is available in Figure 2.8, where eight objects are plotted in the two-dimensional feature space. The dendogram of the same Figure identifies the process of clusters merging, depending on the increasing distance between objects. With a stopping criteria of two clusters, it is possible to see that the objects 7 and 8 will be placed in one cluster, while the remaining objects will be placed in the other.

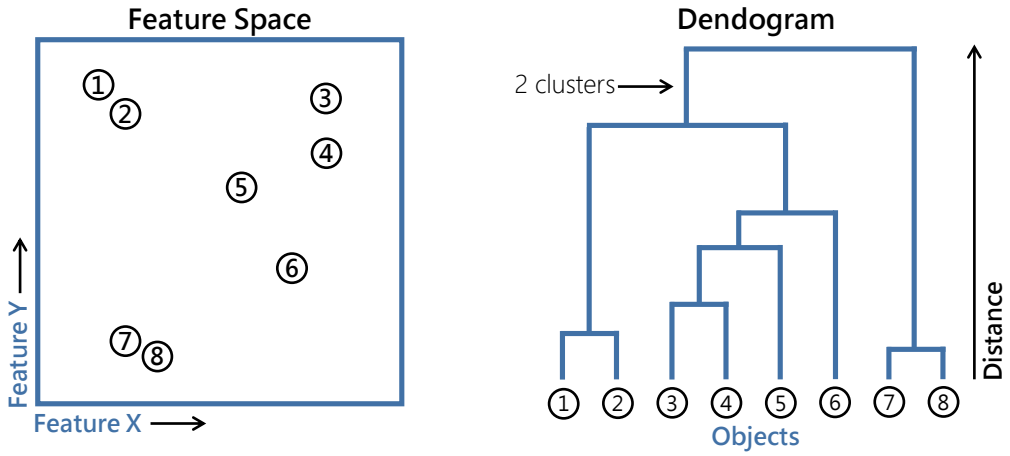


Figure 2.8: Illustration of a dendogram for a specific dataset. The dataset has eight objects (1 to 8), characterised by two features (X and Y). The results of the application of the agglomerative clustering algorithm are represented in the dendogram of the Figure, where each object has initially its own cluster. Progressively, with the increase of the accepted distance between objects, the clusters are merged, until a stopping criteria is met.

2.6.2.2 K-Means

K-Means is the most popular and simplest partitioning clustering algorithm [74]. It requires as input K, the number of clusters to divide the dataset, and the distance metric, where it is usually applied the Euclidean distance. Each cluster is represented by its centroid, which corresponds to the mean of the features' values of all objects within the cluster. The steps of K-Means algorithm are the following [73, 74]:

1. K points are set as initial centroids, among the unlabelled data.
2. Every object is assigned to its closest centroid.
3. Computation the new centroid for each cluster.
4. Repetition of steps 2 and 3 until all cluster memberships stabilise.

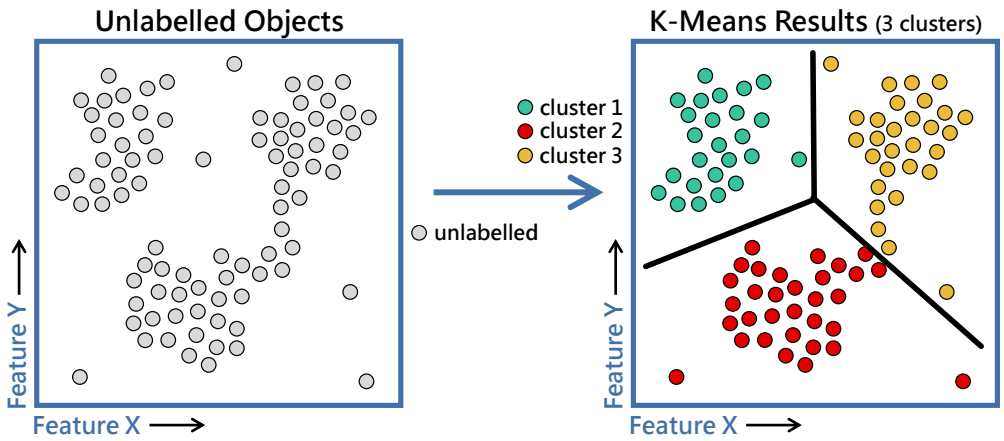


Figure 2.9: Illustration of the results of K-Means algorithm for a specific dataset, characterised by two features (X and Y). A number of three clusters was predefined, and the resulting segmentation is available in the right part of the Figure. The objects of each cluster are identified by a different colour, as it is explained in the legend placed in the centre of the Figure.

The results of K-Means process are illustrated in Figure 2.9, with a simple example of a dataset with two features for three clusters. The three segmented clusters are identified by a different colour, where each object is more approximate of its cluster centroid than to the others.

One of the problems of K-Means is the fact that the number of clusters has to be chosen beforehand. If the clustered dataset is completely unknown, the number of clusters is then impossible to predict. To overcome these limitations, the following methods were proposed to try to identify the ideal number of clusters:

- **Elbow method**: This method computes a given evaluation metric, as the sum of the within cluster variance, for different numbers of clusters, usually from 1 to 10, and plots a curve with the obtained results. The ideal number of clusters is the one that adding another cluster does not improve significantly the value of the evaluation metric. This point is known as the elbow of the curve. However, the identification of the ideal number of clusters is done visually, so it cannot be done unambiguously.
- **Curvature-based method**: With the purpose of eliminating the uncertainty of the ideal number of clusters identification of the elbow method, Zhang et al. [78] developed the curvature-based method, where the curvature of each point of the

elbow method graph is computed. The curvature index κ is the amount by which a geometric object deviates from being flat, defined as the following equation:

$$\kappa = \frac{\left|y''\right|}{(1+y'^2)^{3/2}} \tag{2.6}$$

where $y = f(x)$, which represents the curve obtained by the within cluster variances for each number of clusters. The elbow of the curve is then the point with the maximum curvature index.

2.6.2.3 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that aims to separate high density regions from low density regions. The density of a region is established by the number of objects that are less than a distance from another object. This distance is the input parameter *Eps*. The minimum number of objects inside a region, *MinPts*, is also given as input. The process of DBSCAN is the following [77]:

1. For each object in the dataset, its density is computed, which corresponds to the number of objects under a distance *Eps*.
2. Classification of every object as one of the next three categories:
   - **Core object**: Happens when the number of objects under *Eps* is bigger than the *MinPts* threshold.
   - **Border object**: This object is not a core object, but falls into the neighbourhood of a core object.
   - **Noise object**: A noise object is neither of the previous categories, since it is far from all the other objects.
3. Identification of the detected clusters and removal of outliers, the noise objects.

The results of the DBSCAN process to the same dataset of Figure 2.9 are schematised in Figure 2.10, where two clusters were constructed, identified by a different colour. The outliers of the dataset are identified in white. This algorithm only needs one iteration to cluster the data, but has some disadvantages. Being *Eps* a distance between objects that is required as parameter, it is very difficult to define in datasets with high number of features. While in Figure 2.10 the objects are characterised by two features, as they are represented in two dimensions, it is easy to get the desired *Eps*. However, in a dataset with a fifty features, for example, it might be impossible to get the ideal parameter. This problem also happens if the dataset is not well known.

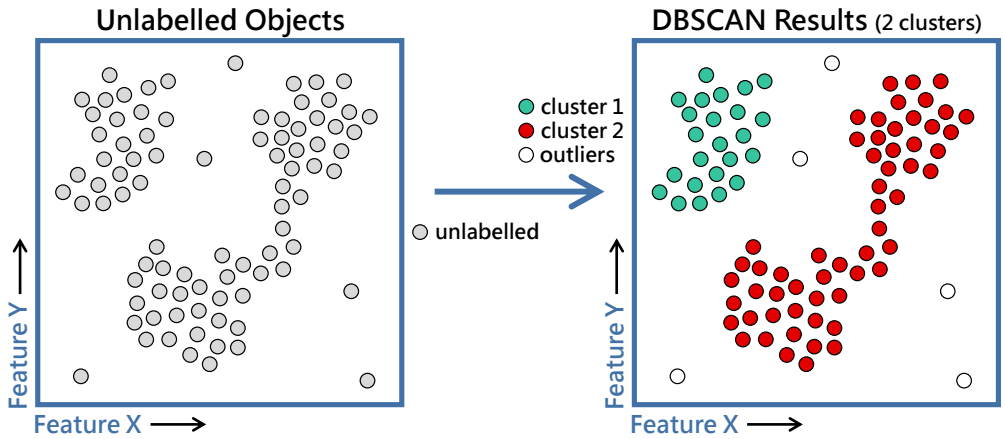


Figure 2.10: Illustration of the results of DBSCAN algorithm for a specific dataset, characterised by two features (X and Y). Since DBSCAN does not require the input of the number of clusters, the algorithm segmented the data in two clusters, considering its density-based approach. Both clusters and the outliers of the dataset are identified by a different colour, as it is explained in the legend placed in the centre of the Figure.

Besides the fact that the final number of clusters is not required as input, the great advantage of DBSCAN is the fact that is able to find clusters with arbitrary shapes and sizes, also being resistant to noise in the dataset. Contrarily, algorithms as K-Means attribute a cluster to every object and form globular clusters.

2.6.2.4 HDBSCAN

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is an improvement of DBSCAN, since it does not require the *Eps* parameter as input. Instead of computing the number of objects that are under *Eps* as in DBSCAN, HDBSCAN computes for every object the distance required to contain the minimum number of points, thus creating new concept of density. The algorithm then merges in the same cluster all the connected points within a density. Different densities are hierarchically tested to optimise the value that gives the best results [79].

CHAPTER 3

Proof of Concept with Simulated Data

The first phase of this thesis consisted in a Proof of Concept (PoC), that was developed to demonstrate the feasibility of the use of crowdsourced data for the automatic construction of buildings' floor plans and environmental fingerprints, trough the detection of similarities between signals. To achieve this, the developed PoC relied on simulated environmental data, designed to be as approximate as possible to the real acquired data. This first approach was motivated by the fact that, to reach the final objectives, this thesis will rely on the complex topic of processing non-annotated data. Thus, the use of controlled data, that is not suitable to the variability of real contexts, allowed not only the verification of the feasibility of the final algorithm, but also the mastery of the techniques required to develop it.

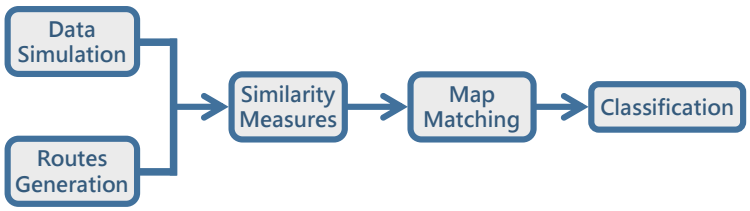


Figure 3.1: Scheme of the PoC workflow.

The workflow of this PoC is available in the scheme of Figure 3.1. The first stage of the developed algorithm simulates the environmental distribution of the magnetic field and the Wi-Fi radio data, within a physical space. In the next stage, random routes are generated in the same space, and the environmental data is annotated, to replicate the acquisition process. Then, similarity metrics are applied to the obtained routes, to identify common segments between them, named overlaps. After this stage, the map matching process tries to construct maps with the combination of all routes, in order to prove the accuracy of the overlap identification process. Due to the obtained results,

a final stage was required, so the correct map, among all the constructed maps, can be identified. The algorithm was developed in Python 2.7.

In this PoC, the physical space is represented by a two dimensional grid of 100 m^2 (10x10 m), divided in evenly disposed 0.04 m^2 (0.2x0.2 m) cells, i.e., a square matrix of 50 by 50 cells. As it can be seen in Figure 3.2, the centre of the matrix represents the origin point of the simulated physical space.

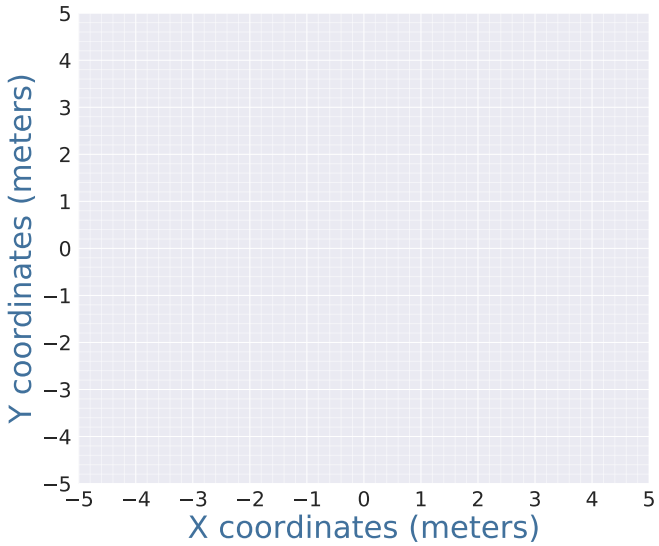


Figure 3.2: Representation of the physical space used in this PoC. Each cell represents 0.04 m^2, where the coordinates of every position vary from the centre of the grid, defined to be the origin point.

3.1 Simulation of Environmental Data

The simulation of environmental data in this PoC was made to create a solution that reduces the variability and unpredictability of real scenarios. If a system does not work using information that behaves as physics models predict, then it would be possible to infer that the use of data highly suitable to variations would not improve the system.

Thus, instead of using real data collected from smartphones, environmental data was simulated. As it will be used in the final solution, this PoC relies on environmental data of magnetic field and Wi-Fi radio.

3.1.1 Magnetic Field Fingerprints Simulation

The use of the magnetic field for the purposes of this work requires that inferences exist, to create singular patterns that can be identified by similarity measures. To recreate this, the simulation of the magnetic field was done using an algorithm previously developed at Fraunhofer AICOS. This algorithm places dipoles in chosen locations within or outside

the simulated space, to replicate the magnetic interferences that would be created by the construction materials, for example. The magnetic field value within each cell element has three components, which correspond to the three axes of the Cartesian coordinate system [66].

In Figure 3.3, it is possible to see the magnetic field behaviour of the simulated space, represented by its corresponding magnetic fingerprints. In this case, two dipoles were placed in the coordinates (0.0, 0.0) and (5.0, 5.0), in the (x, y) plane, where z was fixed as 0.0. Both dipoles were vertically oriented, across the z axis.

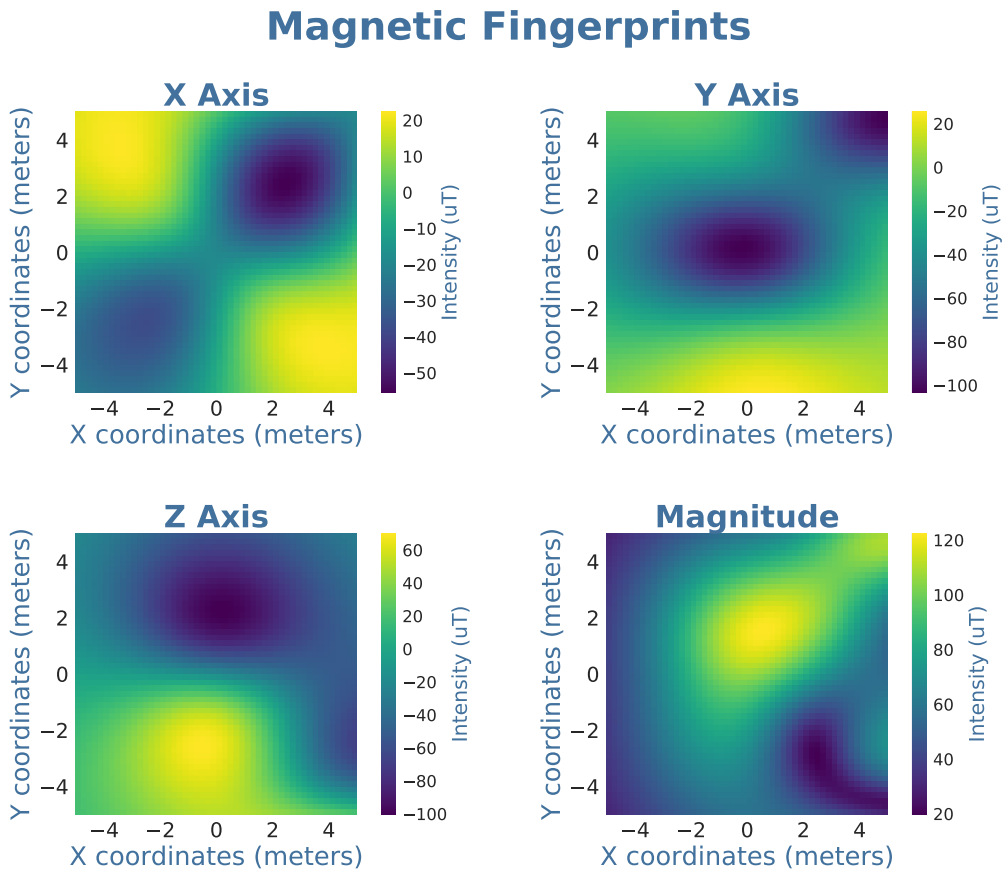


Figure 3.3: Magnetic fingerprints used in this PoC, with each axis represented in its fingerprint, as well as their magnitude. The dipoles were placed at (0.0, 0.0) and (5.0, 5.0) coordinates of the (x, y) plane, where z was fixed as 0.0. Both dipoles were vertically oriented, across the z axis. Each fingerprint is computed for the same physical space simulated in Figure 3.2. Their values vary differently, as it is explained by the scales positioned in their right.

3.1.2 Wi-Fi Radio Fingerprints Simulation

For the Wi-Fi radio simulation, a number of APs are placed in the simulated space, in chosen positions. Considering that the Wi-Fi decay pattern with no interferences follows

a Gaussian distribution, as it is described in Section 2.3.2.1, the signal that is expected to be read in the surroundings is computed for each AP.

In this simulation, the chosen positions of the two placed APs in the (x, y) plane were (-2.0, 2.0) and (2.0, 3.0), with z fixed as 0.0. Figure 3.4 shows the fingerprints of the Wi-Fi radio signal for both APs.

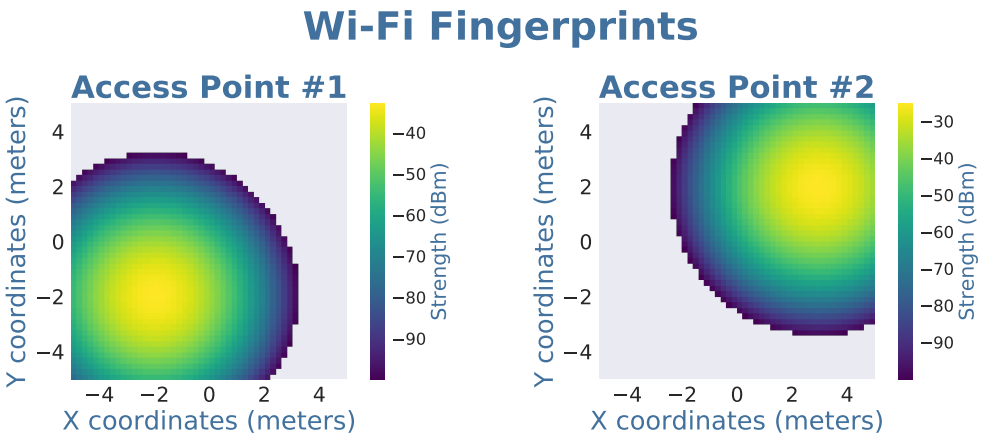


Figure 3.4: Wi-Fi fingerprints used in this PoC, with the decay of each AP represented in its fingerprint. APs 1 and 2 were placed at (-2.0, 2.0) and (2.0, 3.0) coordinates of the (x, y) plane, respectively. The z axis was fixed as 0.0. Each fingerprint is computed for the same physical space simulated in Figure 3.2. Their values vary differently, as it is explained by the scales positioned in their right.

3.2 Generation of Random Routes

The simulation of routes was created to eliminate the error created by the noisy inertial sensors of smartphones. The originated routes only have straight line paths with rigid turns of 90 degrees to the right or to the left. The length of each step was defined to be of one meter. However, a straight line always has even number of steps. This is done to approximate this simulation to the human behaviour, since the walking patterns in real situations consist mainly in straight segments, with few turns, when necessary. Consequently, the full length of every route is always a even number. The process of routes generation is the following:

1. Selection of a random pair of coordinates on the created grid (see Figure 3.2), to be the starting point of each route.

2. Selection of a random length for the straight line path, considering the even number of steps, within the grid's borders.

3. Random selection of the turning direction, to the left or to the right, depending on the available possibilities.

4. Repetition of steps 2 and 3 until a minimum total length of each routes is met, established as 16 meters.

Contrarily to the final solution, that will use a large number of routes collected with crowdsourcing techniques, this PoC will be done using only three routes, for simplicity. To ensure that overlapping areas exist, new routes will be generated until at least one overlap is found. Figure 3.5 shows the three simulated routes that will be used as example to explain the algorithm of this PoC.

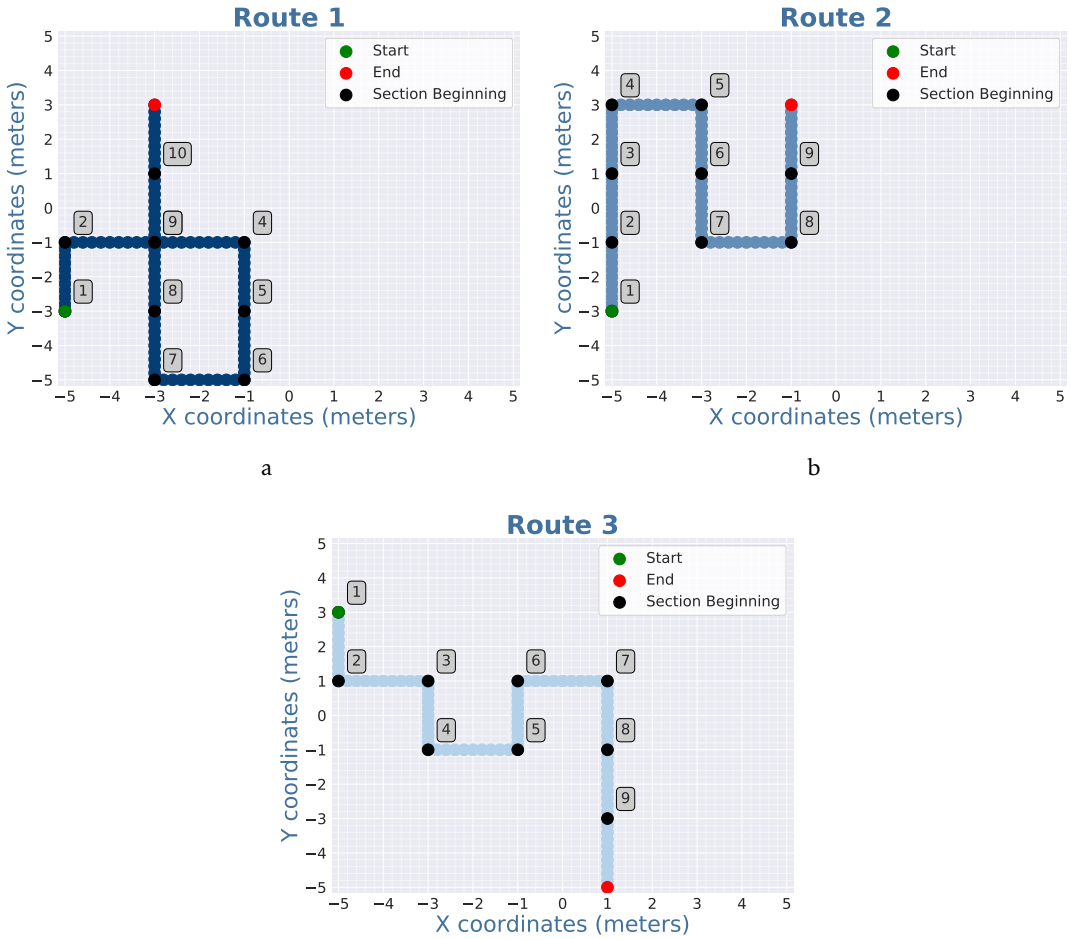
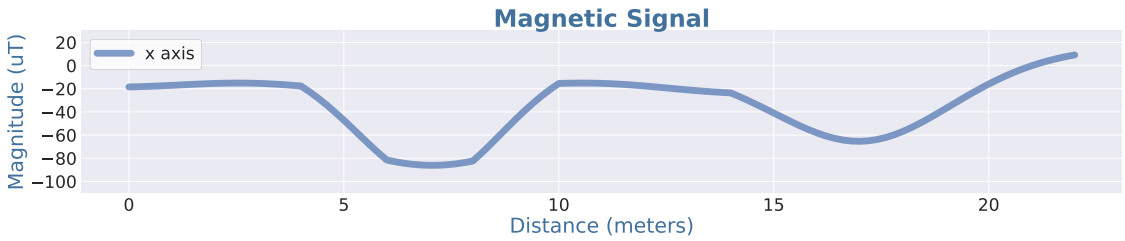


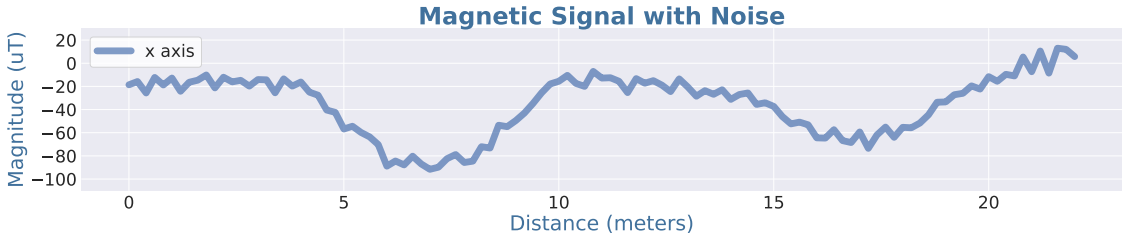
Figure 3.5: Trajectories of the three simulated routes in this PoC, one for each Subfigure. They were computed within the limits of the physical space of Figure 3.2. The green and red dots represent, respectively, the start and end of each route. The black dots represent the beginning of the two meter straight sections that will be further used in Section 3.3.

After the routes are generated, the data acquisition process has to be simulated. This process consists on retrieving the values of the magnetic and Wi-Fi radio fingerprints, for the positions of each route. Thus, since it is assumed that a step has a length of one meter, but the resolution of the considered grid is one fifth of that value, every step is composed by five values of environmental data.

Data collected in real environments has great variability, due to noisy sensors and other factors, as the difference on the walking velocity of the users. However, in this PoC those problems do not happen. Then, the process of identifying similarities, between two overlapping sections of two different routes, would be too direct if the collected signals of the simulated fingerprints were compared. Thus, to approximate this PoC to the reality, Gaussian noise with mean 0 and variance 5 is added to the registered magnetic field and Wi-Fi radio signals. In Figure 3.6, it is possible to see the differences on a sequence of magnetic field data from the x axis, from one of the simulated routes.



(a)



(b)

Figure 3.6: Sequence from the x axis of the simulated magnetic field, before the addition of Gaussian noise (Figure 3.6a), and after (Figure 3.6b). The Gaussian noise follows a normal distribution with mean 0 and variance 5.

With all the necessary information available, every route is translated to the origin point, at the (0, 0) coordinates. This is done to approximate this PoC to the final solution, where the absolute positions of the collected routes are not known.

3.3 Similarity Measures

The identification of overlapping sections between the simulated routes is made by segmenting every route in two meter sections and comparing them to each other. In other words, the segmented sections of the first route are compared to the sections of the second and third routes, and the sections of the second route are compared to the sections of the third. Besides this, since overlaps could happen in opposite ways, all sections are also compared with the inverted ones. For these comparisons, the following methods were applied, for both the magnetic field and the Wi-Fi radio data:

- **Magnetic Field**
  - **Dynamic Time Warping**: This measure is very useful to identify similarities between signals with a temporal distortion, caused by the different velocities of the users during the acquisitions. Although this problem is not verified in this PoC, DTW was still implemented, to understand its mechanism and test its potentialities. Thus, DTW is computed for every pair of segmented sections, and the cost functions are registered. This measure is implemented in a package for Python[1].
  - **Difference of means**: The mean of the magnetic signal of each section might be useful, since the interferences created by the dipoles create a decay pattern that influence the mean of the signal. Considering this, the difference of the means between every pair of sections is computed, for every axis and their corresponding magnitude.
- **Wi-Fi Radio**
  - **Difference of RSSs' means**: The difference of the means of measured RSSs also gives useful information about the similarity between sections, since the strength of the Wi-Fi signal decreases with the distance from the APs.
  - **Coherence of APs**: Since the similarity identification by the difference of RSSs might be affected by the presence of noise, this method tests if the APs detection is consistent in both sections, returning a Boolean (consistent/not consistent). The test will return True if the detected APs are the same in both sections being analysed. Otherwise, it will return False.

For each method, heatmaps for every pair of routes are built to visually evaluate the metrics results. As it can be seen in Figure 3.7, where each heatmap represents a comparison between two routes, and each cell has the mean of the distances computed with DTW, for all axes of the magnetic field. The axes of each heatmap identify the straight section numbers of each route, as it is described in the trajectories of Figure 3.2. The numbers followed by an apostrophe (') identify the sections that were inverted before the comparison, so overlaps in opposite directions can be identified. The different results are displayed in different colours, as the scale of each map reveals. The smaller the distance, the greater the similarity. For example, in the cell corresponding to the section 3 of the first route and the section 7 of the second route, this minimum distance corresponds to an overlap.

After the computation of all methods for every pair of straight sections, it is necessary to evaluate them, so the overlaps can be identified. To do this, the following process was applied:

[1] Available in https://github.com/pierre-rouanet/dtw (visited on 09/17/2018).

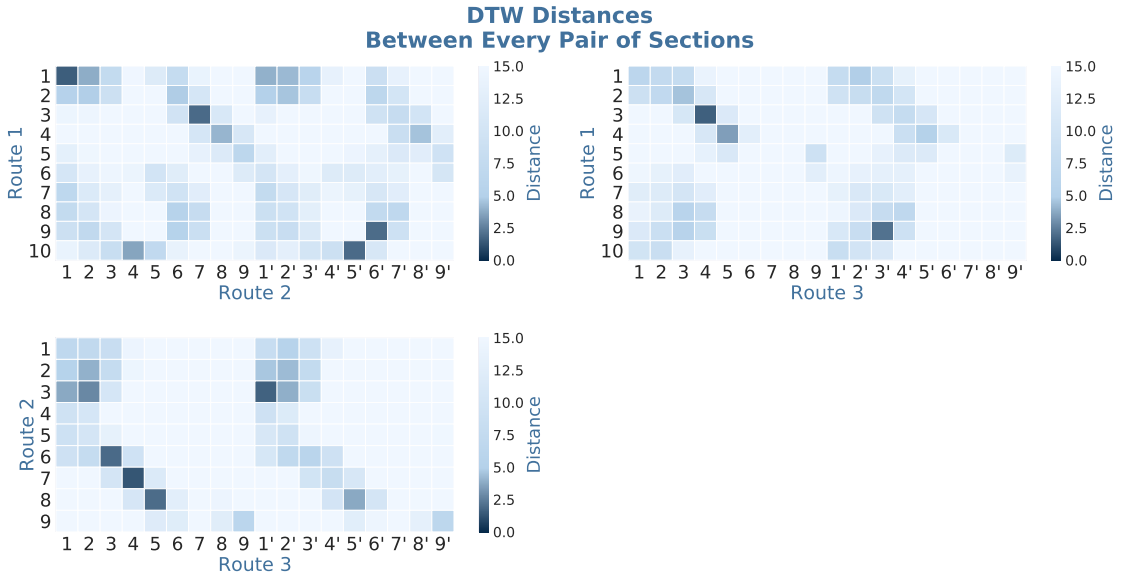


Figure 3.7: Resulting heatmaps from the magnetic field comparisons with DTW. Each heatmap has the results of the comparisons between two routes, where each cell contains the distance between two straight sections of different routes, identified by the corresponding section numbers. The sections identified by its number and an apostrophe (') were inverted before the comparison, to identify overlaps in opposite ways. The sections of each route are also available in the trajectories of Figure 3.2. As it is showed in the scales in the right of each heatmap, the colour of each cell denotes a different distance. The darker the colour, the lesser the distance and consecutively, the higher the similarity.

1. The metrics' results are converted to a normalised value, between 0 and 1. This is done using a sigmoid function, so similar sections will obtain a high parameter value, while different sections will obtain a low parameter value. The variables that define the shape of the sigmoid were empirically chosen for each method.

2. Taking into consideration the fact that some metrics give more trustworthy results, a weighted mean of the normalised values obtained in step 1 is computed, applying the following equation to every comparison:

$$final\ result = \frac{\sum_{i=1}^{N} \left(result_i \times weight_i \right)}{\sum_{i=1}^{N} weight_i} \tag{3.1}$$

   where N represents all the metrics applied. For the magnetic field, the weights were defined as 1.0 for the DTW and 0.8 for the difference of means. Regarding the Wi-Fi radio, the chosen weights were 0.8 for the difference of RSSs' means and 1.0 for the coherence of APs. Therefore, similar sections will have a higher *final result* value.

3. The identified overlaps are retrieved by analysing the *final results*. This process consists on identifying the maximum value among all set of comparisons. Then, all

the comparisons that have a value higher than 80% of the maximum are registered as overlapping sections.

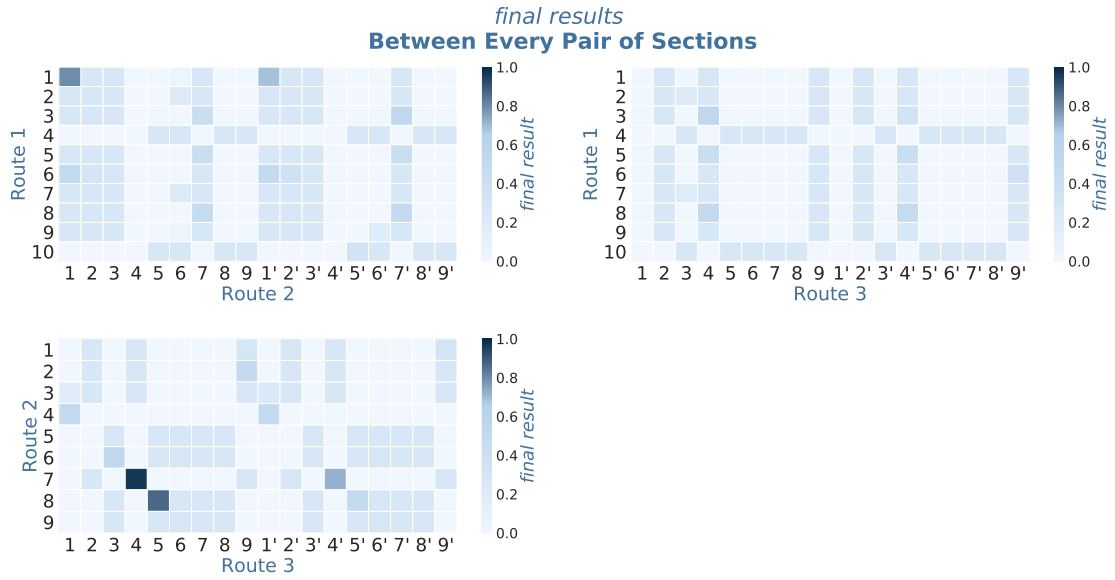


Figure 3.8: Resulting heatmaps with the computed *final results* for the comparisons between every pair of routes. Each cell contains the *final result* obtained for two sections, identified by the corresponding section numbers in the heatmaps' axes. The sections identified by its number and an apostrophe (') were inverted before the comparison, to identify overlaps in opposite ways. The sections of each route are also available in the trajectories of Figure 3.2. As it is showed in the scales on the right of each heatmap, the colour of each cell denotes a different value. The overlaps are in the darker cells, with a *final result* value approximate to 1.

After step 2, a new heatmap is computed with the *final results* of the comparisons between every pair of routes, as it can be seen in Figure 3.8. Again, the straight sections of each route are identified by their numbers and the comparisons that were computed between a section and the reverse of another are represented with an apostrophe ('). From these heatmaps, the overlaps are retrieved, depending on the darkness of the colour of each cell. Then, the created list of overlaps is sent to the map matching process, ordered by the number of route. For each overlap, the *final result* obtained from the comparisons is also stored in the list, since it will be important for the classification phase.

3.4 Map Matching

The map matching phase of this PoC aims to reconstruct the map of the simulated space, given by the combination of the trajectories of the created routes. After this process, it is possible to verify the quality of the overlaps identified by the similarity measures.

It is expected that the process of identifying overlaps has some error, where not only wrong overlaps might be identified, but also true overlaps may stay unidentified. Since

the only method that produces a different value when comparing sections in both directions is the DTW for the magnetic field, sometimes overlaps of the same sections in both directions are identified. Thus, the map construction must consider that not all identified overlaps are correct. To do so, the algorithm constructs as many maps as the necessary to include all overlaps. Even if in this PoC's example the number of constructed maps might not be too high, since it only processes overlaps for 3 routes, the number of maps increases considerably with the number of routes.

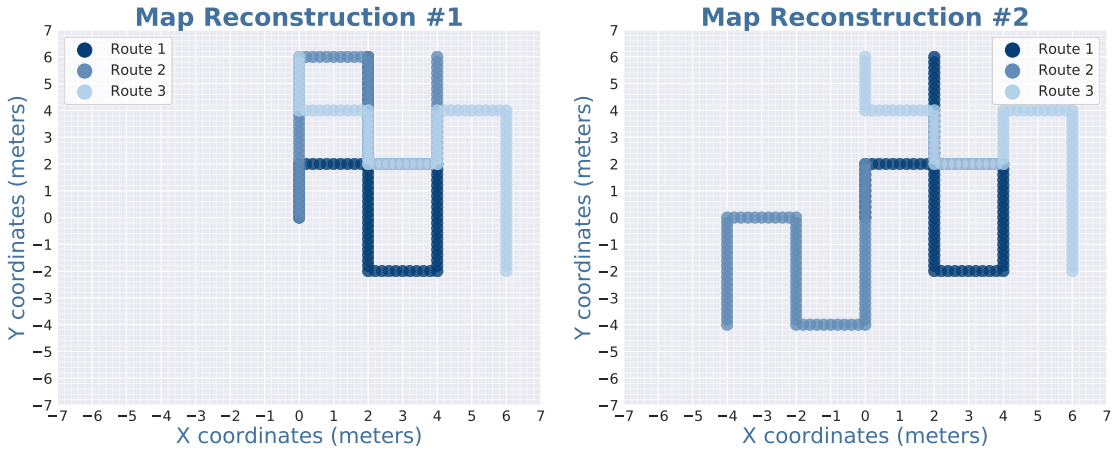
The map matching process is initialised with the random mapping of a route in a grid, similar to the one available in Figure 3.2, which is the map where the reconstruction will take place. Therefore, this first route will provide the basis coordinates for every map, with every route being transformed to match this one. After this, the algorithm iteratively searches for overlaps between the route being analysed and all the available routes, among the previously obtained list.

Then, for each identified overlap, the new route is transformed to match the coordinates of the route already mapped. If a new overlap produces a different configuration of routes, the original map with the first route is copied, and a new map that considers the new overlap is produced. After this point, every new overlap will be matched in all available maps. The map matching finishes when all the identified overlaps are mapped.

The process of adding a new route to a map is done using the information of the original coordinates of the new route and the coordinates of the route already in the map. Then, the orientations of the overlapping sections are computed. If these two sections have different orientations, then the new route needs to be rotated to match the orientation of the first route's section. This is done by multiplying a rotation matrix by the coordinates of the new route. The rotation matrix, defined in Equation 3.2, is computed with an angle given by the difference of orientation between both sections, $\theta = (\theta_1 - \theta_2)$. With both overlapping sections in the same orientation, the new route is translated into the positions of the route already mapped, by subtracting from all points, the difference between the overlapping sections.

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.2}$$

After the map matching process has been applied to the routes used in this PoC, two maps were created, as it is possible to see in Figure 3.9. Thus, by comparing these maps to the positions of the original routes, is it possible to conclude that Figure 3.9a has represented the accurate combination of all created routes, in the simulated physical space.



(a) (b)

Figure 3.9: Reconstructed maps from the map matching process. Depending on the combination of all existing overlaps, a variable number of maps is constructed, by the successive transformation of routes.

3.5 Classification

The map matching process originates a set of maps, where only one represents correctly the combination of all routes in the simulated space. To help on deciding which map is the correct, a supervised classifier was trained. This binary classifier has the capability of deciding if a map is or is not correct.

To train the classifier, an algorithm was developed to provide the ground truth information about every constructed map. The algorithm builds the map that correctly combines all the simulated routes, and then compares its coordinates with the maps obtained after the map matching process. If the difference between them is always a constant value, then the constructed map is the correct. This evaluation attributes a label to the training set of features.

As it is explained in Subsection 2.6.1, classifiers require a set of labelled features to the training phase. The features that characterise each constructed map are:

- **Number of overlapping sections**: The number of sections that overlap in the constructed map is used as feature. It includes every pair of overlapping sections in the map, not only the overlaps identified by the similarity measures.
- **Number of routes**: This feature is also important to consider, since it is more probable to find overlaps in a map with more routes.
- **Statistics**: Regarding all the found overlaps in the map, the following statistical features are computed, using the *final results* previously obtained in Section 3.3:
  - **Maximum *final result***
  - **Minimum *final result***

- **Mean of overlaps' *final results***
- **Standard deviation of overlaps' *final results***
- **Root mean square of overlaps' *final results***

Three different classifiers were tested in an adaptation of a framework previously developed at Fraunhofer AICOS, which trains a set of classifiers to test their performance. At the end, it returns the classifier with the best accuracy. The following classifiers were tested:

- K-Nearest Neighbours (Subsubsection 2.6.1.1)
- Decision Tree (Subsubsection 2.6.1.2)
- Random Forest (Subsubsection 2.6.1.3)

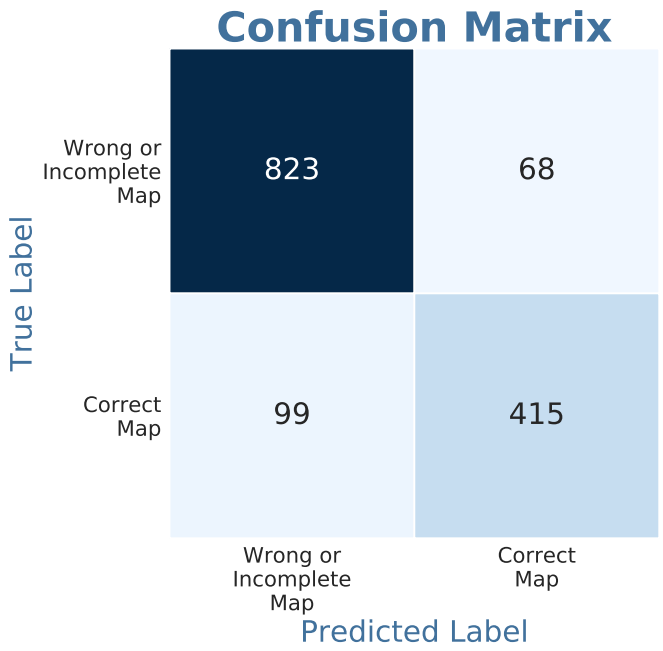


Figure 3.10: Confusion matrix obtained with the Random Forest classifier, after it has been trained with 1000 cycles. The accuracy obtained with this classifier was approximately 88%.

The classifiers were trained with a dataset of 1000 cycles, with different numbers of simulated routes. As it was previously explained, the number of resulting maps varies depending on the quality of the retrieved overlaps in the end of Section 3.3. The classifier that gave the best results was the Random Forest, with an accuracy of approximately 88%. The resulting confusion matrix is available in Figure 3.10. As it can be seen, the number of maps that was classified as wrong or incomplete is explained by the fact that among all of the created map, at best only one represents the correct combination of routes. For

the resulting maps of this PoC's example, the only map that was classified as the right one was the first, which corresponds to the truth.

After the classification process, more than one floor plan can be classified as correct. Among the hypothetical set of "correct" floor plans, the final floor plan to be returned is the one that has the highest mean of all overlapping results. However, this problem was not verified in this example. In Figure 3.11 it is possible to see the final returned map, after the merge of all routes.

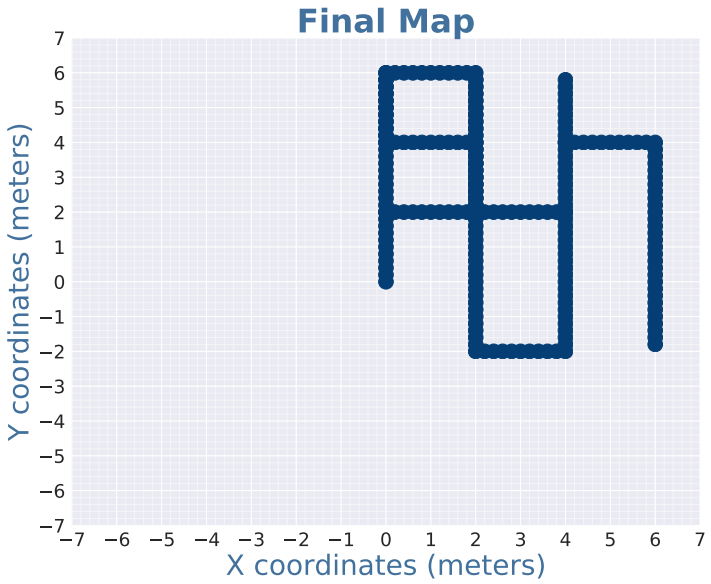


Figure 3.11: Returned map after the classification process. As it can be confirmed by the trajectories of Figure 3.5, the map matching process, as well as the classification phase produced the expected results. The physical space is not the same of Figure 3.2, since the combination of routes does not produce an absolute reference to the original positions.

3.6 Discussion

The results obtained in this PoC sustain the hypothesis that the development of a system that constructs automatic floor plans might be possible to develop. Although the used environmental data does not correspond to the collected data in real situations, its characteristics allow the approximation. While the fact that humans most commonly walk in straight lines is contemplated in this PoC, it does not consider the high variability of the direction changes. Despite the fact that the segmentation process must be adapted to the real context, the similarity measures that can guarantee an accurate identification of overlaps may be applied with the same principles. Regarding the map matching process, some modifications have to be made, to ensure the scalability of the solution, since the number of routes used to construct the map will be very much higher. Thus, Chapter 5 describes how the final solution handles this differences, with some different methods applied in each phase of the process.

Chapter 4

Data Acquisition

The algorithm proposed on this thesis aims to automatically construct indoor floor plans and environmental fingerprints, without any human effort. To do this, data will be opportunistically collected via crowdsourcing. In practice, this means that large amounts of data will be acquired from users' smartphones, who will walk freely during their daily activities, inside the building to be mapped.

To support the development of this algorithm, an acquisition protocol was designed and the respective dataset was collected to simulate the human movement during real life situations where individuals can express "normal" behaviour. The tests performed during the algorithm development process were done using a dataset acquired at Fraunhofer AICOS's Lisbon office. Its floor plan is available in Figure 4.1. This building will be referred in this thesis as test building.

Figure 4.1: Floor plan of the test building.

To allow the visualisation of the results obtained at each stage of the algorithm of Chapter 5, as well as to other necessities of the PIL solution, the floor plan of Figure 4.1 was converted to a schematic that translates the walkable areas of the building. This

new simpler floor plan is represented in Figure 4.2, where the white coloured zones represent the existing areas of the building. The blue colour translates the impossible areas, originated by the furniture, walls, or simply zones outside of the building. The red colour identifies sections with stairs.

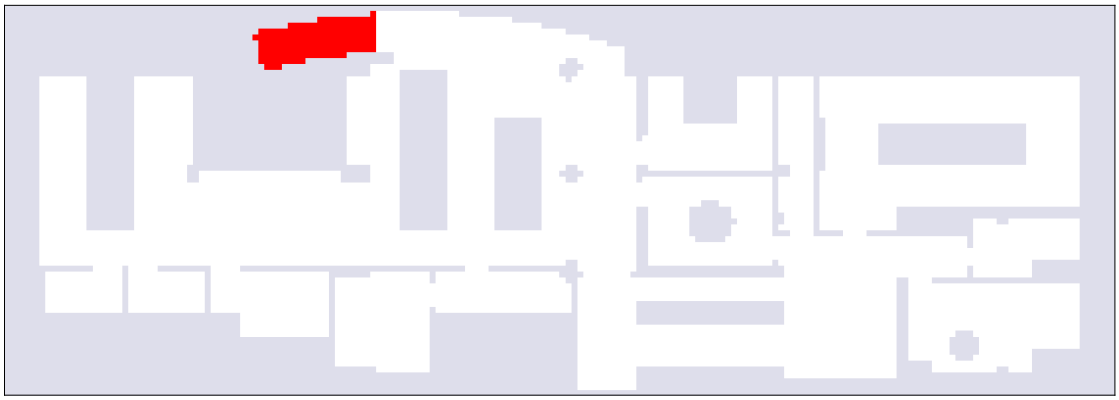


Figure 4.2: Floor plan of the test building, to be used in the development of the algorithm. The white colours translates the walkable areas, while the blue colour designates the building's constrains. The red colour identifies a stairs section.

4.1 Routes Design

In order to achieve the natural movement of users in the test building, where numerous different trajectories happen, 22 different routes were designed. These routes cover different parts of the building and were designed to have overlaps in different zones. These overlaps are fundamental to the construction of floor plans process. Since most buildings have common areas and are constructed with main corridors, where a great part of the users passes by, it can be assumed that overlaps will always exist indoors. The overlaps between routes were designed in both directions, to further increase the reliability of the dataset.

The coordinates of the designed routes were annotated with the purpose of providing a ground truth basis for intermediate tests, during the development phase. In Figure 4.3 it is possible to see four routes of the dataset. The full design is available in Appendix A.

Regarding the nomenclature of the routes in the dataset, each acquisition has a unique name, defined as **HCrowdXX_YY**. **XX** and **YY** are the values that identify unequivocally each acquisition. **XX** is given by the number of the route's design, and assumes values from **00** to **21**. **YY** represents the repetition number of the acquisition for a given route, incrementally varying from **01**.

Overlaying all routes of the dataset in the floor plan, it is possible to obtain the covered areas of the building, where each area has at least one route passing by. Ideally, the proposed algorithm will produce a floor plan equal to Figure 4.4, which shows the covered areas of the test building.

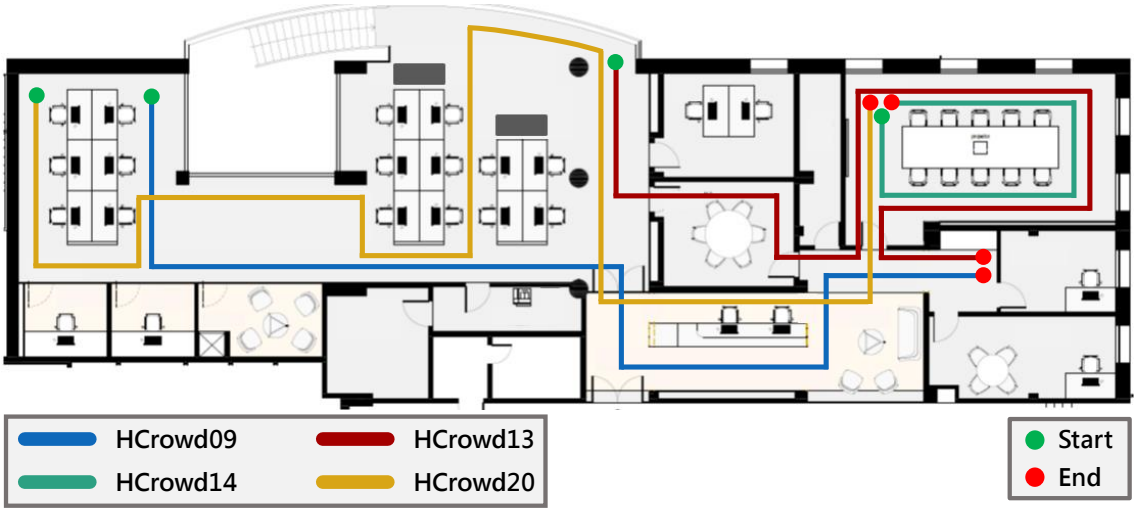


Figure 4.3: Design of four routes of the dataset, displayed in front of the test building's floor plan. Each route is identified by a different colour, as it is described in the Figure's legend. The starting and ending point is as well identified by the green and red circles, respectively.

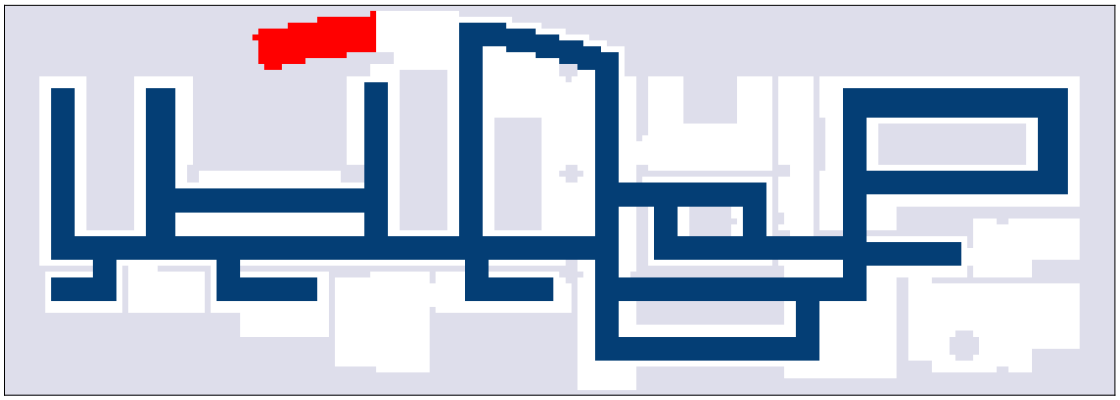


Figure 4.4: Illustration of the areas of the test building covered with routes. An ideal algorithm would construct a floor plan similar to the covered areas of the building.

4.2 Data Acquisition

The data acquisition process was designed considering the characteristics of the acquisitions in a real environment. Six healthy subjects, aged from 22 to 27 years old, collected data through the day and in different days, to consider its variability. A total of 135 acquisition were collected, with several repetitions of the same route, but always in different conditions, whether by different subjects or different acquisition devices. Overall, 95 minutes of data were acquired. The smartphones used for the acquisition were a Google Nexus 5 and a Google Nexus 6P, both running on Android system. These smartphones are equipped with all the necessary sensors to this project.

To acquire the data for this dataset, both smartphones had installed a mobile application responsible to manage all the smartphones' sensors and register all the information

collected by them. This application, named *Recorder*, was previously developed at Fraunhofer AICOS.

It is important to collect data from different smartphones, since their sensors have some different characteristics, although the collected information is the same [30]. Readings from the magnetometer usually have an offset between different smartphones. The sampling frequency of each sensor is another parameter that varies among different smartphones. Table 4.1 shows the sampling frequencies registered for each sensor on each smartphone, in one acquisition of the same route.

Table 4.1: Sampling frequencies of different sensors for each used smartphone. These values were obtained for one acquisition in the **HCrowd00** route, using the Nexus 5 and Nexus 6P smartphones.

	Sampling Rates (Hz)			
	Accelerometer	**Magnetometer**	**Gyroscope**	**Wi-Fi Radio**
Nexus 5	198.82	49.80	198.77	0.32
Nexus 6P	396.25	49.66	396.45	0.67

4.2.1 Sensors Acquired

The data acquired from smartphones' sensors is used in two main phases of this work. Subsection 2.2.2 explains how the sensors from the IMU are used to reconstruct the movement. The sources of information described in Subsection 2.3.2 are not only used in the fingerprinting processes, but will also be very useful for this work in the identification of similarities phase.

Thus, the collected dataset has data from the accelerometer, the magnetometer and the gyroscope, all part of the smartphones' IMU. Regarding the environmental data, the magnetometer also provides the necessary magnetic field information. Besides this, readings from the local Wi-Fi radio signal are also registered.

4.2.2 Acquisition Protocol

The acquisition of the dataset used in this work followed some rules to ensure the integrity and fidelity of the data. Every participant on the acquisition was instructed to walk normally in the designed routes, starting at the initial position, identified by a green dot, and finishing at the end of the route, identified by a red dot (see Figure 4.3). The subject also had to hold the smartphone in texting position, since the variations on the position of the device affect the dead reckoning results, as it is explained in Section 4.3.

Since the collected data will serve as ground truth for the intermediate phases of this work, the acquisitions were duly annotated. The *Recorder* application allows the annotation of the moments of desired events, with a tap on the smartphone's screen.

Thus, every subject was instructed to tap the screen in every turn, providing with great certainty the information of the moments where the subject passed by those points.

4.3 Dead Reckoning Parameters Retrieval

The processing of sensed data from the accelerometer, the gyroscope and the magnetometer allows the reconstruction of the users' trajectories, as it is explained in Subsection 2.2.2, being also used in other steps of the entire process. This data processing is the starting point of this work. The parameters retrieved from dead reckoning techniques are the moments when steps were taken, their corresponding length, and finally, the direction of each step. Alternatively, some algorithms compute the relative direction between a step and its previous. These parameters are all that is necessary for the trajectories reconstruction.

In a real context, the device that performs the data collection for the localisation purposes, automatically computes the dead reckoning parameters immediately after the data has been sensed, and reacts accordingly. In the development of this work, these parameters are not immediately obtained, since the data is processed *a posteriori*. Thus, they are computed by a framework developed at Fraunhofer AICOS, that processes the data collected from the IMU's sensors, as if it was coming in real time, and returns a list with the aforementioned parameters. Relatively to the direction, the used framework computes the direction of every step, by merging the data coming from the magnetometer and the gyroscope. Thus, the movement characteristics are considered to be given as input to the algorithm developed in this thesis.

Chapter 5

Proposed Algorithm and Results

In this chapter, a new algorithm for the automatic construction of indoor floor plans and environmental fingerprints will be presented, along with the results obtained during its development. This algorithm aims to reduce the extensive effort of the floor plans and fingerprints construction process, viewed as one of the major problems of current infrastructure-free IPS. This work was developed in Python 2.7 and its implementation can be divided in several stages, as it can be seen in Figure 5.1.

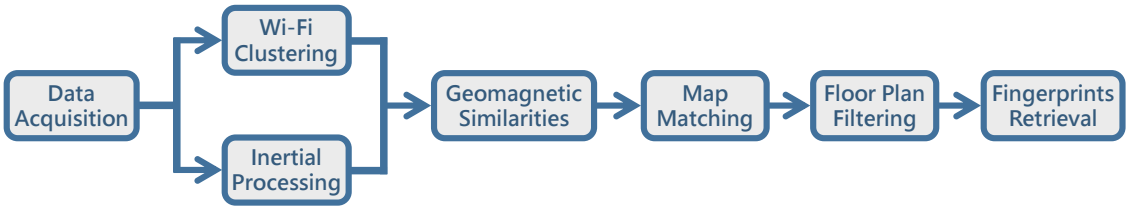


Figure 5.1: Scheme of the proposed algorithm workflow.

After the data acquisition process, the Wi-Fi clustering and the inertial processing stages aim to process the respective types of data, before the identification of geomagnetic similarities of stage three. While the Wi-Fi radio data is clustered to reduce the search area for similarities, the inertial data is processed to understand the walking patterns of the users. After these stages, the floor plan construction will be done in the map matching stage, followed by its filtering. Finally, the environmental fingerprints are obtained from the collected data in the fingerprints retrieval stage.

This chapter addresses every stage of this process, where an explanation will be given, together with the obtained results. Finally, the constructed floor plan and the corresponding fingerprints for the dataset acquisitions' building, described in Chapter 4, will be tested in a real scenario.

5.1 Wi-Fi Radio Clustering

The first main stage of this project consists on the processing of the Wi-Fi radio data, followed by its evaluation by unsupervised machine learning techniques. The clustering of data allows the identification of clusters with similar information. Each cluster represents a smaller area of the building.

The collected Wi-Fi data is received in batches of APs replies. A batch consists on a singular scan of the reachable APs in the area. Depending on the sampling frequencies of the smartphone, after a batch is received, a new Wi-Fi radio request is sent. The APs that are close enough to perceive the request, reply with the information of their signal, as its strength, frequency channel and MAC address. After a predefined interval, a list with all available WLANs is obtained, representing a batch. Every batch is characterised by a timestamp of the instant of the AP reply.

The clustering of the Wi-Fi data is applied to the received batches of each collected route. After the clustering process, a cluster is attributed to each batch. The batches of the same cluster have similar information, meaning that they have coherent strengths for the same APs. With the clustering results, it is expected to be possible to divide the building in small areas, one area for each cluster. Although this process is not indispensable to the final objective, it has a major role on decreasing the processing time, especially in large buildings, as it will be explained in Section 5.3. Before the application of the clustering algorithms, the features to be evaluated must be extracted.

5.1.1 Wi-Fi Data Pre-Processing

Machine learning algorithms work by evaluating the set of features that characterise every object. In this specific case, the objects are the obtained Wi-Fi batches. Before the extraction of features, the data is pre-processed following these steps:

1. A search through the measured Wi-Fi signals is conducted to identify all APs with coverage inside the building. This search considers the following aspects:
    - Networks that transmit only in the 5 GHz band are not considered, since they are not detectable by every smartphone.
    - An AP is considered only if at least two WLANs are detected in a batch. This is done to remove variable APs, as a smartphone hotspots, based on the fact that common APs transmit in more than one WLAN.
2. For each batch, the detected WLANs are analysed. In the cases where more than one WLAN from the same AP is detected, the mean of their strength values is computed.
3. Some APs have weak signal indoors. Therefore, when an AP is missing in a certain batch, this work assumes that their RSS is equal to -100 dBm.

The result of this pre-processing is a list of batches that contains the Wi-Fi signal strengths for all the detected APs in the building.

A further optional processing is implemented, where an evaluation of the obtained list is performed, and the Wi-Fi signals strengths readings with low values are neglected. A low value is due to a reading that was received far from the AP, being for this reason very susceptible to interferences. For the tested dataset, this threshold was established as -90 dBm. With this, values below -90 dBm are replaced by -100 dBm. After this, the APs that do not have meaningful values are removed from the list.

5.1.2 Features Extraction

With the Wi-Fi radio information properly organised, the study of the best features can be done. Different methods compare the data in different ways, reason why distinct clustering results are expected. Thus, several features were implemented to test which one gives the best results:

- **RSS values**: This method does not require any further processing. The features that characterise each batch are the Wi-Fi signals strengths values in the list previously obtained.

- **Difference of RSSs**: Based on the fact that readings from different devices might have some variability, comparing directly the RSS values might not give the best results. Thus, to only consider the relative differences between the APs of a batch, a different approach is required. The developed method computes, for all batches, the differences between the registered RSSs of every pair of APs.

- **Exponential representation**: Considering that signals collected far from the AP are more suitable to noise, Torres-Sospedra et al. [65] introduced an exponential representation, in order to give higher values to higher RSSs, as well as the opposite, in a non-linear way. This representation of the data is also implemented, where the RSS values are converted with the following equation:

$$Exponential_i(RSS_i) = \exp\left(\frac{RSS_i}{\alpha}\right) \tag{5.1}$$

  where α was defined by the authors as 24, through an empirical process.

- **Sigmoid representation**: In the sequence of the last representation, the RSS values can be converted by a sigmoid function, where the values higher than a threshold are valued as 1, while the values below a lower threshold are valued as 0. The sigmoid function is given by:

$$Sigmoid_i(RSS_i) = \frac{-1}{1+\exp(\alpha \cdot x - \beta)} + k \tag{5.2}$$

where α was empirically defined as 0.145, β as -9.3 and k as 1. With these parameters, values above -40 dBm are converted to 1, and for lower RSSs, the converted value decreases until it reaches 0, at approximately -90 dBm.

- **Difference of new values**: With the same premise of the difference of RSSs method, the same process is implemented for the exponential and sigmoid representations.

The study of the best method can only be done after the implementation of the clustering algorithms, since it is the only way to evaluate if each method gives a good relation between the data.

5.1.3 Clustering Algorithms

The unsupervised machine learning techniques are exploratory in nature. These algorithms aim to find an organisation among the data, resorting in clusters that separate the data according to their similarity. The number and the shape of resulting clusters can be previously foreseen if the characteristics of the dataset are well known. However, the idea of the algorithm to be developed in this thesis is to automatise the process of floor plans construction and fingerprints mapping.

As it was explained, the application of clustering in the collected Wi-Fi data aims to divide the building into smaller areas that have coherent information. Considering that this solution aims to be applied in any unknown building, it is impossible to predict both the number of clusters and their shape. Thus, the clustering algorithms applied cannot require the number of clusters as input. Besides this, since the shape of the clusters also cannot be predicted, the evaluation of the results, to decide the best clustering algorithm, must be done visually, considering the available dataset.

The following algorithms were tested for the acquired data, with a detailed description of their functioning available in Subsection 2.6.2:

- **Agglomerative Clustering**: The fundament of this algorithm implementation to cluster the Wi-Fi data was to test if a hierarchical approach would give proper results. It was implemented using the *scikit-learn* package for Python [80]. This specific implementation has a predefined number of clusters as stopping criteria.

- **K-Means**: Being one of the most used clustering algorithms, K-Means was applied to the Wi-Fi data. This partitioning approach is also implemented in *scikit-learn* package [80], which has as input the number of clusters on which the data will be divided.

- **DBSCAN**: This density-based approach was also tested in this work, in order to understand if the identification of high density regions, with similar information, would give good results. This method also allows the identification of outliers, which are objects that do not match any of the clusters. This algorithm is available in *scikit-learn* package [80], having *Eps* and *MinPts* as input parameters.

- **HDBSCAN**: This algorithm is a variation of DBSCAN. It uses a hierarchical approach to optimise the *Eps* parameter of DBSCAN, since it is very difficult to predict for a high number of features. HDBSCAN is implemented in a package developed for Python that has a set of tools to perform HDBSCAN [81]. The only required parameter is the minimum cluster size.

5.1.4 Clustering Results

The identification of the best algorithm to cluster the Wi-Fi data was done through an iterative process, where different algorithms and different features were tested. The best algorithm is the one that is capable of dividing the Wi-Fi batches in consistent clusters, where the cluster attributions are coherent through the badges of every route. The number of clusters is secondary, since it is preferable to have fewer clusters with great consistency, than the opposite.

In order to evaluate the quality of the combinations of different algorithms and results, a visualisation tool was developed. For each designed route, using their annotated coordinates, it was possible to compute their full length. Then, a linear regression between the full time that took to a user to acquire the data, and the timestamp of every batch, it was possible to identify the approximate coordinate where the batch was acquired, within the trajectory. Figure 5.2 displays a map of the test building of this work. On top, four acquisitions of the dataset, corresponding to four different routes, are mapped. Their design is available in Figure 4.3. The acquired Wi-Fi batches of each route are scattered in Figure 5.2, where each polygon represents the batches of a route, as it can be seen in the legend of the map. This map, together with the same routes, will be used to visually demonstrate the clustering results.

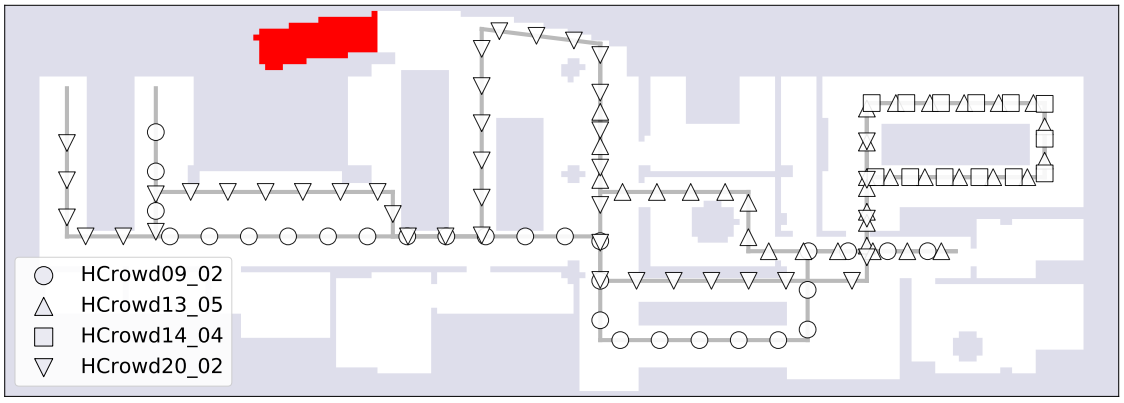


Figure 5.2: Four routes from the dataset with collected Wi-Fi batches, before the clustering process, plotted in the corresponding positions of the test building. The batches of each route are identified by the polygon described in the plot's legend.

Since the goal was to implement a method that does not require the number of clusters, DBSCAN was the first algorithm to be tested. The *MinPts* parameter was established at 10, a small value, to allow the creation of clusters that would represent a small area in

the building. Since the *Eps* parameter cannot be easily predicted in datasets with a high number of features, its value was progressively changed until a solution with satisfying results would be obtained. However, it was not possible to obtain a *Eps* that produced satisfactory results. The *Eps* value is very hard to predict when the number of features is large, since it works as a radius on which an object will search for close neighbours. The variations on the type of the features did not improve the results. Testing progressively incremented *Eps* values, it was possible to understand that, for a acceptable number of clusters for the test building, as five for example, the number of outliers reached 86% of the total number of batches. In Figure 5.3 it is possible to see this result, where five clusters were obtained with an *Eps* value of 2.2. The outlier batches are identified by the white colour, while each cluster is coloured differently. In this example, the used features were the original RSS values.

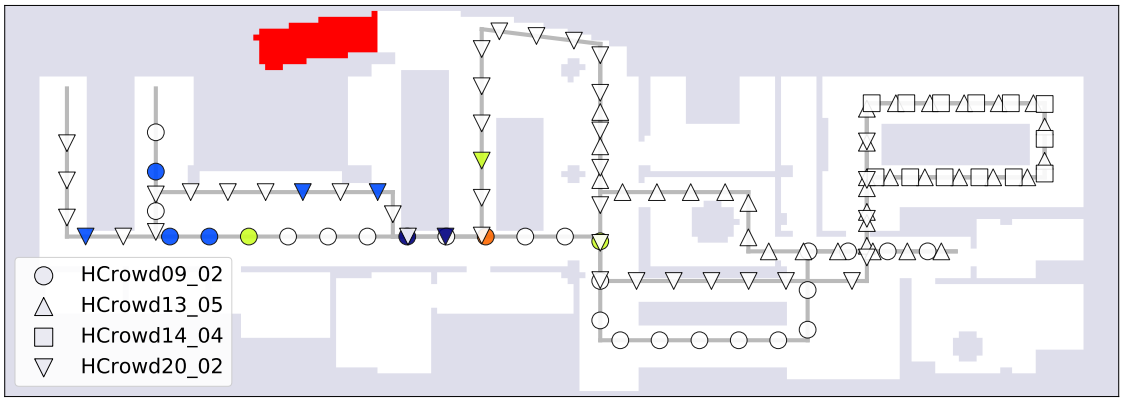


Figure 5.3: Results of DBSCAN algorithm on four routes of the dataset, plotted in the corresponding positions of the test building. The original RSS values were used as features, and the *Eps* value was established as 2.2. The number of obtained clusters was 5, each represented by a different colour. The white filled polygons are the outliers.

HDBSCAN is an algorithm that hierarchically implements DBSCAN to avoid the definition of the *Eps* value as parameter. Initially, *MinPts* was defined as 10, the same number used in DBSCAN. However, the algorithm divided the batches in 28 clusters that were not uniformly distributed. Besides this, the number of outliers still was too high, reaching almost 31% of the batches. Increasing the *MinPts* value, the acceptable number of clusters, established as five, was obtained with *MinPts* as 75, where the cluster distribution for the routes of the example is available in Figure 5.4. The type of features did not influenced the results. However, the high percentage of outliers, almost 67%, and the distribution still were not satisfying, which is why the search for new algorithms continued.

With the aforementioned experiments, it is possible to understand that a density-based approach does not work for this type of data. The hierarchical approach, implemented by the agglomerative clustering method, was tested next. Since the algorithm starts by attributing a cluster to each batch, and progressively merges them, the final

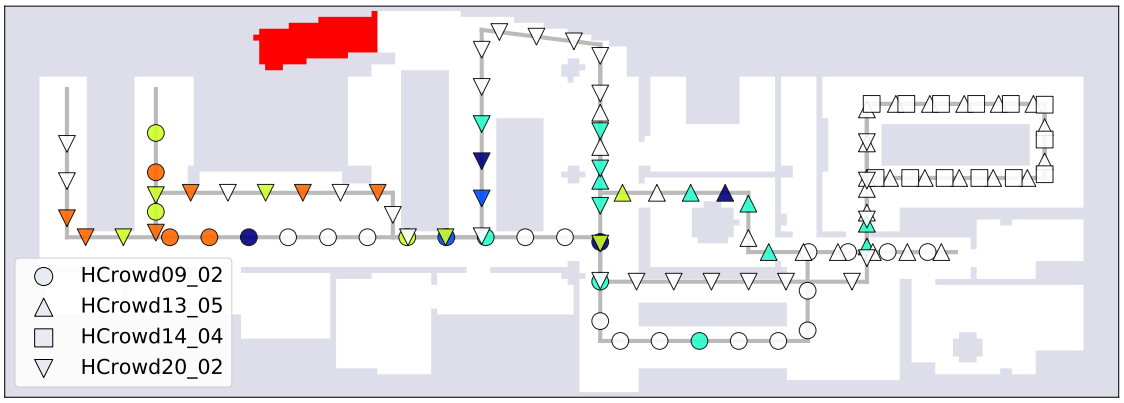


Figure 5.4: Results of HDBSCAN algorithm on four routes of the dataset, plotted in the corresponding positions of the test building. The differences between the RSS values were used as features, and the *MinPts* value was established as 75. The number of obtained clusters was 5, each represented by a different colour. The white filled polygons are the outliers.

result will not have outliers. The only parameter of this method is the final number of clusters. Although this requirement may become a problem in a final solution, it was still worth to test this hierarchical approach. Starting at the five clusters, the algorithm was tested with the dataset. The results showed that this number of clusters was to high, since the cluster distribution were not very coherent, for every type of features. Progressively decreasing, the best outcome happened when the number of clusters was defined as two, with the RSS values converted by a sigmoid, as it can be seen in Figure 5.5.

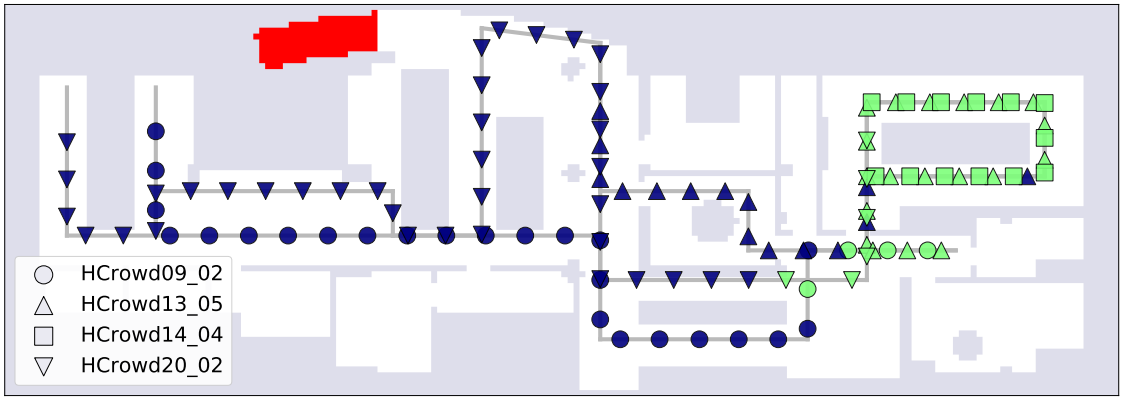
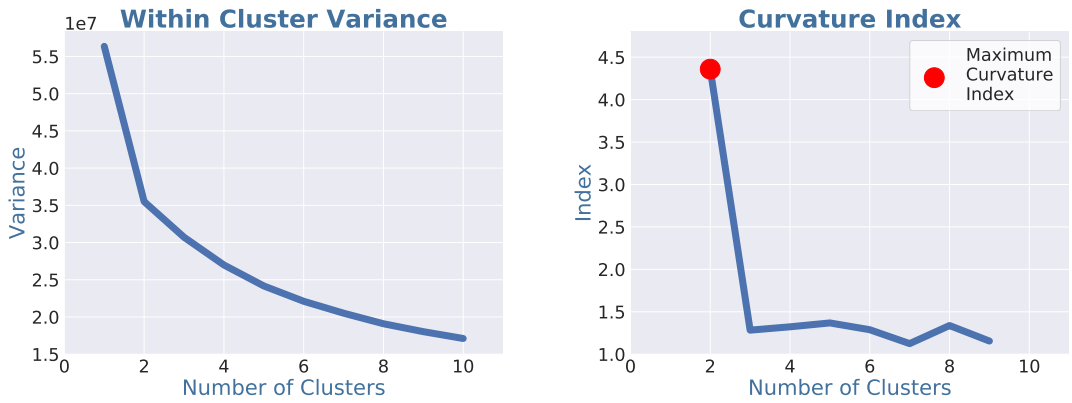


Figure 5.5: Results of agglomerative clustering algorithm on four routes of the dataset, plotted in the corresponding positions of the test building. The RSS values converted by a sigmoid were used as features, and the number of clusters was established as two. Each cluster is represented by a different colour.

Since it is not possible to use the agglomerative clustering without having the number of clusters as inputs, another algorithm was tested. As a partitioning algorithm, K-Means iteratively adjusts its cluster attributions until a stable solution is found. The used implementation also has the number of clusters as input. However, some methods were

developed that aim to optimise the final number of cluster. The elbow and the curvature-based methods (Subsubsection 2.6.2.2) were thus implemented, where the tested number of clusters varied from one to ten. The best results for this algorithm were obtained using the differences between the RSS values as features. As it is possible to see in the plots of Figure 5.6, the ideal number of clusters for this dataset is two, which corresponds to the maximum index of Figure 5.6b. It is also possible to verify that this number of clusters is identified in Figure 5.6a by the elbow, the point where the biggest variation of curvatures exist. The final distribution of clusters in the example routes is available in Figure 5.7.



(a) (b)

Figure 5.6: Results from the elbow (Figure 5.6a) and the curvature (Figure 5.6b) methods, after the application of K-Means clustering algorithm, for a number of clusters between one and ten, using this work's dataset.

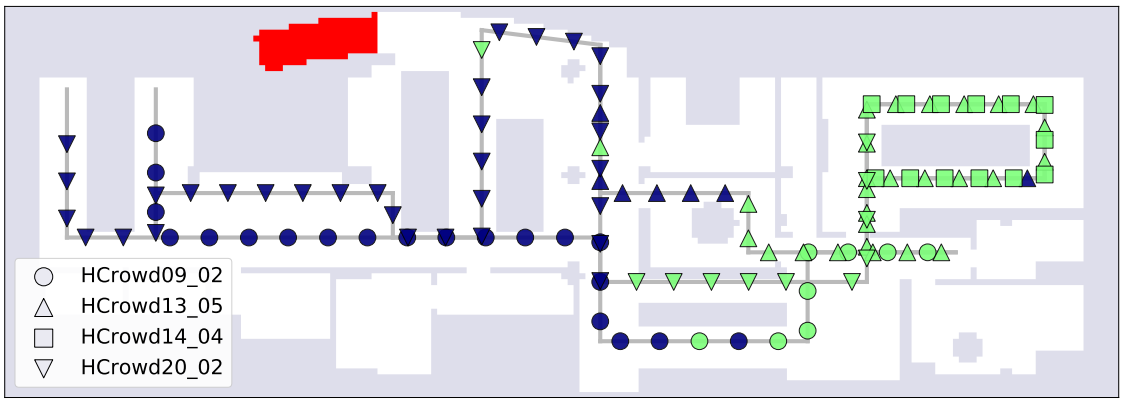


Figure 5.7: Results of K-Means algorithm on four routes of the dataset, plotted in the corresponding positions of the test building. The differences between the RSS values were used as features, and the number of clusters obtained by the elbow and curvature-based methods was two, each represented by a different colour.

From all the tested algorithms, the best results were obtained with K-Means, which is why this algorithm is the one that is used in the development of the algorithm. The differences between the RSS values are the features that will further used, since they

were the ones that provided the best results. However, it is possible to see in Figure 5.7 that some wrong cluster attributions happened, since sometimes a colour that is different from the neighbourhood is shown. Thus, an algorithm was created to correct these incoherent transitions between clusters. It works by changing the cluster of a batch if the three batches before and the three after coincide, and are different from the centre batch. Figure 5.8 shows the final results of clustering process, after the correction process, for the four routes used as example.

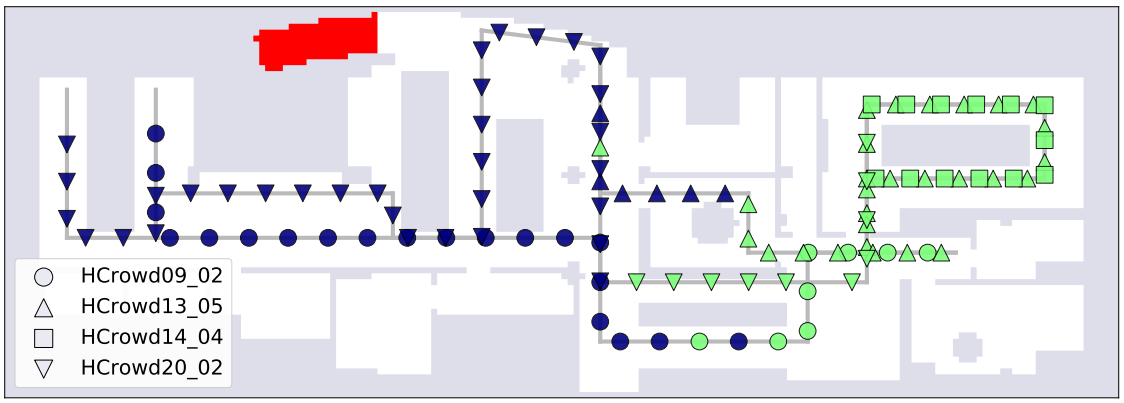


Figure 5.8: Final results of the clustering process after the noisy batches removal, on four routes of the dataset.

With the results from clustering process, it is possible to know which parts of the acquisitions were collected in the same area. With this information, a list is created to store the initial and final timestamps of each segment with the same cluster, in every route. Some requirements are applied to this registry process. As it will be later explained, a minimum length of each partition is required. Thus, the annotation of the timestamps only happens if at least three consecutive batches of the same cluster are identified. Furthermore, since the exact moment of the cluster transition is not known, due to the low sampling frequency of Wi-Fi acquisitions, the timestamp of the batch before the first batch of a cluster is registered, as well as the timestamp of the batch after the last batch of the same cluster. Figure 5.9 illustrates this process. In an acquisition that took 40 seconds, where 22 batches were collect, the results of the clustering process are represented by the green and red filled rhombuses. The limits of the black arrows are the stored timestamps, taking in consideration the aforementioned rules.

5.2 Data Processing

The second stage of this process consists in the processing of the acquired data, in order to prepare it for the next stages of this work. This processing regards the inertial data obtained from the IMU of the smartphones and the collected data from the Wi-Fi radio and the magnetic field is also processed.

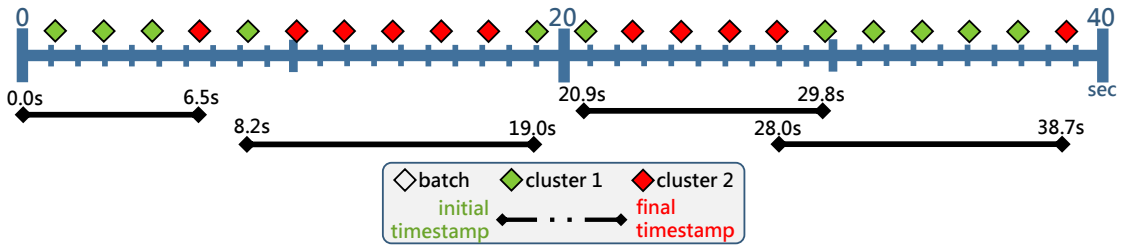


Figure 5.9: Illustration of the timestamps annotation rules with the results of clustering process. For each route, the initial and final timestamps of segments with consecutive concordant clusters are annotated in the black arrows, considering that a minimum of three batches are required. The annotated timestamps are the timestamps of the immediately before and after the first and last batch of the segment.

5.2.1 Trajectories Reconstruction

The inertial data that is firstly processed was previously obtained in the dataset construction process, as it is explained in Section 4.3. The timestamp and the length of each step, as well as their direction, are the parameters retrieved after the dead reckoning techniques.

With these parameters, it is possible to infer the trajectories described by the users. The process that allows the reconstruction is the following:

1. The initial position, in a two dimensional plane, is defined as (0, 0) for every route. The direction of the movement, given in radians, varies accordingly to the illustration of Figure 5.10.

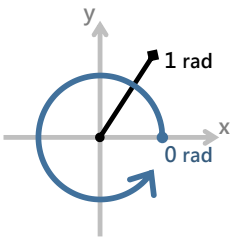


Figure 5.10: Representation of the direction variation circle, in radians.

2. At each step, the next position is obtained by summing to each coordinate of the last position, the coordinates of the displacement of the new step. The displacement has to consider the direction of the movement, so its coordinates are obtained by multiplying a rotation matrix by the length of the step, as if it was done to the 0 rads angle direction. The following Equation describes this process:

$$(x_{next},\ y_{next}) = (x_{prev},\ y_{prev}) + (disp,\ 0) \cdot \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{5.3}$$

where (x_{next}, y_{next}) and (x_{prev}, y_{prev}) are the coordinates of the next and the previous positions, respectively. $disp$ is the displacement of the new step, and θ its direction.

Figure 5.11 shows the reconstruction of one of the routes used as example. As it is possible to see, by comparing this reconstruction to the original designed route, drawn in the background of the map of the same Figure, the reconstruction has some problems. The framework used to compute the dead reckoning parameters has some errors in the computation of the steps' length, reason why the full length of the original and reconstructed routes differs. However, it must be taken into consideration that some of this error is explained by the variability of the users' movement, in relation to the trajectories that were designed. Furthermore, the direction of the movement also originates some error, since the drift originated by the gyroscope might not be completely eliminated by the employed correction mechanism.

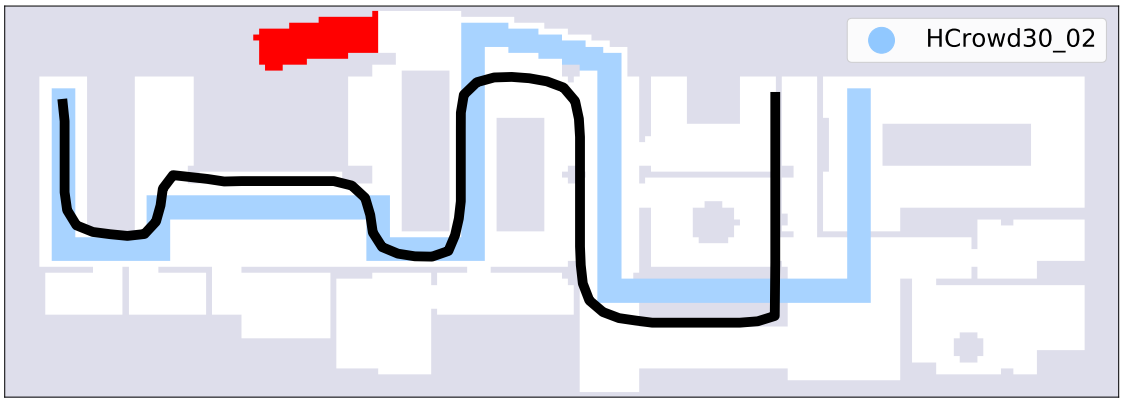


Figure 5.11: Result of the reconstruction process for one route of the dataset, plotted above the buildings map, in black. The dimension of both the map and the route are proportional. The real designed routes is drawn in blue colour.

After the routes are reconstructed, information about their direction will be retrieved. Since the algorithm of the floor plans reconstruction will be based in comparisons between all the crowdsourced data, which is not annotated, it is important that these comparisons are done with high confidence data, to reduce the hypothetical accumulated error.

As it was previously explained, the retrieval of the direction information has some error. Besides this, this values are highly influenced by the user. Several situations exist where a turn in the movement can be detected, while the user continues to walk straight, as a simple tilt on the smartphone, or a hesitation during the movement inside a building, where the user might turn over to read some information, for example. Thus, the information of the changes of direction is not very trustworthy.

To overcome these problems, the comparisons between the collected data will only take place when is certain that the user walks straight, or, in other words, when the direction of consecutive steps remains constant. To do this, at this stage, the initial and

final moments of the straight line paths are annotated in a list, by the registry of the timestamp of the first step where the user walked straight, as well as the last.

Besides this, as in clustering process, a minimum number of steps is defined to the segment to be considered. This value was established as five. An example that illustrates the annotation of the first and last steps of the straight line paths is available in Figure 5.12. As it can be seen in this Figure, the direction information of all steps is plotted, where each step instant is identified by the red dots. Although there are several straight sections, their annotation only occurs if at least five consecutive steps have the exact same direction value, as the limits of the black arrows show. In almost every identified straight section, the neighbouring steps have an approximate direction. However, they are not considered to avoid at most the susceptibility to comparison errors.

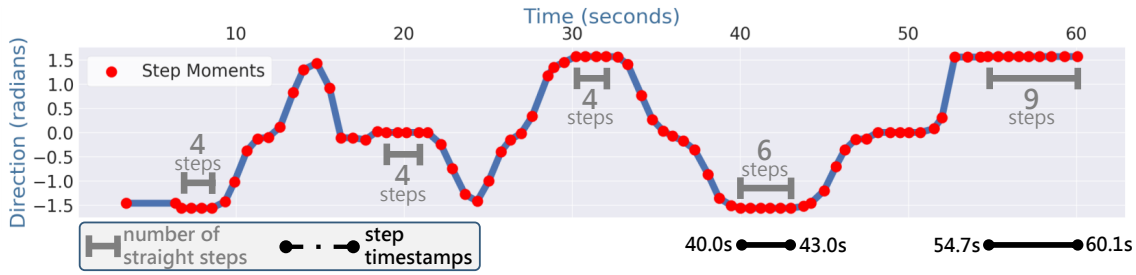


Figure 5.12: Illustration of the straight sections identification rules, with information of the direction of every detected step. A straight section is identified every time at least five consecutive steps with the same direction are detected.

5.2.2 Time to Distance Domain Conversion

Before the comparison of the magnetic data and the later fingerprints retrieval, some processing has to be done. The walking pattern of every user has great variability, where different conditions affect the way the users walk. For example, an elder person will walk slower than a young adult. Furthermore, a person that knows the building will walk with more confidence than a person that is there for the first time. Then, the acquisition of the data will be affected. The higher the speed of the user, the lesser the points that will be collected through the same distance.

To solve these issues, the data is converted to the distance domain. This means that the data will not vary with time, which is dependent of the sampling rates of each sensor, but will vary with distance, by a fixed step value. This way, all the data is interpolated to match the corresponding distance points. Due to the nature of the data, the magnetic field and the Wi-Fi radio are processed in different ways.

The domain conversion of the magnetic field data can be done practically in any chosen step distance, due to the high sampling rates of common magnetometers. The step value was established as 10 centimetres. The conversion process identifies the approximate timestamp that corresponds to every distance point, progressively incremented by 10 centimetres, until the full length of the route is reached.

To do this, the algorithm processes at each time a pair of two consecutive steps. By knowing the accumulated displacement at each step, the algorithm computes a linear interpolation between the known timestamps and accumulated displacements, to obtain the timestamp that corresponds to the distance point being analysed. After the obtainment of the timestamp that corresponds to the 10 centimetre multiple distance point, the magnetic data of every axis is annotated. An illustration of the process of identifying the timestamp that corresponds to the displacement of 5.30 m is illustrated in Figure 5.13. In this example, the use of the timestamps and displacements of steps number 7 and 8, allows the computation of the unknown timestamp, given by the result of Equation 5.4:

$$t_{unknown} = t_{step7} + \left(d_{point} - d_{step7}\right) \frac{t_{step8} - t_{step7}}{d_{step8} - d_{step7}} \tag{5.4}$$

where t_{step7} and t_{step8} refer to the timestamps of steps number 7 and 8. Their displacements are represented by d_{step7} and d_{step8}, respectively. d_{point} refers to the displacement of the distance point that is being interpolated, to obtain the corresponding timestamp, $t_{unknown}$.

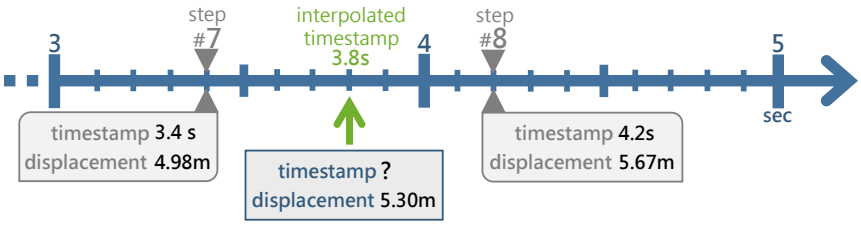
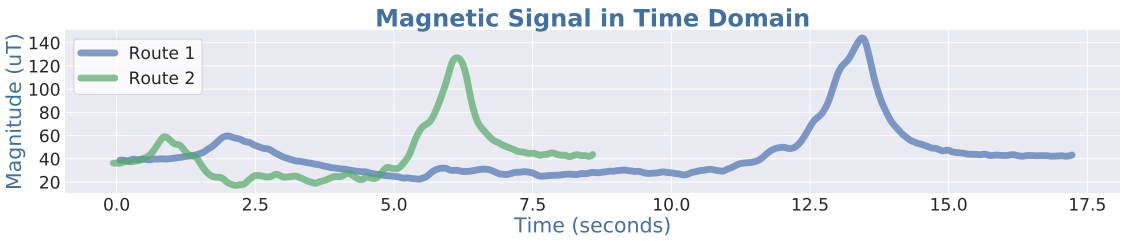


Figure 5.13: Illustration of the interpolation of one distance point, in the time to distance conversion process. For each displacement of the distance domain, as 5.30 m in this example, it is possible to obtain the corresponding timestamp of the time domain. It is done by applying the linear interpolation of Equation 5.4, with the information of the neighbouring steps.

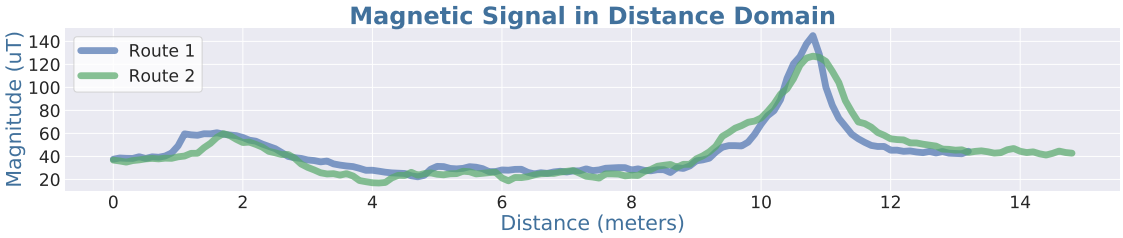
After the conversion process, it is possible to deal with the data as if it was acquired between intervals of 10 centimetres, instead of the varying time intervals. Figure 5.14 shows the final result of the domain conversion, for two routes collected in the same section of a building, but at different walking velocities.

The conversion of the Wi-Fi radio data to the distance domain is done differently than the magnetic field. As it was already explained, the sampling rate of the Wi-Fi signal is lower than the magnetic field's, where the most common value in smartphones is 0.29 Hz. Between two batches, the users may walk several meters. Therefore, a sampling distance of 10 centimetres cannot be obtained.

Instead, the process of conversion works in a simpler way. It just computes the approximate distance that corresponds to the timestamp of every batch. Thus, for a batch, an iteration over consecutive pairs of steps of its route is done. The iteration stops when the timestamp of the batch is included in the time interval between two consecutive steps.



(a)



(b)

Figure 5.14: Results of the magnetic signal conversion from time to distance domain, for two routes collected in the same sections of a building. As Figure 5.14a shows, both routes were collected in different walking velocities, since the length of both signals is different. After the conversion, Figure 5.14b shows both routes in the distance domain, as it can be seen from the alignment of both signals.

Then, a linear interpolation is done, considering the timestamps and the accumulated displacements of the referred pair of steps.

Finally, through the same process of the Wi-Fi radio data conversion, the lists that store the timestamps that identify sections with the same cluster and the timestamps of the straight sections are also converted to the distance domain.

5.3 Geomagnetic Similarities

The third stage of this process regards the identification of similarities between the collected data. This identification aims to find overlaps between routes. An overlap happens when two segments of two different routes are collected in the exact same place. Since there is no annotation of the absolute positions where the routes are acquired, the identification of overlaps relies on the evaluation of the data collected from the sensors of the smartphone.

As it was introduced in Subsubsection 2.3.2.2, the magnetic field pervasively available in every building has great potential for the similarities identification. The unique patterns created by the construction materials and electrical equipment in every building motivate the comparisons between magnetic field data, to identify similar signals, which correspond to overlaps between routes. For this reason, the process of identifying overlaps is done, in this work, with comparisons between the magnetic field data.

After the magnetic data is obtained, and before the further mentioned processes, a

smoothing filter was applied to the data, in order to reduce the acquisition noise. The filter in this algorithm is implemented in the *novainstrumentation* package for Python[1].

5.3.1 Data Segmentation

Before the data comparison, a segmentation process divide the data into interesting sections, using the values previously obtained by the clustering stage and the identification of straight sections.

This process deals with the clustering results first, by segmenting the magnetic data by the intervals retrieved at this stage. These intervals are now represented by the distances that correspond to each timestamp. Then, the segmentation by straight sections takes into consideration the already segmented data. Within these segments, the algorithm will verify which portions are included between the intervals of the straight sections list, to segment the data again.

However, the previous requirement of a minimum number of steps is still considered, and only the new segments that fulfil the minimum of five steps are retrieved. This requirement is very important for the further comparisons. For example, if two very large segments are identified to be similar, it is possible to be more sure that they are effectively an overlap, than if two very small segments are identified, since the small pattern might happen in more than one place of a building.

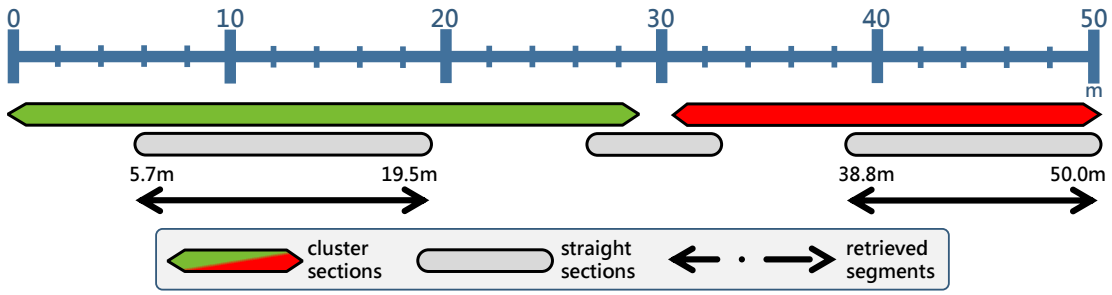


Figure 5.15: Illustration of the data segmentation process, considering the information of the previously defined sections with the same cluster and sections with the same direction, as it is described in the legend. The segmentation only happens when the new segments have a minimum number of five steps.

After the segmentation process, the retrieved segments to be compared in the next stage are a small portion of the full acquisitions, as it can be seen in Figure 5.15. However, this few number of sections to be compared do not represent a problem, since the data collected with crowdsourcing comes in large quantities. Moreover, it is preferable to only use reliable data in the floor plans and fingerprints construction process, to avoid the addition of errors.

[1] Available in https://github.com/hgamboa/novainstrumentation (visited on 09/17/2018).

5.3.2 Time Series Similarities

The process of identifying overlaps between the collected data is one of the most important stages of this thesis. It is done through the comparison of the previously retrieved segments. The algorithm compares every pair of segments in the three axes, as long as they are labelled with the same cluster.

Since the magnetic field data is available in the distance domain, instead of the time domain, it is possible to consider that all the data is aligned. For this reason, both lock-step and elastic measures (Subsection 2.5.1) can be used for this purpose. Having this in mind, and due to the fact the DTW is computationally heavy, the WPA was the chosen measure, being able to find the optimal alignment between a window of a given time series to another full time series.

Before the data comparison, WPA requires the identification of the window to be slid. Since this comparison aims to take advantage of the unique pattern of the magnetic field, the window identification process starts by identifying the maximum peak of every segment's magnitude. Then, the window for each segment is retrieved around its identified peak, with a defined length of 50 points (5 meters), except in the cases that the segments are smaller.

With these non-annotated data, it is impossible to know if two acquisitions were retrieved in the same or opposite directions. To solve this, each window will slide through every segment in both directions, as long as they have the same cluster. The result of the WPA measure is a function that stores the distance of Equation 2.5, for each alignment of the window and the signal. The potential overlapping point is the index where the function has the minimum value. An overlap is identified every time the value of the minimum index is below a threshold, that was defined empirically to be 0.005, but can be changed depending on the need of more or less overlaps. If the minimum of the computed function is below the threshold, the algorithm tries to extend the window, within the limits of both segments that are being compared, and recomputes the WPA measure.

Every overlap is characterised by an importance index, given by the minimum distance of the last WPA function, obtained with maximum achieved length of the window. The lower the index, the more the confidence in the overlap.

After the identification process, this stage returns a list that contains every identified overlap, with information about their characteristics, to be processed in the map matching stage. For each overlap, the list stores the identification of the segments of both routes, the positions of the overlap in both segments, the direction of the comparison, the size of the window and finally the distance value that characterises the quality of the overlap.

For the dataset of this work, the similarities identification process applied to the 172 segmented straight sections found 414 overlaps. In order to verify the quality of this results, to every segment of each route an index was attributed, that identifies the corridor of the building where they were collected. This process was done automatically

by a developed algorithm that processes the annotated coordinates of all routes, and the taps on the smartphones made by the users, as it is described in Subsection 4.2.2. Since a tap was made in every turn, which registered their timestamps, it was possible to identify the corridor where the segment was collected. Thus, by comparing the location indexes of the overlaps, it was possible to conclude that, among all identified overlaps, only two were not correct.

Besides this evaluation, a total of 6048 overlaps actually happened. Although the 414 identified overlaps might seem low comparing to the total number of real overlaps, it is preferable to have more certainty in the used data, discarding some part of it, than the opposite. With the large amount of available data, that continues to grow with the deployment of the system, it is possible to only rely on a small portion of data, to obtain the best results. As it will be seen in the next stages, these assumptions are sustained by the results that were obtained.

5.4 Map Matching

The fourth stage of this algorithm consists on the construction of the buildings' floor plan, a process from now on called map matching. Contrarily to this thesis PoC, where a map for every possible combination of overlaps is constructed, the amount of data that is dealt in the solution with real data does not allow this approach. Instead, the algorithm processes one overlap at the time, and tries to conjugate all the information to construct as few maps as possible.

The constructed maps are two dimensional matrices, where every cell represents a relative position (x, y). The resolution of the constructed maps was defined as one square meter, since it is considered to be a sufficient approximation to provide satisfactory localisation results. The construction will begin at the centre of the matrix, where the first route will be mapped. Then, every cell will store the number of routes that passed by. The final constructed map can thus be viewed as a greyscale image, where the denser cells represent the places of the building where more users passed. These cells are expected to be the main corridors of a building.

An example of the map matching process is illustrated in Figure 5.16. The algorithm starts by ordering the retrieved overlaps from the last stage, by their importance index. This way, the overlap with the highest similarity value is processed first, reducing the possibility of error. Stage 1 of the first iteration matches the routes of the first overlap, where their common segment is identified with the red colour in Figure 5.16. Then, at stage 2, the algorithm searches over all overlaps to select the ones that have one of the already matched routes, with the same segment in common. Then, the corresponding routes of these retrieved overlaps are matched. After this process, in stage 3 of the same Figure, the algorithm retrieves, from the remaining overlaps, every overlap that has in common the segments of the routes already in the map. When the remaining overlaps

have neither of the routes in the map, a final map is reached. Until all the overlaps are mapped, this process restarts to create as many final maps as necessary.

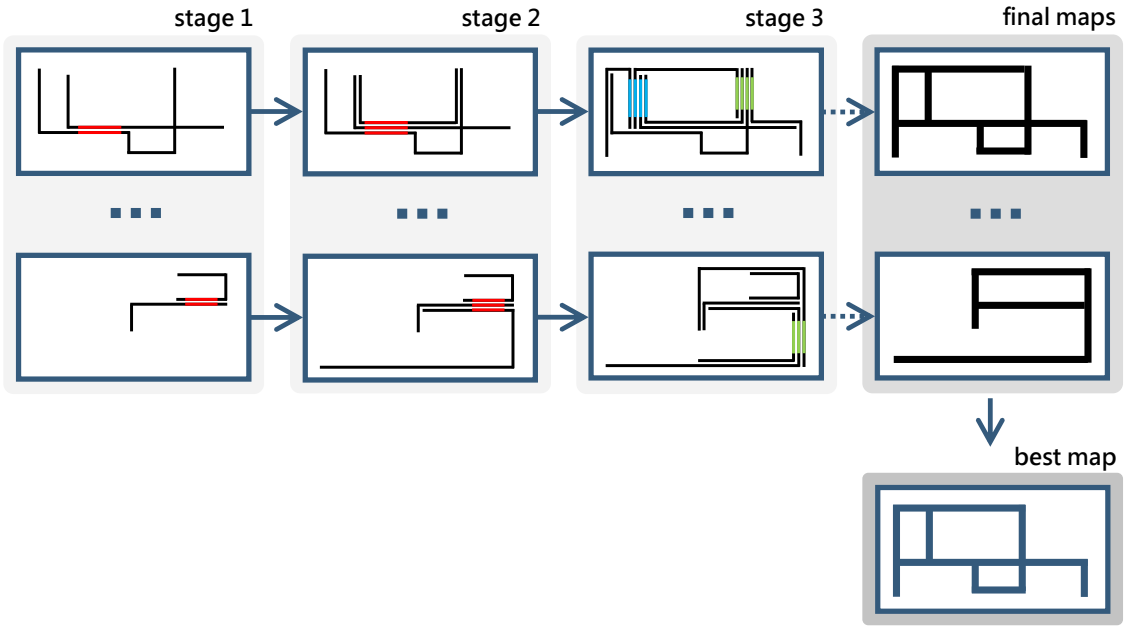


Figure 5.16: Illustration of the map matching process, where several maps are constructed. In stage 1 the best non-mapped overlap is chosen, and the corresponding routes are matched. The stage 2 maps every overlap that happened in the same segment, identified by the red colour. In stage 3, overlaps that happened in the other segments of the mapped routes will also be mapped, until the final map is reached, where no more overlaps can be mapped. Finally, the best produced map is chosen.

In more detail, an overlap identifies common segments between two routes. The overlaps can only be mapped when one of the routes is already in the map. The matching process firstly identifies the positions of the overlapping segment in the already mapped route. The new route is matched by the transformation of its positions, ensuring that the overlapping segment has the same coordinates in both routes. If the overlapping segment in both routes obtained different orientations in the trajectories reconstruction process, due to erroneous estimations of the dead reckoning techniques, a rotation matrix given by the difference between the two segments' directions is applied in order to rotate the new route into the same orientation of the mapped one. Additionally, a translation is also applied, ensuring the same coordinate system between routes. Finally, the new route, that already has the same coordinates of the mapped on the overlapping segment, is added to the matrix that stores the map. Each cell of the matrix, representing a coordinate of the space, stores the number of times that a route passed by. Thus, the process of adding a route to the map consists on making an increment to the values of the matrix's cells with the same coordinates of the positions of the new route.

After the map matching process is finished, a list with all the created maps is obtained. The possible high number of maps is mainly due to the high variability of the data, which

might affect the overlaps identification in some areas. Although all the constructed maps might carry correct information about the building, the map with the highest number of routes is chosen as the final floor plan, since it is the one that contains more information. In Figure 5.17, the final result of the map matching process is represented.

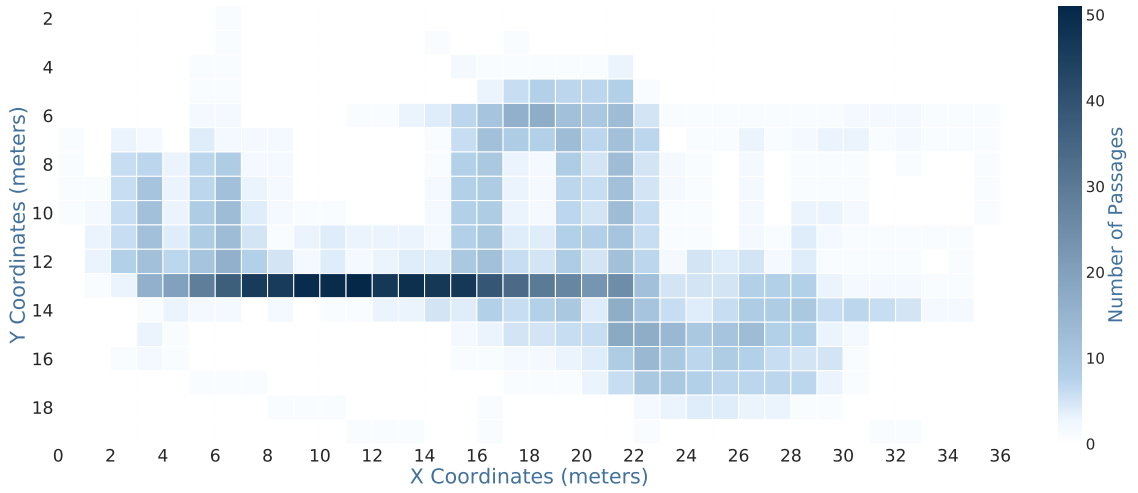


Figure 5.17: Resulting floor plan from the map matching process. Each cell of the matrix stores the number of times that a route passed by, as it is possible to understand in the scale of the Figure.

The obtained map contains the reconstructed trajectories of 58 routes. Comparatively to the total of 135 routes, this number represents 43% of the collected dataset. This apparently low number does not represent a problem, since the large amount of available data allows the exclusion of acquisitions with low confidence. Even with the major part of the dataset excluded from the constructed floor plan, the shape of the building is still defined, as it can be seen in Figure 5.17. The excluded routes happened for two reasons. The first is the fact that the collected signals have high variability, be it for the different sensors' characteristics of the used smartphones, or for the conditions of the acquisition, as the time of the day, or the presence of non-fixed objects that cause interferences. The second reason is the fact that the segmentation constrains sometimes eliminated all the data of a route, due to its small segments.

5.5 Floor Plan Filtering

As it is possible to conclude from the evaluation of Figure 5.17, the produced floor plan has an approximate shape of the covered areas of the test building, available in Figure 4.2. However, it is possible to see that floor plan has its denser areas blurred, mainly due to the dead reckoning errors. To improve this aspect, and to produce a clean floor plan suitable to be used in fingerprinting-based IPS, the results from map matching are subject to a process of filtering.

The final output of this stage will be a binary floor plan, that identifies the walkable and non-walkable areas. To decide which filtering protocol is the best for the produced floor plans, several methods were tested. The following methods were implemented and tested individually and in combinations:

- **Noise Removal**: This operation aims to remove the pixels, or the positions, that have a value below a threshold. To avoid the manual determination of these threshold, the following measures were tested:
  - Mean of all map's cells.
  - Mean of a cell's neighbourhood.
  - Median of all map's cells.
  - Percentile of several values, for all map's cells.
- **Closing**: This morphological operation (Subsection 2.5.2) is applied with the objective of removing the gaps between the cells, wrongly erased by the noise removal operations.

After an evaluation of all the aforementioned techniques, the combination that produced the best results consisted in the application of a noise removal below the mean of the full map, followed by a new noise removal below the median. Finally, a closing operation was applied to correct the errors created by the noise removal processes. The closing operation uses an implementation from the *scikit-image* package for Python [82]. Figure 5.18 shows the final floor plan obtained with the developed algorithm.

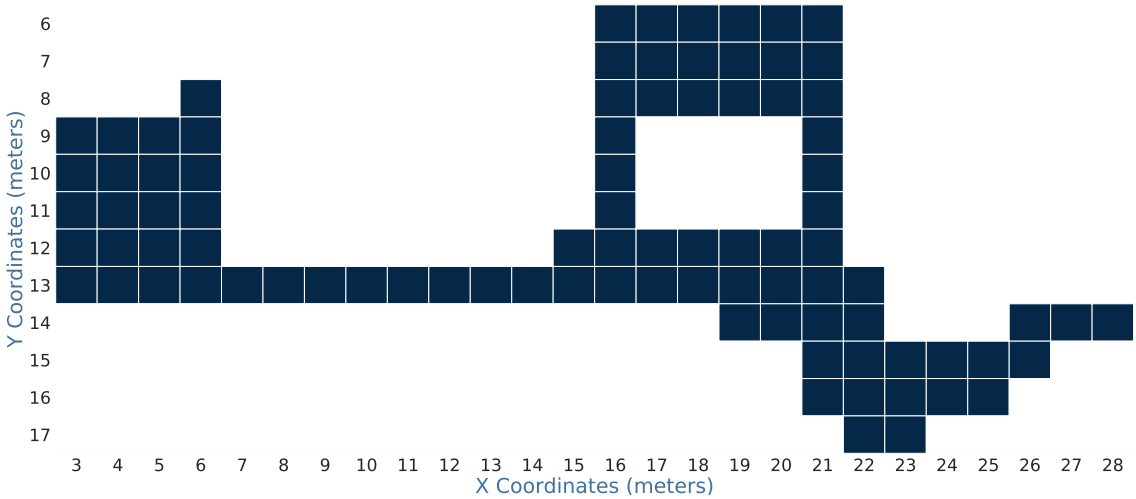


Figure 5.18: Final floor plan obtained after the filtering process. This matrix stores a binary value, that is set to 1 (blue colour) in the reconstructed areas of the building. The coordinates are relative, since the algorithm does not have an absolute reference of the location of the building.

It is possible to evaluate this result with a comparison between this floor plan and the one that was expected, given by the building's map with all the covered areas, available in

Figure 4.4. The main conclusion is that the constructed floor plan has represented most of the areas of the real building, with approximate dimensions. However, two problems can be detected with the floor plan evaluation.

The first problem consists on the missing areas of the original floor plan. This problem was originated by the fact that the dataset did not contemplate a sufficient number of routes in the missing areas, as in the rest of the building. Besides this, the walkable areas of the used building are not very long, comparing to a hospital, for example. The lower limits applied in the retrieval of the sections of the same cluster and with the same heading discarded segments of several routes, where more overlaps could have been found.

The second detected problem happens in mapped areas, where in reality they are map constrains. The filtering caused this effect, and is due to the fact that the corridors of the dataset's building are very close, which caused the closing operation to merge near separate areas. However, in reality, this error has a very small effect in the final deployment of this solution, since it only has a few meters.

5.6 Fingerprints Retrieval

After the construction of floor plans, the environmental fingerprints of the collected data have to be retrieved, in order to accomplish the objectives of this thesis.

The fingerprints retrieved in this work are matrices that store in each cell the information about the collected signal, with a resolution of one square meter. They have the same shape as the floor plans, and their retrieval process works similarly both for the magnetic field and the Wi-Fi radio data.

For the magnetic field, four fingerprints are retrieved, one for each axis of the magnetometer, and a fourth that stores the magnitude values. The algorithm iterates over all routes that are mapped in the final floor plan. With the magnetic data in the distance domain, each displacement point is interpolated with the route's positions on the floor plan, to compute the corresponding coordinates of each point. Then, if the computed position is included on the floor plan, the data that corresponds to the displacement point is stored in the same coordinates in the fingerprint of each axis. After the iteration is finished, the algorithm finds the mean value of the data points stored in every position of the fingerprints. Figure 5.19 shows the obtained geomagnetic fingerprint of the magnitude of all axes.

In order to evaluate the obtained fingerprints, Figure 5.20 shows the corresponding fingerprint of Figure 5.19, collected by the traditional methods, with a resolution of 0.04 m^2. As it can be seen, the values of the same zones of both floor plans are very similar, even though the floor plans have considerable variations. The left area of the obtained fingerprint corresponds to the left area of the original fingerprint, where both intensities are approximate. Although the darker area around the cell (52, 27) in Figure 5.20, considering the (x, y) referential, is not visible in the corresponding zone of Figure

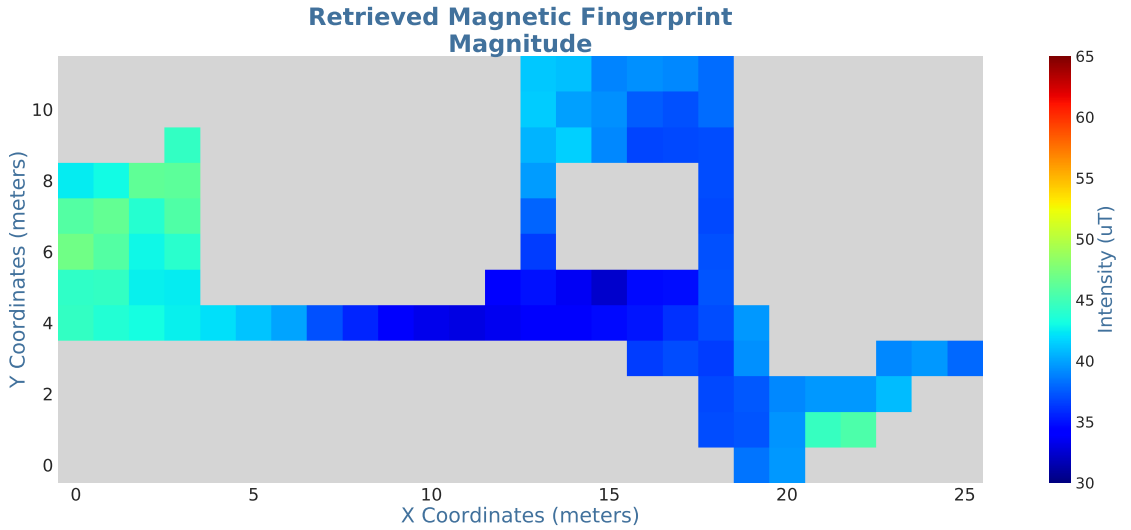


Figure 5.19: Retrieved geomagnetic fingerprint for the magnitude of all axes. The fingerprint has a resolution of one square meter. Each colour represents a different magnetic field intensity, as it is described in the Figure's scale.

5.19, it is possibly due to the fact that the upper limit of the constructed floor plan is lower than the real. Furthermore, in the area corresponding to the coordinates (21, 1) and (21, 2) of Figure 5.19, the intensity does not reach the value of the same area in the original fingerprint, due to the lower resolution of the constructed fingerprints, where the high intensity peak is diluted by its neighbours.

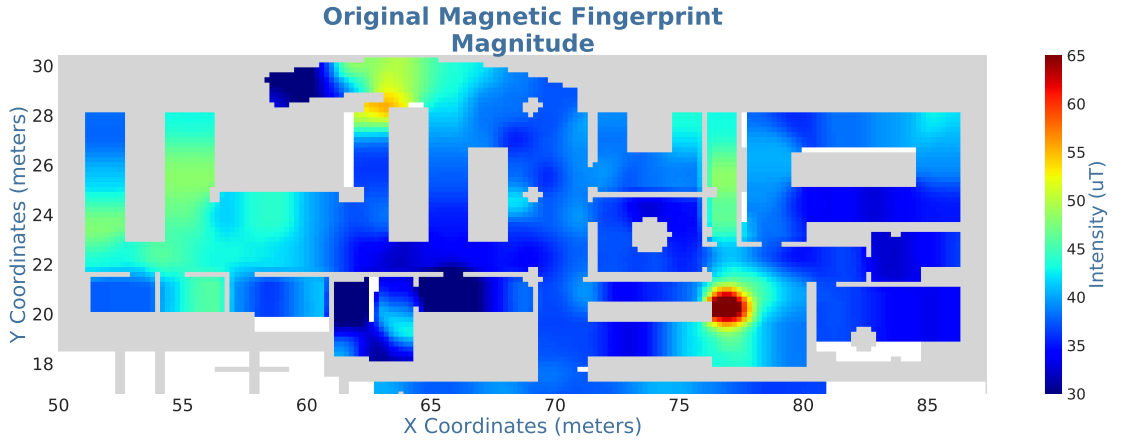


Figure 5.20: Geomagnetic fingerprint for the magnitude of all axes, collected by the traditional methods. The resolution of the fingerprints is of 0.04 m^2. The intensity of the magnetic field in each cell is represented by a different colour, as it is described in the Figure's scale.

Regarding the Wi-Fi radio, the process of retrieving fingerprints is the same as the magnetic field. However, the fingerprints are different. Here, a fingerprint for every AP, both in the 2.4 and 5 GHz bands, stores the RSS values in dBm. Then, a final filtering is done to eliminate fingerprints that have few data points, since their usability is limited.

In Figure 5.21, a Wi-Fi radio fingerprint is displayed with the information of one AP of the test building, in the 2.4 GHz band, identified by the BSSID 84:b8:02:fc:12:9.

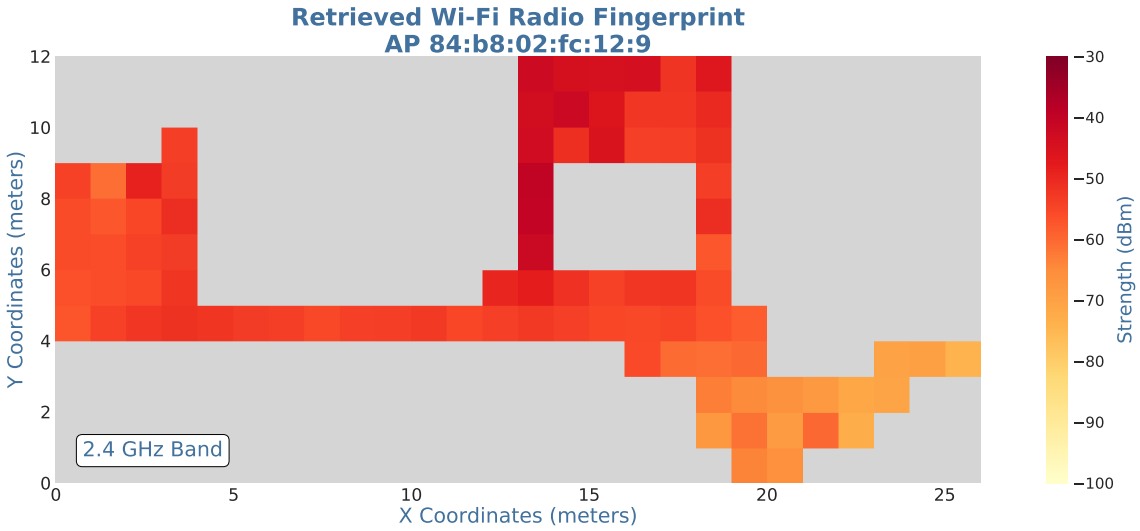


Figure 5.21: Obtained Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9 of the test building, with the signal transmitted in the 2.4 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

Again, with the objective of evaluating the retrieved fingerprint, Figure 5.22 shows the original fingerprint for the same AP, in the same frequency band. The original fingerprint is also represented with a resolution of one square meter. This lower resolution is explained by the low sampling rate of the Wi-Fi radio data (see Table 4.1), which hinders an accurate collection of data for a resolution of 0.04 m^2, as in the magnetic field fingerprints. Besides this, the variability of the Wi-Fi radio signal is not so expressive then the magnetic field's, which allows the reduction of data points. The original fingerprint is often represented outside the map's borders due to the difference of resolutions between the floor plan (0.04 m^2) and the fingerprint (1 m^2). However, this difference does not affect the localisation process. After the comparison of the fingerprints of Figures 5.21 and 5.22, it is possible to conclude that both decay patterns are very consistent, where the area around the (13, 10) coordinate of the first Figure and the area around the (62, 27) coordinate of the second Figure, in the (x, y) plane, are approximately the same position of the building, and have the higher strength for the represented AP. For this reason, it is expected that these positions are the location of the AP in the building.

To support the presented results, which sustain the hypothesis that is possible to construct fingerprints with crowdsourced data, the remaining retrieved fingerprints are represented in Appendix B, both for the magnetic field and the Wi-Fi radio data. Along with every retrieved fingerprint, the original fingerprints for the same axis, APs and radio bands are also represented with the objective of providing a way of comparing the obtained results.

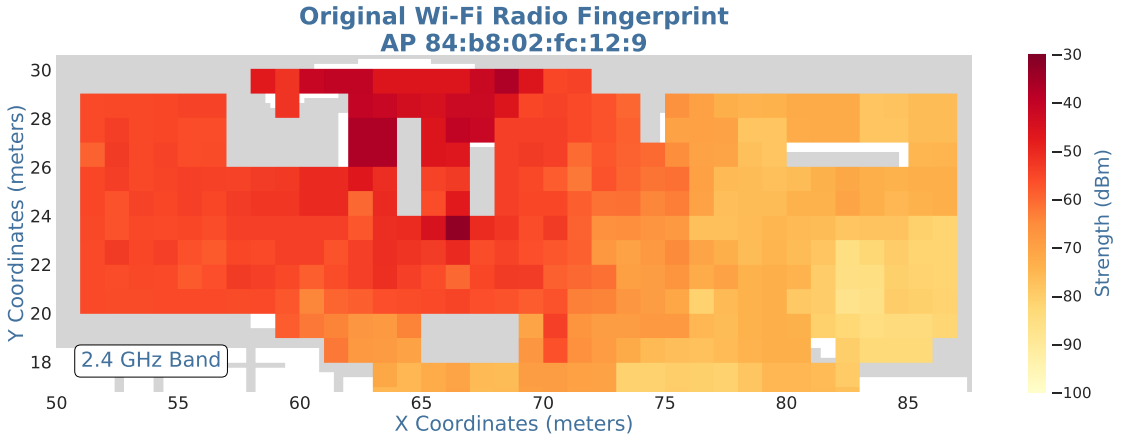


Figure 5.22: Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9, collected with the traditional methods. The signal is transmitted in the 2.4 GHz radio band and the strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.

5.7 Real Scenario Test

One of the main advantages of collecting fingerprints via crowdsourcing is that every point of a fingerprint stores information of several acquisitions. Contrarily, the traditional method of collection uses data from only one passage in each position, as it is explained in Subsection 2.3.1. Due to the extensiveness of this process, sometimes data from missing positions is interpolated, considering the decay of their neighbours.

For this reason, it is expected that crowdsourced fingerprints can provide a better basis for infrastructure-free IPS, thus providing a better accuracy in the positioning phase. To verify these assumptions, an experiment was conducted. Using the PIL solution, the obtained and the original fingerprints were tested with the data of all routes that pass through the existing areas of the reconstructed map of Figure 5.18. Since it would not be fair to test the localisation with routes that pass in non-mapped zones, the routes **HCrowd00**, **HCrowd03**, **HCrowd04**, **HCrowd06**, **HCrowd10**, **HCrowd17**, **HCrowd19**, **HCrowd20**, **HCrowd22**, **HCrowd25** and **HCrowd27** are the ones that can be tested, as it is possible to verify in Appendix A.

To provide the localisation, the PIL solution uses magnetic field fingerprints with a resolution of 0.04 m^2 and Wi-Fi radio fingerprints with a resolution of 1 m^2. To meet these requirements, the fingerprints retrieved in Section 5.6 were processed. The geomagnetic fingerprints were recomputed to consider the new resolution. However, this process resulted in an incomplete fingerprint, where several gaps appeared between the data points. To overcome this problem, the algorithm that interpolates the collected fingerprints by the traditional methods (see Subsection 2.3.1) was applied to compute the unmapped positions. The result of this process was the interpolated fingerprints of every axis. Figure 5.23 shows the interpolated geomagnetic fingerprint for the magnitude of all axes, in the new resolution of 0.04 m^2. The resulting interpolations for the remaining

magnetometer axes are also available in Appendix B.

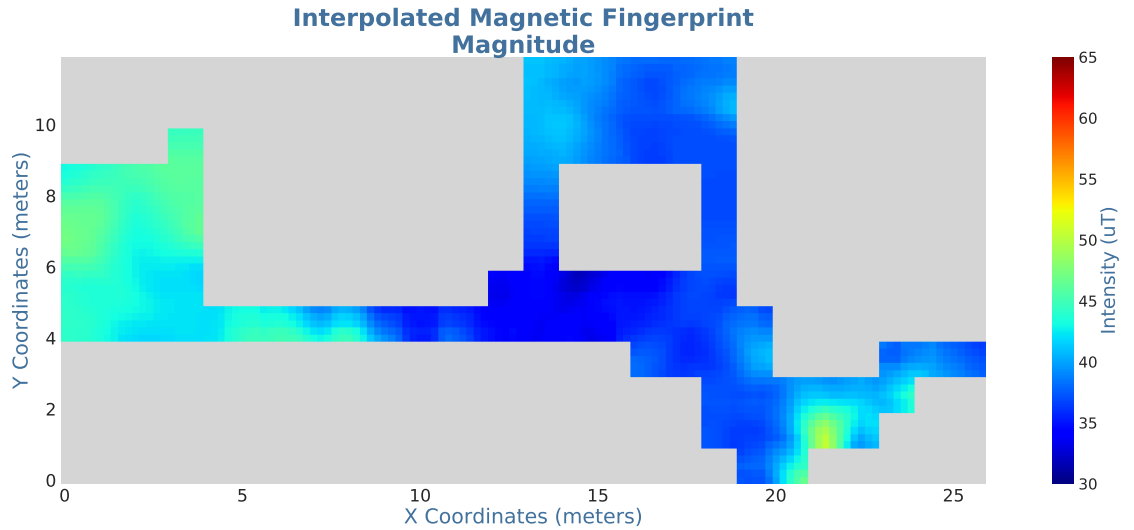


Figure 5.23: Interpolated geomagnetic fingerprint for the magnitude of all axes, from the values of Figure 5.19, with the new resolution of 0.04 m^2. The colours in the Figure's scale represent the different magnetic field intensities.

The Wi-Fi radio fingerprints required for the test are already in the same resolution, so a conversion is not necessary. However, the low sampling rate of the Wi-Fi signal sometimes originates sparse data points. Although this problem is expected to disappear with a larger dataset, the retrieved fingerprints that verified this problem were corrected. To achieve this, the same algorithm that interpolates the Wi-Fi radio fingerprints was used, and the resulting interpolations are available in Appendix B, along with the corresponding obtained and original fingerprints. The fingerprint of Figure 5.21 is already complete, so the applied algorithm produced an interpolation equal to the original.

With the fingerprints in the required resolutions, the test was conducted. The PIL algorithm processes the data of the given acquisitions to identify the dead reckoning parameters. Then, the algorithm uses the reconstruction of the trajectories and the magnetic and Wi-Fi radio fingerprints to retrieve the location of each step. Using an implemented particle filter (see Subsection 2.3.1), the progressively propagated particles are compared to the fingerprints. Then, the returned position corresponds to the weighted centroid of all meaningful particles. To evaluate the accuracy of the localisation, the coordinates of the trajectories are annotated to serve as ground truth. With the identification of the cumulative displacement of each step, the corresponding ground truth coordinates are computed. Then, the difference between the retrieved location and the ground truth position identifies the error of each step. Table 5.1 has identified the mean of the errors of all tested routes, where the mean and the maximum errors of each route were progressively computed.

The outcomes of the real scenario test are satisfactory, where the obtained results for both retrieved and original fingerprints are very similar. Although the errors for

Table 5.1: Results obtained in the real scenario test with the retrieved and original fingerprints. Each cell represents the mean and the standard deviation for the mean and maximum errors of all tested acquisitions.

	Retrieved Fingerprints	Original Fingerprints
Mean Error (m)	4.37 ± 1.98	3.66 ± 1.81
Maximum Error (m)	6.53 ± 2.62	5.71 ± 2.25

the constructed fingerprints are higher than the errors for the original ones, the values differ less than a meter. This difference does not affect the localisation purposes in most applications, since a mean error of 4.37 m would probably still conduct the user to the correct space, as an office or a room. Furthermore, the continuous growing of the dataset will allow the enhancement of the results. While the currently available floor plan hinders some real locations to be returned, a more accurate floor plan, where all or almost all of the areas of the building are mapped will solve this issue. Besides this, with the conjugation of the environmental data of more routes will produce more accurate fingerprints.

5.8 Discussion

The results obtained through the development of this algorithm support the hypothesis that is possible to implement a system to provide infrastructure-free indoor localisation, without the extensive effort of mapping fingerprints. Furthermore, since this solution does not require the floor plan of the building to map the fingerprints, the costs of the implementation are practically non-existent.

Although the final obtained floor plans and fingerprints have some faults, it is possible to verify that they almost accurately represent the test building. Besides this, the fact that a system like this can collect data continuously, the floor plans and the fingerprints can be continuously improved, providing increasingly better localisation results.

Comparatively to the acquisition time of the fully deployed solution, where it is expected that the system is utilised by users almost permanently, the acquired 95 minutes of the used dataset represent a very small part of the total data that will be available. For these reasons, the further development of this solution is supported, toward the creation of a system capable of using the continuously coming data to improve and update the floor plans and fingerprints, when necessary.

Chapter 6

Conclusions and Future Work

In this final chapter, an overview over the developments achieved during this thesis is presented, as well as a balance of the obtained results. Then, some future guidelines are suggested to the continuous improvement of the developed solution.

6.1 Conclusions

The ageing of the world's population has been motivating the scientific community to develop new technological solutions, to ensure an improved quality of life to the older adults. One of the innovative technologies that has a major role in the AAL tasks aims to provide localisation-based services indoors, that can be used to guarantee the safety of the elderly, or simply to ease their daily life.

However, the current available IPS are not easily deployed, due to the high costs on installation and maintenance. Infrastructure-free systems do not require the investment on equipment for every building, but they demand an extensive setup phase, where the fingerprints of the buildings need to be acquired.

The objective of this project was to improve this process, through the elimination of the human intervention on the mapping of fingerprints. With the use of crowdsourcing, this thesis proposes an algorithm for the automatic construction of floor plans and environmental fingerprints, to be deployed in any building. To do that, the algorithm relies on the processing of inertial and environmental data, collected without any annotation from smartphones. While the inertial data is used to infer the trajectories described by the users, the environmental data is compared to each other, to find similarities that identify overlapping sections. Then, these overlaps are used to match the inferred trajectories, to obtain the buildings' floor plans and fingerprints.

The first stage of this thesis consisted in testing the viability of the use of environmental data, such as the Wi-Fi radio and the magnetic field, to find the required similarities. For this purpose, a PoC was developed, that simulates both the inertial and the environmental data, thus avoiding the high variability of real scenarios. The obtained results sustained the initially proposed hypothesis, where similarities between different simulated routes were correctly identified.

For real scenarios, a dataset was constructed considering the features of crowdsourcing data. However, an annotation protocol was created, to allow the validation of the progressively obtained results.

Using the acquired data, the final algorithm was developed, and the floor plan and the fingerprints of a test building were obtained. To achieve this, several steps where taken. Firstly, the Wi-Fi radio data was clustered using an unsupervised machine learning technique, which allowed the division of the dataset into similar groups, that represent small areas of the building. After the processing of the inertial data, where the trajectories described by the users were inferred, the data was segmented in straight line sections. Then, the retrieved segments were compared to each other using the magnetic field data and a time series similarities measure, the WPA, to identify overlaps between different routes. Next, the floor plan of the building was constructed and filtered, and finally, the fingerprints were obtained. Although the developed algorithm was tested in an office building, the created process allows the application of this algorithm in any building.

The results achieved in this thesis show that the developed method has significant potential to be applied in infrastructure-free IPS. Although some faults have been pointed out to the obtained floor plans and fingerprints, it is expected that, with a larger dataset, these problems will disappear.

As a final conclusion, the accomplishments achieved during the development of this project support the proposed hypothesis, that it is possible to automatise the setup phase of infrastructure-free IPS, thus extending the range of applications of indoor location, especially in the AAL scope.

6.2 Future Work

Although the satisfactory results obtained during the development of this thesis, the created algorithm has some room to improve.

Firstly, the process of adding new routes to the fingerprints should be considered, where new acquisitions are continuously being received, and the floor plans and fingerprints are progressively modified. In order to deal with buildings with several floors, the proposed algorithm should be adapted. For example, by using the variation of atmospheric pressure to detect the transitions between floors, it will be possible to identify the Wi-Fi clusters of each floor and build with the necessary different maps.

Still regarding this topic, some decisions have to be done, to establish the way the final solution operates. It is necessary to understand the minimum number of acquired routes

to start the system, so a preliminary map can be obtained. This number might be low, where a simple map with the merge of two routes can be initially constructed. However, in this way, the deployed solution will initially offer a low accuracy, that will be increased with the introduction of new data to the fingerprints. Contrarily, if the minimum number of routes is higher, the solution will require an initial acquisition time, where the system collects the crowdsourced data, but cannot provide the localisation feature. Still, this setup phase might not be too long, being reduced to a few hours, for example, in a large building as a hospital, where hundreds of people pass by daily. Thus, some tests have to be conducted to understand which process is the best, considering the requirements of each application.

Furthermore, the process of updating the floor plans and fingerprints has to be chosen. It is necessary to understand which characteristics and behaviours of the upcoming data will trigger the modification process. For example, the minimum number of new routes that pass by an unknown area has to be decided, so this new area can be classified as truly existent, instead of erroneous, and consequently added to the floor plan. On the other hand, since layout modifications on the building can happen, where previously existent corridors might be eliminated, it is also important to understand the required time to eliminate an existing zone of the floor plan.

Besides the tests and decisions regarding the technical features of this system, some stages of the algorithm can still be improved. The features used in the Wi-Fi radio clustering allowed the obtainment of satisfactory results. However, other features might also be applied, as well as other unsupervised machine learning algorithms. The data segmentation, which only considers straight line sections, can also be modified, to study the possibility of using the curves to retrieve useful information. Concerning the comparison of the magnetic field data, different time series similarities measures can be implemented, as the DTW, to substitute or confirm the results of the applied WPA. Moreover, new environmental sources can be added to the system in order to improve its accuracy, as the ambient sound or the atmospheric pressure.

Finally, after all the decisions have been made, and before the deployment of the developed solution in any fingerprinting-based IPS, final tests have to be done, with respect to the performance of the algorithm, as its usability and computational complexity.

Bibliography

[1] V. Fuchsberger. "Ambient Assisted Living: Elderly People's Needs and How to Face Them." In: *Proceeding of the 1st ACM International Workshop on Semantic Ambient Media Experiences - SAME '08*. 2008, pp. 21–24. ISBN: 9781605583143. DOI: 10.1145/1461912.1461917.

[2] E. Commission. *Objectives of Active and Assisted Living Programme*. URL: http://www.aal-europe.eu/about/objectives/ (visited on 09/17/2018).

[3] Eurostat. *demo_mlexpec Dataset: Life expectancy by age and sex*. URL: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=demo_mlexpec&lang=en (visited on 09/17/2018).

[4] Eurostat. *proj_15nalexp Dataset: Assumptions for life expectancy by age, sex and type of projection*. URL: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=proj_15nalexp&lang=en (visited on 09/17/2018).

[5] Eurostat. *demo_pjangroup Dataset: Population on 1 January by age group and sex*. URL: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=demo_pjangroup&lang=en (visited on 09/17/2018).

[6] Eurostat. *proj_15npms Dataset: Population on 1st January by age, sex and type of projection*. URL: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=proj_15npms&lang=en (visited on 09/17/2018).

[7] V. Guimarães, L. Castro, S. Carneiro, M. Monteiro, T. Rocha, M. Barandas, J. Machado, M. Vasconcelos, H. Gamboa, and D. Elias. "A Motion Tracking Solution for Indoor Localization Using Smartphones." In: *2016 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2016*. 2016, pp. 1–8. ISBN: 9781509024254. DOI: 10.1109/IPIN.2016.7743680.

[8] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier. "The Internet of Things for Ambient Assisted Living." In: *2010 Seventh International Conference on Information Technology: New Generations*. 2010, pp. 804–809. DOI: 10.1109/ITNG.2010.104.

[9] H. Hwangbo, J. Kim, Z. Lee, and S. Kim. "Store Layout Optimization Using Indoor Positioning System." In: *International Journal of Distributed Sensor Networks* 13.2 (2017). ISSN: 15501477. DOI: 10.1177/1550147717692585.

[10] M. Xiong, Y. Wu, Y. Ding, X. Mao, Z. Fang, and H. Huang. "A Smart Home Control System Based on Indoor Location and Attitude Estimation." In: *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)* (2016), pp. 1–5. DOI: 10.1109/CITS.2016.7546460.

[11] M. B. del Rosario, S. J. Redmond, and N. H. Lovell. "Tracking the Evolution of Smartphone Sensing for Monitoring Human Movement." In: *Sensors (Switzerland)* 15.8 (2015), pp. 18901–18933. ISSN: 14248220. DOI: 10.3390/s150818901.

[12] Belmonte-Fernández, A. Puertas-Cabedo, J. Torres-Sospedra, R. Montoliu-Colás, and S. Trilles-Oliver. "An Indoor Positioning System Based on Wearables for Ambient-Assisted Living." In: *Sensors* 17.1 (2017). ISSN: 1424-8220. DOI: 10.3390/s17010036.

[13] M. Er Rida, F. Liu, Y. Jadi, A. A. A. Algawhari, and A. Askourih. "Indoor Location Position based on Bluetooth Signal Strength." In: *2nd International Conference on Information Science and Control Engineering, ICISCE 2015*. 2015, pp. 769–773. ISBN: 9781467368506. DOI: 10.1109/ICISCE.2015.177.

[14] M. Teran, J. Aranda, H. Carrillo, D. Mendez, and C. Parra. "IoT-Based System for Indoor Location Using Bluetooth Low Energy." In: *IEEE Colombian Conference on Communications and Computing (COLCOM2017)* (2017), pp. 0–5. DOI: 10.1109/ColComCon.2017.8088211.

[15] T. Fernandes. "Indoor Localization using Bluetooth." In: *6th Doctoral Symposium in Informatics Engineering* (2011), pp. 1–10. URL: http://paginas.fe.up.pt/~prodei/dsie11/images/pdfs/s5-4.pdf.

[16] J. Bordoy, A. Traub-ens, A. Sadr, J. Wendeberg, F. Höflinger, C. Schindelhauer, and L. Reindl. "Bank of Kalman Filters in Closed-Loop for Robust Localization using Unsynchronized Beacons." In: *IEEE Sensors Journal* 16.19 (2016), pp. 7142–7149. DOI: 10.1109/JSEN.2016.2597967.

[17] T. Pfeifer and D. Elias. "Commercial Hybrid IR/RF Local Positioning System." In: *KiVS Kurzbeiträge* (2003), pp. 1–9. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.96.2526.

[18] S. Hijikata, K. Terabayashi, and K. Umeda. "A Simple Indoor Self-Localization System Using Infrared LEDs." In: *INSS2009 - 6th International Conference on Networked Sensing Systems*. 2009, pp. 45–51. ISBN: 9781424463145. DOI: 10.1109/INSS.2009.5409955.

[19] S. Pergoloni, S. Member, Z. Mohamadi, A. M. Vegni, and S. Member. "Metameric Indoor Localization Schemes Using Visible Lights." In: 35.14 (2017), pp. 2933–2942. DOI: 10.1109/JLT.2017.2706527.

[20] P Kemppi, J Pajunen, and T Rautiainen. "Use of Artificial Magnetic Anomalies in Indoor Pedestrian Navigation." In: *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd* 1000 (2010), pp. 1–5. DOI: 10.1109/VETECF.2010.5594106.

[21] L. Bedogni, F. Franzoso, and L. Bononi. "A Self-Adapting Algorithm Based on Atmospheric Pressure to Localize Indoor Devices." In: *2016 IEEE Global Communications Conference (GLOBECOM)*. 2016, pp. 1–6. DOI: 10.1109/GLOCOM.2016.7841545.

[22] J. Liu, Y. Chen, A. Jaakkola, T. Hakala, J. Hyyppä, L. Chen, J. Tang, R. Chen, and H. Hyyppä. "The Uses of Ambient Light for Ubiquitous Positioning." In: *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*. 2014, pp. 102–108. DOI: 10.1109/PLANS.2014.6851363.

[23] M. Azizyan, I. Constandache, and R. Roy Choudhury. "SurroundSense: Mobile Phone Localization via Ambience Fingerprinting." In: *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*. MobiCom '09. 2009, pp. 261–272. DOI: 10.1145/1614320.1614350.

[24] R. Leonardo. "Contextual Information Based on Pervasive Sound Analysis." MSc. FCT/NOVA, 2017.

[25] P. Bahl and V. N. Padmanabhan. "RADAR: An In-Building RF-Based User Location and Tracking System." In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 2. 2000, pp. 775–784. DOI: 10.1109/INFCOM.2000.832252.

[26] L. Chen, K. Yang, and X. Wang. "Robust Cooperative Wi-Fi Fingerprint-Based Indoor Localization." In: *IEEE Internet of Things Journal* 3.6 (2016), pp. 1406–1417. ISSN: 23274662. DOI: 10.1109/JIOT.2016.2609405.

[27] C. M. Vikas, S. Rajendran, A. Pattar, H. S. Jamadagni, and R. Budihal. "WiFi RSSI and Inertial Sensor Based Indoor Localisation S: a Simplified Hybrid Approach." In: *2016 International Conference on Signal and Information Processing, IConSIP 2016* (2017), pp. 2–7. DOI: 10.1109/ICONSIP.2016.7857443.

[28] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao. "Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic Wi-Fi Sensing." In: *IEEE Journal on Selected Areas in Communications* 33.7 (2015), pp. 1443–1457. ISSN: 07338716. DOI: 10.1109/JSAC.2015.2430274.

[29] Q. Xu, R. Zheng, and S. Hranilovic. "IDyLL: Indoor Localization Using Inertial and Light Sensors on Smartphones." In: *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp 2015* (2015), pp. 307–318. DOI: 10.1145/2750858.2807540.

[30] B. Viel and M. Asplund. "Why is Fingerprint-Based Indoor Localization Still so Hard?" In: *2014 IEEE International Conference on Pervasive Computing and Communication Workshops, PERCOM WORKSHOPS 2014* (2014), pp. 443–448. DOI: 10.1109/PerComW.2014.6815247.

[31] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. "Zee: Zero-Effort Crowdsourcing for Indoor Localization." In: *Proceedings of the 18th annual international conference on Mobile computing and networking - Mobicom '12* (2012). DOI: 10.1145/2348543.2348580.

[32] C. Wu, Z. Yang, and Y. Liu. "Smartphones Based Crowdsourcing for Indoor Localization." In: *IEEE Transactions on Mobile Computing* 14.2 (2015), pp. 444–457. ISSN: 15361233. DOI: 10.1109/TMC.2014.2320254.

[33] J. Niu, B. Wang, L. Cheng, and J. J. Rodrigues. "WicLoc: An Indoor Localization System Based on WiFi Fingerprints and Crowdsourcing." In: *2015 IEEE International Conference on Communications (ICC)*. 2015, pp. 3008–3013. ISBN: 9781467364324. DOI: 10.1109/ICC.2015.7248785.

[34] J. Chen, Y. Zhang, and W. Xue. "Unsupervised Indoor Localization Based on Smartphone Sensors, iBeacon and Wi-Fi." In: *Sensors (Switzerland)* 18.5 (2018), pp. 1–18. ISSN: 14248220. DOI: 10.3390/s18051378.

[35] M. Alzantot and M. Youssef. "CrowdInside: Automatic Construction of Indoor Floorplans." In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '12. Redondo Beach, California, 2012, pp. 99–108. ISBN: 978-1-4503-1691-0. DOI: 10.1145/2424321.2424335.

[36] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. "Walkie-Markie: Indoor Pathway Mapping Made Easy." In: *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13)*. 2013, pp. 85–98. URL: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/shen.

[37] H. Shin, Y. Chon, and H. Cha. "Unsupervised Construction of an Indoor Floor Plan Using a Smartphone." In: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 42.6 (2012), pp. 889–898. ISSN: 10946977. DOI: 10.1109/TSMCC.2011.2169403.

[38] H. Luo, F. Zhao, M. Jiang, H. Ma, and Y. Zhang. "Constructing an Indoor Floor Plan Using Crowdsourcing Based on Magnetic Fingerprinting." In: *Sensors (Switzerland)* 17.11 (2017). ISSN: 14248220. DOI: 10.3390/s17112678.

[39] C. Luo, H. Hong, and M. C. Chan. "PiLoc: A Self-Calibrating Participatory Indoor Localization System." In: *IPSN 2014 - Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (Part of CPS Week)*. 2014, pp. 143–153. ISBN: 9781479931460. DOI: 10.1109/IPSN.2014.6846748.

[40] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. "No Need to War-Drive: Unsupervised Indoor Localization." In: *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services - MobiSys '12*. 2012, pp. 197–210. ISBN: 9781450313018. DOI: 10.1145/2307636.2307655.

[41] A. M. Desai and R. H. Jhaveri. "A Review on Applications of Ambient Assisted Living." In: *International Journal of Computer Applications (0975 – 8887)* 176.8 (2017), pp. 1–7. ISSN: 0974-5645. DOI: 10.17485/ijst/2016/v9i48/108433.

[42] S. Norton, F. E. Matthews, D. E. Barnes, K. Yaffe, and C. Brayne. "Potential for Primary Prevention of Alzheimer's Disease: An Analysis of Population-Based Data." In: *The Lancet Neurology* 13.8 (2014), pp. 788–794. ISSN: 14744465. DOI: 10.1016/S1474-4422(14)70136-X.

[43] P. Rashidi and A. Mihailidis. "A Survey on Ambient-Assisted Living Tools for Older Adults." In: *IEEE Journal of Biomedical and Health Informatics* 17.3 (2013), pp. 579–590. ISSN: 21682194. DOI: 10.1109/JBHI.2012.2234129.

[44] O. C. Ann and L. B. Theng. "Human Activity Recognition: A Review." In: *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*. 2014, pp. 389–393. DOI: 10.1109/ICCSCE.2014.7072750.

[45] E. Kim, S Helal, and D Cook. "Human Activity Recognition and Pattern Discovery." In: *Pervasive Computing, IEEE* 9.1 (2010), pp. 48–53. ISSN: 1536-1268. DOI: 10.1109/MPRV.2010.7.

[46] O. D. Lara and M. A. Labrador. "A Survey on Human Activity Recognition using Wearable Sensors." In: *IEEE Communications Surveys & Tutorials* 15.3 (2013), pp. 1192–1209. DOI: 10.1109/SURV.2012.110112.00192.

[47] Y. T. Zhang, C. C. Poon, C. H. Chan, M. W. Tsang, and K. F. Wu. "A Health-Shirt using E-Textile Materials for the Continuous and Cuffless Monitoring of Arterial Blood Pressure." In: *Proceedings of the 3rd IEEE-EMBS International Summer School and Symposium on Medical Devices and Biosensors, ISSS-MDBS 2006* (2006), pp. 86–89. DOI: 10.1109/ISSMDBS.2006.360104.

[48] N. Bowditch. "Dead Reckoning." In: *The American Practical Navigator*. 2002nd ed. 1802. Chap. 7, pp. 113–118.

[49] V. Radu and M. K. Marina. "HiMLoc: Indoor Smartphone Localization via Activity Aware Pedestrian Dead Reckoning with Selective Crowdsourced WiFi Fingerprinting." In: *2013 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013* (2013), pp. 28–31. DOI: 10.1109/IPIN.2013.6817916.

[50] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang. "An Effective Pedestrian Dead Reckoning Algorithm Using a Unified Heading Error Model." In: *IEEE/ION Position, Location and Navigation Symposium*. 2010, pp. 340–347. DOI: 10.1109/PLANS.2010.5507300.

[51] Q. Ladetto. "On Foot Navigation: Continuous Step Calibration Using Both Complementary Recursive Prediction and Adaptive Kalman Filtering." In: 2000, pp. 1735–1740. URL: https://www.researchgate.net/publication/37409574/.

[52] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. "A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors." In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. UbiComp '12. 2012, pp. 421–430. DOI: 10.1145/2370216.2370280.

[53] P Davidson, J Collin, and J Takala. "Application of Particle Filters for Indoor Positioning Using Floor Plans." In: *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*. 2010, pp. 1–4. ISBN: 9781424478798. DOI: 10.1109/UPINLBS.2010.5653830.

[54] K. Kaemarungsi. "Distribution of WLAN Received Signal Strength Indication for Indoor Location Determination." In: *2006 1st International Symposium on Wireless Pervasive Computing*. 2006, pp. 1–6. DOI: 10.1109/ISWPC.2006.1613601.

[55] A. O. S. project. *Scan Result*. URL: https://developer.android.com/reference/android/net/wifi/ScanResult (visited on 09/17/2018).

[56] R. Ma, Q. Guo, C. Hu, and J. Xue. "An Improved WiFi Indoor Positioning Algorithm by Weighted Fusion." In: *Sensors* 15.9 (2015), pp. 21824–21843. DOI: 10.3390/s150921824.

[57] IPMA. *Geomagnetism*. URL: https://www.ipma.pt/en/enciclopedia/geofisica/geomagnetismo/index.html (visited on 09/17/2018).

[58] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti. "Crowdsourcing with Smartphones." In: *IEEE Internet Computing* 16.5 (2012), pp. 36–44. ISSN: 10897801. DOI: 10.1109/MIC.2012.70.

[59] L. Yu. "Daren C. Brabham: Crowdsourcing." In: *Genetic Programming and Evolvable Machines* 15.2 (2014), pp. 219–220. ISSN: 1573-7632. DOI: 10.1007/s10710-014-9215-3.

[60] R. Priemer. "Signals and Signal Processing." In: *Introductory Signal Processing*. World Scientific, 1990. Chap. 0, pp. 1–9. ISBN: 978-9971-5-0919-4. DOI: 10.1142/0864.

[61] P. J. Brockwell and R. A. Davis. "Stationary Time Series." In: *Time Series: Theory and Methods*. Springer Series in Statistics. Springer New York, 1987, pp. 1–2. ISBN: 978-1-4899-0006-7. DOI: 10.1007/978-1-4899-0004-3.

[62] T. Ferenti. "Biomedical Applications of Time Series Analysis." In: *2017 IEEE 30th Neumann Colloquium (NC)*. 2017, pp. 83–84. ISBN: 978-1-5386-4636-6. DOI: 10.1109/NC.2017.8263256.

[63] J. Serrà and J. L. Arcos. "An Empirical Evaluation of Similarity Measures for Time Series Classification." In: *Knowledge-Based Systems* 67 (2014), pp. 305–314. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2014.04.035.

[64] X. Wang, A. Mueen, H. Ding, G. Trajcevski, and P. Scheuermann. "Experimental Comparison of Representation Methods and Distance Measures for Time Series Data." In: *Data Mining and Knowledge Discovery* 26.2 (2013), pp. 275–309. DOI: 10.1007/s10618-012-0250-5.

[65] J. Torres-Sospedra, R. Montoliu, S. Trilles, Ó. Belmonte, and J. Huerta. "Comprehensive Analysis of Distance and Similarity Measures for Wi-Fi Fingerprinting Indoor Positioning Systems." In: *Expert Systems with Applications* 42.23 (2015), pp. 9263–9278. ISSN: 09574174. DOI: 10.1016/j.eswa.2015.08.013.

[66] D. Folgado. "Measuring Repetitive Tasks using Inertial Sensors." MSc. FCT/NOVA, 2015.

[67] N. Nunes. "Algorithms for Time Series Clustering Applied to Biomedical Signals." MSc. FCT/NOVA, 2011.

[68] R. C. Gonzalez and R. E. Woods. "Introduction." In: *Digital Image Processing*. 2nd ed. Prentice Hall, 1987. Chap. 1. ISBN: 0-201-11026-1.

[69] J. C. Russ. "Processing Binary Images." In: *The Image Processing Handbook*. 4th ed. CRC Press, 1992. Chap. 7. ISBN: 0-8493-1142-X.

[70] O. Simeone. "Machine Learning." In: *A Brief Introduction to Machine Learning for Engineers*. 2017, pp. 5–7. URL: http://arxiv.org/abs/1709.02840.

[71] C. Figueira. "Body Location Independent Activity Monitoring." MSc. FCT/NOVA, 2015.

[72] S. B. Kotsiantis. "Supervised Machine Learning: A Review of Classification Techniques." In: *Informatica (Ljubljana)* 31 (2007), pp. 249–268. URL: https://datajobs.com/data-science-repo/Supervised-Learning-[SB-Kotsiantis].pdf.

[73] P. Rai and S. Singh. "A Survey of Clustering Techniques." In: *International Journal of Computer Applications* 7.12 (2010), pp. 1–5. DOI: 10.5120/1326-1808.

[74] A. K. Jain. "Data Clustering: 50 Years Beyond K-Means." In: *Pattern Recognition Letters* 31.8 (2010), pp. 651–666. ISSN: 01678655. DOI: 10.1016/j.patrec.2009.09.011.

[75] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah. "Time-Series Clustering - A Decade Review." In: *Information Systems* 53 (2015), pp. 16–38. ISSN: 03064379. DOI: 10.1016/j.is.2015.04.007.

[76] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data Clustering: A Review." In: *ACM Comput. Surv.* 31.3 (1999), pp. 264–323. DOI: 10.1145/331499.331504.

[77] P.-N. Tan, M. Steinbach, and V. Kumar. "Cluster Analysis: Basic Concepts and Algorithms." In: *Introduction to Data Mining*. 1st ed. Addison-Wesley Longman Publishing Co., Inc., 2005. Chap. 8. ISBN: 0321321367. URL: https://www-users.cs.umn.edu/~kumar001/dmbook/ch8.pdf.

[78] Y. Zhang, J. Mandziuk, C. H. Quek, and B. W. Goh. "Curvature-Based Method for Determining the Number of Clusters." In: *Information Sciences* 416 (2017), pp. 414–428. DOI: 10.1016/j.ins.2017.05.024.

[79] R. Campello, D. Moulavi, A. Zimek, and J. Sander. "Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection." In: *ACM Transactions on Knowledge Discovery from Data* 10.1 (2015), pp. 1–51. ISSN: 15564681. DOI: 10.1145/2733381.

[80] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: http://scikit-learn.org/stable/index.html.

[81] L. McInnes, J. Healy, and S. Astels. "HDBSCAN: Hierarchical Density Based Clustering." In: *The Journal of Open Source Software* 2.11 (2017). DOI: 10.21105/joss.00205.

[82] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. "Scikit-image: Image Processing in Python." In: *PeerJ* 2 (2014), pp. 1–18. ISSN: 2167-8359. DOI: 10.7717/peerj.453.

Appendix

A

Dataset's Routes Design

This Appendix presents the design of the created routes within the test building. With several acquisitions of each route, collected by several users using both used smartphones (Nexus 5 and Nexus 6P), a total of 135 routes were acquired. The following Figures show the trajectories of routes from **HCrowd00** to **HCrowd21**:

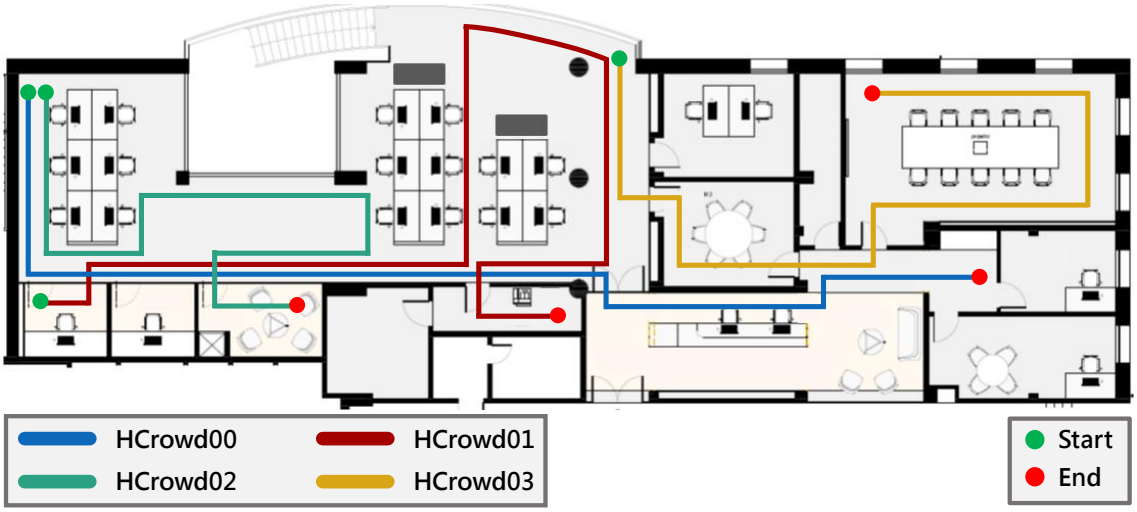


Figure A.1: Design of routes HCrowd00 to HCrowd03 of the dataset. Each route is represented by a different colour, as it is explained in the Figure's legend. The green and red circles express the start and end of each route, respectively.

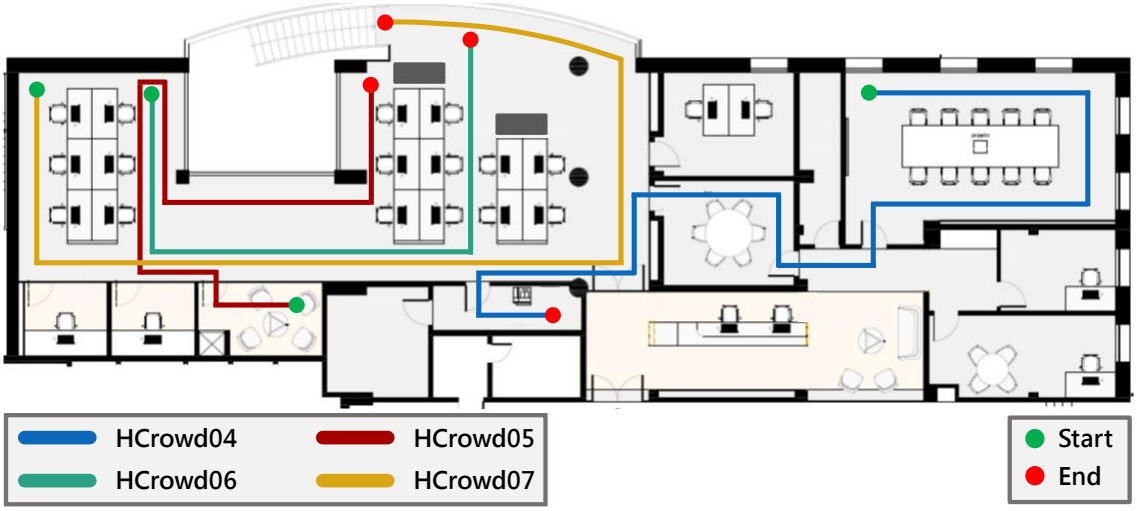


Figure A.2: Design of routes HCrowd04 to HCrowd07 of the dataset. Each route is represented by a different colour, as it is explained in the Figure's legend. The green and red circles express the start and end of each route, respectively.

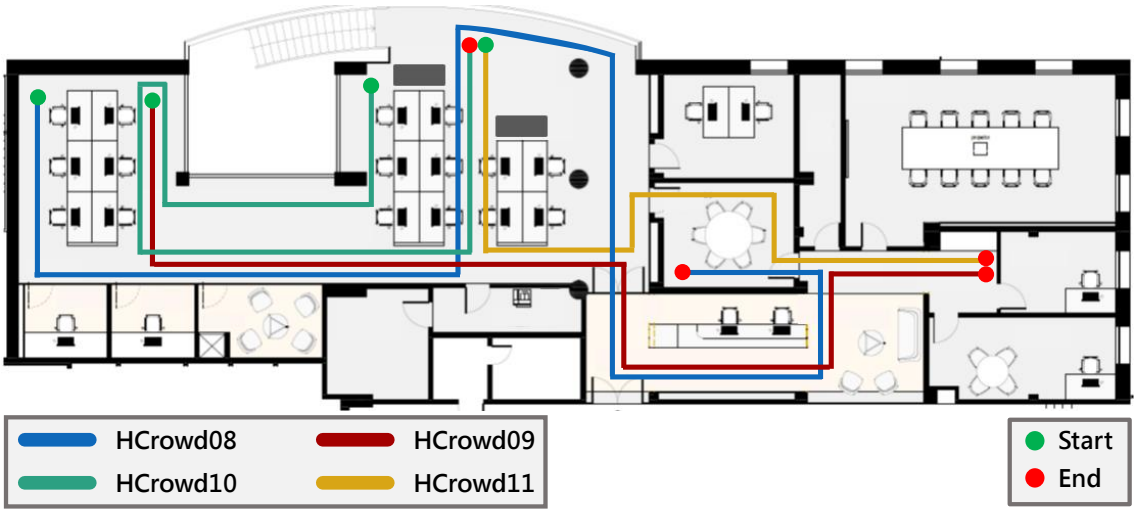


Figure A.3: Design of routes HCrowd08 to HCrowd11 of the dataset. Each route is represented by a different colour, as it is explained in the Figure's legend. The green and red circles express the start and end of each route, respectively.

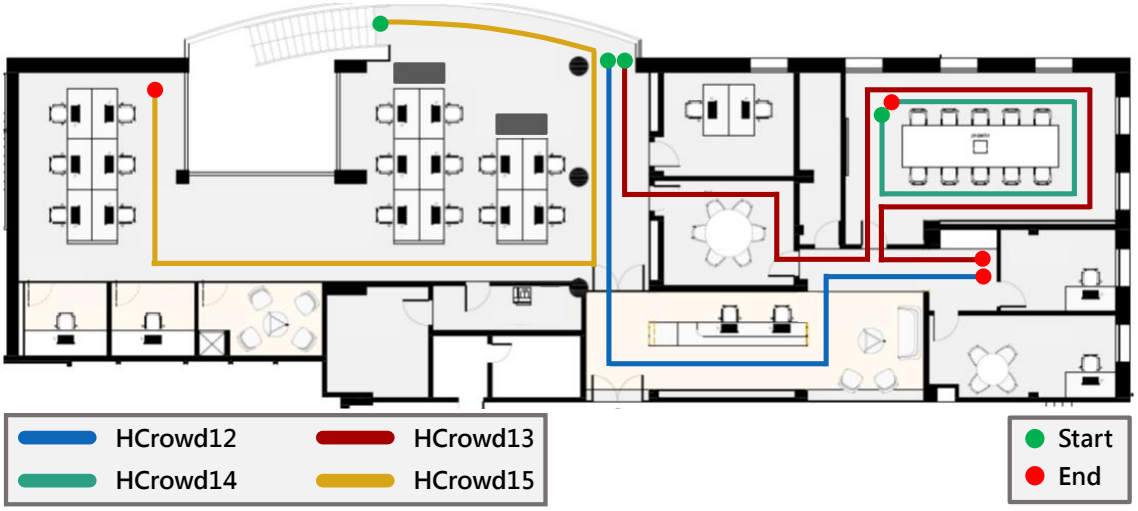


Figure A.4: Design of routes HCrowd12 to HCrowd15 of the dataset. Each route is represented by a different colour, as it is explained in the Figure's legend. The green and red circles express the start and end of each route, respectively.

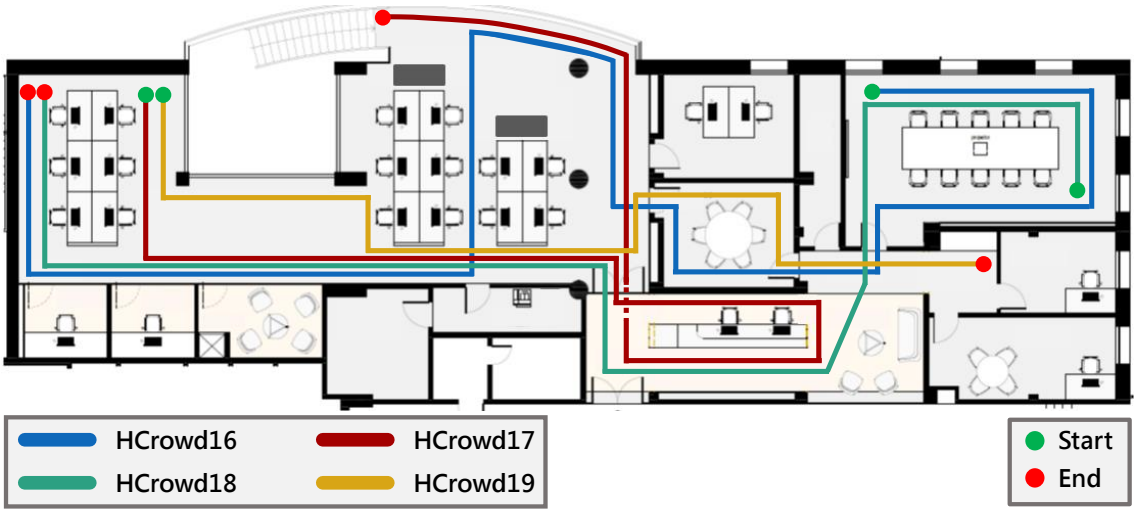


Figure A.5: Design of routes HCrowd16 to HCrowd19 of the dataset. Each route is represented by a different colour, as it is explained in the Figure's legend. The green and red circles express the start and end of each route, respectively.

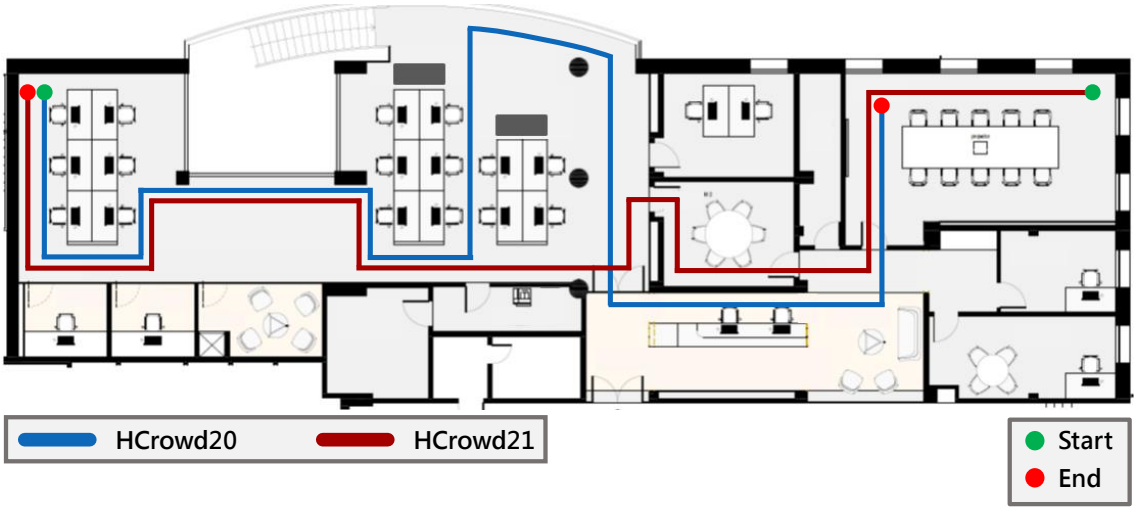


Figure A.6: Design of routes HCrowd20 and HCrowd21 of the dataset. Each route is represented by a different colour, as it is explained in the Figure's legend. The green and red circles express the start and end of each route, respectively.

Appendix B

Retrieved, Interpolated and Original Fingerprints

This Appendix aims to provide a way of evaluating the results obtained with the developed algorithm of Chapter 5. This is done through a representation of all the fingerprints retrieved in several stages of the algorithm, for the collected environmental data, to be used in any infrastructure-free IPS. For the magnetic field data, the results of the three axes of the magnetometer are displayed, as well as their magnitude. For the Wi-Fi radio, the fingerprints of each detected AP in the 2.4 and 5 GHz bands are also shown.

The following Figures are organised in sets three to allow a proper evaluation of the results. The top Figure of each page identifies the retrieved fingerprints of Section 5.6, after the floor plan construction process. These fingerprints are mapped within the constrains of the floor plan, since the localisation process will not allow the positioning in impossible zones.

Then, the corresponding interpolated fingerprints computed in Section 5.7 are displayed in the middle Figure, which were used to test the results of the algorithm in a real scenario. The interpolation of the magnetic field creates fingerprints with a higher resolution of 0.04 m^2, to match the requirements of the PIL solution. On the other hand, the Wi-Fi radio interpolation generates fingerprints with the same resolution of one square meter, but is done to compute the missing positions, considering the Gaussian decay pattern of the Wi-Fi signal.

Finally, to visually compare the outcomes of the fingerprint retrieval process to the traditional method of collection, the original fingerprints are shown in the last Figure of each page. They correspond to the same axis, AP and radio band of the remaining fingerprints in the same page.

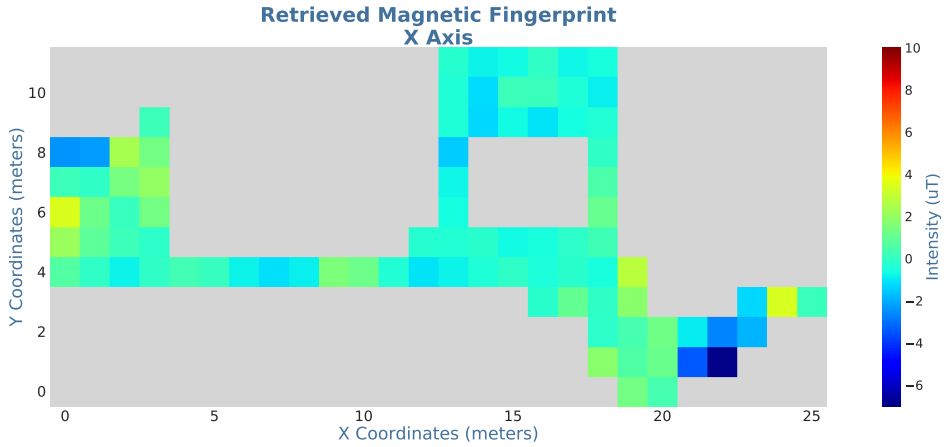


Figure B.1: Retrieved geomagnetic fingerprint for the X axis, with the defined resolution of one square meter. Each colour represents a different magnetic field intensity, as it is described in the Figure's scale.

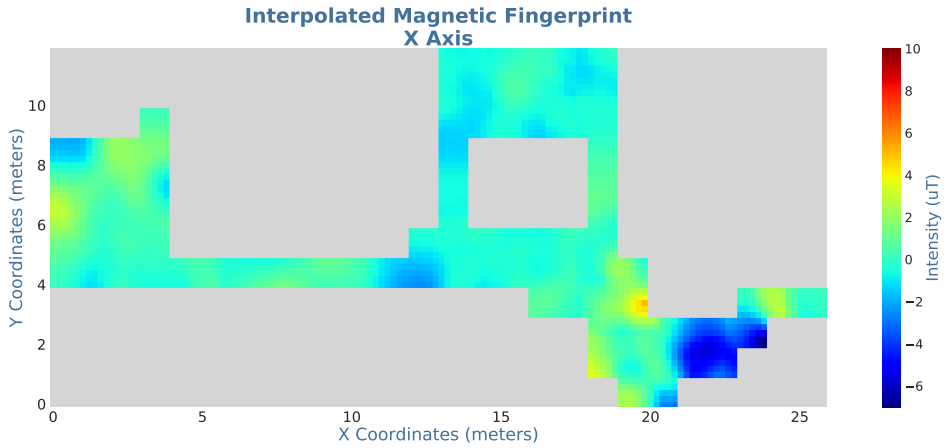


Figure B.2: Interpolated geomagnetic fingerprint for the X axis, from the values of Figure B.1, with the new resolution of 0.04 m^2. The colours in the Figure's scale represent the different magnetic field intensities.

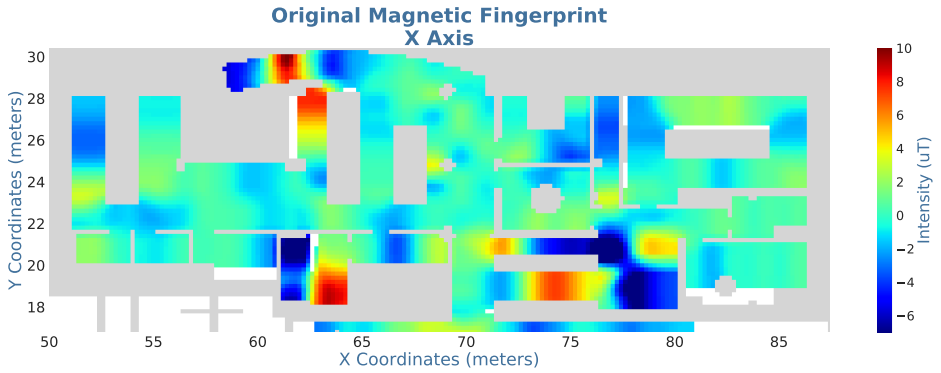


Figure B.3: Geomagnetic fingerprint for the X axis, collected with the traditional methods. The resolution of the fingerprint is 0.04 m^2. The intensity of the magnetic field in each cell is represented by a different colour, as it is described in the Figure's scale.

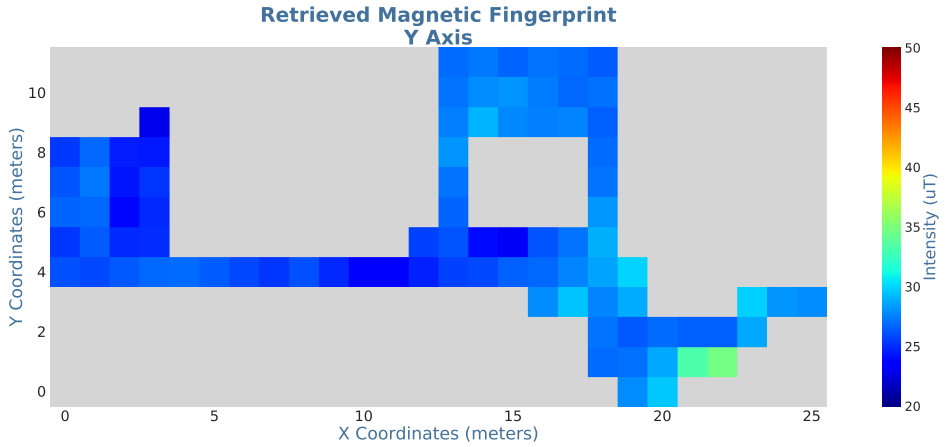


Figure B.4: Retrieved geomagnetic fingerprint for the Y axis, with the defined resolution of one square meter. Each colour represents a different magnetic field intensity, as it is described in the Figure's scale.

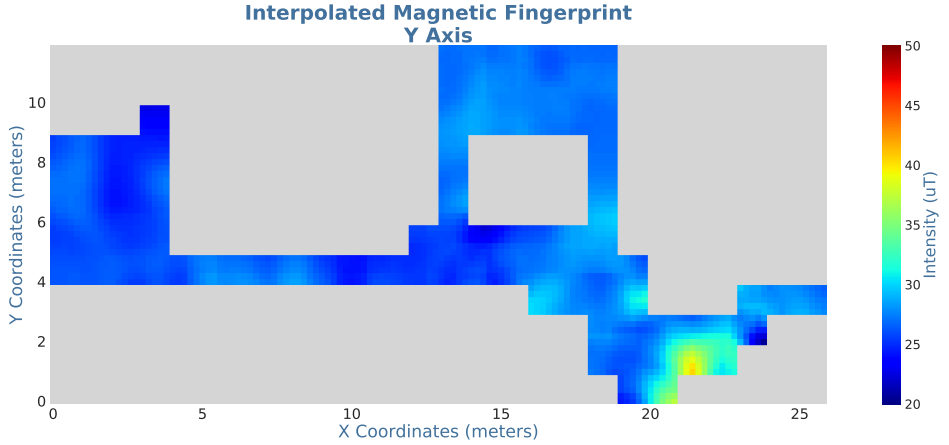


Figure B.5: Interpolated geomagnetic fingerprint for the Y axis, from the values of Figure B.4, with the new resolution of 0.04 m^2. The colours in the Figure's scale represent the different magnetic field intensities.

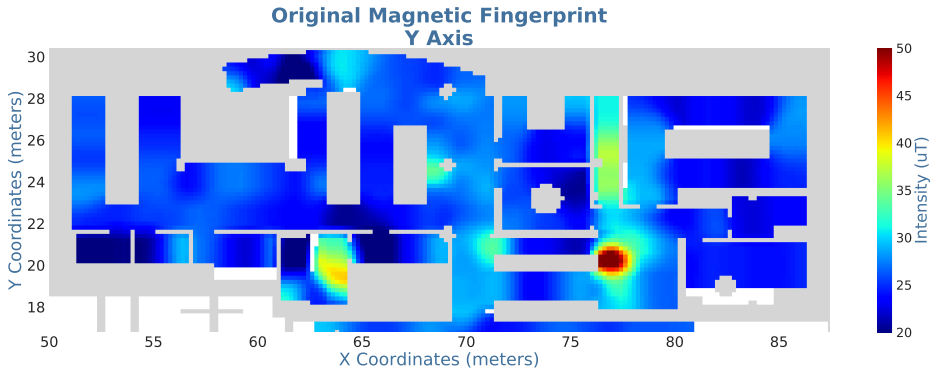


Figure B.6: Geomagnetic fingerprint for the Y axis, collected with the traditional methods. The resolution of the fingerprint is 0.04 m^2. The intensity of the magnetic field in each cell is represented by a different colour, as it is described in the Figure's scale.

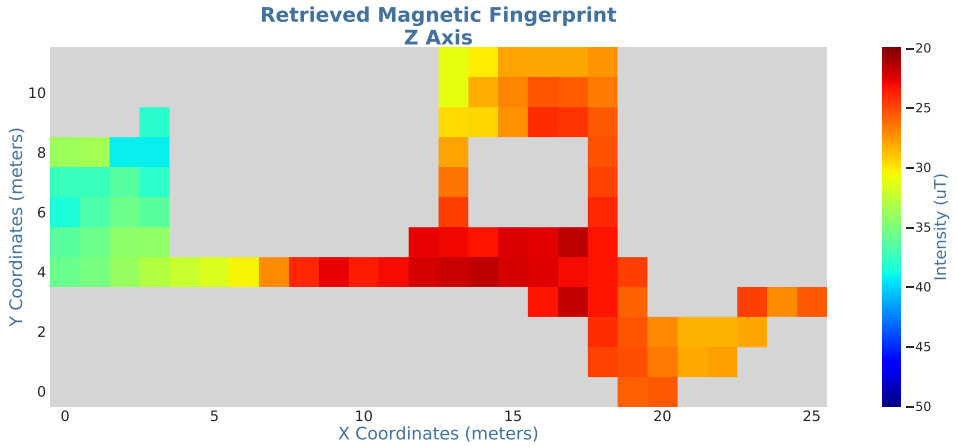


Figure B.7: Retrieved geomagnetic fingerprint for the Z axis, with the defined resolution of one square meter. Each colour represents a different magnetic field intensity, as it is described in the Figure's scale.

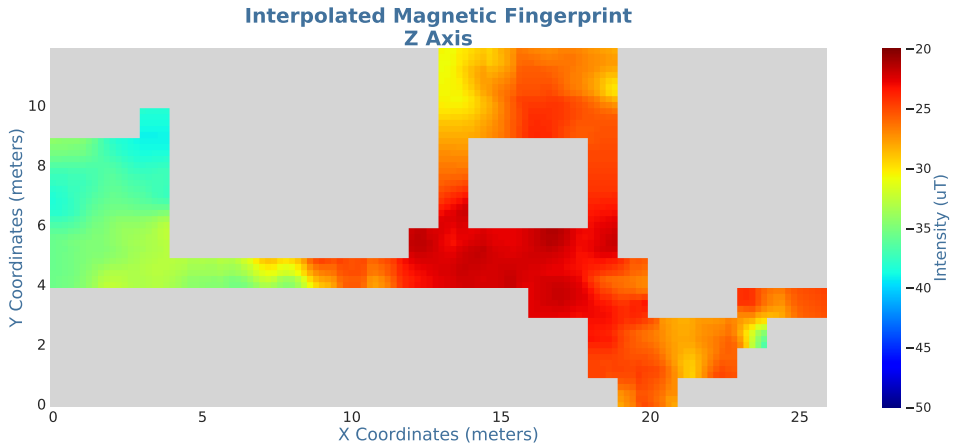


Figure B.8: Interpolated geomagnetic fingerprint for the Z axis, from the values of Figure B.7, with the new resolution of 0.04 m^2. The colours in the Figure's scale represent the different magnetic field intensities.

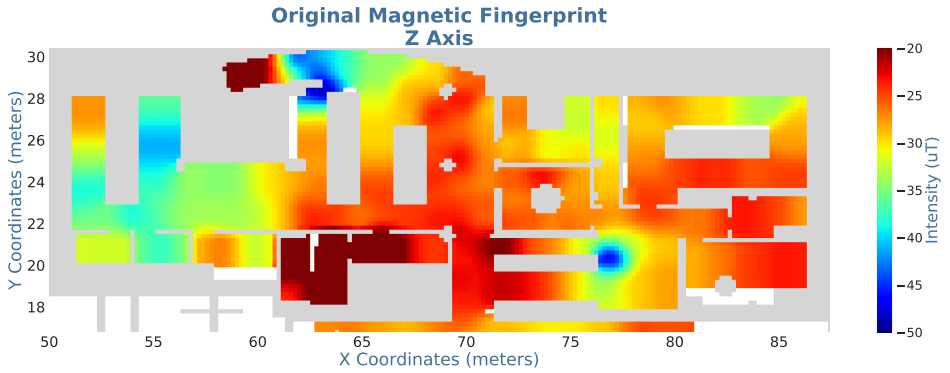


Figure B.9: Geomagnetic fingerprint for the Z axis, collected with the traditional methods. The resolution of the fingerprint is 0.04 m^2. The intensity of the magnetic field in each cell is represented by a different colour, as it is described in the Figure's scale.

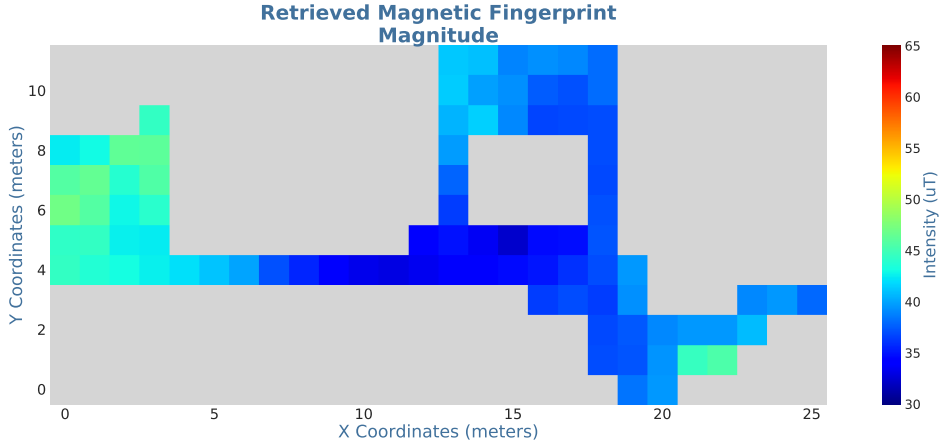


Figure B.10: Retrieved geomagnetic fingerprint for the magnitude of all axes. The fingerprint has a resolution of one square meter. Each colour represents a different magnetic field intensity, as it is described in the Figure's scale.

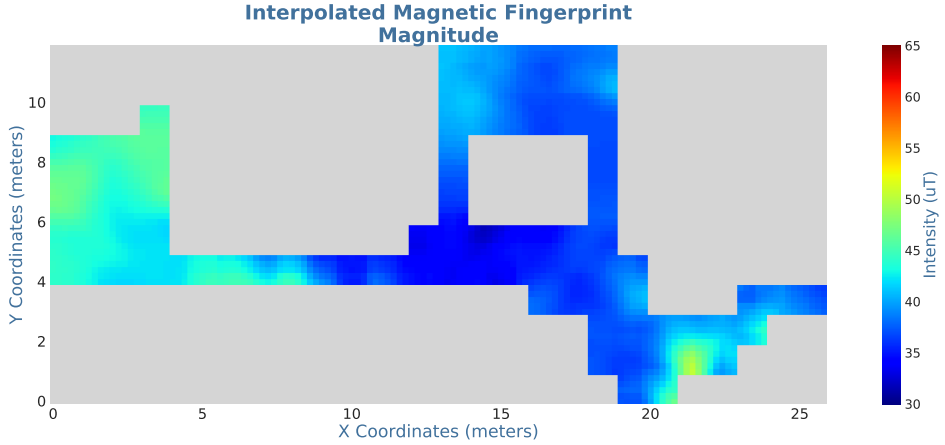


Figure B.11: Interpolated geomagnetic fingerprint for the magnitude of all axes, from the values of Figure B.10, with the new resolution of 0.04 m^2. The colours in the Figure's scale represent the different magnetic field intensities.

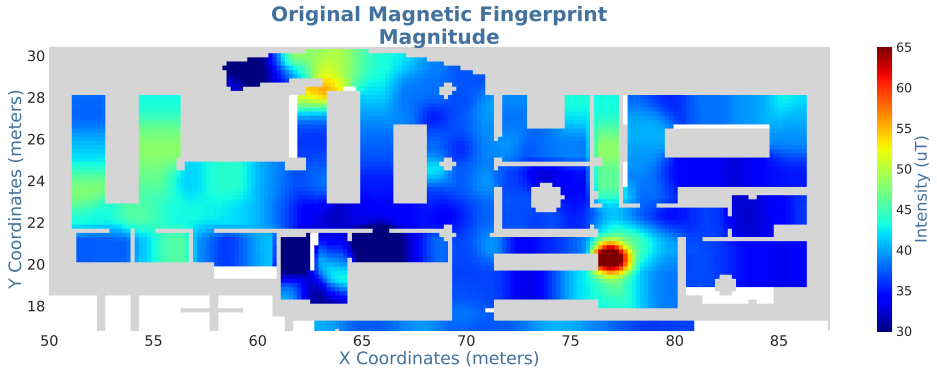


Figure B.12: Geomagnetic fingerprint for the magnitude of all axes, collected with the traditional methods. The resolution of the fingerprints is of 0.04 m^2. The intensity of the magnetic field in each cell is represented by a different colour, as it is described in the Figure's scale.

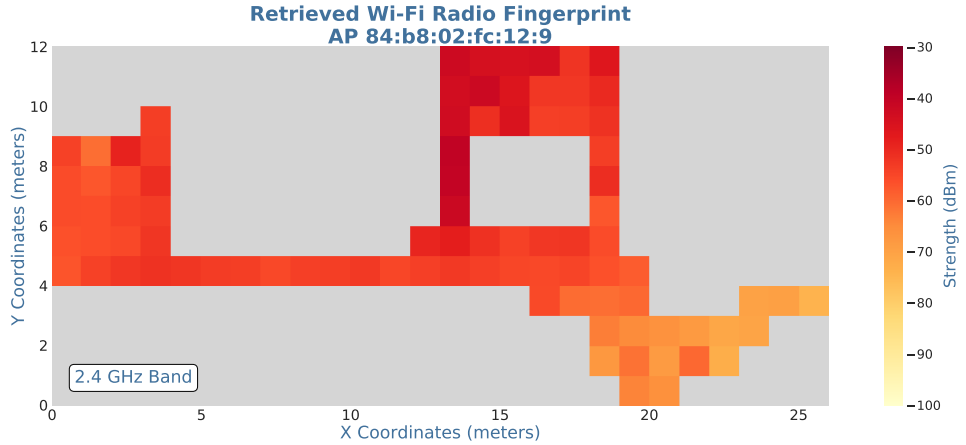


Figure B.13: Obtained Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9 of the test building, with the signal transmitted in the 2.4 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

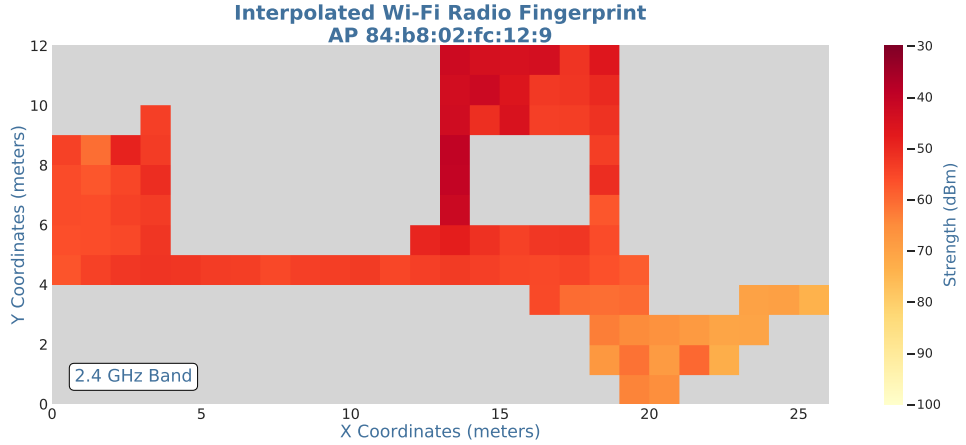


Figure B.14: Interpolated Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9 in the 2.4 GHz radio band, from the values of Figure B.13. The fingerprint has a resolution of one square meter and the colours in the Figure's scale represent the different strengths.

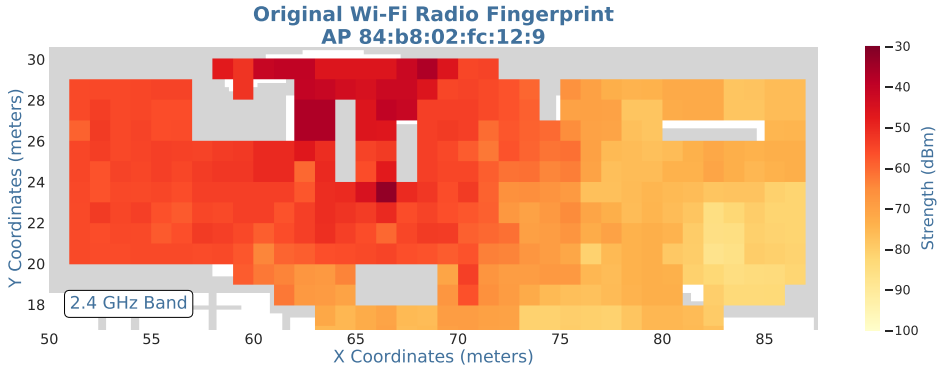


Figure B.15: Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9 in the 2.4 GHz radio band, collected with the traditional methods. The resolution of the fingerprint is one square meter. The strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.

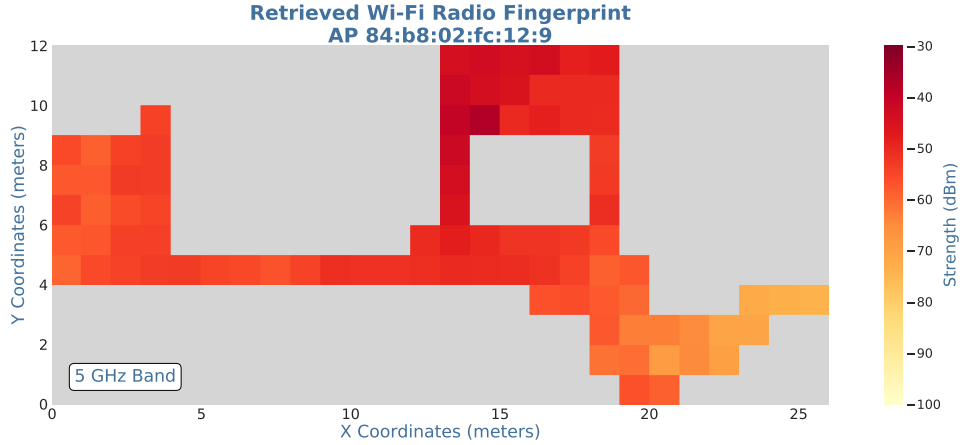


Figure B.16: Obtained Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9 in the 5 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

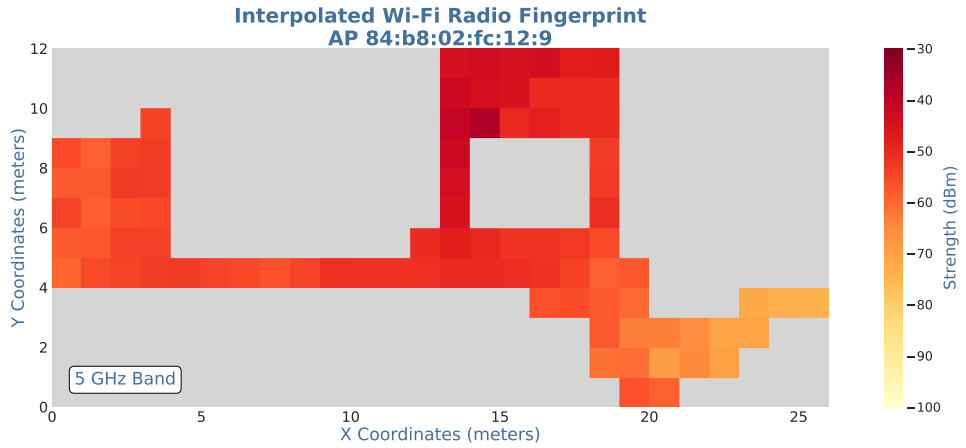


Figure B.17: Interpolated Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9 in the 5 GHz radio band, from the values of Figure B.16. The fingerprint has a resolution of one square meter and the colours in the Figure's scale represent the different strengths.

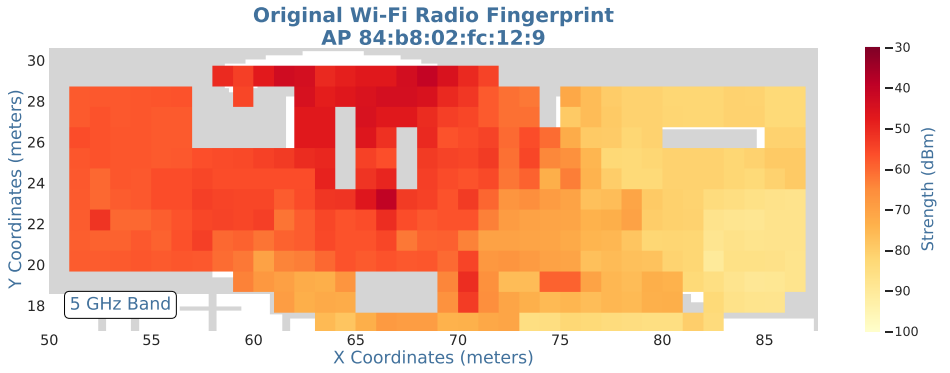


Figure B.18: Wi-Fi radio fingerprint for the AP 84:b8:02:fc:12:9 in the 5 GHz radio band, collected with the traditional methods. The resolution of the fingerprint is one square meter. The strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.

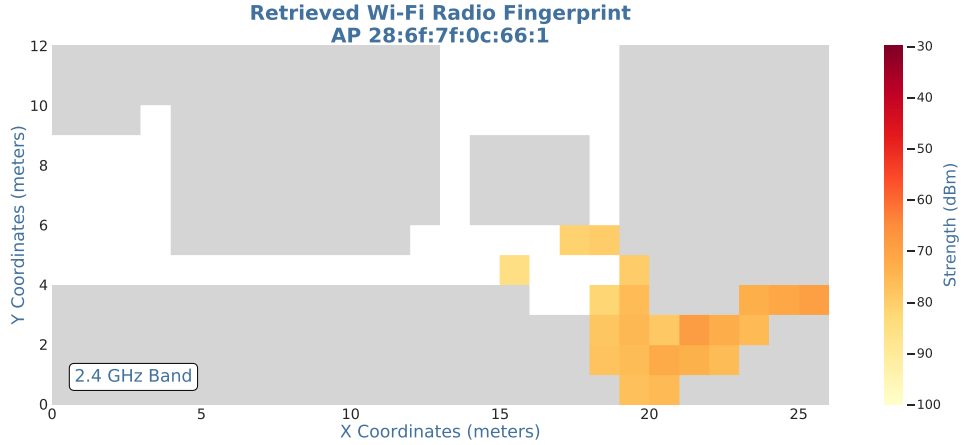


Figure B.19: Obtained Wi-Fi radio fingerprint for the AP 28:6f:7f:0c:66:1 in the 2.4 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

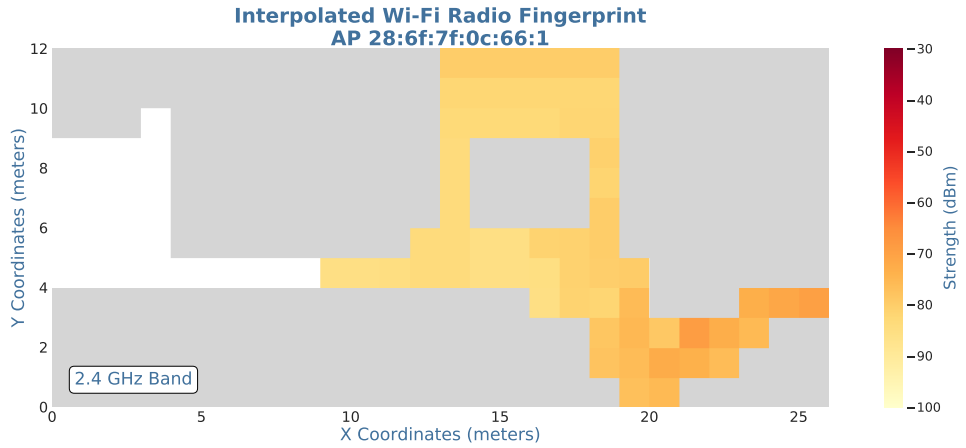


Figure B.20: Interpolated Wi-Fi radio fingerprint for the AP 28:6f:7f:0c:66:1 in the 2.4 GHz radio band, from the values of Figure B.19. The fingerprint has a resolution of one square meter and the colours in the Figure's scale represent the different strengths.

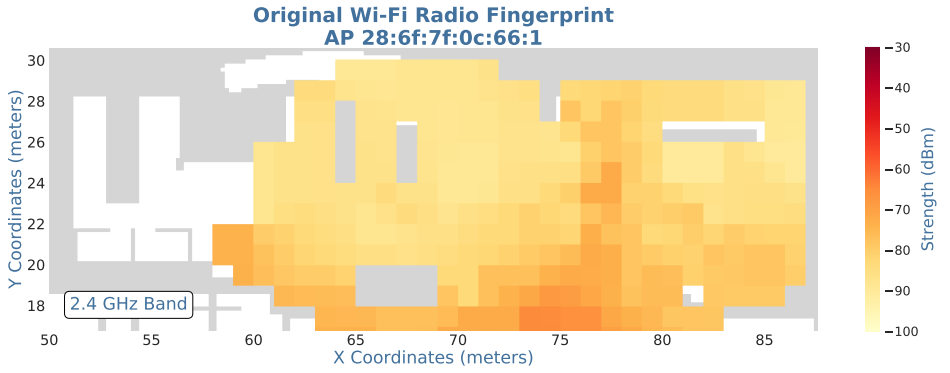


Figure B.21: Wi-Fi radio fingerprint for the AP 28:6f:7f:0c:66:1 in the 2.4 GHz radio band, collected with the traditional methods. The resolution of the fingerprint is one square meter. The strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.

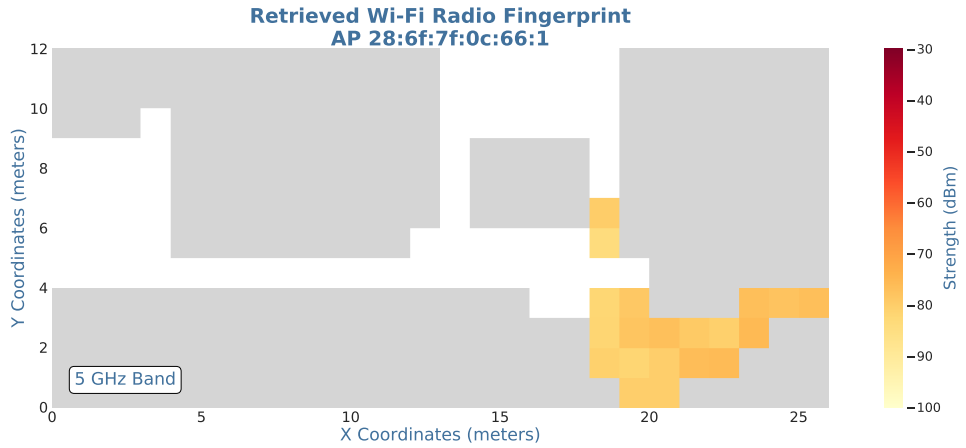


Figure B.22: Obtained Wi-Fi radio fingerprint for the AP 28:6f:7f:0c:66:1 in the 5 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

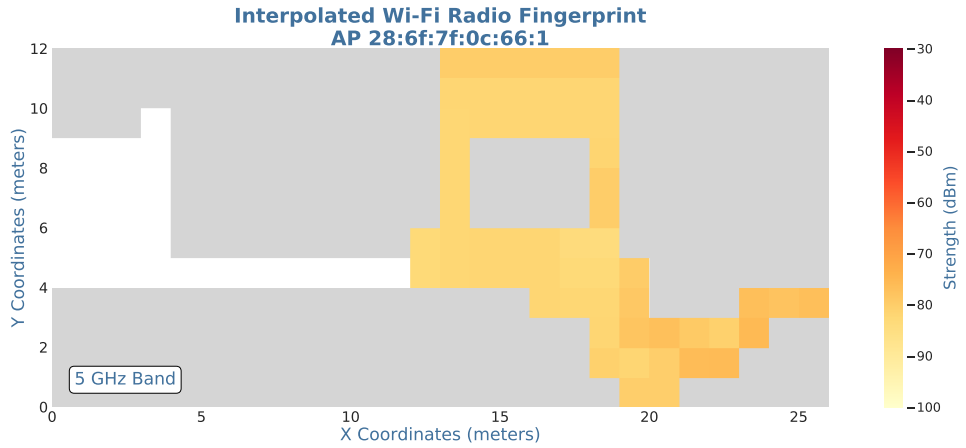


Figure B.23: Interpolated Wi-Fi radio fingerprint for the AP 28:6f:7f:0c:66:1 in the 5 GHz radio band, from the values of Figure B.22. The fingerprint has a resolution of one square meter and the colours in the Figure's scale represent the different strengths.

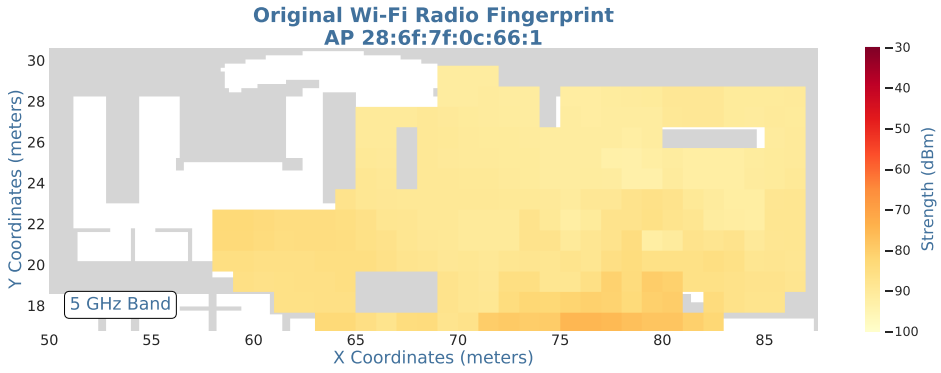


Figure B.24: Wi-Fi radio fingerprint for the AP 28:6f:7f:0c:66:1 in the 5 GHz radio band, collected with the traditional methods. The resolution of the fingerprint is one square meter. The strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.

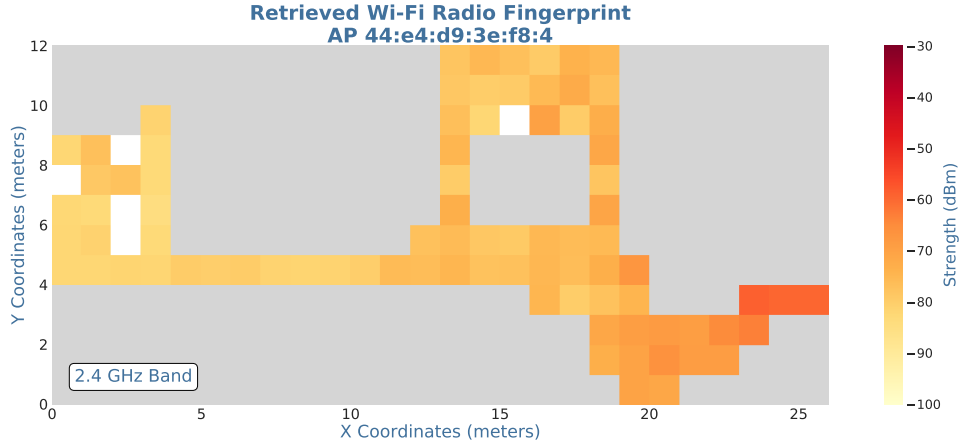


Figure B.25: Obtained Wi-Fi radio fingerprint for the AP 44:e4:d9:3e:f8:4 in the 2.4 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

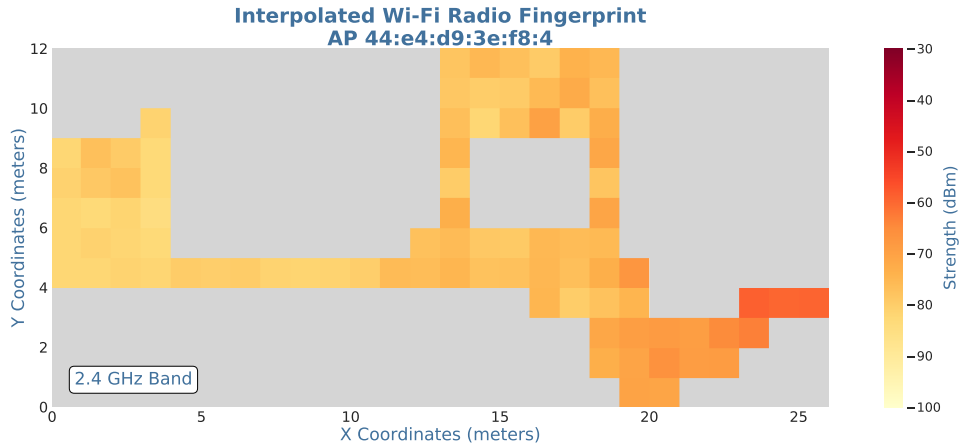


Figure B.26: Interpolated Wi-Fi radio fingerprint for the AP 44:e4:d9:3e:f8:4 in the 2.4 GHz radio band, from the values of Figure B.25. The fingerprint has a resolution of one square meter and the colours in the Figure's scale represent the different strengths.

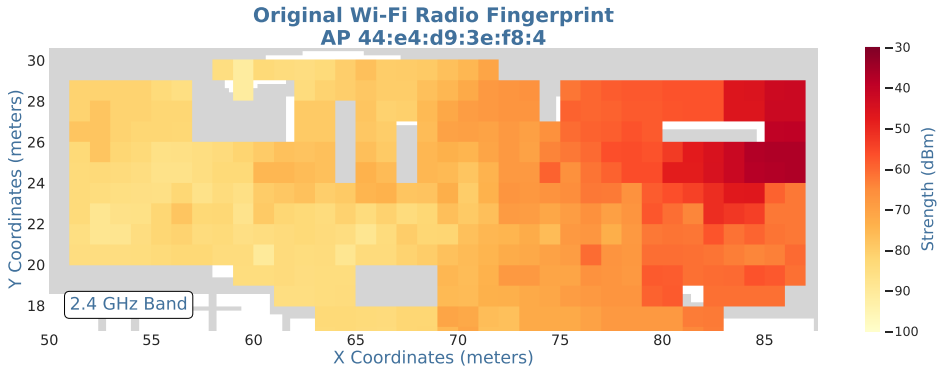


Figure B.27: Wi-Fi radio fingerprint for the AP 44:e4:d9:3e:f8:4 in the 2.4 GHz radio band, collected with the traditional methods. The resolution of the fingerprint is one square meter. The strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.

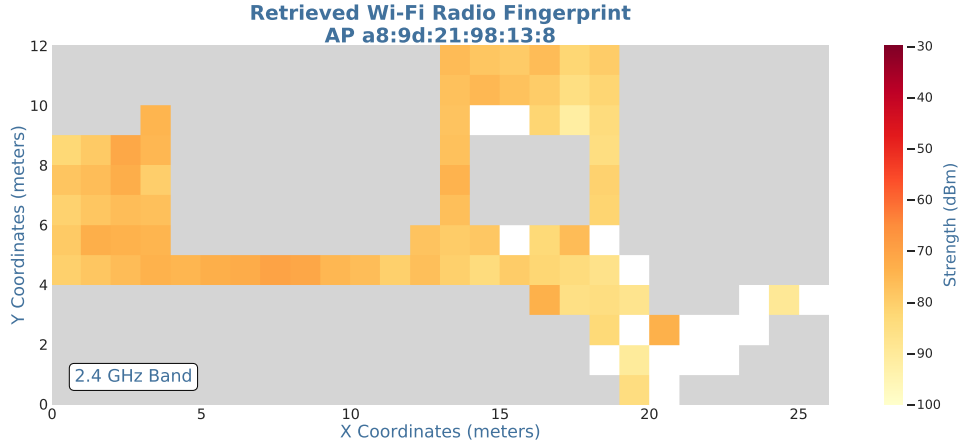


Figure B.28: Obtained Wi-Fi radio fingerprint for the AP a8:9d:21:98:13:8 in the 2.4 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

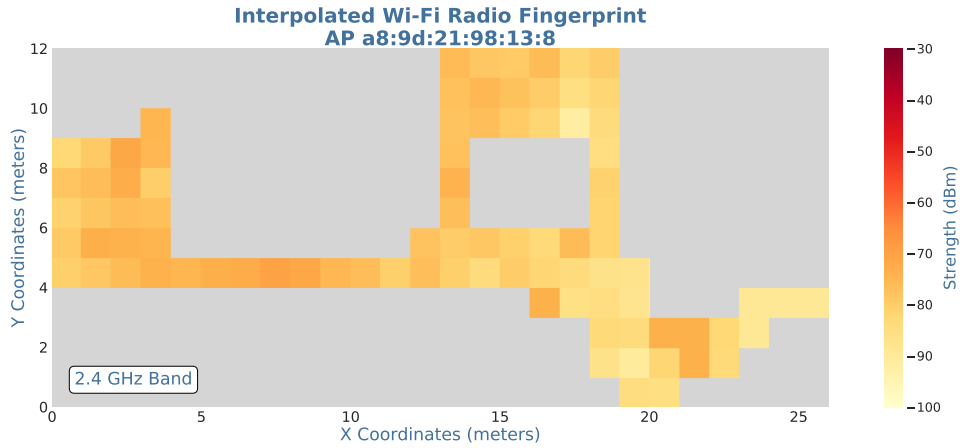


Figure B.29: Interpolated Wi-Fi radio fingerprint for the AP a8:9d:21:98:13:8 in the 2.4 GHz radio band, from the values of Figure B.28. The fingerprint has a resolution of one square meter and the colours in the Figure's scale represent the different strengths.

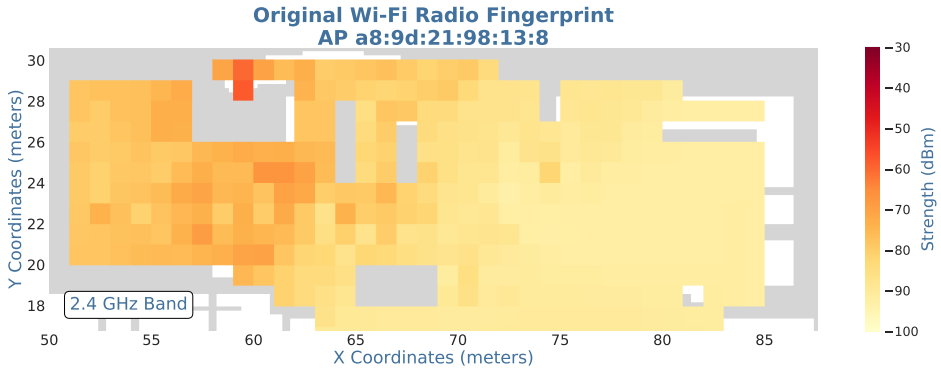


Figure B.30: Wi-Fi radio fingerprint for the AP a8:9d:21:98:13:8 in the 2.4 GHz radio band, collected with the traditional methods. The resolution of the fingerprint is one square meter. The strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.

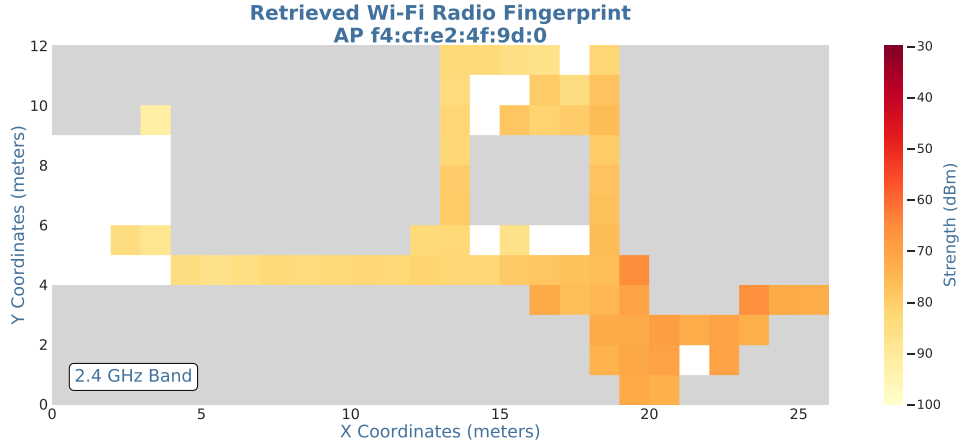


Figure B.31: Obtained Wi-Fi radio fingerprint for the AP f4:cf:e2:4f:9d:0 in the 2.4 GHz radio band. The fingerprint has a resolution of one square meter, where the colour of each cell identifies a different strength, as it is given by the Figure's scale.

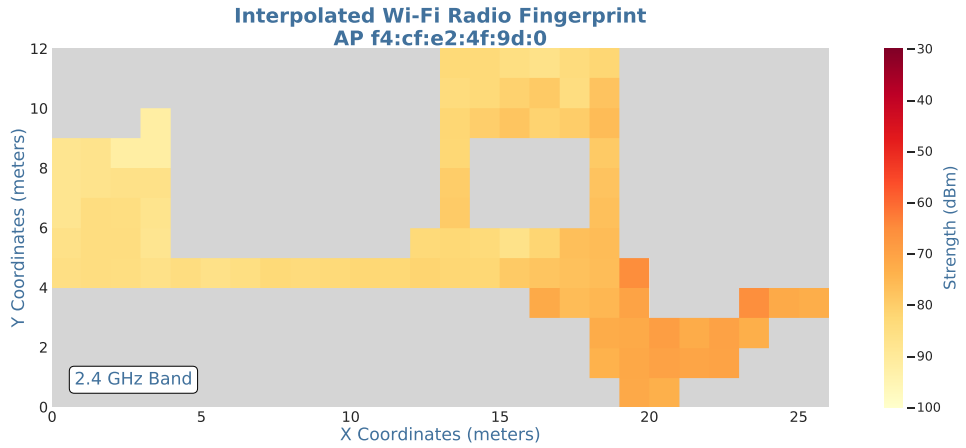


Figure B.32: Interpolated Wi-Fi radio fingerprint for the AP f4:cf:e2:4f:9d:0 in the 2.4 GHz radio band, from the values of Figure B.31. The fingerprint has a resolution of one square meter and the colours in the Figure's scale represent the different strengths.

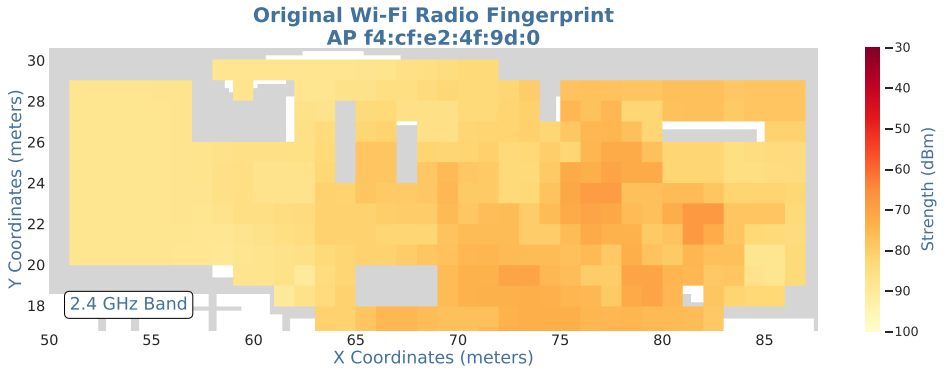


Figure B.33: Wi-Fi radio fingerprint for the AP f4:cf:e2:4f:9d:0 in the 2.4 GHz radio band, collected with the traditional methods. The resolution of the fingerprint is one square meter. The strength of the Wi-Fi radio in each cell is represented by a different colour, as it is described in the Figure's scale.