UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE ELECTRÓNICA

“Smart Hierarchical WiFi Localization System for Indoors”

Author

Noelia Hernández Parra

Supervisors

Dr. D. Manuel Ocaña Miguel
Dr. D. José María Alonso Moral

2014

Doctoral Thesis

“A person who never made a mistake
never tried anything new” - Albert Einstein

Agradecimientos

Después de tanto tiempo, por fin me encuentro escribiendo estos agradecimientos, ese momento que marca el final de esta etapa, o al menos el primer paso para terminarla. Y a pesar de tener tantas personas a las que nombrar, aquí estoy, mirando a la pantalla sin saber muy bien por donde empezar. Pensándolo bien, quizás estas sean las líneas más difíciles de escribir de una tesis.

Así que voy a empezar por mis directores de tesis, Manuel y José. A vosotros os tengo que dar las gracias por vuestro apoyo y constancia, por haber sabido encontrar el equilibrio entre estrés y paciencia que tanta falta hace, especialmente en la etapa final de la tesis. Por creer en mí y haberme guiado hasta aquí.

A toda la gente del Robe-Isis, por tener ese buen humor y estar siempre dispuestos a meter caña. A mis chicos del laboratorio Llama, Edu y Fer porque sin el mal humor de las mañanas a primera hora y el buen humor de media mañana escribir la tesis no habría sido lo mismo. A mis chicos del laboratorio de arriba, Javi y Rober, por darme ánimos todas y cada una de las veces que os habéis cruzado conmigo. A mis chicos del Este, Carlos y Raúl (ąy Llorca!), a pesar de que últimamente no os veo mucho (sé que soy una desertora). Y a los "jefes", Luismi y Sotelo, porque siempre habéis estado ahí.

También quiero dar las gracias a todos los que se fueron, pero especialmente a vosotros: Edu, qué habría hecho yo sin esas conversaciones con un café a cualquier hora del día (ągracias por seguir ahí a pesar de estar lejos!). Sergi (o Tony...), por ser como eres, por esos buenos ratos después de comer. Osquitar, por ser una de las personas que más me han hecho reír durante estos años, esos momentos valen oro (ąMonooo!). Iván, por ser mi compañero de penas y alegrías durante los años pre-tesis. ąSin vosotros esto no es lo mismo!

Y llega tu turno. Tengo que agradecértelo especialmente a ti, porque esta tesis es casi tan tuya como mía, porque sin tu ayuda, tus ideas y tu apoyo me habría quedado en el camino. Porque todos sabemos que terminar la tesis es una locura, y tú has estado ahí en lo bueno y en lo malo. Porque más allá de la tesis has hecho que todo vuelva a tener sentido. Esta vez, vamos a evitar "la maldición de los agradecimientos de la tesis" no poniendo nombres. Para qué, si no hacen falta...

Por supuesto, tengo que agradecer TODO a mis padres, porque gracias a vosotros soy como soy, porque habéis sabido entender que haya desaparecido estos últimos meses, porque me habéis apoyado en todas mis decisiones. Mamá, papá, ąya acabo!

Y para terminar, tengo que dar las gracias a todos aquellos que de una forma u otra se cruzaron en mi camino y aportaron su granito de arena.

Gracias a todos.

Resumen

En los últimos años, el número de aplicaciones para *smartphones* y *tablets* ha crecido rápidamente. Muchas de estas aplicaciones hacen uso de las capacidades de localización de estos dispositivos. Para poder proporcionar su localización, es necesario identificar la posición del usuario de forma robusta y en tiempo real. Tradicionalmente, esta localización se ha realizado mediante el uso del GPS que proporciona posicionamiento preciso en exteriores. Desafortunadamente, su baja precisión en interiores imposibilita su uso.

Para proporcionar localización en interiores se utilizan diferentes tecnologías. Entre ellas, la tecnología WiFi es una de las más usadas debido a sus importantes ventajas tales como la disponibilidad de puntos de acceso WiFi en la mayoría de edificios y que medir la señal WiFi no tiene coste, incluso en redes privadas. Desafortunadamente, también tiene algunas desventajas, ya que en interiores la señal es altamente dependiente de la estructura del edificio por lo que aparecen otros efectos no deseados, como el efecto multicamino o las variaciones de pequeña escala. Además, las redes WiFi están instaladas para maximizar la conectividad sin tener en cuenta su posible uso para localización, por lo que los entornos suelen estar altamente poblados de puntos de acceso, aumentando las interferencias co-canal, que causan variaciones en el nivel de señal recibido.

El objetivo de esta tesis es la localización de dispositivos móviles en interiores utilizando como única información el nivel de señal recibido de los puntos de acceso existentes en el entorno. La meta final es desarrollar un sistema de localización WiFi para dispositivos móviles, que pueda ser utilizado en cualquier entorno y por cualquier dispositivo, en tiempo real.

Para alcanzar este objetivo, se propone un sistema de localización jerárquico basado en clasificadores borrosos que realizará la localización en entornos descritos topológicamente.

Este sistema proporcionará una localización robusta en diferentes escenarios, prestando especial atención a los entornos grandes. Para ello, el sistema diseñado crea una partición jerárquica del entorno usando K-Means. Después, el sistema de localización se entrena utilizando diferentes algoritmos de clasificación supervisada para localizar las nuevas medidas WiFi. Finalmente, se ha diseñado un sistema probabilístico para seguir la posición del dispositivo en movimiento utilizando un filtro Bayesiano. Este sistema se ha probado en un entorno real, con varias plantas, obteniendo un error medio total por debajo de los 3 metros.

Palabras clave: Servicios Basados en Localización, Localización WiFi en Interiores, Aprendizaje Automático, Computación Flexible.

Abstract

Recent years have seen a rapid growth of smartphone and tablet applications. Many of these applications make use of the localization capabilities of these devices in what are called Location Based Services. To be able to provide this kind of services, a reliable and real time identification of the user location is needed. Traditionally, global localization has been carried out through GPS, which provides accurate localization when working outdoors. Unfortunately, the use of GPS is affected by Non-Line-Of-Sight, making GPS localization in indoor environments not suitable.

Different technologies are being used to provide indoor localization, among them, WiFi is a common choice due to its important advantages: there are WiFi access points in most buildings and measuring WiFi signal is free of charge even for private WiFi networks. Unfortunately, it also has some disadvantages: when working indoors the signal strength is strongly dependent on the building structure and some other non-desired effects appear, such as the multipath effect, signal absorption and the small scale variations. Moreover, since WiFi networks are deployed with the goal of maximizing connectivity and disregarding localization tasks, there are usually many access points distributed over the environment increasing the so-called co-channel interferences, which cause high variations in the received signal strength from the access points.

The goal of this thesis is the localization of mobile devices in indoor environments using as the only available information the signal received from the already existing access points in the environment. Since WiFi is pre-installed in most of the buildings, there is no need to either modify the environment or add new devices to it. Then, the final research objective is to achieve robust WiFi real-time localization for mobile devices, available to be deployed in any environment and to be used by any device.

To achieve this objective, a hierarchical fuzzy-based approach is proposed to perform localization in topologically described environments. This new approach is able to deal with multi-floor large environments that have been previously neglected in the literature. To do so, the system creates a hierarchical partition of the environment using similarity clues in a K-Means-based approach. Then, the localization system is trained using different supervised learning algorithms to classify the new WiFi samples through the hierarchical tree of the environment partition. Finally, a Bayesian filter to track the position of a device in motion has been designed. This approach was tested in a multi-floor real environment, obtaining an overall mean error distance under 3 metres.

KeyWords: Location Based Services, WiFi Indoor Localization, Machine Learning, Soft Computing.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ABSYNTHE	Abstraction, Synthesis and Integration of Information for Human-Robot Teams.
AP	Access Point.
BPF	Backtracking Particle Filter.
CDF	Cumulative Distribution Function.
ECSC	European Centre for Soft Computing.
EKF	Extended Kalman Filter.
FDT	Fuzzy Decision Trees.
FDT-S	Fuzzy Decision Trees with Simplification.
FRBC	Fuzzy Rule-Based Classifier.
FURIA	Fuzzy Unordered Rule Induction Algorithm.
GP	Gaussian Process.
GPS	Global Positioning System.
GSM	Global System for Mobile Communications.
GUAJE	Generating Understandable and Accurate fuzzy models in a Java Environment.
HILK	Highly Interpretable Linguistic Knowledge.
HMM	Hidden Markov Model.
IEEE	Institute of Electrical and Electronics Engineers.
IMU	Inertial Measurement Unit.
ISM	Industrial, Scientific and Medical.
KF	Kalman Filter.
KNN	K-Nearest Neighbour.

LBS	Location-Based Service.
MDP	Markov Decision Process.
NLOS	Non-Line-Of-Sight.
PF	Particle Filter.
PFDT	Pruned Fuzzy Decision Trees.
PFDT-S	Pruned Fuzzy Decision Trees with Simplification.
POMDP	Partially Observable Markov Decision Process.
QR	Quick Response code.
RF	Radio Frequency.
RFID	Radio Frequency IDentification.
RSS	Received Signal Strength.
SFP	Strong Fuzzy Partition.
SNR	Signal-to-Noise Ratio.
SOM	Self-Organizing Map.
SVM	Support Vector Machine.
UAH	University of Alcalá.
UWB	Ultra Wide Band.
VRC	Variance Ratio Criterion.
WAF	Wall Attenuation Factor.
WLAN	Wireless Local Area Network.
WM	Wang and Mendel.
WM-S	Wang and Mendel with Simplification.

Chapter 1

Introduction

1.1 Motivation

The development and history of the mobile phone has seen a tremendous number of changes since the first cell phones were introduced. It was at the beginning of the 1980s when mobile phone technology started to be deployed commercially. Since then, there have been many new mobile phone systems introduced, and many improvements have been made in this form of radio communications technology. The mobile phones themselves as well as the associated equipment have become much cheaper and far smaller.

The first systems to be launched were based on analogue technology. These early phones were very large and could certainly not be placed in a pocket like the phones of today (Figure 1.1).

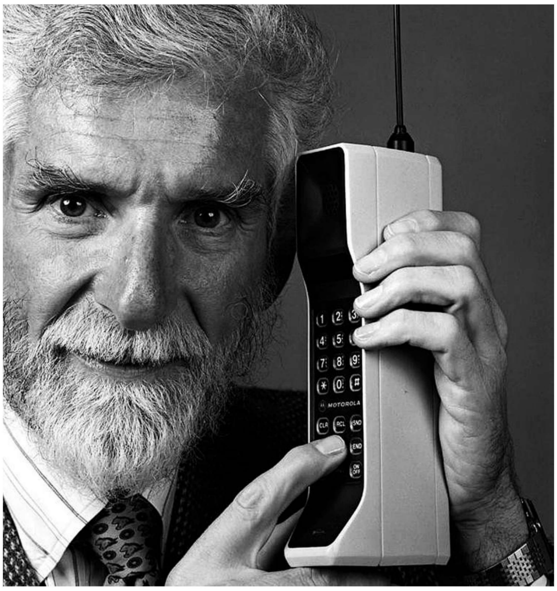


Figure 1.1: Martin Cooper with the handset he used to make the first mobile phone call on April 1973.

The first mobile phones helped people to be connected everywhere. While the initial phones had reduced connectivity capabilities, allowing only to make calls or send SMS, as the usage of phones increased new possibilities emerged using the phones for data transfer. They could be used to download information from the Internet or to send video but, at a very low rate speed. The introduction of the first commercial mobile phone with a built-in *Global Positioning System (GPS)*, developed by the Finnish company Benefon at the end of 1999, allowed the use of these phones for localization. The third generation (3G) systems aim was to provide a relatively high-speed data transfer capability. These 3G systems were able to provide a significant improvement in capability over the previous ones.

At this point, the first Apple's iPhone made everybody realise that connectivity alone was not enough. With the introduction of a multi-touch touchscreen and the app store, where many applications could be found, a new understanding of the mobile phone industry appeared. The mobile phones were not just to make calls anymore, but they could be used for gaming, being connected to social media, etc. That is how the era of the smartphones started (Figure 1.2).

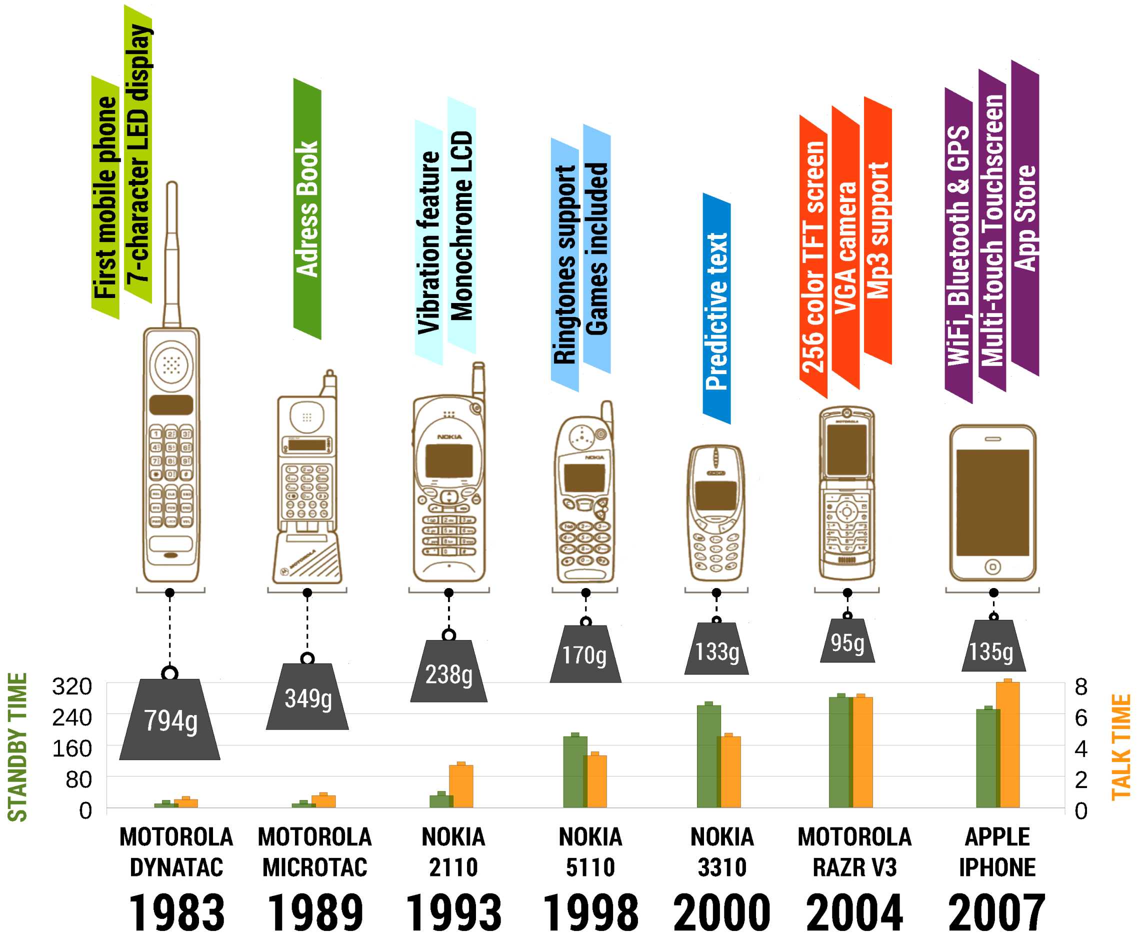


Figure 1.2: Mobile phones evolution.

By 2009, it had become clear that, at some point, 3G networks would be overwhelmed by the growth of bandwidth-intensive applications like streaming media. Consequently, the industry began looking to data-optimized fourth generation (4G) technologies, with the promise of speed improvements up to 10-fold over existing 3G ones.

With all these advances in the mobile phone industry, the use and availability of smartphone and tablet applications have grown rapidly [BI Intelligence, 2013] (Figure 1.3).

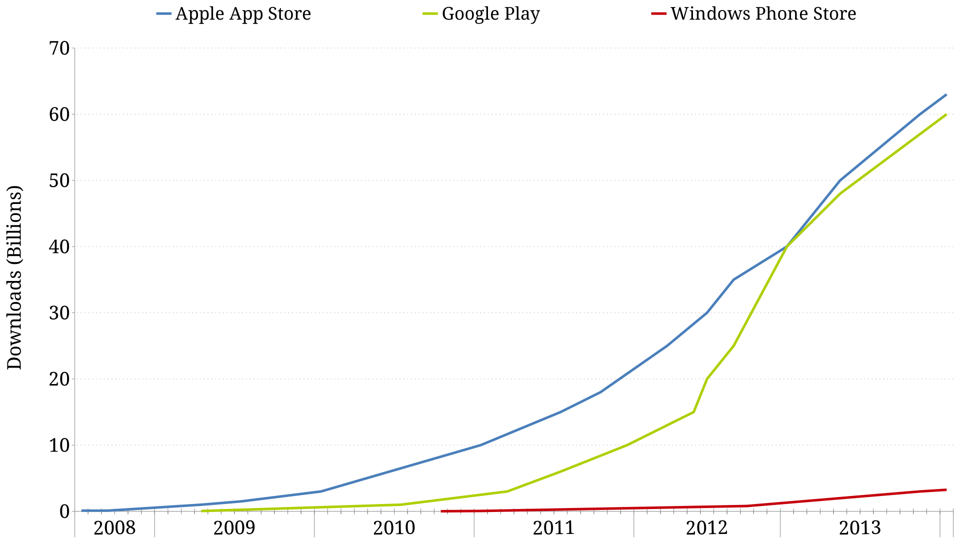


Figure 1.3: Cumulative number of apps downloaded from the Apple App Store, Android Google Play and Windows Phone Store from June 2008 to January 2014.

Many of these applications provide *Location-Based Services (LBSs)* making use of the localization capabilities of these devices. An LBS can be defined as a service that integrate a mobile device's location or position with other information to provide added value to the user.

As part of its annual Mobile Life study [TNS, Kantar Group, 2012], TNS found a 19% of the world's 6 billion mobile users were already using LBSs, with a 62% of the non LBS's users aspiring to do so in the future. Navigation with maps and GPS is currently the most popular motivation behind LBSs (46%). But, besides the classic navigation, the use of LBSs allows for applications in very different areas:

- Marketing:
  - Store or services locators. Using location-based intelligence, customers can quickly find the nearest store location, restaurant, bank, movie theatre, etc.
  - Proximity-based marketing. Local companies can push ads only to individuals within the same geographic location. Location-based mobile marketing delivers ads to potential customers within that city who might actually act on the information.

- Social:
  - Social events recommendation based on the user's location and preferences [Quercia et al., 2010].
  - Places recommendation based on the users' location and their social network profile [Saiph Savage et al., 2012] [Cheng et al., 2013].
- Healthcare systems:
  - Staff and equipment location and tracking in hospitals [Molina and Alba, 2011] [Ekahau, 2014].
  - Patient location and tracking. Tracking the position and actions of the patients is required for medical observation or accident prevention [Pourhomayoun et al., 2012].
  - Healthcare workers assignment. Assigning the closest home healthcare worker to assist a patient's call [Christensen et al., 2007].
- Intelligent Transportation Systems:
  - Travel information. An LBS can deliver real-time information to the smartphone, such as traffic updates or weather reports, so that the user can plan accordingly [Stamoulakatos and Sykas, 2003].
  - Roadside assistance. In the event of a blown tire or accident, many roadside assistance companies provide an application that allows them to track the user exact location without the need for giving directions [eCall, 2007].
  - Toll charging. Automatic payment for the use of road infrastructures through a smartphone LBS [Satelise, 2010].
- Security:
  - Object search. An LBS can track the position of different objects allowing anti-theft/anti-lost of them or preventing of stray children [Jeon and Kim, 2013].
  - Fraud prevention. An LBS can create another level of security by matching a customer's location through the smartphone to a credit card transaction. Tying the smartphone's location to a credit card allows to flag transactions made across several geographic locations over a short time [Choey et al., 2003].
- Logistics:
  - Mobile workforce management. For companies that employ individuals out in the field or at multiple locations, an LBS allows employees to check-in at a location using their mobile device.
  - Inventory control at warehouses [Zhao and Zhang, 2011].
- Guidance at museums or public buildings [Hammadi et al., 2012].

- Gaming: The user location can be part of the game play increasing the user experience [Guo et al., 2012].

- Augmented reality: Virtual information depending of the location is added to real environments to simplify its interpretation or provide it in a more attractive way [Chang, 2011] [Alappanavar et al., 2013] [Google Glass, 2014].

To be able to provide this kind of services, LBSs need to accurately identify the location of the user. This has boosted the need for a reliable and real time localization for mobile devices.

Traditionally, global localization has been carried out through GPS [Enge and Misra, 1999], which provides accurate localization when working outdoors. This way, GPS has become the main technology for positioning in outdoor environments. Encouraged by the accuracy of the GPS outdoors and due to the fact that almost every new smartphone and tablet have built-in GPS receivers, most of the currently existing LBSs are oriented to outdoor environments. However, in indoor environments LBSs are of equal interest in a wide range of personal and commercial applications (Figure 1.4).

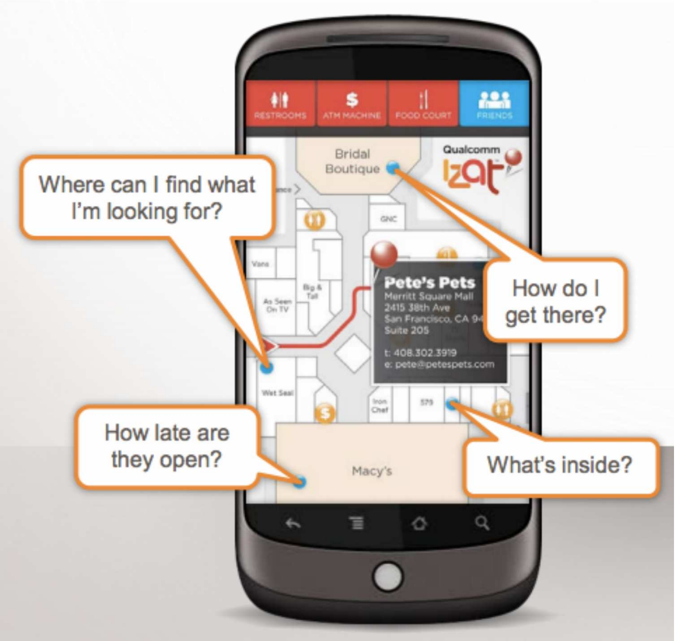


Figure 1.4: Indoor location-based services application.

All these applications require accurate indoor localization for the strategic planning of the navigation or to provide guidance to the final target. Unfortunately, the use of GPS is affected by *Non-Line-Of-Sight (NLOS)*, satellite signals are attenuated and scattered by roofs, walls and other objects making GPS localization in indoor environments not suitable. Providing indoor localization requires the use of other technologies.

Different technologies are being used to provide indoor localization: infrared [Want et al., 1992], ultrasound [Priyantha et al., 2000], laser [Barber et al., 2002], computer vision [Krumm et al., 2000], *Radio Frequency (RF)* [Bahl and Padmanabhan, 2000] [García-Valverde et al., 2013] or *Global System for Mobile Communications (GSM)* [Parodi et al., 2006].

The decision about using one of these technologies is mainly determined by the accuracy required by the final application and the cost of the system deployment. The required accuracy for indoor LBSs is usually in the order of metres [Mautz, 2012].

Some of the previously named technologies, such as ultrasound or laser, may accomplish the accuracy requirements, but they require the use of additional hardware. Some other, such as *Ultra Wide Band (UWB)* or *Radio Frequency IDentification (RFID)* require the installation of artificial marks in the environment increasing the cost of the system and forcing for an specific deployment on every target environment.

If the goal is to provide a generic location service available on any device (smartphones, tablets, laptops, etc.) and in any environment, an already available technology in both the devices and the environments should be used. With this requirement in mind, the following technologies can be used:

- **GSM**: it is a common choice to provide LBSs outdoors. Although some attempts have been made in providing indoor localization using GSM as in [Varshavsky et al., 2007], the minimum achieved error was around 7 meters for a 95^{th} percentile for a floor and, to date, this kind of systems have not been proved useful in distinguishing between building floors. Moreover, depending on the building structure, it does not work properly in indoor environments due to signal blockage.

- **Cameras**: computer vision techniques applied to the images obtained from a camera are commonly used to provide indoor positioning in different areas. But, when applying these techniques to perform localization using a mobile phone, different problems appear: computer vision techniques are computationally expensive and need the camera to be pointing to the environment continuously, decreasing the usability of the system. Moreover, these systems perform position tracking, so the initial position is needed in order to be tracked. They are also strongly dependent on the illumination conditions and have to be calibrated to get accurate information.

  Some camera-based systems use the mobile device's built-in camera to scan codes, such as *Quick Response codes (QRs)*, to obtain the position of the device [Humanes et al., 2013]. This kind of systems requires the installation of QR codes over the environment and to actively scan the codes to update the device's position.

- **RF**: A major group of indoor positioning systems utilizes RF signals emitted by common wireless communication networks. Among all the technologies, WiFi is arising as the most popular one. This is probably due to the advantages of using WiFi for indoor localization: WiFi *Access Points (APs)* are deployed in almost every building and measuring the WiFi signal is free of charge even for private networks. This fact allows to install a localization system based on WiFi without doing any modifications in the environment. Moreover, almost every device is already equipped with a WiFi interface, and no special requirements are needed to perform the localization. This way, almost every device can benefit from indoor LBSs using WiFi.

Unfortunately, using WiFi for indoor localization also has some disadvantages: although the *Received Signal Strength (RSS)* decays logarithmically on free space, the multipath effect [Rappaport, 1996], obstacles and the small scale effect [Youssef and Agrawala, 2003] make the RSS a complex function of the distance. In addition, the presence of people heavily affects the RSS absorbing part of the electromagnetic signal [Bahillo et al., 2009]. As a result, it is very difficult to model the RSS in indoor environments and the provided accuracy is lower than using other technologies such as laser.

Although lot of research has taken place in WiFi indoor localization systems, it remains as an open problem, and their accuracy can still be improved.

1.2 Scope of this thesis

Since 2002, the researchers of the RobeSafe (Robotics and eSafety) Research Group at the Department of Electronics of the University of Alcalá have been working on the problem of indoor localization at different areas. Important results have been achieved mainly in the robotics area, where a navigation system based on WiFi signal strength and ultrasounds was developed [Ocaña, 2005].

The RobeSafe Group has focused its efforts on some important aspects in order to develop these localization systems. The group is interested in developing non-invasive systems, which means to use the own infrastructure of the environment without adding extra devices or technologies. RobeSafe also aims for systems which do not depend on a specific technology and can be applied to a high range of devices such as, cellphones, mobile robots, etc. Finally, developing low-cost solutions is always a constraint for the RobeSafe group.

This thesis is part of *Abstraction, Synthesis and Integration of Information for Human-Robot Teams (ABSYNTHE)* project [Alonso et al., 2012]. The ABSYNTHE project goal is the development of novel tools and approaches to facilitate communication and coordination in human-robot teams. Localization is one of the most important information when human-robot teams are collaborating. Each team member must be aware of its own location but also of the location of the others. Robot localization must be accurate (in the range from zero to three meters) while human localization does not need to be so accurate but understandable because humans are able to manage poor quality information about their locations. This thesis is focused in the human localization stage, but it can also be directly used by robots, or even improved by adding the information from the robot's own sensors.

The goal of this thesis is the localization of mobile devices in indoor environments using as the only available information the signal received from the already existing APs in the environment. Since WiFi is pre-installed in most of the buildings, there is no need to either modify the environment or add new devices to it. Then, the final research objective is to develop a robust WiFi real-time localization for mobile devices, available to be deployed in any environment and to be used by any device.

1.3 Document structure

After the introduction in Chapter 1, Chapter 2 contains a brief review of the most significant research on WiFi indoor localization.

In Chapter 3 an analysis of the WiFi signal behaviour is exposed. The most important characteristics of WiFi technology are reviewed and the most important signal variations to take into consideration when designing a WiFi localization system are analysed.

The problem of the small scale variations on static positions will be tackled using Fuzzy Rule-Based Classifiers in Chapter 4.

In Chapter 5, the challenge of designing a WiFi localization system for large environments, crowded with APs and not deployed for localization purposes, will be faced.

Chapter 6 presents an improvement of the system to localize a device in motion using a Bayesian filter framework. Results for experiments under real conditions are presented and discussed.

Chapter 7 contains the conclusions and main contributions of this work, and future research lines that may spring from it.

Finally, Appendix A describes the software developed for topology-based localization that implements the work described in this thesis and Appendix B summarizes the main publications derived from this PhD dissertation.

Chapter 2

State of the Art

Indoor localization has been one of the most active fields of research for the last decade. But, as it was mentioned in the introduction, WiFi indoor localization is still an open problem. At present, there are some available indoor localization systems based on WiFi. One of the most famous ones is the Google localization service (Figure 2.1) which combines GPS, WiFi and GSM on indoor Google Maps [Indoor Google Maps, 2014] to provide positioning in buildings, but its accuracy is not enough to provide an indoor guidance service yet. Other systems such as the Active Badge [Want et al., 1992], the Cricket [Priyantha et al., 2000] and Ekahau positioning engine [Ekahau, 2014] rely on especially designed hardware. These kind of purpose-built systems can be expensive and hard to implement on a world-wide scale.

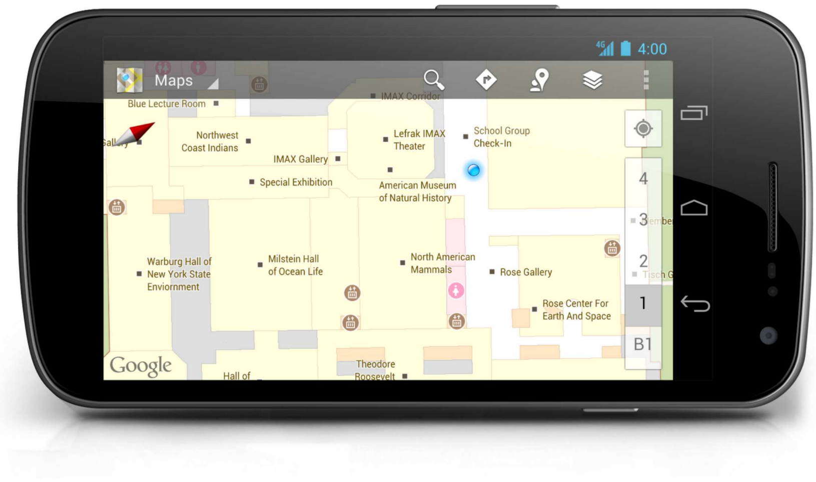


Figure 2.1: Indoor Google Maps.

This chapter presents a brief survey of the state of the art in WiFi indoor localization. First, the different approaches for WiFi indoor localization are described, providing an overview of the most remarkable methods and those that are related to the contents of the following chapters of this thesis. Then, a discussion on the revised research is carried out to, finally, point out the specific objectives of the thesis.

2.1 WiFi localization systems

Given the diversity of WiFi localization methods published in the literature, different classifications of them are possible [Liu et al., 2007]. Even though classification of some of the methods is not evident and there is a certain degree of overlapping between groups, they can be classified in terms of the algorithms that are used to solve the localization problem. Using this criterion, the studies can be separated in the following categories:

- **Deterministic:**
  - **Propagation model based:** Localization is carried out estimating the distance to nearby APs by means of a WiFi signal propagation model [Kotanen et al., 2003] [Bose and Foh, 2007] [Mazuelas et al., 2009] [Yang and Chen, 2009] [Herranz, 2013]. Propagation models describe how the signal is propagated in the environment and they are used to translate the RSS into a distance. The APs location in the environment is normally known a priori. Using the distance to the APs and their positions in the environment, the typical choice is to use lateration algorithms to perform the localization.
  - **Fingerprint based:** These systems use a fingerprint database stored in a training stage to obtain the estimated position of the device by means of different classification algorithms [Bahl and Padmanabhan, 2000] [Yim, 2008] [Mengual et al., 2010] [García-Valverde et al., 2012]. The fingerprint database stores information, typically the RSS, at certain locations of the environment, modelling the characteristics of the signal using either discrete (fingerprints) or continuous (surfaces) representations. Classification algorithms are the common choice for localization in these kind of systems.
- **Probabilistic:** These methods keep track of the position of the device maintaining a probability distribution over the positions or coordinates of the environment [Ocaña, 2005] [Youssef and Agrawala, 2008] [Fang and Lin, 2010] [Biswas and Veloso, 2010].

Generally, these systems provide localization using a map as reference. Two map representations have been traditionally used: discrete and continuous. On a discrete map representation, the environment is divided into discrete positions and the localization is usually obtained in an estimation stage comparing the measures with a previously stored pattern (fingerprint based methods) [Bahl and Padmanabhan, 2000] [Youssef et al., 2003]. When the discrete positions are selected based on their topological significance it is called a topological representation. Topological representations [Kuipers and Byun, 1988] [Kortenkamp, 1993] discretise the environment using nodes that correspond to a differentiating feature of the environment. These approaches have been especially useful in WiFi-based localization systems where no movement models are available and topological information is more relevant than a metric one (e.g. been at the doorway of office 15 versus being at coordinates x,y,z). On a continuous map representation the environment is considered continuous and the position is usually obtained using a propagation model or updating a probabilistic distribution of the position through action and observation models as in particle filters [Fox et al., 2003] [Hightower and Borriello, 2004]. Continuous maps are more often used in robotics where the propagation and actuation models are known, although some attempts have been made to model the human movement using *Inertial Measurement Units (IMUs)* as described in [Woodman and Harle, 2008].

2.1.1 Deterministic

2.1.1.1 Propagation model based methods

Since the year 2000, different studies have been presented to estimate a WiFi device position using propagation models in indoors [Bahl and Padmanabhan, 2000] [Kotanen et al., 2003] [Bahillo et al., 2009]. Generally, the exact location of the APs in the environment is needed, so usually, a database with the APs location is built in an offline process. But sometimes, the position of the APs is unknown or hard to obtain, so the location of the APs can be estimated using the device's pose and a model of the signal propagation [Sichitiu and Ramadurai, 2004] [Caballero et al., 2008] [Zhang et al., 2011].

Signal propagation models are usually adjusted to calculate the signal propagation path loss. But, due to the multipath effect and shadowing present indoors, path loss models become environment-specific. So, theoretical models are being used to translate the difference between the transmitted and the RSS into a distance estimation looking for the minimum model adjustment requirements.

[Kotanen et al., 2003] proposed the Hata-Okumura model which has become popular in the last decade for WiFi technology signal modelling in indoors. The RSS from an AP is converted to distance d as follows:

$$\log(d) = \frac{1}{10\gamma}\left(P_{TX} - P_{RX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) - X_{\alpha}\right) \tag{2.1}$$

where, d is the estimated distance between the AP and the receiver in metres, P_{TX} is the transmitted power level and P_{RX} is the power level measured at the receiver. G_{TX} and G_{RX} are the antenna gains of the transmitter and the receiver respectively in dBi, λ is the wavelength of the signal in metres and the γ value denotes the influence of walls and other obstacles. Error is also included in the equation since X_{α} is a normal random variable, whose standard deviation equals to α.

The curve in Figure 2.2 represents the tuned propagation model by [Kotanen et al., 2003] and every dot corresponds to the mean of measured RSS. As can be seen, there are differences between the measured values and the model, being the mean absolute error of 1.41 metres.

Hata-Okumura model was revised in [Bose and Foh, 2007] determining the following values for the variables when using the model in indoors. Since WiFi frequency is 2.4 GHz, λ can be estimated to be 12.5 cm. The standard deviation of X_{α} is in the range

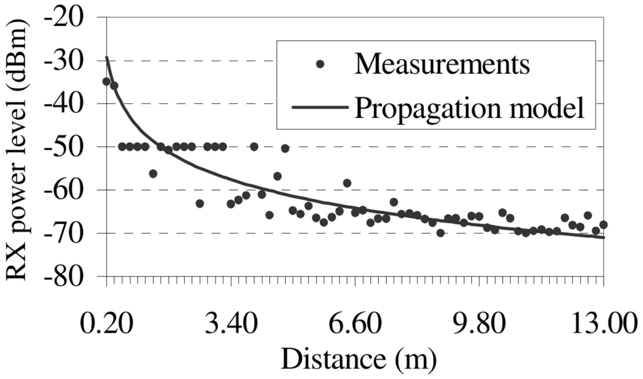


Figure 2.2: Hata-Okumura propagation model tuned by [Kotanen et al., 2003].

of 3 dB up to 20 dB, depending on building construction and the number of obstacles between the AP and the receiver. For free space, γ equals 2, but for obstructed paths in buildings, γ is between 4 and 5.

Although the Hata-Okumura is a well-known model, it is difficult to reproduce the effect of obstructions, especially by walls. [Bahl and Padmanabhan, 2000] proposed the *Wall Attenuation Factor (WAF)* which provides flexibility in accommodating different building layouts while taking into account large-scale path loss. The authors designed the propagation model to have a good trade-off between simplicity and accuracy. The WAF model computes the distance d as follows:

$$P(d) = P(d_o) - 10\gamma \log d \left(\frac{d}{d_0}\right) - \begin{cases} nW * \text{WAF} & , \quad nW < C \\ C * \text{WAF} & , \quad nW \geq C \end{cases} \tag{2.2}$$

where, γ is the path loss exponent, $P(d_o)$ is the signal power in dBm at some reference distance d_o and d is the distance between the wireless device and the AP. C is the maximum number of obstructions (walls) up to which the attenuation factor makes a difference, nW is the number of obstructions (walls) between the transmitter and the receiver, and WAF is the wall attenuation factor. In general the values of γ and WAF depend on the building layout and construction material, and are derived empirically. The value of $P(d_o)$ can either be derived empirically or obtained from the wireless network hardware specifications.

[Yang and Chen, 2009] utilized linear regression to discover the relationship between the RSS and the distance from a wireless device to an AP. This way they are able to adjust the parameters of the propagation model to their environment. The authors chose polynomial regression adjusted by least squares. They used M training points (d_i, RSS_i) where d_i is the distance between the wireless device and an AP and RSS_i is the corresponding signal strength reading at the training point to approximate the polynomial. This approach is more accurate than the previous ones (improving the error rate around 33%) but it is also less general to be used in different environments.

Once the propagation model has been established, the position of the device has to be estimated. Lateration is the most common approach for estimating the position using the distances to multiple APs. Lateration is defined as the method that estimates the position of an object by measuring the distance to multiple reference positions [Hightower and Borriello, 2001]. Calculating the position of an object in two dimensions requires at least the distance measurements to three non-collinear points (Figure 2.3). In three dimensions, at least the distance measurements to four non-coplanar points are required, but the number of required distance measurements can be reduced in some specific applications. For example, applications based on GPS can estimate the position of a device by using only three measurements from satellites (Figure 2.4), since it can be assumed that one of the solutions is almost impossible.

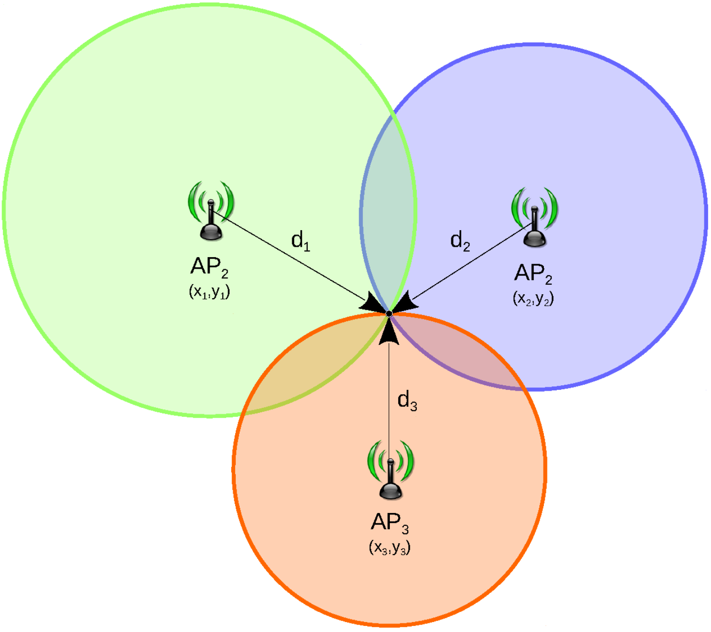


Figure 2.3: 2D Lateration, where d_1, d_2 and d_3 denote the distance to APs 1, 2 and 3 respectively. The crossing point of the circumferences is the estimated position.

[Bahl and Padmanabhan, 2000] proposed RADAR, an RF-based system for locating and tracking users inside buildings using a basic tri-lateration algorithm. It used three APs and an empirical propagation model to estimate a mobile position. The RADAR proposal obtained an accuracy of 4.3 metres for the 50^{th} percentile.

Similarly, [Bose and Foh, 2007] used a lateration algorithm to estimate the location of a mobile device in a two dimensional plane. The system obtained an average error of 2.9 metres for a NLOS environment. [Mazuelas et al., 2009] also applied a standard lateration technique with an optimized propagation model to estimate a robot position. It achieved a mean error of 4.1 metres for the 50^{th} percentile.

An optimization approach has also been studied to solve the lateration problem. Non-Linear and Linear Least Squares methods have been proposed in [Yang and Chen, 2009] to estimate a mobile location. Both methods minimize the sum of the square error of

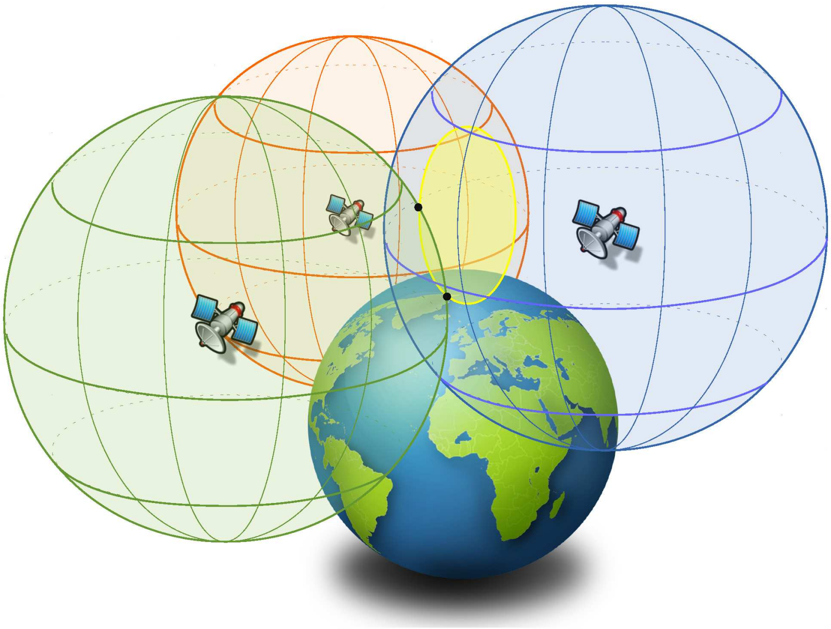


Figure 2.4: 3D Lateration. The crossing points of the spheres denote the possible positions, since one of them is not located on earth it can be discarded.

Equation 2.3 in order to obtain an estimation $(\hat{x}, \hat{y})$.

$$(\hat{x}, \hat{y}) = \arg\min_{x,y} \sum_{i=1}^{N} \left(\sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i \right)^2 \tag{2.3}$$

where, d_i represents the range distances, (x_i, y_i) is the known position of the i^{th} AP, N is the number of APs, and (x, y) is the position of the mobile device to be estimated. Both algorithms were validated in a real indoor environment using a WiFi network, improving the accuracy of standard lateration methods. The median error obtained by Linear Least Squares was 3.66 metres in contrast to the 3.05 metres error obtained by Non-Linear Least Squares methods (50^{th} percentile).

Despite these methods are being applied to indoor environments they still have to face some challenges. The main challenge of these systems is the difficulty to formulate a reliable radio propagation model due to the fact localization is not carried out in static indoor environments and the signal usually goes through obstacles that are not known a priori [Ocaña et al., 2005]. Typical obstacles are opened and closed doors, windows, pieces of furniture, people, etc. Among these obstacles, the presence of people can be considered one of the most significant ones, since human body is mainly made up of water, which absorbs part of the signal, and it may dim significantly the RSS as exposed in [Bahillo et al., 2009].

The localization systems described in this section have been tested in indoor environments with a low number and a very uniform distribution of APs over the environment, which suggests that they were deployed for localization purposes. This may have affected the localization results, improving the results in comparison with a real environment where the APs are deployed for communication.

2.1.1.2 Fingerprint based methods

Fingerprints have been used to have a spatial representation of the signal strength readings from the surrounding APs. These systems rely on a training phase where the RSS at the target areas is measured and stored. This way, the characteristics of the WiFi signal at different areas of the environment are captured and the complex adjustment of the signal propagation model is avoided. However, such data collection requires significant human labour and the fingerprint databases have to be stored.

There are two stages for fingerprint based location systems: an offline stage denoted as “training” stage and an online stage which is called “localization” stage.

Training stage

During this stage, a fingerprint database is built. Its construction begins by dividing the environment in cells with the help of a floor plan (Figure 2.5). These cells can either be uniformly distributed over the environment, or arranged covering interest areas in a topological approach.

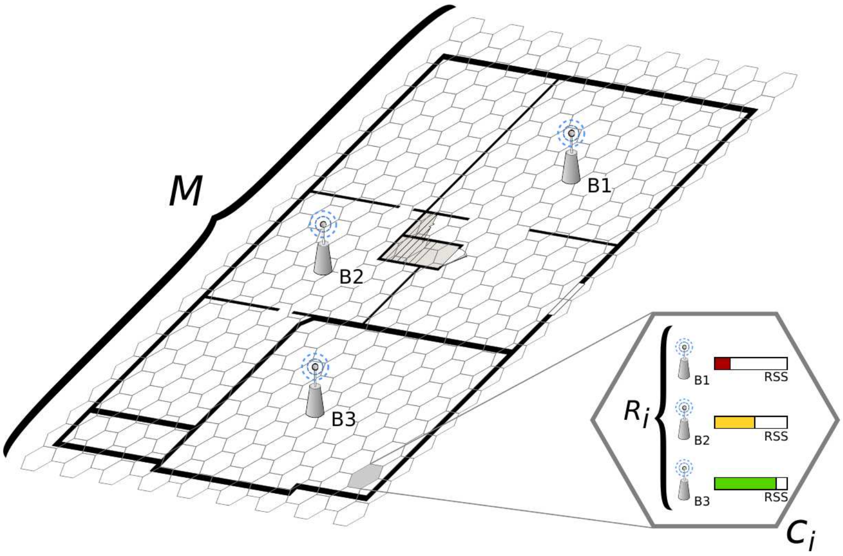


Figure 2.5: Environment division for fingerprint based systems.

Then, a site survey is performed and the RSS from the APs is collected at each cell for a certain period of time and stored into a fingerprint database. The i^{th} element in the fingerprint database has the form:

$$M_i = \Bigg(\underbrace{\Big\{ R\vec{S}S_{ij} \mid \forall AP_j \in APs \Big\}, \theta_i}_{R_i \in R}, C_i \Bigg), \qquad i = 1, \ldots, M \tag{2.4}$$

where C_i is the position of the i^{th} cell. Vector $R\vec{S}S_{ij}$ holds the RSS values measured from the AP_j. The parameter θ_i contains any other information needed in the localization stage. This can be for example the orientation $\theta_i \in \{north, south, east, west\}$ of the device, such as in the RADAR system [Bahl and Padmanabhan, 2000]. The i^{th} fingerprint is denoted by R_i and the set of all fingerprints by $R = R_1, \ldots, R_M$.

The fingerprint database can be modified or preprocessed before the localization stage. The motivation can be the reduction of the memory requirements to store the fingerprint database, the reduction of the computational cost of the localization stage or the accuracy improvement in the localization stage. In addition, different localization methods use different characteristics.

Most of the systems collect statistical values such as the mean, the standard deviation, and the median of the corresponding signal strength values [Bahl and Padmanabhan, 2000] [Alonso et al., 2009]. Other systems use histograms [Ladd et al., 2005] or Gaussian models [Haeberlen et al., 2004]. Some systems also collect data at different times to generalize the training stage [Prasithsangaree et al., 2002].

In [Mengual et al., 2010] the authors modelled the signal strength by means of a calibration stage to overcome the relative effect of doors and walls. The calibration stage was divided into three stages. First, the RSS measurements were normalized to identify the relative effect of walls and obstacles. Second, neural networks computed the normalized values to group the measurements in clusters. Finally, the physical topology was used to optimize the clusters.

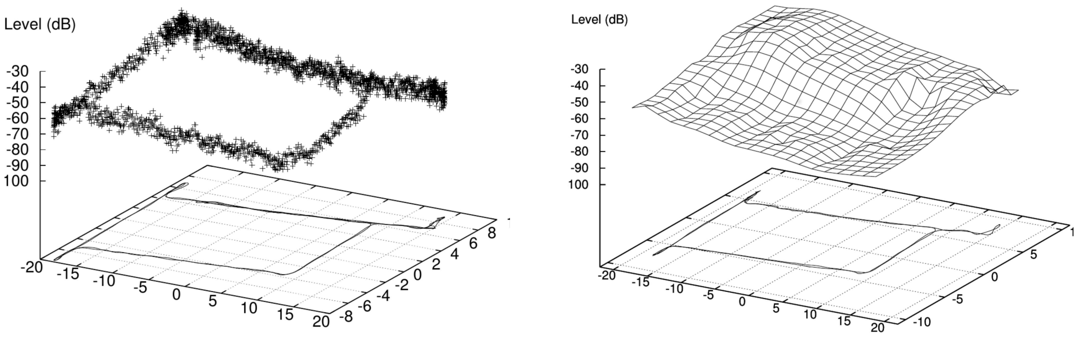
[Ferris et al., 2006] approached the modelling problem using *Gaussian Processes (GPs)*. The key idea underlying GPs is the requirement that the function values at different points are correlated. This dependency can be specified via an arbitrary covariance function, or kernel $k(\mathbf{x}_p, \mathbf{x}_q)$ (Equation 2.5).

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2l^2} |\mathbf{x}_p - \mathbf{x}_q|^2 \right) \tag{2.5}$$

where $\mathbf{x}_p$ and $\mathbf{x}_q$ are the input values, σ_f^2 is the signal variance and l is the length scale that determines how strongly the correlation between points drops off. Both parameters (σ_f^2 and l) control the smoothness of the functions estimated by the GP. The posterior GP was estimated from a calibration trace of signal strength measurements annotated with their locations. Assuming independence between different APs, a GP was estimated for each AP separately.

[Fink and Kumar, 2010] continued the idea of GPs and used online methods to improve the quality of the models as the environment was explored. Also, other approaches such as graph representation [Biswas and Veloso, 2010] or neural networks [Paul and Wan, 2009] have been proposed. In [Biswas and Veloso, 2010] the authors modelled the world as a WiFi signature map with geometric constraints and introduced a continuous perceptual model of the environment generated from the discrete graph-based WiFi signal strength sampling. [Paul and Wan, 2009] proposed to use a radial basis function neural network to fit nonlinear maps between known calibration locations and RSS measurements.

Besides the previously explained representations, some studies focused on continuous representations are being used for robotic applications. Usually, continuous fingerprint surfaces are generated through experimental sampling using interpolation methods for regions without data samples (Figure 2.6). These surfaces can be used in different ways to infer the mobile position.



(a) RSS recorded by a robot during a trajectory over the environment [Howard et al., 2006].

(b) Interpolated RSS surface using the data set shown in (a) [Howard et al., 2006].

Figure 2.6: RSS interpolation to create fingerprint surfaces.

[Howard et al., 2006] proposed a solution based on a simple interpolation kernel. An exhaustive study of how to generate fingerprint surfaces was presented in [Zàruba et al., 2007], which modelled the signal propagation like a map of the expected RSS measurements in the environment. The surface was built in several steps. First, a scaled floor-plan with all the walls, doors, and windows (and other major obstacles) of the environment was entered. The number of cells and their positions were defined and added to the floor-plan, and signal-strength measurements were taken at the same physical locations. A parametric ray-tracing algorithm [Hassan-Ali and Pahlavan, 1998] was used to provide a description of signal strengths at the measurement points (e.g., how many different obstacles do radio wave rays pass through and/or reflect off until they reach the measurement points). This parametric representation of the signal at the measurement points was approximated using the real measurement values. Finally, the surface was recalculated by using ray-tracing, but this time with the inserted transmission and reflection properties of the obstacles obtained in the previous step.

Localization stage

Given the fingerprint database, the objective of the localization stage is to compare the measures obtained online with the stored ones to infer the location of the device

(Figure 2.7).

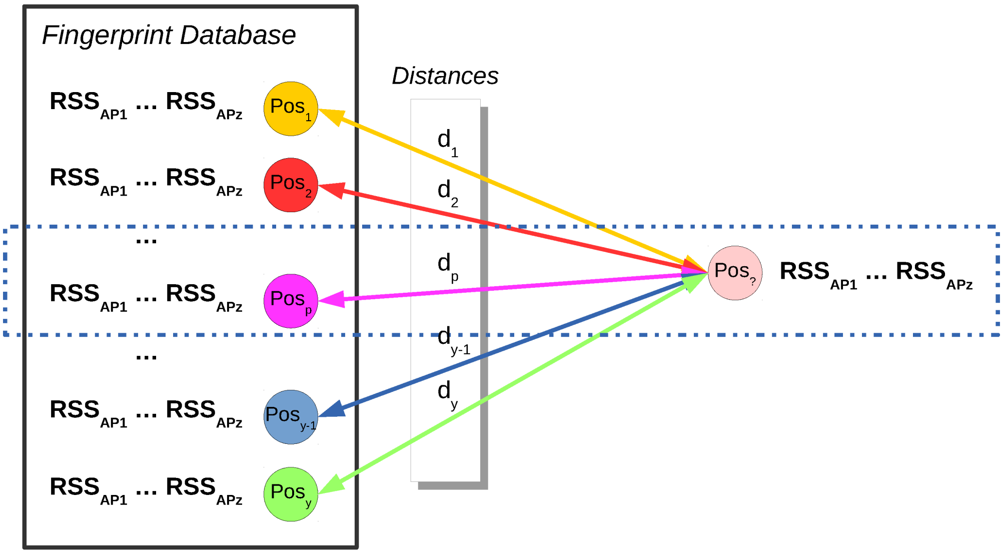


Figure 2.7: Localization stage for fingerprint based systems.

In the last years, machine learning research has greatly developed, and its advances have been applied to the localization problem.

[Liu et al., 2007] refers as classifier-based methods to the algorithms that first use a training stage to collect fingerprints of an environment and then estimate the location of an object by matching online measurements with the closest fingerprint location.

[Bahl and Padmanabhan, 2000] is one of the first remarkable work in this area. The authors proposed RADAR, an in-building user location and tracking system which adopts a *K-Nearest Neighbour (KNN)* approach. The error of the RADAR system was 2.94 metres for the 50^{th} percentile and around 9 metres for the 90^{th} percentile. In their following work [Bahl et al., 2000], RADAR was enhanced adding a tracking stage by a Viterbi-like algorithm. It got an error of 3.16 metres for the 90^{th} percentile.

[Fang and Lin, 2008] presented a more complex machine learning approach using discriminant adaptive neural networks, which took the RSS from the APs as inputs to infer the client position. The nonlinear relationship between RSS and the position is modelled during the network learning phase. Useful information was extracted into discriminative components. Thus, the nonlinear relationship between RSS and the position was accurately constructed by incrementally inserting the discriminative components and recursively updating the weightings in the network until no further improvement was required. The authors compared this approach with other machine learning methods obtaining an improvement of 21.07% for the mean error and 19.89% for the standard deviation with an error of 4.91 metres for the 90^{th} percentile.

Along the neural networks line, [Mengual et al., 2010] also proposed a solution based on *Self-Organizing Maps (SOMs)* neural networks. The authors used the RSS measurements and the information about the regions (cells) clustered by n SOMs to estimate a mobile position. The mobile position was figured out based on three criteria: Exact location estimation, when the n SOMs output agreed about the same position; region estimation, when the n SOMs output a region which is possibly different for each SOM, what is settled by a voting algorithm; optimized region estimation according to the physical distribution of the space, as above, but only the regions that have points that are close together in the physical space are taken into account before the result is estimated. By using these criteria the system was able to obtain a classification rate of 74% which ups to 85% for 80% of the cases.

Other work from the field of machine learning proposed the use of *Support Vector Machine (SVM)*. [Brunato and Battiti, 2005] proposal was based in the main idea of the SVM which is to map the input vectors into a feature space with a higher number of dimensions, and to find an optimal separating hyperplane in the feature space. The SVM algorithm displayed an error rate of 5.12 metres for the 90^{th} percentile. Moreover, SVM presented a low algorithmic complexity in the normal operating phase with respect to other algorithms. [Figuera et al., 2012] studied how including a priori information in the learning machine algorithm can enhance the performance of the location system. The authors proposed three advanced SVM-based algorithms which include a priori information obtaining a mean error of 3 metres.

Decision tree algorithms are also being used with fingerprint databases. The basic idea involved in decision tree algorithms is to break up a complex decision into the union of several simpler decisions, hoping that the final solution would resemble the desired one. [Yim, 2008] proposed a decision tree method which was built during the training phase. The study also compared the complexity of decision trees and other classifier methods such as KNN and neural networks concluding that decision trees are a much more efficient solution than the other ones in terms of complexity.

Despite the fact that fingerprint methods are able to solve the indoor localization problem, most of these systems do not manage well enough the noise that affects wireless signals in indoors. Fuzzy logic [Zadeh, 1965] [Zadeh, 1973] is used in a number of studies [Astrain et al., 2006] [Alonso et al., 2009] [Parwekar and Reddy, 2013] to deal with this uncertainty, especially with the small scale effect [Youssef and Agrawala, 2003]. These systems take advantage of the robustness of fuzzy logic to infer a mobile position without a high number of samples. The feasibility of fuzzy logic in real-scenarios was demonstrated by obtaining a classification rate close to 90%. Moreover, the results showed that fuzzy logic-based system can use a lower number of samples to estimate a mobile position [Alonso et al., 2011].

[Dharne et al., 2006] proposed a *Fuzzy Rule-Based Classifier (FRBC)* able to get good results while reducing the computation time thanks to the use of a grid-based map describing the environment under consideration. Moreover, they reduced the computational cost by taking into account only significant grid-points. [Chen et al., 2008] proposed the use of an adaptive FRBC which updated a manually created set of rules using online measures. Rules were created using the relations of the RSS from four different APs achieving a classification rate of 59%.

More interesting is the system developed in [García-Valverde et al., 2012]. They also created an adaptive FRBC in a real environment with up to 90 APs with no prior knowledge about their locations. They proposed an incremental online learning method to extend the rule base to adapt the system to new environment conditions. They obtain a classification rate of 76.40% using training and test data collected on the same day without using the adaptive method, getting down to 10% when using data from different days to train and test the system. Using the online learning method the system was able to maintain the accuracy around 77%.

Similarly to the presented propagation model based localization systems, the localization systems described in this section, except the system developed in [García-Valverde et al., 2012], were carried out in indoor environments with a low number and a very uniform distribution of APs over the environment, which suggests that they were deployed for localization purposes. This may have affected the localization results, improving the results in comparison with a real environment where the APs are deployed for communication.

2.1.2 Probabilistic methods

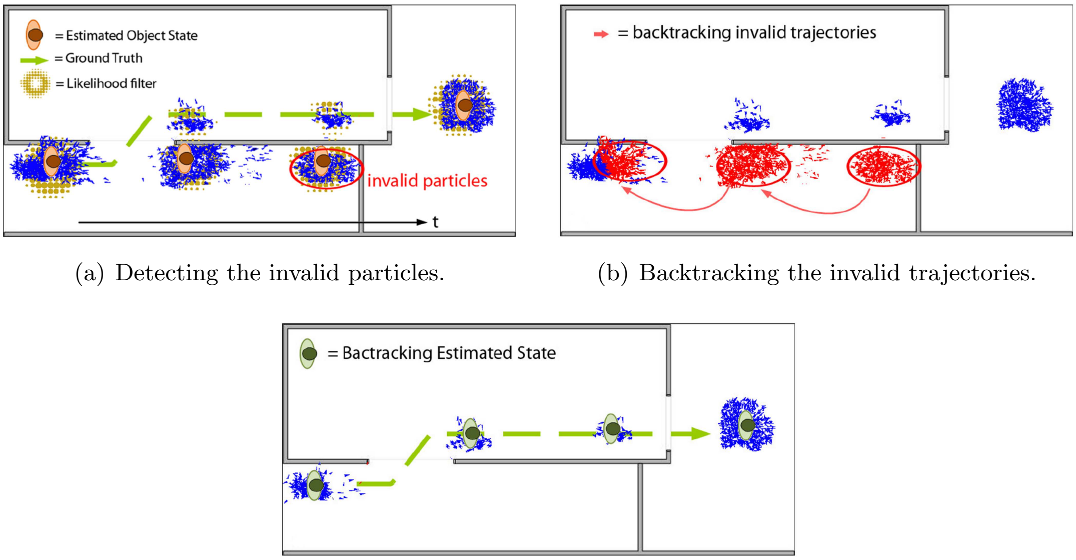
Probabilistic methods have been used in a vast number of applications in order to track the dynamic system's state from observable and noisy measurements [Fox et al., 2003]. These techniques maintain a probability distribution that captures the knowledge about the state of the system at a given instant of time. The distribution changes over time, following the transition model of the system and is updated with each observation by means of a probabilistic sensor model. Standard filtering techniques, such as Kalman or Particle filters have been used to solve the localization problem with WiFi technology. Typically, these techniques require to have an observation model of the signal such as propagation models or fingerprint databases.

The Horus system [Youssef and Agrawala, 2008] proposed a joint clustering technique for location estimation, which used a probabilistic distribution to model the noise of WiFi technology. Each candidate location coordinate was regarded as a class or category. In order to minimize the distance error, a location was chosen while its likelihood was the highest. The experiment results showed that this technique acquired an accuracy of more than 90% within 2.1 metres. The authors suggested that increasing the number of samples at each sampling location could improve the system accuracy.

[Yim et al., 2008] presented the design and implementation of a WLAN-based *Extended Kalman Filter (EKF)* positioning method. It used a RF propagation loss model to estimate the distances between the mobile and the APs. The EKF performance was compared with trilateration and fingerprinting methods. The experimental results showed that the proposed EKF positioning method improved the results of the trilateration method without the filtering (3.52 metres versus 4.1 metres mean error). However, using a simple KNN (1-NN) fingerprinting algorithm they obtained more accurate results (2.4 metres mean error) than their proposed EKF method. Along the same line, [Wu et al., 2007] used an EKF in combination with neural networks to recursively estimate the position of a mobile. The authors chose the EKF because it can blend the information optimally minimizing the variance of the estimation error. They showed that the EKF provided a solution with an error of 2 metres in a fully controlled environment. In [Fang and Lin, 2010], the authors exploited the information about the system state with an EKF which used a fingerprint representation of the environment. The EKF obtained a mean error of 3.5 metres for the 75^{th} percentile.

Other authors proposed solutions based on *Particle Filters (PFs).* PFs can cope with arbitrary distributions enabling global localization, or can maintain multi-modal distributions to deal with ambiguities. Moreover, the probabilistic observation model of WiFi sensors is strongly nonlinear and leads to distributions that could be barely approximated by using Gaussians only.

[Zàruba et al., 2007] proposed the use of PFs which used a fingerprint database representation of the expected RSS measurements in the environment. The authors chose PFs techniques to efficiently estimate the multi-modal distribution of the mobile's position. Simulated experiments showed that the filters estimate the mobile's location obtaining an average precision around 2.1 metres. [Widyawan et al., 2008] proposed a variant of PFs, the so-called *Backtracking Particle Filter (BPF).* BPF is a technique for refining state estimates based on exclusion of invalid particle trajectories (Figure 2.8). Categorization of invalid trajectory determined during importance sampling step of the PF. BPF obtained an enhancement in the mean error (1.34 metres) up to 25% compare to PF only (1.82 metres) in a simulated experimentation.



(a) Detecting the invalid particles.

(b) Backtracking the invalid trajectories.

(c) Backtracking the estimated states.

Figure 2.8: Backtracking Particle Filter used in [Widyawan et al., 2008].

In [Biswas and Veloso, 2010], the authors also considered a PF-based localization approach where a WiFi fingerprint database along with a robot's odometry is used to obtain the robot's location. The system got an error of 2.5 metres for the 90^{th} percentile. Finally, in [Ferris et al., 2006] the authors incorporated the likelihood model into a PF operating in indoor environments using a signal representation based on GPs. The average error was 2.12 metres obtained over a 3 km trajectory.

2.2 Discussion

Previous sections have introduced a number of published methods for indoor localization using WiFi technology. This section presents a discussion and a comparison of the most important work for WiFi indoor localization.

WiFi localization research arose with the introduction of the RADAR system [Bahl and Padmanabhan, 2000] which provides good performance and robustness. The *RADAR lateration* method allows a fast implementation without the need of arduous training stages. The *RADAR fingerprint* approach provides a solution for any problem that requires to determine the position of a mobile in medium-size indoor environments. Both solutions were an inspiring start point for many other researches and, to date, it is still being used as a baseline to compare new systems.

Localization with WiFi technology is still a very open problem. In order to show a general view of the performance of these methods, Table 2.1 shows a comparison of some of the algorithms referred above in terms of the used positioning algorithm and the achieved results. Error values are given by authors in several different forms, usually as the mean error or the maximum error for a data percentile. Some fingerprint systems that estimate the position using a discrete number of cells provide their results as the percentage of the data correctly classified into the corresponding cell.

Since the experimental evaluation of these systems has been performed using datasets specifically captured for each of them, results are not directly comparable. Most of them compare their solutions with “standard” algorithms published in the literature, which limits the validity of a real comparison because results in WiFi localization systems are very environment dependent. In addition, many authors do not provide numeric values for the size of their environments, and only present pictures or graphs of them. To be able to get a general view of the achieved results by the different systems, the environment size is approximately indicated in the “Environment size” column of the table, and the cell size (important in the fingerprint systems) is shown in the “Cell size” column of the table.

Several conclusions can be drawn from the analysis carried out in previous sections and the information in Table 2.1:

- Propagation model based methods are the most precise systems if the propagation model is well adjusted for the environment. Unfortunately, this is hard to achieve in indoors, being the main reason of the low acceptance of this kind of systems for WiFi indoor localization. Theoretically, one propagation model could be used for localization in different environments; this is true when working outdoors but in indoors it is usually necessary to do some site-survey to adjust the model, loosing their main advantage.

  The error achieved using these systems is around 4 metres for the 50^{th} percentile, being reduced to 3 metres for the 50^{th} percentile if an environment-adjusted propagation model is used.

Reference	Positioning Algorithm	Environment Size	Cell size	Results
Propagation model based methods				
[Bahl and Padmanabhan, 2000]	Trilateration: WAF	$979m^2$	*Continuous*	$4.3m$ (50^{th} percentile)
[Bose and Foh, 2007]	Lateration: Hata-Okumura	$\simeq 100m^2$	*Continuous*	$2.9m$ (mean error)
[Mazuelas et al., 2009]	Lateration: Optimization	$\simeq 2500m^2$	*Continuous*	$4.1m$ (50^{th} percentile)
[Yang and Chen, 2009]	Lateration: Non-linear opt.	$3438m^2$	*Continuous*	$3.05m$ (50^{th} percentile)
Fingerprint based methods				
[Bahl and Padmanabhan, 2000]	KNN	$979m^2$	$0.5m - 5m$	$2.94m$ (50^{th} percentile)
[Bahl et al., 2000]	KNN+Viterbi algorithm	$979m^2$	$1.75m - 3.5m$	$3.16m$ (90^{th} percentile)
[Fang and Lin, 2008]	Neural Networks	$690m^2$	$1m - 2m$	$4.91m$ (90^{th} percentile)
[Mengual et al., 2010]	Neural Networks	$\simeq 350m^2$	$2m - 5.5m$	74% (classification rate)
[Brunato and Battiti, 2005]	SVM	$750m^2$	$1m - 1.5m$	$5.12m$ (90^{th} percentile)
[Figuera et al., 2012]	SVM	$559m^2$	$1m$	$3m$ (mean error)
[Yim, 2008]	Decision Tree	$\simeq 80m^2$	$\simeq 1m$	$2.3m - 3.9m$ (mean error)
[Chen et al., 2008]	FRBC	$1762m^2$	Room size	59% (classification rate)
[García-Valverde et al., 2012]	FRBC	$\simeq 50m^2$	Room size	77.22% (classification rate)
Probabilistic methods				
[Youssef and Agrawala, 2008]	Fingerprints: Bayesian	$1766m^2/424m^2$	$\simeq 1.5m/2.1m$	$0.86m/1.32m$ (90^{th} percentile)
[Yim et al., 2008]	Prop. Model: EKF	$\simeq 2100m^2$	*Continuous*	$3.5m$ (mean error)
[Wu et al., 2007]	Prop. Model: EKF	$109m^2$	*Continuous*	$2m$ (mean error)
[Fang and Lin, 2010]	Fingerprints: EKF	$690m^2$	$1m$	$3.5m$ (75^{th} percentile)
[Biswas and Veloso, 2010]	Fingerprint surface: PF + odometry	$\simeq 3000m^2$	*Continuous*	$2.5m$ (90^{th} percentile)
[Ferris et al., 2006]	Fingerprint surface: PF	$3000 * 3$ floors m^2	*Continuous*	$2.1m$ (mean error)

Table 2.1: A comparison of WiFi indoor localization methods.

- Fingerprint based systems are only valid for the environment where they were trained, but with the necessary training stage they can be used in environments with any characteristics.

  The main disadvantage using this kind of systems is the necessity of site-survey for the fingerprint database construction, especially if the required resolution is high (which means more cells to site-survey). But recently, some systems are arising for automate or at least simplify the site-survey task [Chintalapudi et al., 2010] [Wu et al., 2012] [Wang et al., 2012] [Rai et al., 2012] [Yang et al., 2012] allowing to develop this kind of system reducing the effort required during the training stage.

  Among the different algorithms used in fingerprint based systems the classic KNN remains as a good choice while SVM and FRBCs are arising as the ones providing the best results.

  The error of fingerprint based systems in indoors is lower than the error obtained using propagation models, reaching an error around 5 metres for the 90^{th} percentile.

- Probabilistic methods improve the performance of the WiFi localization systems, getting a mean error from 2 to 3 metres. Using the WiFi RSS along with the information provided from other sensors reduces the error in the localization. It does not seem to be one probabilistic method overcoming the others, so the selection of one of them depends on the restrictions of the system, the available information or the results achieved in an specific environment.

2.3 Objectives

After the review of the state the art, and considering the discussion presented in the introduction, the objectives of this thesis are as follows:

1. To study the WiFi signal behaviour in indoor environments. Finding the main challenges to face when using WiFi RSS to develop a localization system.

2. To record sufficiently representative datasets in different real indoor environments with different characteristics. To date, there are no available datasets of WiFi RSS recorded in real indoor environments. These datasets will be used to test the methods proposed in this work, and made available to the public. Compare the designed system with other systems presented in the literature.

3. To develop a WiFi indoor topology-based localization system taking into account the previous conclusions. It must comply with the restrictions of a production system (work with different devices, real-time execution, robustness to signal interferences, tested in real environments with several floors) which are not dealt with in most of the systems in the state of the art. Assess the performance of the proposed system with different configurations.

4. To improve the developed system using probabilistic based methods to locate a device in motion. It must comply with the previously described restrictions. Perform

experiments during different trajectories to get an idea of the maximum expectable accuracy.

5. To develop a WiFi localization application to perform user-friendly localization available for different devices (smartphones, tablets, laptops, etc.).

Chapter 3

WiFi Signal Analysis

WiFi was not originally intended to be used as a localization technology, so some intrinsic performance limitations appear when it is used for this purpose. Most of the commercial devices equipped with WiFi technology use 802.11b/g standards which work at 2.4 GHz. This is a free frequency, where some other devices such as bluetooth and microwave ovens work, making the WiFi RSS a noisy signal. Although in outdoor environments the WiFi RSS decreases with the distance to the emitter [Rappaport, 1996], when working indoors the RSS is strongly dependent on the building structure and some other non-desired effects appear. Most of these effects are due to the multipath effect [Elnahrawy et al., 2004]. Another important issue is the absorption of part of the signal by people moving around in the environment, which significantly diminishes RSS [Bahillo et al., 2009].

In addition, the small scale variations cause the RSS vary when the WiFi device moves distances in the range of the wavelength ($\lambda = 12.5cm$). This effect makes very difficult to estimate the correct location because small variations in the position can lead to high variations in the RSS.

Moreover, since WiFi networks are deployed with the goal of maximizing connectivity and disregarding localization tasks, there is usually a high number of APs distributed over the environment increasing the so-called co-channel interferences, which cause high variations in the RSS from the APs.

Some interesting conclusions regarding the WiFi signal behaviour have been pointed out in previous researches: In [Bahl and Padmanabhan, 2000], the authors determined that the device orientation could cause a variation up to 5 dB. The authors of [Kaemarungsi and Krishnamurthy, 2004] found that the RSS is noisier when people are present during the measurement.

In Section 3.1, the most important characteristics of WiFi technology will be reviewed. Next, Sections 3.2 to 3.5 will show a deeper analysis of the most important effects to take into consideration when designing a WiFi localization system. The conclusions extracted from this analysis will be taken into account to develop the system with the necessary knowledge about WiFi signal behaviour.

3.1 WiFi technology

The WiFi Alliance defines WiFi as “any *Wireless Local Area Network (WLAN)* product that is based on the *Institute of Electrical and Electronics Engineers (IEEE)* 802.11 standards”. IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) for implementing WLANs in the 2.4, 3.6, 5 and 60 GHz frequency bands. They were created and maintained by the IEEE 802 Standards Committee. While each amendment is officially revoked when it is incorporated in the latest version of the standard, the commercial world tends to maintain them because they concisely denote the capabilities of their products.

There are multiple versions of the IEEE 802.11 standard, being the following ones the most important among them:

- 802.11b: It offered transmission rates of 11 Mbps at 2.4 GHz and it was ratified in 1999.
- 802.11g: This version also works on the 2.4 GHz band, increasing the maximum transmission rates to 54 Mbps. The 802.11g amendment was ratified by 2003.
- 802.11n: It operated on both the 2.4 GHz and 5 GHz bands being the support for the 5 GHz band optional. Since most of the devices on the market were already using 2.4 GHz, some of the 802.11n APs did not include the hardware needed to work on 5 GHz. It provides maximum data rates of 600 Mbps and it appeared in 2009.
- 802.11ac: This version works on the 5 GHz band at a maximum data rate of 1 Gbps. It was ratified in January 2014.

The most extended are the 802.11b/g versions, working at the 2.4 GHz frequency band (Figure 3.1). This is the *Industrial, Scientific and Medical (ISM)* international band, which is a free band available worldwide. However, because of this choice of frequency band, 802.11b/g equipment may occasionally suffer interferences from other devices working at the same frequency, such as Bluetooth, microwave ovens and cordless telephones.

Inside the 2.4 GHz ISM frequency band, there are multiple channels utilizing the 2.4 - 2.5 GHz spectrum. The spectrum is sub-divided into 14 channels of 22 MHz each spaced 5 MHz apart, so adjacent channels overlap (Figure 3.2).

Channel 14 is only allowed in Japan, while channels 12 and 13 are allowed in most parts of the world, except the USA, where only Channels 1 to 11 are legal to use. So, there are only three non-overlapped channels allowed worldwide: 1, 6 and 11.

APs coverage range is usually from 25 to 140 metres for 802.11b/g standards, but it depends on the hardware and the environment where the AP is located.

Finally, RSS values collected by the WiFi cards are discretised in integral steps of 1 dB, usually ranging from -10 dBm to -100 dBm.

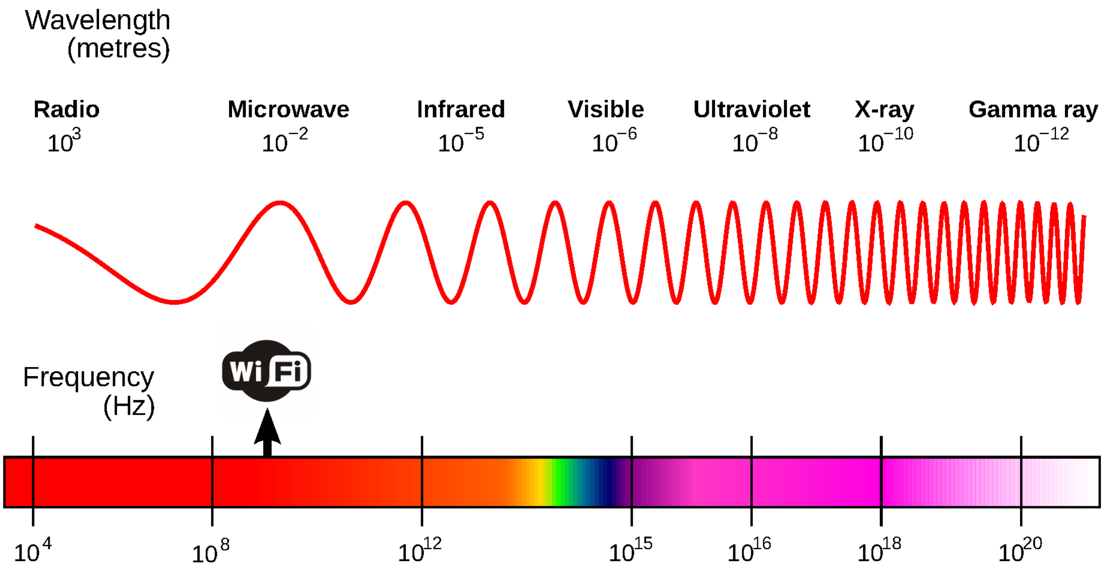


Figure 3.1: WiFi technology in the frequency spectrum.

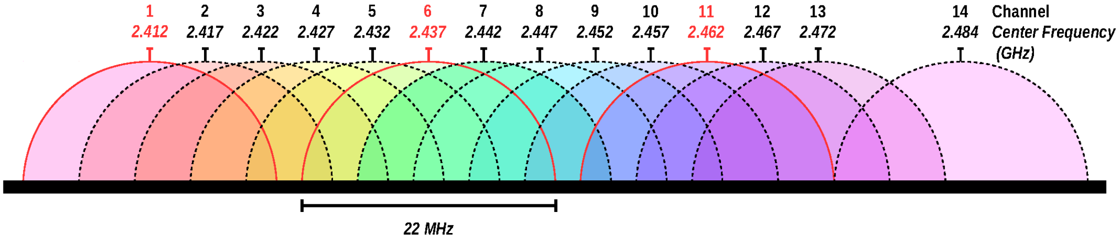


Figure 3.2: WiFi channels in the 2.4 GHz band.

3.2 Co-Channel interferences

Co-channel interferences are crosstalk from two different APs using the same frequency (working at overlapped channels). To study how the co-channel interferences affect the signal, the RSS from an AP emitting in channel 6 was measured during 4 hours, while an AP emitting in channel 7 was on in the surroundings. The second AP was configured to be turned off after two hours.

Figure 3.3 shows how the RSS from an AP is altered when another AP, working in an overlapped channel, is turned off (sample 7100). As can be seen, the RSS kept stable, around -61 dBm ± 2 dB, during the first two hours. After the second AP was turned off, the RSS fell around 10 dB, keeping stable again around -71 dBm ± 2 dB. This variation in the RSS can increase error in localization since changes in one AP can lead to changes in the near ones.

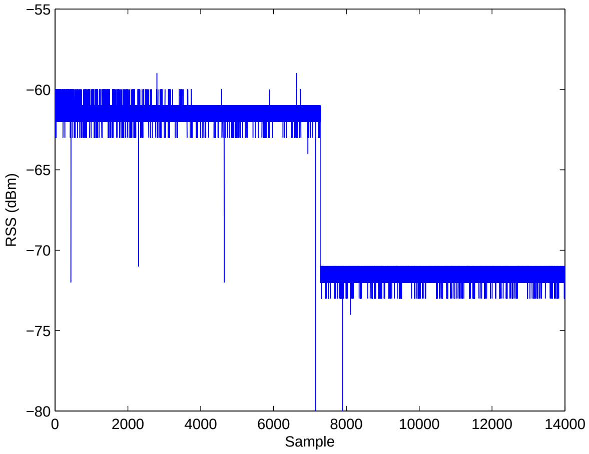


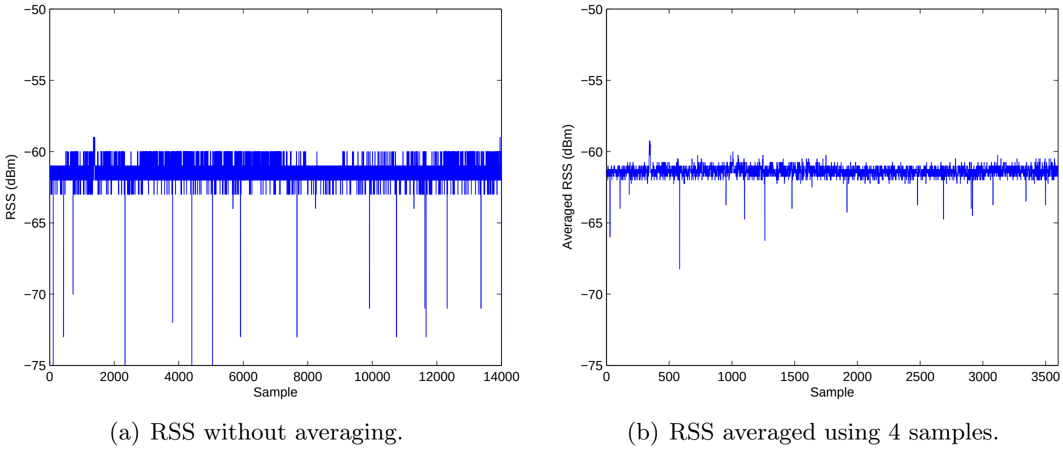
Figure 3.3: Co-channel interferences.

3.3 Temporal variations

Temporal variations are those which are produced when the measurement interface stands at a fixed position, being the time the only variable that changes between the different samples.

These variations are usually caused by environment changes, such as people walking around, opening or closing doors, etc. To obtain a reliable localization system, these variations must be minimal.

To study the effect of temporal variations, the RSS from different APs were measured during 24 hours at different positions. Figure 3.4(a) depicts the RSS from an AP at one of the positions. Figure 3.4(b) shows the same measures, but averaged using 4 samples. As can be seen, the RSS variation is ± 2 dB and about 1 dB when the measures are averaged. This effect is similar for the RSS from all the APs at all the positions. This shows that the RSS at a static position is stable enough to assure the feasibility of localization systems development using WiFi. To support this statement the spatial stability, introduced in [Ocaña, 2005], is studied.



(a) RSS without averaging.

(b) RSS averaged using 4 samples.

Figure 3.4: Temporal variations: RSS received from an AP at a fixed position.

The spatial stability is defined as the RSS stability for an AP at each position of the environment for measurements collected at different times. To analyse the spatial stability, 300 samples were measured at 9 positions (reference data), obtaining another set of measurements at the exact same positions on a different day (test data). Both sets of measurements were averaged for each position and dataset. These measurements have been collected avoiding the presence of any people or RF devices to avoid undesired effects.

Figure 3.5 shows the RSS from an AP for both datasets. As can be seen, the maximum deviation between them is 0.5 dB at position 5, suggesting that the RSS tends to be stable for long periods of time, as long as other undesired effects do not affect the RSS.

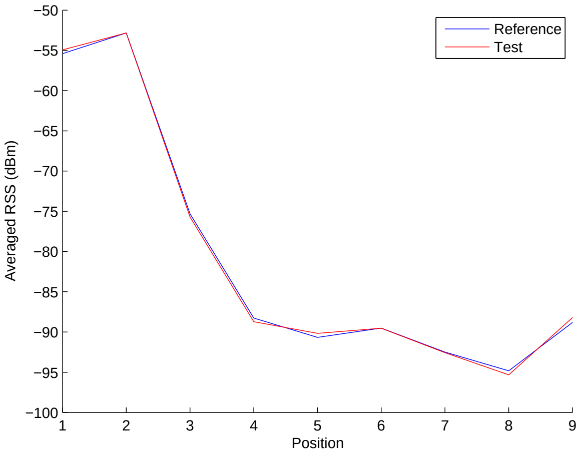


Figure 3.5: Spatial stability for two datasets at the exact same positions on different days.

3.4 Large scale variations

The large scale variations causes the RSS decrease with the distance to an AP due to attenuation of the signal [Rappaport, 1996]. These variations are desirable as they lead to differentiation between the locations of the environment. This has been used for localization purposes especially in outdoor environments.

To analyse this effect, the RSS from 4 APs was measured using a robot moving along a corridor (Figure 3.6). The frequency of acquisition was 4Hz and the robot speed 0,2 m/s, this way a new sample from all the four APs was obtained every 5 centimetres.

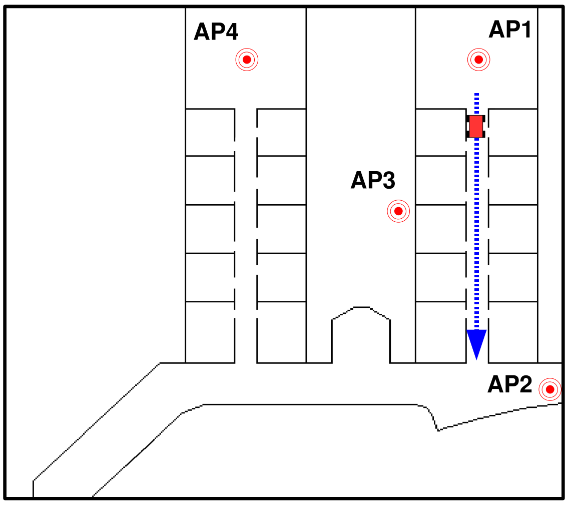


Figure 3.6: Large scale variations: Experimental set-up.

Figures 3.7 and 3.8 show the variation of the RSS from the 4 APs. As can be seen, the RSS varies with the distance to the AP. This way, as the device gets closer to the position of the AP the RSS increases (a variation of 20 dB corresponds to distances around 6-10 metres). But, it can also be seen that the noise in the RSS is very high, making very difficult to obtain a propagation model to calculate the distance to an AP using the RSS.

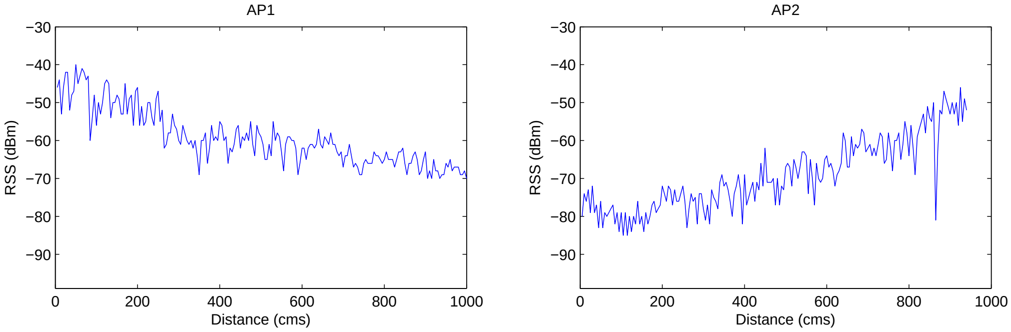


Figure 3.7: Large scale variations: RSS over distance from AP1 and AP2.

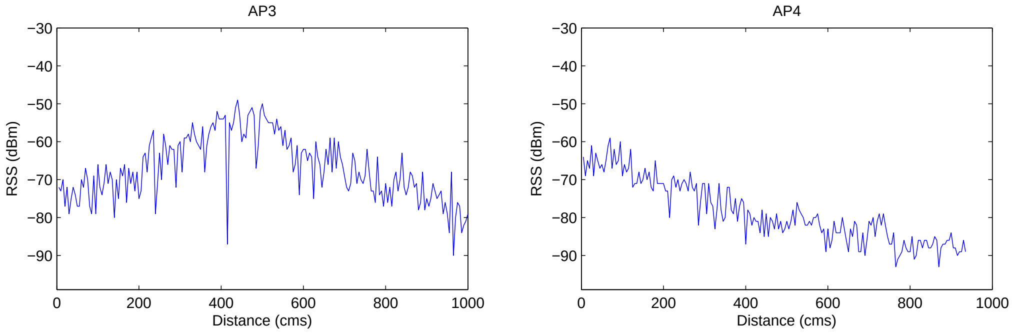


Figure 3.8: Large scale variations: RSS over distance from AP3 and AP4.

3.5 Small scale variations

Small scale variations cause the RSS vary when the WiFi device moves distances in the range of the wavelength [Youssef and Agrawala, 2003]. For 802.11b/g networks, working at 2.4 GHz, the wavelength λ is 12.5 cm (Equation 3.1). This effect makes very difficult to estimate the correct location because small variations in the position can lead to high RSS variations.

$$\lambda = \frac{c}{f} \approx \frac{3 \cdot 10^8 m/s}{2.4 \cdot 10^9 Hz} = 12,5cm \tag{3.1}$$

This effect is one of the main problems to deal with when using WiFi for localization since it is hard and not practical for a person or robot to place in the exact same location where the measurements for the radio-map where collected.

With the aim of studying the small scale variations, the RSS was acquired at equally separated points, at distances under the wavelength.

These measures were acquired as detailed below:

- A grid of 12.5 cm x 12.5 cm divided in 1 cm side squares was created. It is shown in Figure 3.9. This way, the points where the device should be placed at each position to collect the different measures are clearly identified.
- Initially, the device is placed at point A0 (Figure 3.10) and 300 samples are collected. This point is used as reference (λ_0).
- From point λ_0, new measurements are collected in three different directions to check the variations in the RSS caused by the small scale variations:

  1. Horizontally: RSS is measured on λ_0 (A0), $\lambda_0 + 3cm$ (A3), $\lambda_0 + 6cm$ (A6), $\lambda_0 + 9cm$ (A9), $\lambda_0 + 12cm$ (A12). These points are shown with blue circles in Figure 3.9.

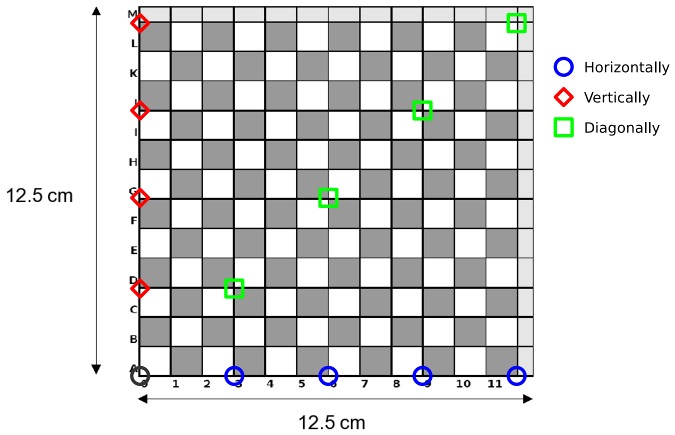


Figure 3.9: Measurement points for small scale variations analysis.

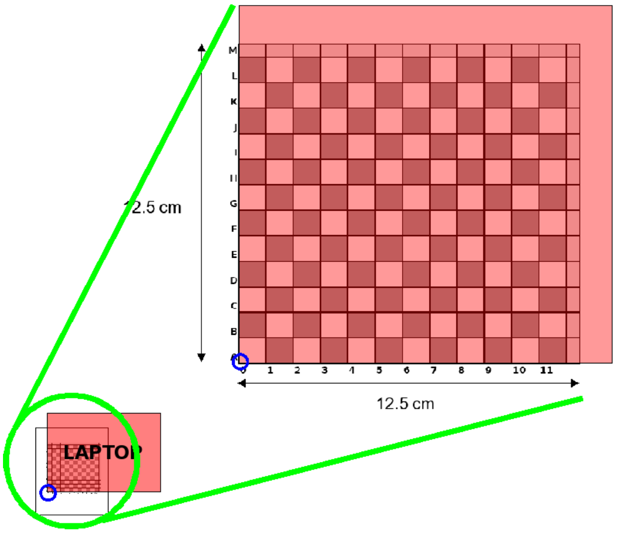


Figure 3.10: Small scale testing grid (Laptop at reference position A0).

2. Vertically: RSS is measured on λ_0 (A0), $\lambda_0{+}3cm$ (D0), $\lambda_0{+}6cm$ (G0), $\lambda_0{+}9cm$ (J0), $\lambda_0 + 12cm$ (M0). These points are shown with red diamonds in Figure 3.9.

3. Diagonally: RSS is measured on λ_0 (A0), $\lambda_0{+}3\sqrt{2}cm$ (D3), $\lambda_0{+}6\sqrt{2}cm$ (G6), $\lambda_0 + 9\sqrt{2}cm$ (J9), $\lambda_0 + 12\sqrt{2}cm$ (M12). These points are shown with green squares in Figure 3.9.

These measurements were collected at different locations of the environment trying to avoid other effects, controlling which APs were emitting in the environment and without people moving around.

Firstly, the variations in the RSS from a single AP are analysed. Figure 3.11 shows the RSS histogram from the closest AP at all the test points. As can be seen, small scale variations may cause differences up to ±10 dB for very close points (under the wavelength λ). A difference about 20 dB is large enough to induce misclassification between two different positions separated in the range of 6 to 10 meters, as explained in Section 3.4 (page 32).

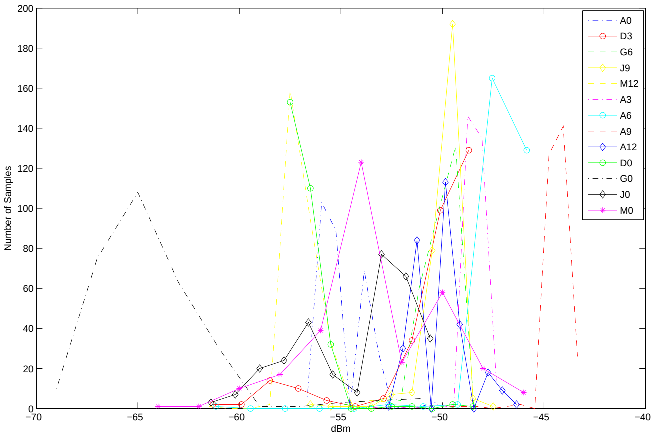


Figure 3.11: Small scale variations: RSS histogram from the closest AP at all the points around a position.

When analysing the histograms of the RSS from different APs at the same location (Figure 3.12), it can be seen that the variations caused by the small scale variations do not follow any pattern. As also stated in [Youssef and Agrawala, 2003], the RSS variations caused by this effect can be assumed chaotic and thus, can not be modelled.

Finally, Figure 3.13 shows the RSS histogram from the same AP at two different locations. As in the previous experiment, the histograms do not follow any pattern.

These experiments were repeated at different positions and using different APs, obtaining the same results. This way, it can be affirmed that the small scale variations are not related either with the position of the environment or the AP from which the measurements are collected.

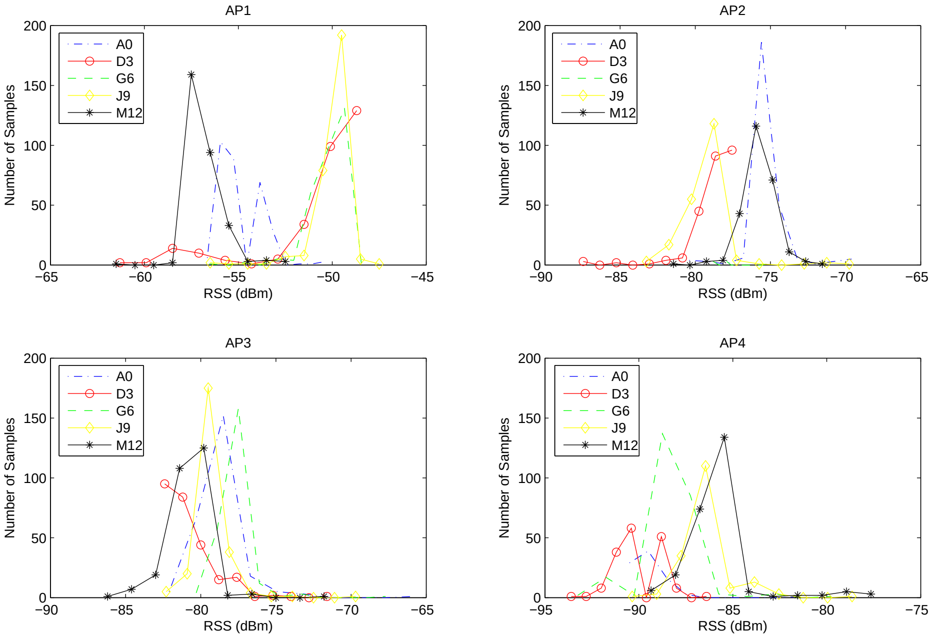


Figure 3.12: Small scale variations: RSS histogram from different APs.

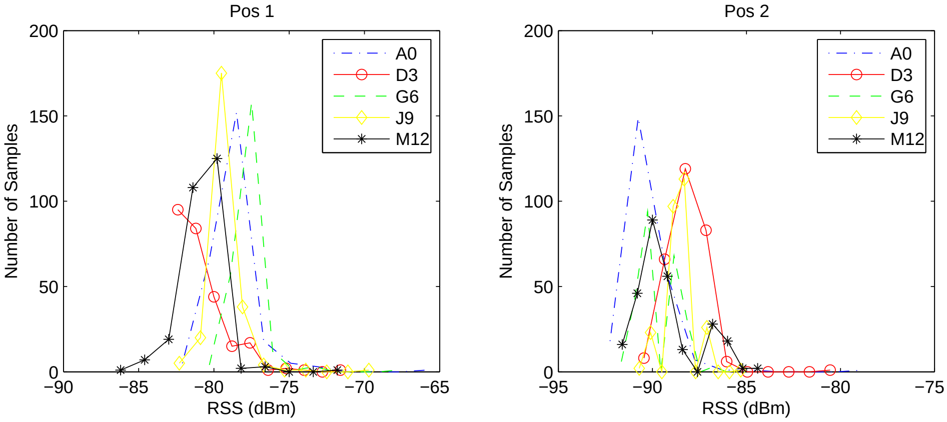


Figure 3.13: Small scale variations: RSS histogram from the same AP at two different positions.

By analysing the previous results it can be seen that small scale variations cause important variations in the RSS, being one of the main causes of error in WiFi localization systems. This way, if the measurements to test the localization system are collected a few centimetres apart from the position where the train measurements were collected, the possibilities of obtaining the correct position decrease.

3.6 Conclusions

In this chapter, the influence of different effects on the WiFi RSS have been analysed. The following conclusions can be drawn:

- Co-channel interferences cause a high variation in the RSS if new APs emitting in overlapped channels with the existing ones are installed in the environment, or if existing APs are removed or turned off. This effect is hard to avoid in the real world, since it can not be assured that an AP will not break or new APs will not be installed, unless the localization system is deployed in a fully controlled environment.

- Spatial stability analysis shows that the RSS at a static position on different days is stable enough to assure the feasibility of localization using WiFi. When introducing other effects, such as people moving around, doors opening or interferences from devices emitting in the same frequency, the RSS is noisier but still stable enough. Although glitches can be observed in the 24 hours RSS analysis, the use of multiple APs should be enough to filter most of them.

- The RSS varies with the distance allowing the differentiation of the positions of the environment. Ideally, if the RSS were only influenced by the large scale variations, it would be possible to estimate the position inside an environment using a generic propagation model. But, since there are other effects that affect the RSS, it is very difficult to adjust a model for an environment, and unlikely that the designed model adjusts well to different environments.

- Small scale variations cause important variations in the RSS, being one of the main sources of localization error. If the test measurements are collected a few centimetres away from the position where the train measurements were collected, the RSS can be very different. This effect is also hard to avoid, since in a realistic application the position of the device can not be forced to the exact same location where the train measures were collected. So, it is necessary to use techniques that cope with this effect to implement a realistic WiFi localization system.

- Some other error sources that affect WiFi RSS, such as the orientation of the device, are considered as not critical because their contribution to the RSS will be masked by more critical error sources like the small scale variations.

Chapter 4

Fuzzy Rule-based Classifiers to deal with Small Scale Variations

Small scale variations have been identified in the previous chapter as one of the main sources of uncertainty when determining the position of a device using WiFi. Different techniques have been used to tackle this problem. The most interesting one can be found in [Youssef and Agrawala, 2003]. In this work the authors proposed “the perturbation technique” to handle the small scale variations. This technique was based on restrictions over the motion a device can suffer. In order to detect small scale variations, the system calculated the distance between the current estimated location and the previous one. If this distance was above a threshold, the system assumed that small scale variations were affecting the signal strength.

To compensate for these small scale variations, the system perturbed the received vector and re-estimated the location using it. For example, if the received signal strength vector was $(RSS_1, RSS_2, \ldots, RSS_n)$, the system perturbed this vector to obtain a set of vectors: $(RSS_1(1+x), RSS_2(1+x), \ldots, RSS_n(1+x))$, where $x \in \{-d, 0, d\}$ was the noise value used to perturb the signal strength (0 for no perturbation). This means building 3^n vectors to cover all the combinations. However, the authors said that since the small scale variations depended heavily on the strength of the RSS, perturbing only the component corresponding to the strongest AP can be enough. This approach has been proved useful in an environment non crowded of APs where the averaged distance error was reduced around a 25%. This technique can be only applied when the device is in motion and the measurements are collected continuously since it makes use of information about the previous position of the device and the distance from it.

In this thesis, the problem of the small scale variations on static positions will be tackled using *Fuzzy Rule-Based Classifiers (FRBCs)*. FRBCs have been selected because of their ability dealing with complex or uncertain information.

In the next sections, a brief introduction to FRBCs design will be presented, the steps and parameters for FRBCs development will be explained and finally, the results using the designed FRBCs will be analysed and compared to other methods.

4.1 Designing fuzzy rule-based classifiers

Fuzzy sets, introduced by Zadeh [Zadeh, 1965], allow a mathematical representation of concepts using imprecise boundaries. Compared to traditional binary sets (where only two crisp values are admissible: true or false, 1 or 0) fuzzy sets may have any value in the 0 to 1 range. The use of binary sets is a strong limitation when dealing with real-world problems where the available information is noisy or uncertain.

Fuzzy modelling [Hellendoorn and Driankov, 1997] has been studied to deal with complex uncertain systems, in which conventional mathematical models may fail to obtain satisfactory results.

An important problem in the development of fuzzy models is to define the membership functions and the fuzzy rules [Mamdani, 1977] [Zadeh, 1973]. These membership functions and rules can be constructed by knowledge extraction from human experts. However, information supplied by humans suffers from certain problems. Firstly, human knowledge is usually incomplete or not organized since different experts usually make different decisions. Even the same expert may have different interpretations of the same observation on different times. Furthermore, knowledge acquisition from experts is not systematic or efficient and even in some problems there is no expert knowledge available. These problems have led researchers to build automated algorithms for modelling systems using fuzzy theories via machine learning and data mining techniques.

The proposed FRBC has been designed and built using *Generating Understandable and Accurate fuzzy models in a Java Environment (GUAJE)* [Alonso and Magdalena, 2011], a free software tool for generating understandable and accurate fuzzy models. It implements *Highly Interpretable Linguistic Knowledge (HILK)* [Alonso et al., 2008], which is a fuzzy modelling methodology that focuses on building comprehensible fuzzy classifiers. Applying fuzzy machine learning techniques, HILK is able to automatically extract useful pieces of knowledge from experimental data. Such knowledge is represented by means of linguistic variables and rules under the fuzzy logic formalism.

The whole modelling process is made up of three steps:

- **Membership functions design:** Automatic generation of fuzzy partitions from data.
- **Rule base learning:** Linguistic rules are automatically extracted from data.
- **Knowledge base improvement:** Iterative refinement of the partitions and rules.

Once the modelling process is finished and the knowledge base is built it is used to **infer the FRBC output**.

4.1.1 Membership functions design

Since GUAJE looks for interpretability as well as accuracy, the system variables, automatically extracted from data, are described by a set of linguistic terms represented by membership functions like the ones in Figure 4.1. As it can be seen the same value *xi* is partially *Low* (0.2) and *Medium* (0.8), but the addition of both membership degrees equals one. This kind of partition, called *Strong Fuzzy Partition (SFP)* [Ruspini, 1969], is the best from the comprehensibility point of view because it satisfy all the semantic constraints (distinguishability, normalization, coverage, overlapping, etc.) demanded to be comprehensible [Mencar and Fanelli, 2008].

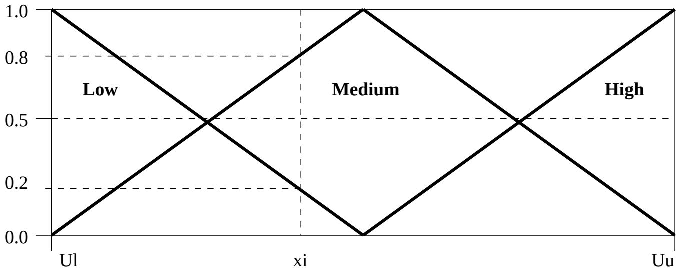


Figure 4.1: A strong fuzzy partition with three linguistic terms.

4.1.2 Rule base learning

Once all the linguistic variables have been defined with a set of linguistic terms and their associated semantics, they can be used to express linguistic propositions like "*RSS from* AP_i *is High*". Then, several propositions are combined to form fuzzy rules describing the system behaviour:

$$r\text{: }\textbf{If}\quad \underbrace{\underbrace{X_1 \text{ is } A_1^i}_{Partial\ Premise\ P_1}\quad \textbf{AND} \ldots \textbf{AND}\quad \underbrace{X_I \text{ is } A_I^j}_{Partial\ Premise\ P_I}}_{Premise}\quad \textbf{Then}\ \underbrace{Y_r \text{ is } C^n}_{Conclusion}$$

where, given rule r, rule premises are made up of tuples (*input variable*, *linguistic term*) where X_a is the name of the input variable a, while A_a^i represents the label i of such variable, with a belonging to $\{1, ..., I\}$ and being I the number of inputs. In the conclusion part, C^n represents one of the possible output classes, i.e., one position in the case of WiFi localization.

For instance,

If *RSS from APi is High* **and** *RSS from APj is Low*
Then *The device is close to Position k*

In order to generate these rules from data, different methods can be found in the literature [Hüllermeier, 2005]. Among them, the two following methods have been chosen to generate the rules from data with the previously defined fuzzy partitions:

- ***Wang and Mendel (WM)*** [Wang and Mendel, 1992]: It generates complete rules (considering all the available variables) which are quite specific. WM starts by generating one rule for each data pair of the training set and then, new rules will compete with existing ones.

- ***Fuzzy Decision Trees (FDT)*** [Ichihashi et al., 1996]: It generates a neuro-fuzzy decision tree (directly from data) which is translated into quite general incomplete rules (only a subset of input variables is considered). In addition, inputs are sorted according to their importance (minimizing the entropy). FDT is a fuzzy version of the decision trees defined by Quinlan in [Quinlan, 1986] and improved in [Quinlan, 1996].

4.1.3 Knowledge base improvement

After defining the linguistic variables and rules, HILK offers a powerful and flexible simplification procedure which affects to the whole knowledge base including both partitions and rules. The goal is getting a more compact and general FRBC, keeping high accuracy while increasing comprehensibility and reducing the system complexity. It starts by looking for redundant elements (terms, variables, rules, etc.) that can be removed without altering the system accuracy. Then, it tries to merge elements always used together. Finally, it forces removing elements not contributing to the final accuracy.

4.1.4 Inferring the FRBC output

Once the knowledge base is created, it is used to infer the system output as follows:

First, given an input vector $x^p = \{x_1^p, ..., x_I^p\}$, the firing degree (for each rule r) is computed as the minimum membership degree of x^p to all the attached A_i^j fuzzy sets, for all the I inputs (Equation 4.1):

$$\mu_r(x^p) = \min_{i=1,...,I} \mu_{A_i^j}(x_i^p) \tag{4.1}$$

Then, the output class C^i is derived from the highest $\mu_{C^i}(x^p)$ (Equation 4.2) which is the membership degree of x^p to the class C^i. It is computed as the maximum firing degree of all rules yielding C^i as output class (Equation 4.3).

$$y_{FRBC}(x^p) = C^i \Leftrightarrow \mu_{C^i}(x^p) = \max_{n=1,...,c} \mu_{C^n}(x^p) \tag{4.2}$$

$$\mu_{C^n}(x^p) = \max_{r=1,...,R} \mu_r(x^p) \Leftrightarrow Y_r \text{ is } C^n \tag{4.3}$$

The output of the FRBC will be one position along with an activation degree that can be understood as a degree of confidence on the system output. Several output classes can be activated since several fuzzy rules can be fired at the same time by the same input vector. This way, the activation degrees of the different classes could be used in order to make an interpolation among several positions. For instance, if the system output says that the device is in position A with a confidence degree of 0.2 and in position B with a confidence degree of 0.8, it can be considered that it is located between A and B but closer to B.

4.2 Experimental analysis

This section describes the experimental results obtained on the FRBC development.

4.2.1 Experimental set-up

Exhaustive experiments were carried out in the test-bed environment established at the Campus of the University of Oviedo, old premises of the *European Centre for Soft Computing (ECSC)* located at Asturias (Spain).

The layout of the ECSC test-bed is shown in Figure 4.2. It has a surface of about 500 m^2 with 15 offices, one main long corridor, and two large open working areas. There are 6 APs distributed over the environment which is discretised into 16 significant topological positions.

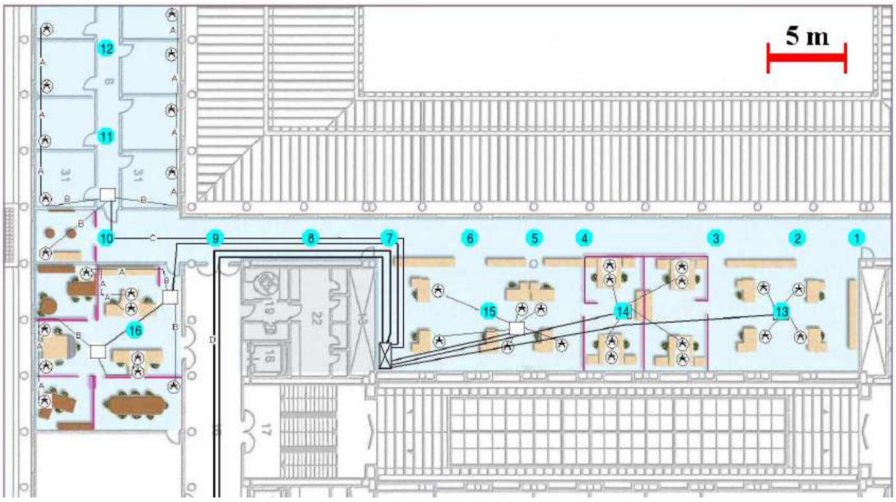


Figure 4.2: Small scale analysis: ECSC environment.

The dataset used for the FRBC generation is composed of the measures corresponding to λ_0 point at each environment position, while the test dataset is composed of the measures collected at all the other points, as explained in Section 3.5 (page 33). This way, the system is tested maximizing the small scale variations effect.

4.2.2 Experimental results

Looking for the FRBC providing the best results, the influence of some parameters has been analysed. These parameters are:

- **Input variables.**

  The RSS and the *Signal-to-Noise Ratio (SNR)* have been collected from the available APs. Thus, two different configurations have been considered: using only the RSS, and using both the RSS and the SNR.

- **Number of linguistic terms defined per input.**

  Since GUAJE generates FRBCs maximizing both the interpretability and the accuracy, it recommends to choose an odd number of linguistic terms equal or smaller to nine. Therefore, four cases have been analysed: three, five, seven and nine linguistic terms defined for each input variable.

- **Rule induction technique.**

  Two rule induction algorithms have been considered, Wang and Mendel (WM) and Fuzzy Decision Trees (FDT), introduced in Section 4.1.2. The WM algorithm has not any configuration parameters, but for the FDT algorithm two cases are evaluated: the whole tree (FDT) and the *Pruned Fuzzy Decision Trees (PFDT)* with a loss tolerance threshold equal to 0.1. Then, the HILK simplification algorithm has been executed for the three cases (*Wang and Mendel with Simplification (WM-S)*, *Fuzzy Decision Trees with Simplification (FDT-S)* and *Pruned Fuzzy Decision Trees with Simplification (PFDT-S)*). Hence, a total of six different methods have been tested; three different rule induction techniques before and after simplification.

- **Number of samples to average at both training and test stages.**

  Six cases are evaluated: 1 (raw data without averaging), 4, 12, 28, 40, and 60 averaged samples. The time spent collecting the samples is not a problem during the training stage of the system because it is made offline. However, it becomes a critical requirement during the localization stage.

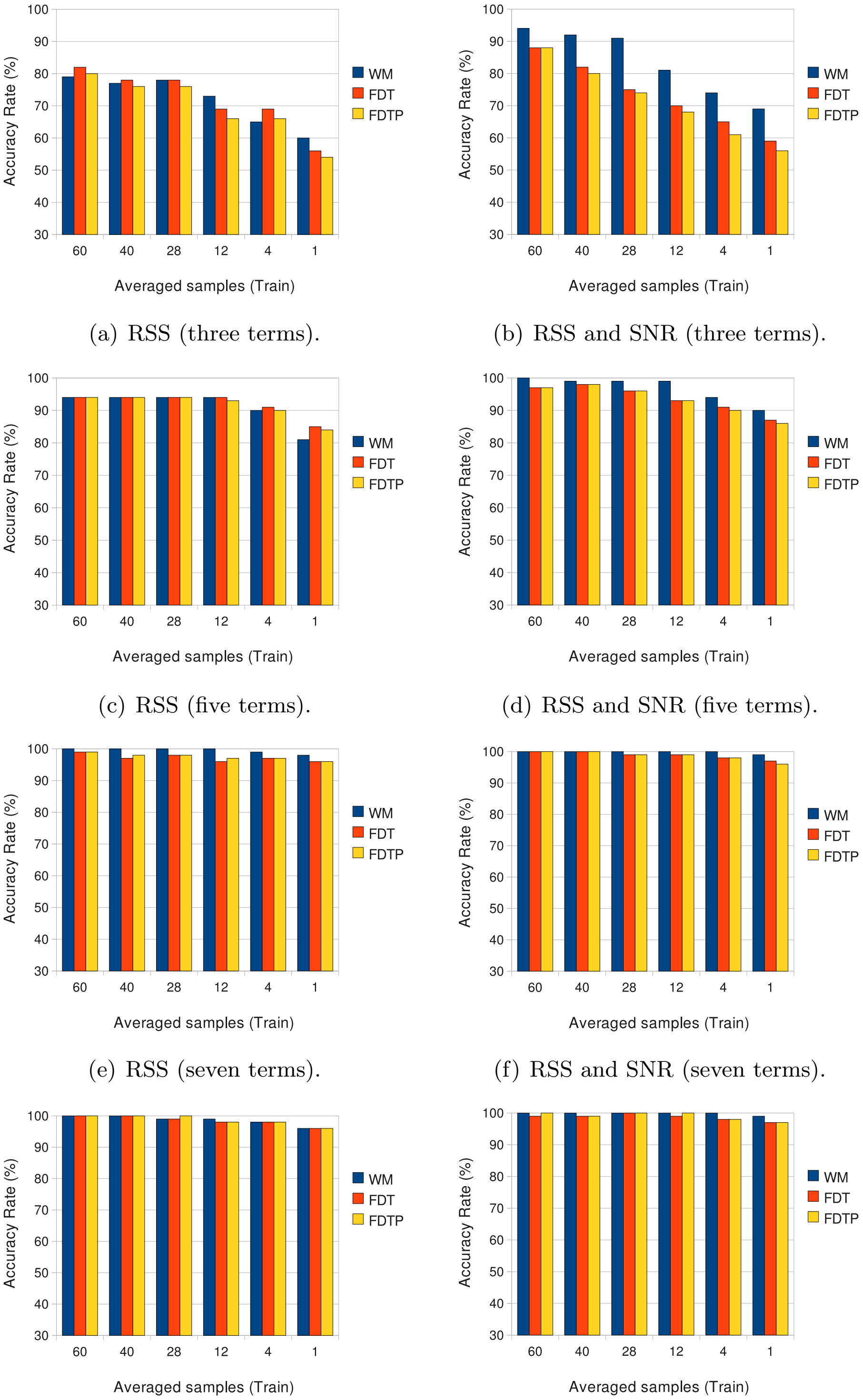
In summary, 288 (2 x 4 x 6 x 6) FRBCs covering all the described situations have been built, evaluating each of them with six test datasets yielding a total of 1728 experiments.

4.2.2.1 Input variables

In this section an analysis of the inclusion of the SNR in the localization system is carried out. The goal is to find out if the measured SNR information is useful for the localization. Figure 4.3 shows the results during the training stage using only the RSS or considering both the RSS and the SNR. On the one hand, the vertical axes represent the accuracy of the analysed FRBC computed as the percentage of correctly classified samples using the training dataset. On the other hand, the horizontal axes show the number of samples averaged to build the training dataset.

From these graphs, comparing left and right columns, it can be observed that the computed accuracy is slightly worse when working with the RSS alone. When using the test data, the results are better using only the RSS what suggests that adding the SNR causes some overfitting effect, as can be seen in Figure 4.4. In Figure 4.4 the horizontal axes are slightly different since they include pairs of numbers showing the samples averaged to build the train and test datasets respectively. For instance, $12 - 4$ means that the classifier is trained using blocks of twelve samples while the test dataset is built averaging four samples.

The decrease in accuracy is produced because the SNR measures do not follow a particular pattern. That is why the generalization ability of the classifiers is strongly



(a) RSS (three terms).

(b) RSS and SNR (three terms).

(c) RSS (five terms).

(d) RSS and SNR (five terms).

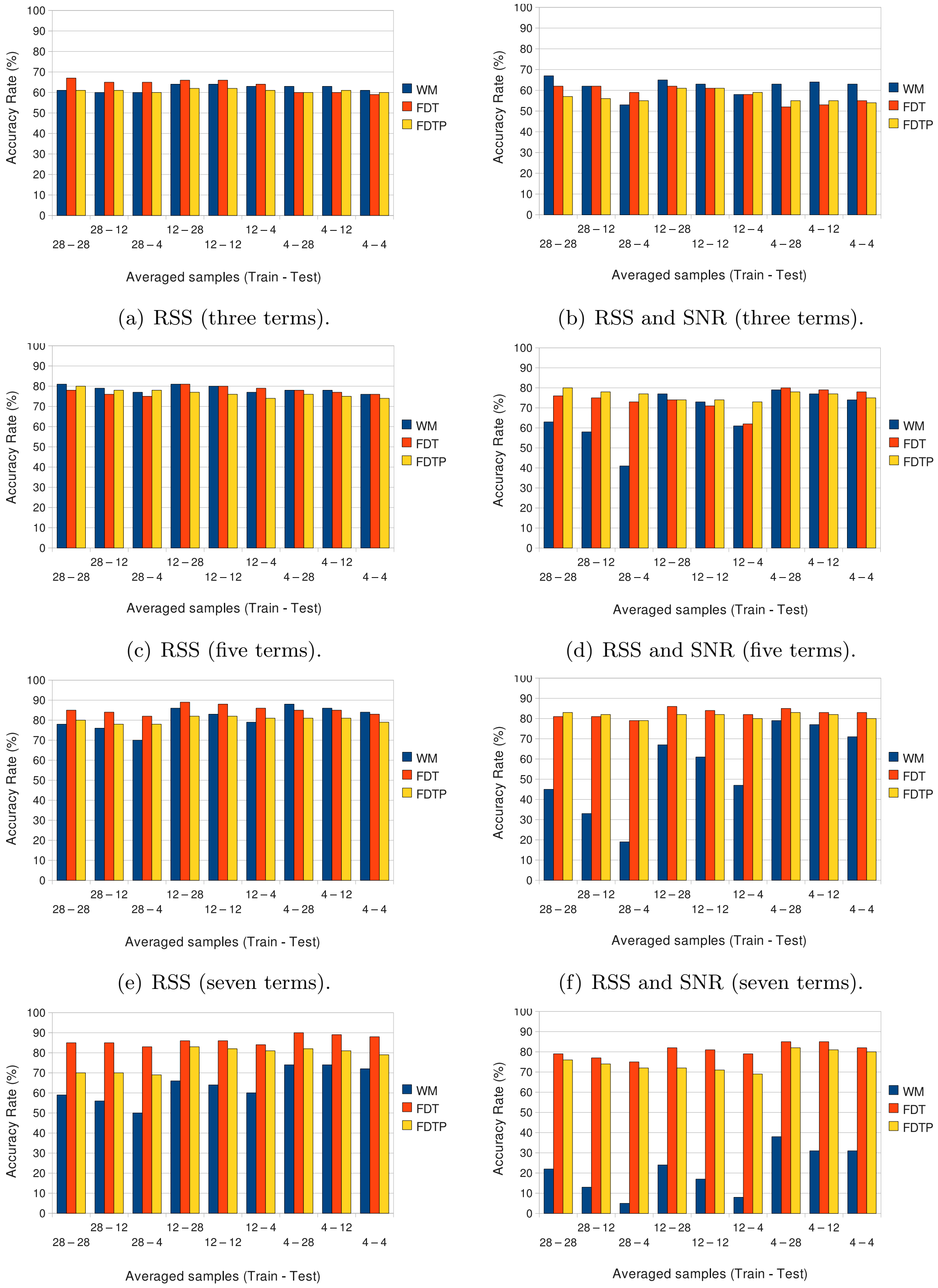
(e) RSS (seven terms).

(f) RSS and SNR (seven terms).

(g) RSS (nine terms).

(h) RSS and SNR (nine terms).

Figure 4.3: FRBC design: Varying the inputs. Training stage.



(a) RSS (three terms).

(b) RSS and SNR (three terms).

(c) RSS (five terms).

(d) RSS and SNR (five terms).

(e) RSS (seven terms).

(f) RSS and SNR (seven terms).

(g) RSS (nine terms).

(h) RSS and SNR (nine terms).

Figure 4.4: FRBC design: Varying the inputs. Test stage.

penalized and the accuracy is reduced for the test data. Moreover, the SNR inputs usually disappear of the rules after simplification for most of the designed classifiers. Therefore, it can be concluded that the SNR does not give any reliable information to the localization system. Furthermore, doubling the number of inputs increase the system complexity without any advantages, so the SNR can be discarded and the FRBC can be built using only the RSS.

4.2.2.2 Rule induction technique and number of linguistic terms

Taking advantage of the conclusions derived from the analysis made in the previous section, a more detailed analysis for both the number of terms and the rule induction algorithms can be made by focusing only on the FRBCs built using the RSS. The goal is to find out the best combination of both the number of terms (3, 5, 7, or 9) and the rule induction method with or without simplification (WM, WM-S, FDT, FDT-S, PFDT, and PFDT-S). Figures 4.5 and 4.6 show the comparison of the designed FRBCs for both the training and the test data respectively. As expected, during the training stage the accuracy increases as the number of terms grows, but the results are almost steady from seven labels on. This behaviour is not always held on test data where an overfitting effect sometimes appears when passing from seven to nine terms. Such effect is due to the excessive specification of rules when working with a large number of linguistic terms.

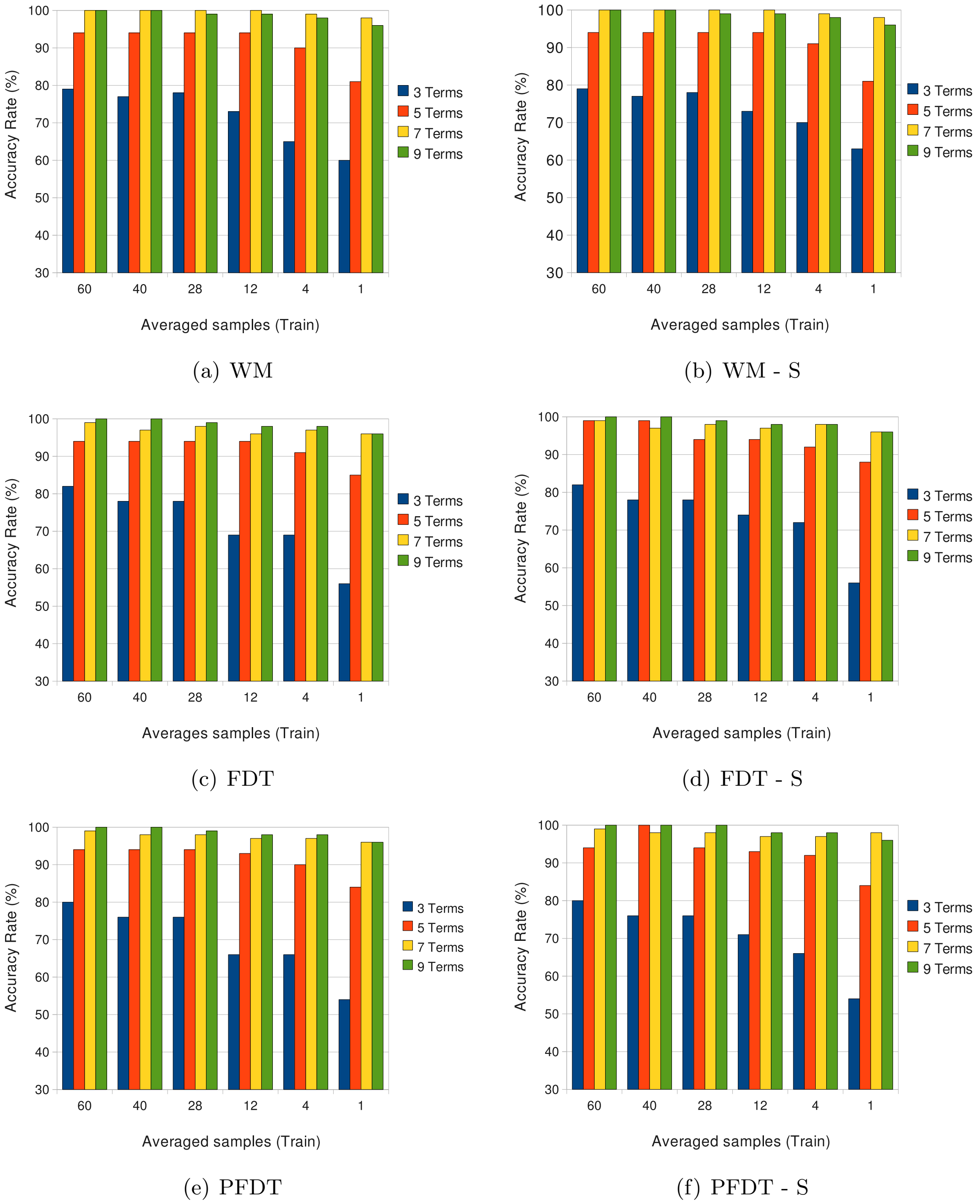
After comparing left and right graphs in Figure 4.5, it can be deduced that the simplification procedure gets more compact FRBCs keeping (and sometimes increasing) the achieved accuracy during the training stage. Nevertheless, this statement is not always true when looking at test results plotted in Figure 4.6. Simplification does not alter accuracy when dealing with WM, but it slightly gets worse accuracy for the FDT and PFDT which usually exhibit a good generalization ability. The simplification procedure enhances the comprehensibility of the final model at the cost of losing some accuracy, which is not admissible for a localization application.

FDT using nine linguistic terms provides the most accurate FRBC for both the training and the test datasets. There are two main reasons for not using more than nine terms per input variable. A large number of terms leads to overfitting and it may decrease the generalization ability of the model, but it also would be less tolerant to slight modifications in the environment (for instance people moving around).

4.2.2.3 Number of samples to average

In this section the influence of the number of samples to average is analysed for both the training and the estimation stages.

Keeping in mind the previously drawn conclusions, this section focuses on the results obtained using the FRBC providing the best results, i.e., the one built considering the RSS only, nine linguistic terms with their associated uniformly distributed SFPs, and linguistic rules automatically generated from training data using the FDT algorithm. Figure 4.7 shows how the accuracy varies depending on the number of samples averaged at the training stage. As expected, the larger the number of samples averaged, the higher accuracy is achieved. The accuracy gets 100% for a number of samples greater or equal



(a) WM

(b) WM - S

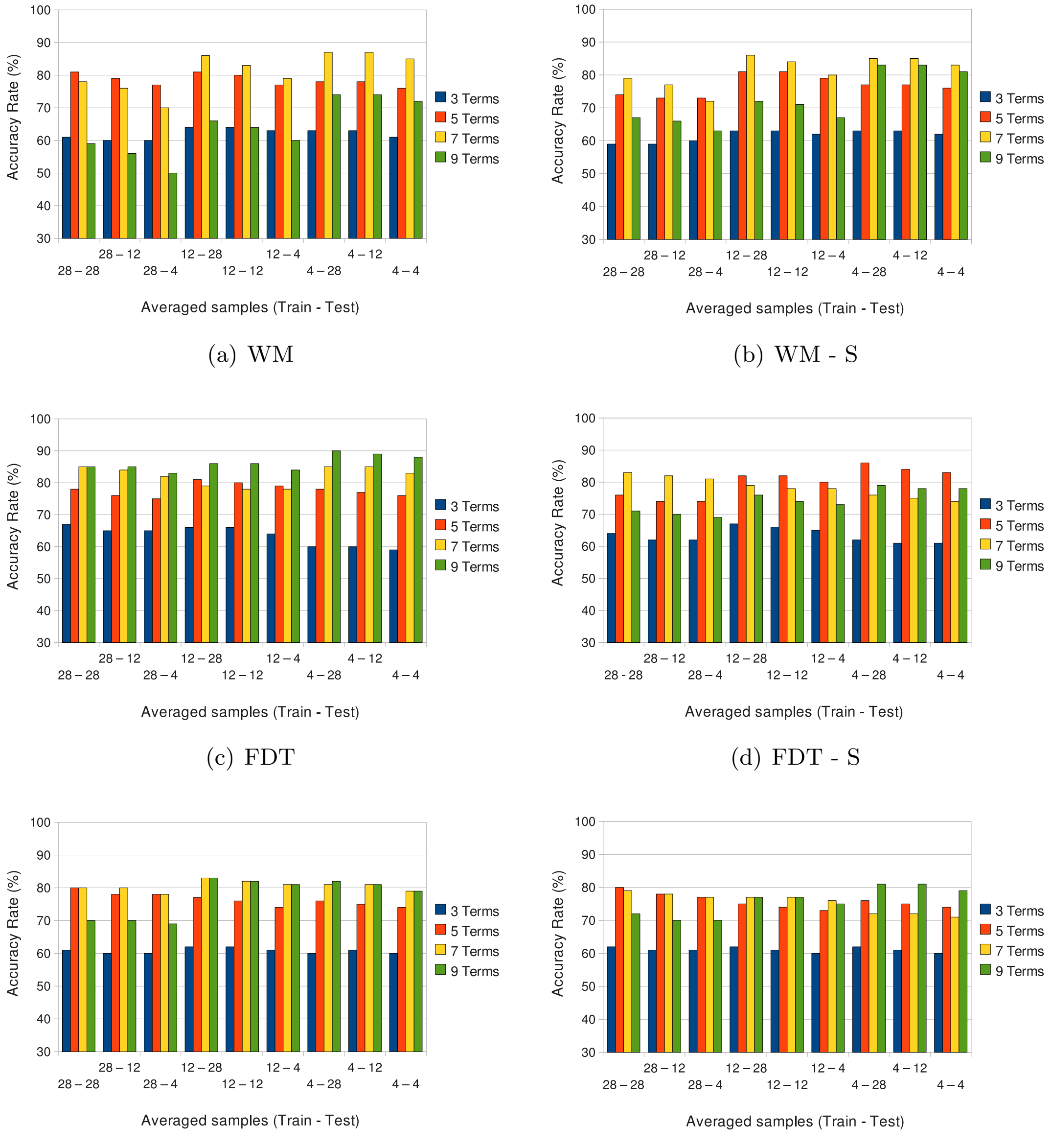
(c) FDT

(d) FDT - S

(e) PFDT

(f) PFDT - S

Figure 4.5: FRBC design: Varying the rule induction method and number of linguistic terms. Training stage.



(a) WM

(b) WM - S

(c) FDT

(d) FDT - S

(e) PFDT

(f) PFDT - S

Figure 4.6: FRBC design: Varying the rule induction method and number of linguistic terms. Test stage.

to 40. As an effect of averaging, the measured variations are smoother and the accuracy is higher, but at the cost of a longer acquisition time and a loss of generalization. One minute is the time needed for acquiring 60 samples if the acquisition frequency is equal to 1 Hz. This time can be acceptable for training but it could be too much for an online localization.

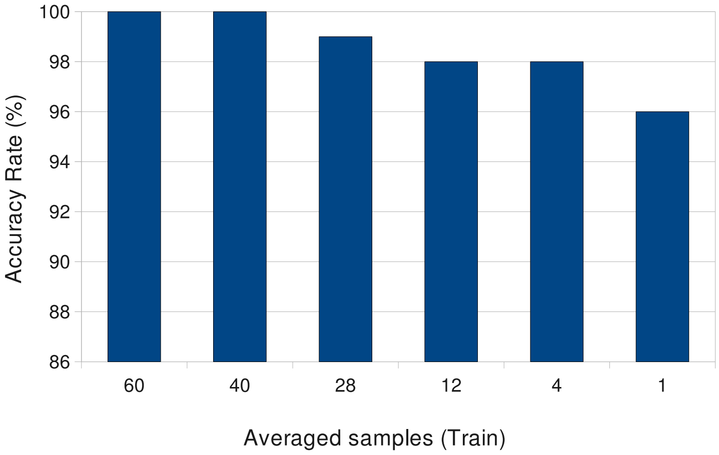


Figure 4.7: FRBC design: Varying the number of samples to average. Training stage.

Figure 4.8 illustrates the results on the test stage using different number of samples to average both the train and the test datasets. It is easy to appreciate that the accuracy increases as the number of averaged samples decreases. This effect is caused because the number of samples available to train the system is reduced as the number of averaged samples increases. This way, having 300 samples per position in the train dataset, makes 300 samples per position in the non-averaged dataset, 75 in the 4 averaged samples dataset, leaving only 5 samples in the train dataset when averaging using 60 samples.

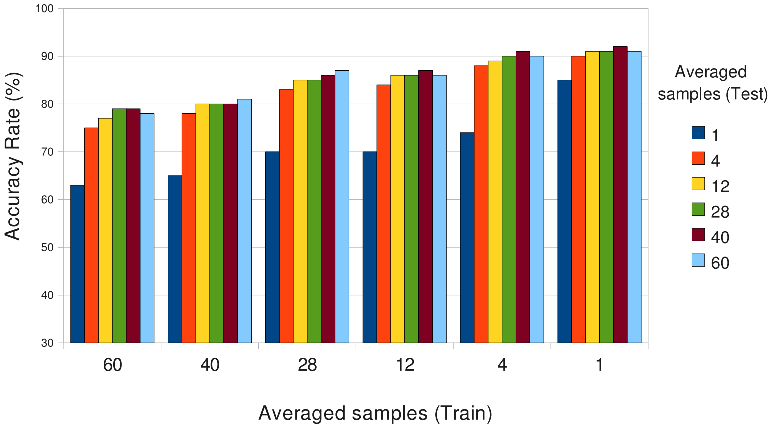


Figure 4.8: FRBC design: Varying the number of samples to average (using all the available samples). Test stage.

To be able to do a fairer comparison, the same number of samples have been used to build all the classifiers. Figure 4.9 illustrates the results on the test stage using the different number of samples to average both the train and the test datasets. It is easy to appreciate that the accuracy on test gets the highest rates when the FRBC is built using 4 samples to average the training data. The most accurate solution is the one obtained when testing with 60 averaged samples (around 85%), what means one minute for acquisition time. From a practical point of view, it is desirable an acquisition time as small as possible during the test stage. Looking at Figure 4.9 it seems reasonable to consider only 4 averaged samples, what decreases the acquisition time to 4 seconds while still keeping a high accuracy around 83% (87% using all the available samples (Figure 4.8)). It yields a good trade-off between the accuracy and the acquisition time.

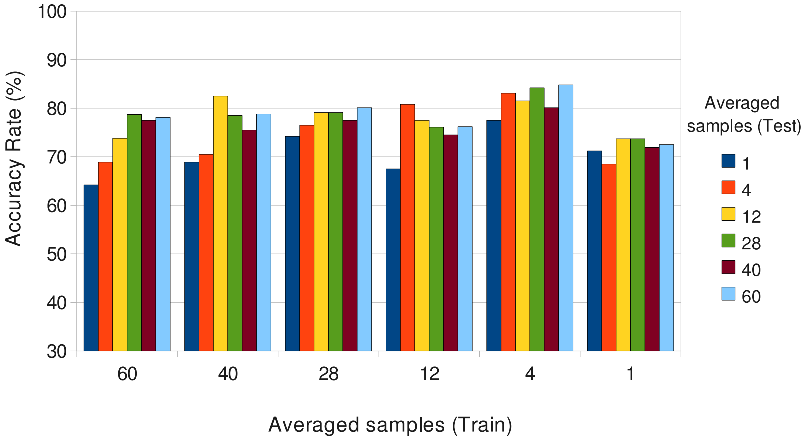


Figure 4.9: FRBC design: Varying the number of samples to average (using the same number of samples for the datasets). Test stage.

4.2.3 Using other classifiers to deal with small scale variations

To test the performance of the FRBC dealing with the small scale variations, different classifiers have been tested using the same train and test datasets. *Fuzzy Unordered Rule Induction Algorithm (FURIA)*, KNN and SVM have been selected to be compared with the designed FRBC (from now on, it will be referred as FDT).

Figure 4.10 shows the achieved accuracy obtained using the four classifiers. As can be seen, the best results in terms of accuracy are achieved using the FDT classifier (86.77%), a 7.44% higher than the results achieved by the KNN algorithm (79.33%) which provides the second highest accuracy.

Figure 4.11 shows an analysis of the distance to the real positions achieved by each classifier. As can be seen, the FDT classifier has the lowest error, obtaining an error of 6 metres for the 95th percentile. Notice that, since a topology-based indoor localization is performed, the error in distance depends on the distance between the topological positions (the minimum distance between two positions of the environment is 3.5 metres).

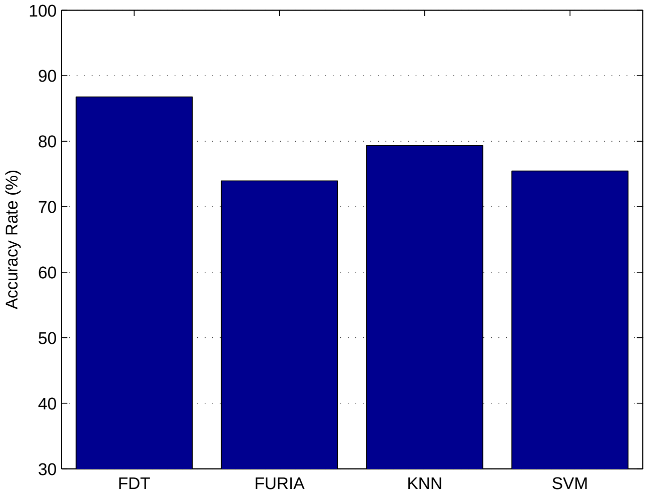


Figure 4.10: Accuracy using different classifiers to deal with the small scale variations.

The mean distance to the real position using the FDT classifier is 6.45 metres for the misclassified samples and 0.85 metres taking into account all samples (both correctly and incorrectly classified) compared to the 7.22 metres and 1.49 metres respectively obtained using the KNN classifier.

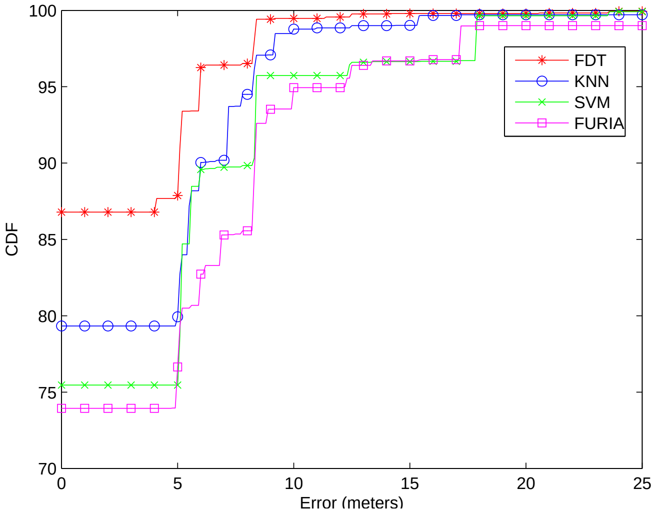
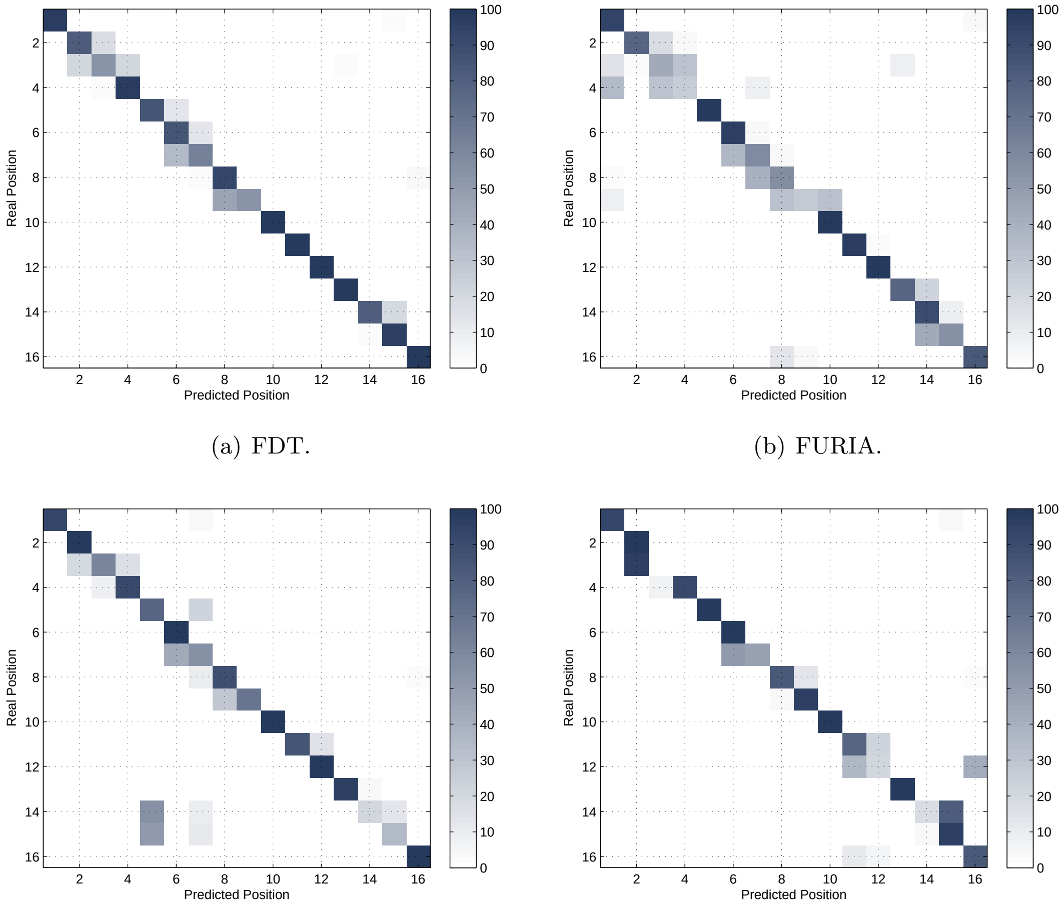


Figure 4.11: CDF of different classifiers to deal with the small scale variations.

Finally, Figure 4.12 shows the confusion matrix of each classifier. It details the predicted positions by the classifier related to the positions where the device really was. Looking at the figure, it can be seen that most of the classification errors occur within the nearest positions for the FDT classifier, while the other classifiers misclassified more separated positions. This points out that the small scale effect is reduced by using the designed FRBC.



(a) FDT.

(b) FURIA.

(c) KNN.

(d) SVM.

Figure 4.12: Confusion matrix of different classifiers to deal with the small scale variations.

4.3 Conclusions

The uncertainty generated by small scale variations has been reduced with the proposed system obtaining an accuracy around to 87% using only 4 averaged samples in the test stage.

Regarding the design of the FRBCs, the following conclusions can be extracted:

- The SNR does not provide any reliable information to the WiFi localization and increases its complexity, so the SNR can be discarded.
- Accuracy increases when adding more linguistic terms per input during the training stage. This is due to the fact that the input space is split into smaller cells providing

a larger granularity and a finer analysis. As a side effect, the number of rules is increased, rules are more specific and the generalization ability of the FRBCs is reduced depending on the available data as well as on the selected rule induction technique (FDT provides more general rules than WM). That is why the larger the number of linguistic terms the higher the accuracy with FDT and PFDT, while using WM or the algorithms with simplification the accuracy is higher with a medium value of linguistic terms.

- In general, the best results are obtained using FDT, in both the training and the test stages.
- Using WM provides high accuracy in the training stage, but during the test stage it is slightly lower than using FDT. This is because WM algorithm generates rules more specific for the available data reducing its generalization.
- Using the simplification procedure with the WM algorithm (WM-S) the accuracy in the test stage is increased, but not enough to overcome the FDT results.
- Simplification slightly reduces accuracy for the FDT algorithm. The simplification procedure enhances the comprehensibility of the final model at the cost of losing some accuracy, which according to localization requirements is not admissible.
- Only 4 samples in both the train and the test stages are needed to achieve an accuracy around 87%, so during the localization stage it would be necessary to spend only four seconds to acquire the required number of samples.
- The best results are achieved with the FRBC built using only the RSS as input, nine linguistic terms with their associated uniformly distributed SFPs, linguistic rules automatically generated from training data by means of the FDT algorithm and averaging the data using 4 samples for both the train and the test stages.

Regarding the use of the proposed FRBCs to reduce the effect of the small scale variations, the following conclusions can be drawn:

- The accuracy was improved around a 7% in comparison with the algorithm providing the second highest accuracy. The mean error was reduced a 10% when taking into account only the misclassified samples, and around a 42% when taking into account all the samples (both correctly and incorrectly classified).
- The misclassified samples using the FDT algorithm were always classified within the nearest positions.
- The generalization ability of the FDT algorithm should also be able to absorb slight modifications on RSS, such as the noise produced by people moving around, punctual interferences from other devices, etc.
- The results were obtained in a relatively small environment, with a small number of APs, at static positions of the environment. Further conclusions can be extracted by using the FDT in larger environments and during an online stage when the device's position is obtained while it is in motion.

Chapter 5

WiFi Indoor Localization in Large Environments

In the previous chapters, the main sources of error in WiFi localization systems were analysed. An FDT algorithm was proposed in Chapter 4 to deal with the main source of error, the small scale variations, achieving an accuracy close to 87% in a small environment non crowded with APs. As explained before, most of the systems in the literature are designed in small or medium sized environments with their APs deployed for localization purposes, following a grid distribution, and in a much smaller number than the expected in a real environment.

In this chapter, the challenge of designing a WiFi localization system for a large environment, crowded with APs not deployed for localization purposes, will be faced. The main challenges in these environments will be analysed and a WiFi localization system able to deal with them will be devised.

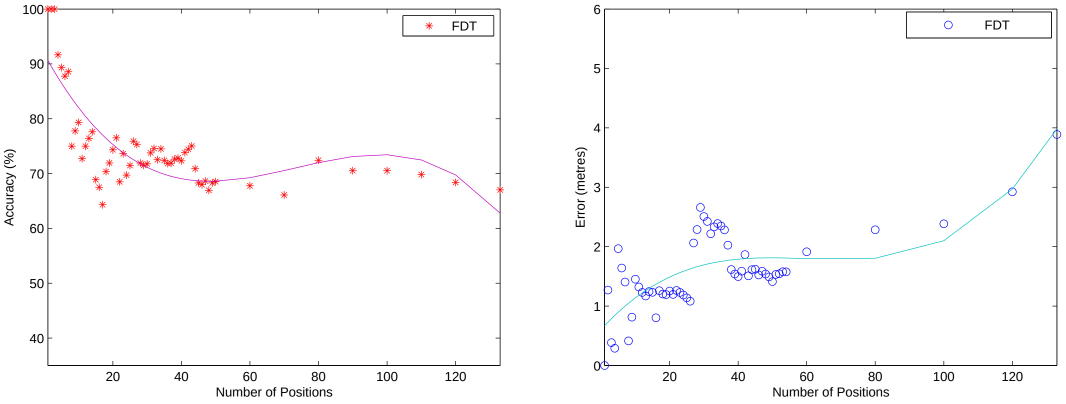
The remaining of the chapter is organised as follows: First, the performance of the system in large environments will be analysed in Section 5.1. Then, Section 5.2 will describe a hierarchical approach proposed to tackle with the loss of performance in large environments. Next, in Section 5.3 the experimental analysis and results will be exposed and, finally, in Section 5.4 a critical discussion about the results will be provided.

5.1 Analysis of the performance in large environments

As result of all the effects described in Chapter 3, 2.4 GHz WiFi signal is extremely noisy. To overcome this problem, the FDT algorithm described in Chapter 4 was proposed to improve the localization performance, obtaining an accuracy close to 87%. These results were obtained in a small environment (16 positions and 6 APs).

To evaluate the behaviour of the system in a more realistic context, a large environment located in the West wing of the Polytechnic School at the *University of Alcalá (UAH)* was site-surveyed. This environment has four floors with a surface of $3000m^2$ each. WiFi measurements were collected on 133 positions distributed over the four floors where 216 APs were detected. For a full description of the environment, please refer to Section 5.3.1.

Figure 5.1 shows the results of the FDT classifier in the large environment. Figure 5.1(a) shows the accuracy variation as the number of the positions increases, while Figure 5.1(b) shows the mean error (computed as the mean distance between the estimated positions and the real ones). As can be seen, accuracy decreases as the number of positions increases. This effect seems to cease after certain number of positions (around 50) is reached. In the same way, the mean error of the system increases with the number of positions.

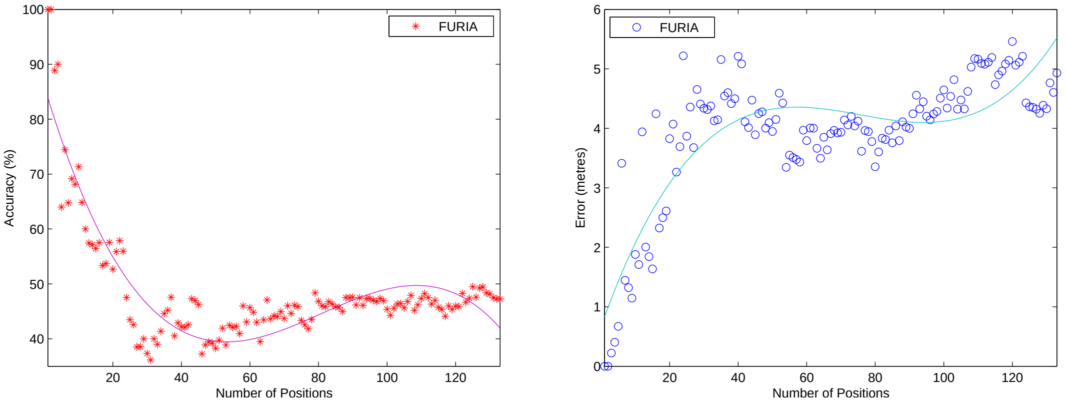


(a) Accuracy variation.

(b) Mean error variation.

Figure 5.1: FDT: Accuracy and mean error variation with the number of positions.

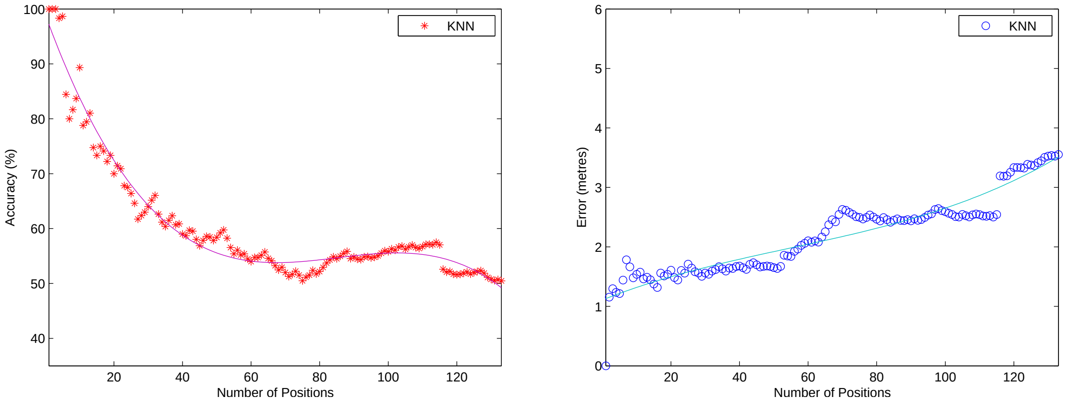
These experiments have been repeated with other classifiers: FURIA, KNN and SVM (Figures 5.2, 5.3 and 5.4). As can be seen, the effect is similar for all the classifiers: accuracy decreases as the number of positions increases and the mean error increases with the number of positions.



(a) Accuracy variation.

(b) Mean error variation.

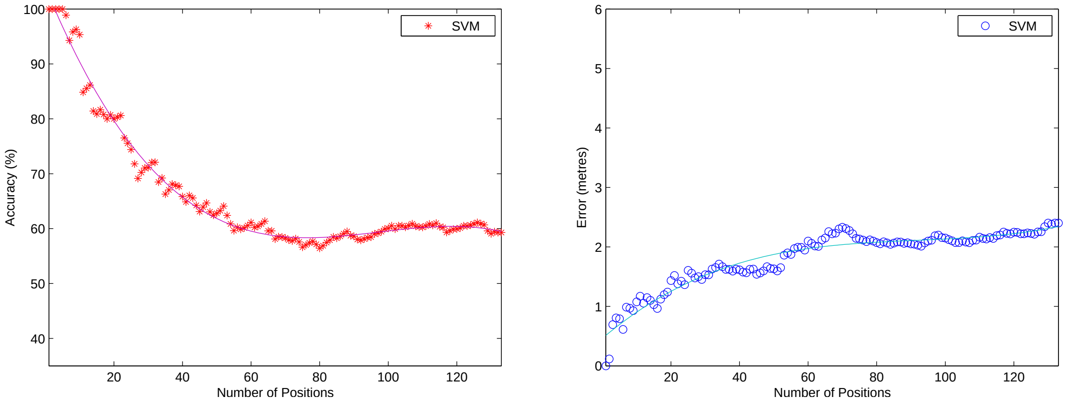
Figure 5.2: FURIA: Accuracy and mean error variation with the number of positions.



(a) Accuracy variation.

(b) Mean error variation.

Figure 5.3: KNN: Accuracy and mean error variation with the number of positions.



(a) Accuracy variation.

(b) Mean error variation.

Figure 5.4: SVM: Accuracy and mean error variation with the number of positions.

To tackle this problem the classifiers' task will be simplified by dividing the environment into smaller sub-zones, in a hierarchical approach, where the classifiers will have to assign each sample to a zone, reducing the number of outputs in the first level and the number of inputs and outputs in the following ones. With this division of the problem, the effects that appear when the number of positions and APs increases are expected to be reduced, improving the performance of the localization system.

5.2 Hierarchical approach

This section presents a description of the proposed hierarchical localization approach. The main objective is to achieve high accuracy even when working in large environments. To do so, the system will create a hierarchical partition of the environment with the objective of improving the localization task while decreasing its complexity by reducing the number of positions in each one of the partitions. First, a hierarchical partition of the environment will be created using a clustering algorithm. Then, different classifiers will be trained to localize the device through the different levels of the hierarchy. This way, the device will be first located inside the higher subzones, to finally decide the position of the device inside the lower ones. A block diagram of the entire system is shown in Figure 5.5.

Both training and localization stages will be thoroughly explained in the next subsections.

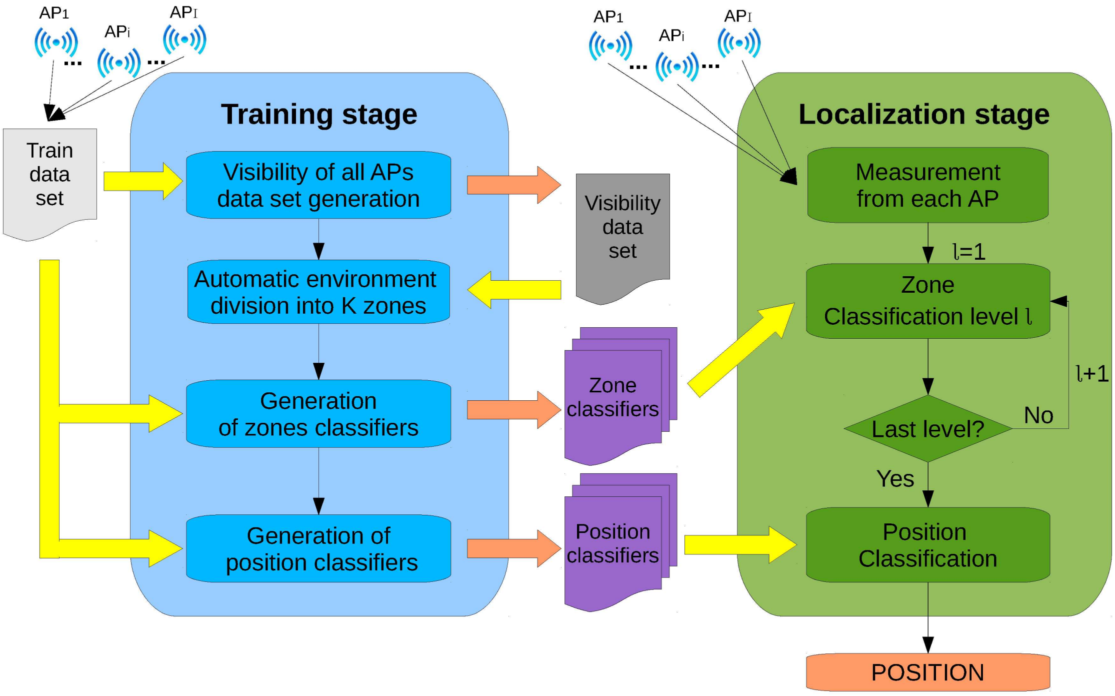


Figure 5.5: Hierarchical approach: General architecture of the system.

5.2.1 Training stage

The goal is to obtain a hierarchical localization tree by dividing the environment into zones. For each zone, a specific classifier will be trained to distinguish between the different zones (zone classifiers) and, in the lowest level of each one of the tree branches, one classifier will be trained to distinguish between the different positions (position classifiers). The training stage consists of the following steps:

- **Visibility dataset generation**: First, the RSS is measured for every position of the environment and stored in $RSS_{TRAINDATA}$ (Equation 5.1).

$$RSS_{TRAINDATA} = \begin{pmatrix} RSS_{AP_1}(P_1,1) & RSS_{AP_2}(P_1,1) & \dots & RSS_{AP_I}(P_1,1) \\ \vdots & \vdots & & \vdots \\ RSS_{AP_1}(P_1,S) & RSS_{AP_2}(P_1,S) & \dots & RSS_{AP_I}(P_1,S) \\ RSS_{AP_1}(P_2,1) & RSS_{AP_2}(P_2,1) & \dots & RSS_{AP_I}(P_2,1) \\ \vdots & \vdots & & \vdots \\ RSS_{AP_1}(P_2,S) & RSS_{AP_2}(P_2,S) & \dots & RSS_{AP_I}(P_2,S) \\ RSS_{AP_1}(P_J,1) & RSS_{AP_2}(P_J,1) & \dots & RSS_{AP_I}(P_J,1) \\ \vdots & \vdots & & \vdots \\ RSS_{AP_1}(P_J,S) & RSS_{AP_2}(P_J,S) & \dots & RSS_{AP_I}(P_J,S) \end{pmatrix} \tag{5.1}$$

where I is the number of APs, J is the number of positions and S is the number of samples collected per position.

The division of the environment is performed using the so-called visibility. The visibility of an AP (AP_i) at a certain position (P_j) is defined by Equation 5.2:

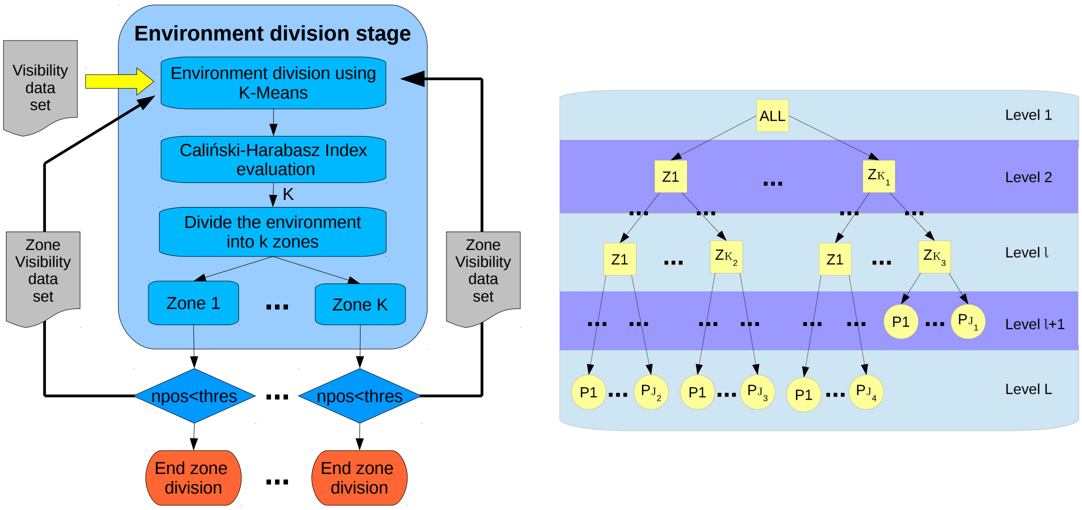
$$VIS_{AP_i}(P_j) = \frac{1}{S}\sum_{s=1}^{S} d_{ij}(s), \quad d_{ij}(s) = \begin{cases} 1 & , \ RSS_{AP_i}(P_j,s) > RSS_{thres} \\ 0 & , \ otherwise \end{cases} \tag{5.2}$$

$VIS_{AP_i}(P_j)$ is computed as the percentage of samples that were collected with $RSS_{AP_i}(P_j,s)$ greater than a predefined threshold RSS_{thres}. Currently, this threshold is set to the minimum value, this way the sample s is taken into account for visibility purposes for any $RSS_{AP_i}(P_j,s)$. This threshold could be used to decrease the visibility of those APs with low RSS.

Once the visibility of all APs for each position is evaluated, the visibility dataset $(VIS_{TRAINDATA})$ is generated as described by Equation 5.3:

$$VIS_{TRAINDATA} = \begin{pmatrix} VIS_{AP_1}(P_1) & VIS_{AP_2}(P_1) & \dots & VIS_{AP_I}(P_1) \\ VIS_{AP_1}(P_2) & VIS_{AP_2}(P_2) & \dots & VIS_{AP_I}(P_2) \\ \vdots & \vdots & & \vdots \\ VIS_{AP_1}(P_J) & VIS_{AP_2}(P_J) & \dots & VIS_{AP_I}(P_J) \end{pmatrix} \tag{5.3}$$

- **Automatic environment partition**: The environment is automatically divided into zones using the K-Means clustering algorithm [MacQueen, 1967] and the Caliński-Harabasz Index [Caliński and Harabasz, 1974] over $VIS_{TRAINDATA}$. Figure 5.6(a) depicts the flow diagram of the procedure. The environment is iteratively divided into zones, in a hierarchical partition that can be represented by a tree (Figure 5.6(b)). Each zone Z_K is divided into K new sub-zones through K-Means, being the value of K determined by the Caliński-Harabasz Index. A zone is no further divided if it has less than 10 positions. This threshold has been experimentally selected.



(a) Partition flow diagram.

(b) Environment division tree.

Figure 5.6: Hierarchical approach: Partition procedure.

- **Zone classifiers training**: Once the environment is divided into hierarchical zones, a classifier is built over the train data with the aim of distinguishing between the different zones belonging to the same level (squares in Figure 5.6(b)). Four classification algorithms (FDT, KNN, FURIA and SVM) have been tested.

- **Positions classifiers training**: Another classifier is trained for each zone in the lowest level of the tree branches. These classifiers find the closest topological position to the current location among all the positions belonging to the lower zones (circles in Figure 5.6(b)). Again, FDT, KNN, FURIA and SVM have been used.

5.2.2 Localization stage

In this stage, the WiFi device will estimate its current position using the RSS from all the APs. The set of classifiers trained in the previous stage are now used to hierarchically localize the device, first in the higher zones and, at the end, determining the position of the device in the lower ones. The localization stage comprises two steps as showed in Figure 5.5:

- **Measurement**: Using a WiFi device, 4 RSS samples are collected from every AP and an averaged sample is generated. This value has been selected from the experimental analysis carried out in Chapter 4 (page 39).

- **Classification**: The averaged sample is classified through the different levels of the hierarchy previously built in the training stage. Starting from the first level of zone classifiers, the system finds out the zone the sample belongs to. Then, the sample is classified again using the second level of classifiers corresponding to the zone previously identified. This procedure continues until the lowest level in the tree branch is reached. At the end, the estimated position of the WiFi device is determined by the position classifier associated to the lowest zone identified in the previous steps.

5.2.3 Learning algorithms

This section provides a brief revision of the algorithms used to test the hierarchical approach.

5.2.3.1 Environment division

K-Means clustering algorithm [MacQueen, 1967] along with the Caliński-Harabasz Index [Caliński and Harabasz, 1974], also known as *Variance Ratio Criterion (VRC)*, is used to obtain the hierarchical partition of the environment. The objective is to create a partition of the environment maximizing the intra-cluster similarity.

K-Means [MacQueen, 1967] follows a simple way to assign the samples of a given dataset through a certain number of clusters (K clusters) fixed a priori. The objective is to define K centroids, one for each cluster. First, these centroids are placed as far away as possible from each other. Next, each sample is associated to the nearest centroid and the K centroids are re-calculated as the barycentre of the clusters resulting from the previous step. This procedure is repeated until the K centroids do not change their location anymore.

Setting the right number of clusters becomes a key task. To do so, the VRC performs a quantitative evaluation of clusters with the aim of finding out the right K. For a solution with N observations and K clusters, VRC is calculated as described in Equation 5.4,

$$VRC_K = \frac{BGSS}{WGSS}\frac{N-K}{K-1} \tag{5.4}$$

where $BGSS$ (Between-Group Sum of Squares) measures the dispersion of the clusters between each other (Equation 5.5) and $WGSS$ (Within-Group Sum of Squares) measures the dispersion within each cluster (Equation 5.6). Compact and well-separated clusters within the feature space are expected to have small values of $WGSS$ and large values of $BGSS$. As a consequence, the better the data partition, the greater the ratio between $BGSS$ and $WGSS$. The normalization term $(N-K)/(K-1)$ prevents this ratio to increase monotonically with the number of clusters, making VRC a maximization criterion with respect to the number of clusters K.

$$BGSS = \sum_{k=1}^{K} n_k \left\| G^{\{k\}} - G \right\|^2 \tag{5.5}$$

where n_k is the number of observations belonging to the cluster k, $G^{\{k\}}$ is defined as the dispersion of the barycentre of each cluster, calculated as the mean of all the variables within the cluster k and G is the barycentre of the whole set of data.

$$WGSS = \sum_{k=1}^{K} \sum_{d \in D_k} \left\| M_d^{\{k\}} - G^{\{k\}} \right\|^2 \tag{5.6}$$

where $d \in D_k$ are the indices of the observations belonging to the cluster k, $M_d^{\{k\}}$ is the sample d belonging to the cluster k and $G^{\{k\}}$ is defined as the dispersion of the barycentre of each cluster, as explained before.

5.2.3.2 Classification

As explained in Sections 5.2.1 and 5.2.2, four different classifiers (FDT, FURIA, KNN and SVM) based on three different kinds of algorithms (rule induction, instance-based and kernel-based) have been tested to classify the RSS measures into zones at each level and positions at the lowest level of the hierarchy.

Rule induction classifiers have been proved as a powerful tool to deal with noisy data [Hühn and Hüllermeier, 2010]. The previously explained Fuzzy Decision Trees (FDT) (Chapter 4, page 39) and Fuzzy Unordered Rule Induction Algorithm (FURIA) [Hühn and Hüllermeier, 2009] have been selected among this kind of algorithms. FURIA is a fuzzy modelling method which extends the well-known RIPPER algorithm [Cohen, 1995], a state-of-the-art rule learner, while preserving its advantages, such as simple rule sets. In addition, it includes some modifications and extensions. In particular, FURIA learns fuzzy rules instead of conventional rules and unordered rule sets instead of rule lists. Moreover, to deal with uncovered examples, it makes use of an efficient rule stretching method.

Among all existent **instance-based** classifiers, K-Nearest Neighbour (KNN) [Kibler and Aha, 1987] is usually used as baseline to compare with WiFi indoor localization systems [Youssef and Agrawala, 2008, Wu et al., 2007]. It is a variation of the nearest neighbour algorithm where the most popular class of the K nearest samples is used for prediction. This prevents a single noisy sample from incorrectly classifying a new one. As the system does not have this information a priori, a popular method is to train and test the system using a variety of K values, and adopting the one that produces the best result. For the following experimentation, the KNN classifier has been configured to use the euclidean distance using one sample ($K = 1$) for the prediction.

Finally, the Support Vector Machine (SVM) [Cortes and Vapnik, 1995] algorithm has been chosen as the most outstanding **kernel-based** classifier. SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space which separates input classes. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. For the following experimentation, the SVM classifier has been configured to use a linear kernel.

5.3 Experimental analysis

This section describes the experiments carried out to validate the hierarchical approach. In addition, a critical discussion of the obtained results is provided.

5.3.1 Experimental set-up

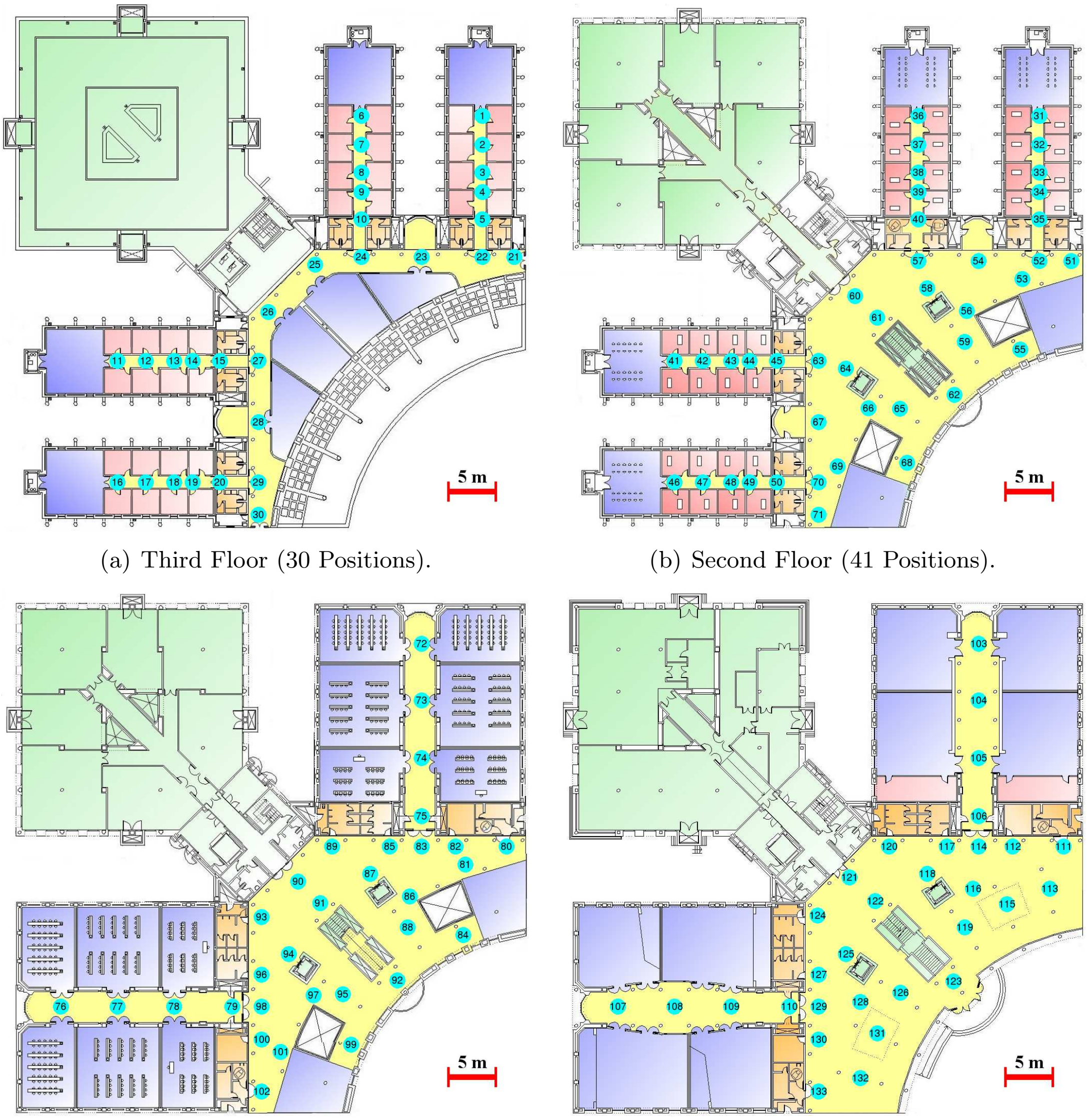
The hierarchical approach has been tested in a complex real-world environment. The experiments have been performed on the West wing of the Polytechnic School at the UAH (Figure 5.7). The environment has four floors with a surface of $3000m^2$ each. In the experiments 216 APs have been detected. They were deployed over the environment with the aim of providing Internet access to the students but disregarding localization purposes. 133 significant topological positions, represented by circled numbers in Figure 5.7, have been considered (distributed over the four floors). In this building, mainly made of concrete, the signal measurement is highly affected by the multipath effect.

With the aim of evaluating the scalability of the proposal two different scenarios have been tested. In the simple scenario, only the third floor (Figure 5.7(a)) has been considered. It represents a relatively small test-bed environment with 30 positions and 105 APs. In the complete scenario, all the floors have been considered. This can be deemed as a large and complex test-bed environment with 133 positions and 216 APs. The system performs the localization with no prior knowledge about the APs physical locations.

The tests have been carried out with a laptop computer using its internal Wireless device acquiring 1 sample per second. Two RSS raw datasets (train and test) have been collected on different days, one week apart, under real conditions. Each raw dataset has 60 samples per position and per AP.

5.3.1.1 Simple scenario. Small test-bed environment

To illustrate the simple scenario division obtained by the proposed system, an environment division tree has been used (Figure 5.8). The horizontal dotted lines show the division between the different levels, the squares represent the zone classifiers and the circles denote the position classifiers, as explained in Section 5.2.1, Figure 5.6(b) (page 62). The number under each circle corresponds to the number of positions in the corresponding zone. Finally, the lines joining the nodes represent the hierarchy between the different zones, showing the number of subzones in which a zone is divided. As can be seen, the scenario has been divided in 3 different levels, obtaining 5 final zones with 3 to 10 positions each.



(a) Third Floor (30 Positions).

(b) Second Floor (41 Positions).

(c) First Floor (31 Positions).

(d) Ground Floor (31 Positions).

Figure 5.7: UAH test-bed environment.

After dividing the environment, the zone and position classifiers are trained for each zone. Then, the test data is classified through the different levels until an inferred position is obtained for each sample.

Table 5.1 summarizes the results achieved in the simple scenario when considering the different algorithms. The results labelled as “Single Classifier” are the results in the case of classifying all the positions without dividing the environment. The column entitled as “Hierarchical Classification” shows the results achieved after applying the proposed hierarchical approach (all the three levels). SCAL stands for “Same Classifier for All Levels” and it reports the accuracy using the same classifier (FDT, FURIA, KNN or SVM)

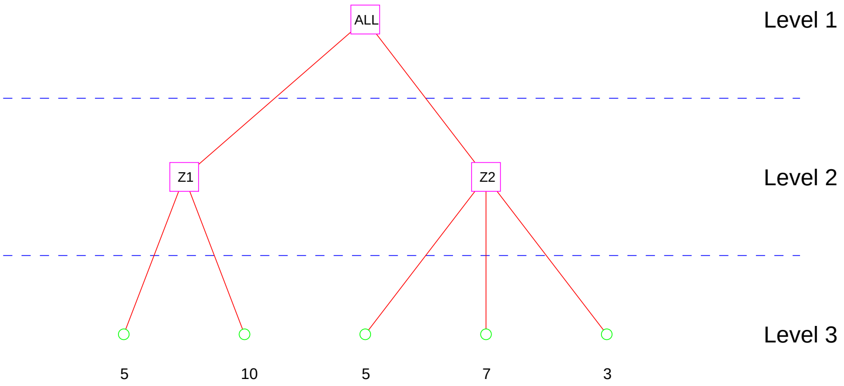


Figure 5.8: Simple scenario division (30 positions in the 3^{rd} floor).

at every level. ZCKNN means "Zone Classifiers using KNN" and it reports the accuracy using KNN in all the zone classifiers and FDT, FURIA, KNN or SVM (the one appearing in the first column) in the position classifiers at the lowest level of the hierarchy. The last column, "Improvement", gives the difference between the two previous columns, which is the increase in accuracy as result of applying the proposed hierarchical localization in contrast to the non-hierarchical approach.

Table 5.1: Summary of results in the simple scenario (30 positions in the 3^{rd} floor).

	Single Classifier	Hierarchical Classification	Improvement
FDT	RSS: 70.67%	SCAL: 79.78%	9.11%
		ZCKNN: 80.67%	10%
FURIA	RSS: 58.22%	SCAL: 59.78%	1.56%
		ZCKNN: 64.00%	5.78%
K-NN	RSS: 63.78%	SCAL: 76.00%	12.22%
		ZCKNN: 76.00%	12.22%
SVM	RSS: 65.33%	SCAL: 83.33%	18.00%
		ZCKNN: 84.89%	**19.56%**

Figure 5.9 gives a more detailed view of the results achieved by the hierarchical system at all the levels of the hierarchy. In the graphs on the left side of the figure, the vertical axis represents the accuracy, while in the graphs on the right side it represents the mean error

of the system computed as the mean distance between the estimated positions and the real ones. The horizontal axis represents the number of levels in which the environment is divided in all the graphs. “1 Level” means all the positions have been classified using only one classifier, without dividing the environment, while the maximum number of levels means that the whole hierarchical partition, shown in Figure 5.8, has been used for the classification steps. The results corresponding to the intermediate number of levels are shown just for comparison purposes and are obtained stopping the division process once the corresponding level is reached.

Two pair of graphs are plotted, each one focusing on the different configurations. On the one hand, the two pictures on top (5.9(a) and 5.9(b)) depict the results using the same classifier (FDT, FURIA, KNN or SVM) at every level of the classification hierarchy. On the other hand, the two pictures at the bottom (5.9(c) and 5.9(d)) show the results using KNN in all the zone classifiers and FDT, FURIA, KNN or SVM at the last classification level (position classifiers).

As can be seen in the graphs on the left side of Figure 5.9, accuracy increases with the number of levels. Accuracy remains almost the same with the first division of the environment into two levels, but it significantly increases with the next hierarchical partition (three levels) for the KNN and SVM classifiers. Just the opposite happens when using the fuzzy classifiers FDT and FURIA, the accuracy significantly increases after the first division, while it remains almost the same (it even decreases using the FURIA classifier) with the next partition. No matter the selected classification technique, adopting the hierarchical approach yields to an improvement of accuracy versus the “1 Level”. FDT, KNN and especially SVM significantly increase accuracy thanks to the hierarchical approach and clearly overcome FURIA. Using the FDT classifier provides the highest initial accuracy, but SVM overcomes its results when the classification is performed using the hierarchical approach.

Looking at the graphs on the right side of Figure 5.9, it can be seen that the mean error is reduced for all the configurations (between 33% and 62%) using the hierarchical approach.

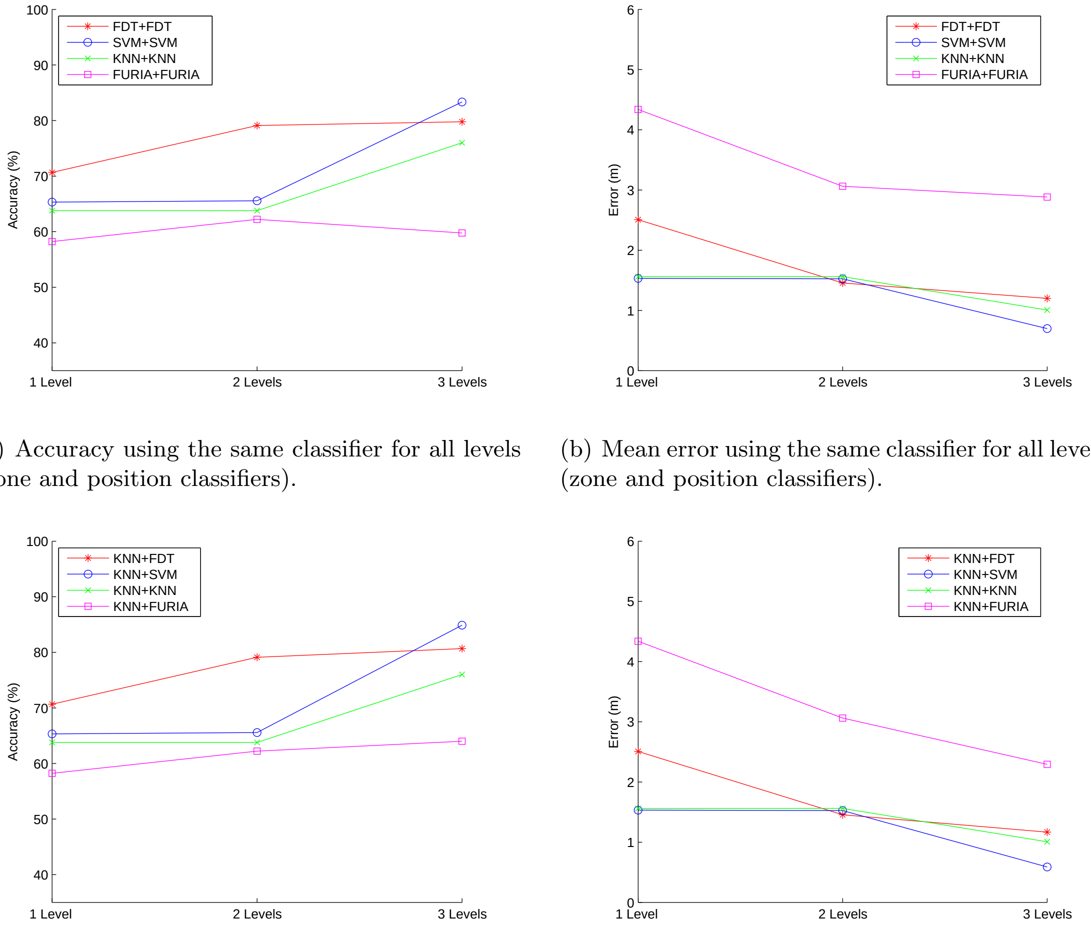
Using KNN in the zone classifiers (Figure 5.9(c) and 5.9(d)), slightly improves the results for all the proposed classifiers (FDT, FURIA and SVM).

The following conclusions can be drawn after looking at Figure 5.9 and Table 5.1:

- The hierarchical classification approach clearly overcomes the single classifier approach. In all cases there is an improvement in the accuracy. Moreover, no matter the selected classification algorithm (FDT, FURIA, KNN, or SVM) there is always at least one configuration yielding a minimum improvement of 5.78% (Table 5.1).

- With respect to FDT, FURIA and SVM, the accuracy increases using KNN in all zone classifiers and FDT, FURIA or SVM only at the lowest level of the hierarchy (position classifiers) versus using FDT, FURIA or SVM in all the classifiers. Such behaviour was expected since the environment was divided into zones using K-Means, the “equivalent” clustering algorithm to KNN.

- The hierarchical approach reduces the mean error of the system at least a 33%, even in the case when this improvement is not reflected in the accuracy. Using



(a) Accuracy using the same classifier for all levels (zone and position classifiers).

(b) Mean error using the same classifier for all levels (zone and position classifiers).

(c) Accuracy using K-NN in the zone classifiers.

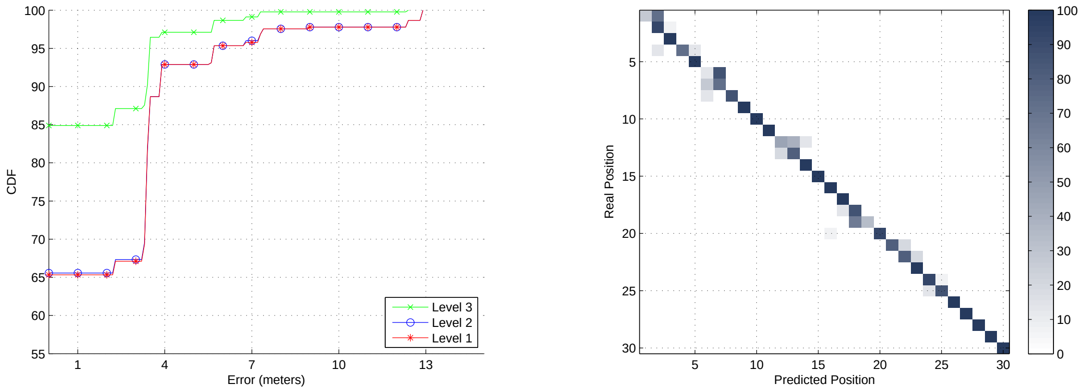
(d) Mean error using K-NN in the zone classifiers.

Figure 5.9: Accuracy and mean error results in the simple scenario (30 positions in the 3^{rd} floor).

FURIA at all the levels of the hierarchy gets an improvement of only 1.56%, but the mean error is reduced from 4.33 to 2.88 metres. This means that thanks to the hierarchical approach the number of classification errors is reduced, but also that the misclassifications occur within closer positions.

- The lowest error (0.58 metres) and the highest accuracy (84.89%) and improvement (19.56%) are obtained using KNN in all zone classifiers and SVM only at the lowest level of the hierarchy (position classifiers).

To fully understand the obtained results, Figure 5.10 shows the *Cumulative Distribution Function (CDF)* along with the confusion matrix for the configuration providing the highest accuracy. The CDF (Figure 5.10(a)) shows an analysis of the distance to the real positions in the different levels of the hierarchical system. As can be seen, the error decreases as the number of levels in the hierarchy increases, obtaining an error under 4 metres for the 95th percentile. The confusion matrix (Figure 5.10(b)) details the predicted positions by the system related to the groundtruth (the positions where the device really was). Looking at the figure, it can be seen that most of the classification errors occur within the nearest positions. Notice that, since a topology-based indoor localization is performed, the minimum error in distance depends on the minimum distance between the topological positions (2.25 metres in this scenario).



(a) CDF.

(b) Confusion Matrix.

Figure 5.10: CDF and confusion matrix using ZCKNN with SVM. Simple scenario (30 positions in the 3rd floor).

The mean distance to the real positions is 3.90 metres for the misclassified samples and 0.58 metres taking into account all samples (both correctly and incorrectly classified), compared to 4.42 and 1.53 metres respectively obtained without applying the hierarchical approach.

5.3.1.2 Complete scenario. Large test-bed environment

Figure 5.11 illustrates the environment division tree obtained by the proposed hierarchical localization approach in the complete scenario (all the four floors depicted in Figure 5.7). The horizontal dotted lines show the division between the different levels, the squares represent the zone classifiers and the circles denote the position classifiers, as explained in Section 5.2.1, Figure 5.6(b) (page 62). The numbers under the circles are the number of positions in the corresponding zone. Finally, the lines joining the nodes represent the hierarchy between the different zones, showing the number of subzones in which each zone is divided. As can be seen, the scenario has been divided into 5 different levels, obtaining 24 position zones with 2 to 9 positions each.

After dividing the environment, all the zone and position classifiers are trained. Then, the test data are classified through the different levels until an inferred position is obtained for each sample.

Table 5.2 summarizes the accuracy results achieved in the complete scenario using the different algorithms. The format of this table is the same than the one described for Table

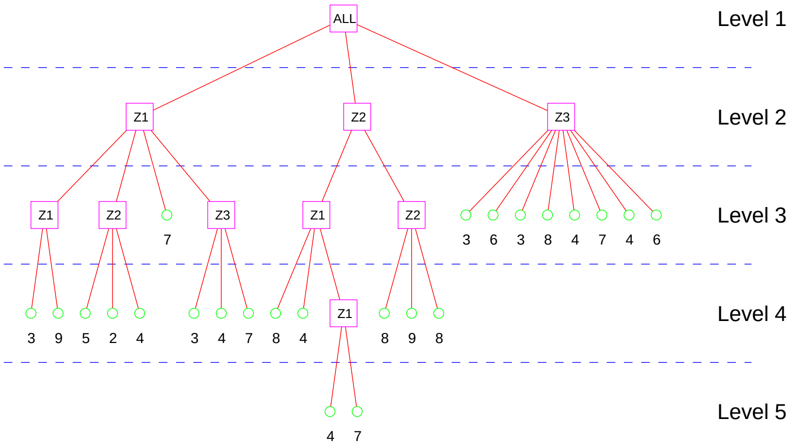


Figure 5.11: Complete scenario division (133 positions).

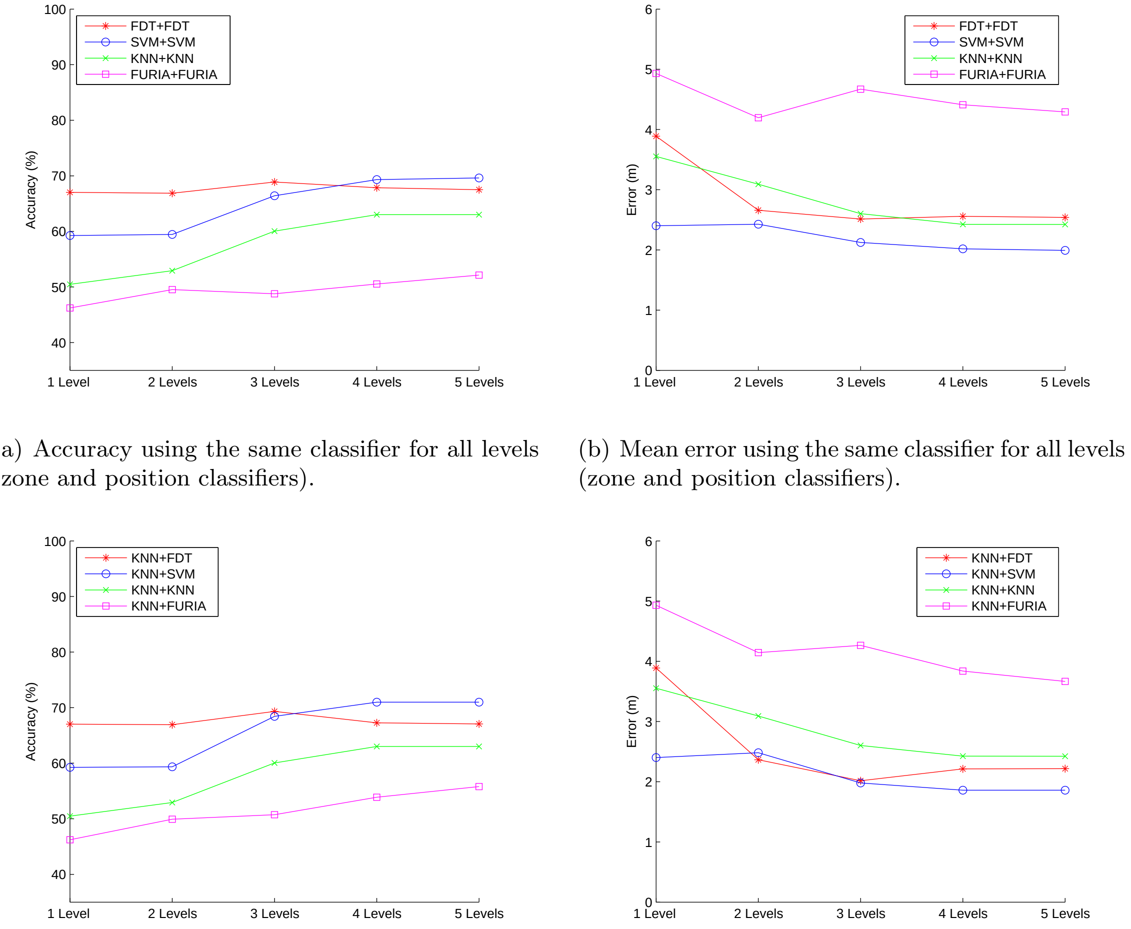
5.1 in the case of the simple scenario. The first column determines the base classification algorithm to be considered (FDT, FURIA, KNN or SVM). The second column, “Single Classifier”, shows the results when the environment is not divided. In the third column, “Hierarchical Classification”, the results provided by the proposed hierarchical approach are shown. SCAL reports the accuracy using the same classifier (FDT, FURIA, KNN or SVM) at every classification level, while ZCKNN reports the accuracy using KNN in all the zone classifiers and FDT, FURIA, KNN or SVM (the one appearing in the first column) in the position classifiers at the lowest level of the hierarchy. The last column, “Improvement”, gives the increase in accuracy as result of applying the proposed hierarchical localization approach in contrast to the non-hierarchical one. It is computed as the difference between the two previous columns.

Table 5.2: Summary of results in the complete scenario (133 positions).

	Single Classifier	Hierarchical Classification	Improvement
FDT	67.02%	SCAL: 67.49%	0.47%
		ZCKNN: 67.07%	0.05%
FURIA	46.22%	SCAL: 52.13%	5.91%
		ZCKNN: 55.79%	9.57%
K-NN	50.48%	SCAL: 63.01%	12.53%
		ZCKNN: 63.01%	12.53%
SVM	59.25%	SCAL: 69.62%	10.37%
		ZCKNN: 70.98%	**11.73%**

Figure 5.12 gives a more detailed view of the results. The format of this figure is the same as the one described for the simple scenario. The horizontal axis represents the number of levels in which the environment is divided ("1 Level" means the environment has not been divided while "5 levels" means that the environment has been fully divided following the proposed hierarchical approach). In the graphs on the left side of the figure, the vertical axis represents the accuracy, while in the graphs on the right side it represents the mean error of the localization system.

Two pair of graphs are plotted, each one focusing on the different configurations. On the one hand, the two pictures on top (5.12(a) and 5.12(b)) depict the results using the same classifier (FDT, FURIA, KNN or SVM) at every level of the classification hierarchy. On the other hand, the two pictures at the bottom (5.12(c) and 5.12(d)) show the results using KNN in all the zone classifiers and FDT, FURIA, KNN or SVM at the last classification level (position classifiers).



(a) Accuracy using the same classifier for all levels (zone and position classifiers).

(b) Mean error using the same classifier for all levels (zone and position classifiers).

(c) Accuracy using K-NN in the zone classifiers.

(d) Mean error using K-NN in the zone classifiers.

Figure 5.12: Accuracy and mean error results in the complete scenario (133 positions).

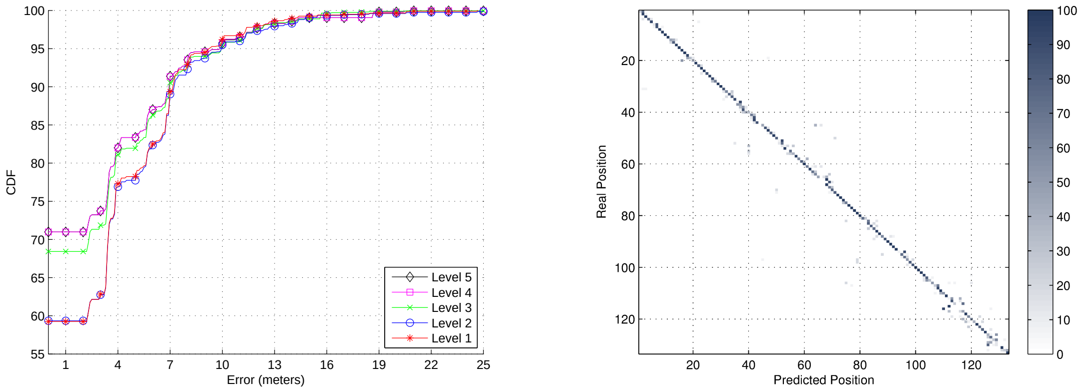
The following conclusions can be drawn after analysing the results in Table 5.2 and Figure 5.12:

- The hierarchical approach always overcomes the non-hierarchical one. Even though using the FDT classifier there is almost no improvement in terms of accuracy, the mean error is reduced from 3.89 to 2.42 metres using FDT in all levels of the hierarchy, and from 3.89 to 2.21 metres using KNN in the zone classifiers and FDT only at the lowest level of the hierarchy.

- Results are slightly better when using KNN in all zone classifiers and FURIA or SVM only at the lowest level of the hierarchy versus using FURIA or SVM in all the classifiers, and slightly worse when using KNN in all zone classifiers and FDT only at the lowest level of the hierarchy versus using FDT in all the classifiers.

- The lowest mean error (1.86 metres) and the highest accuracy (70.98%) and improvement (11.73%) are obtained by using KNN in all zone classifiers and SVM only at the lowest level of the hierarchy (position classifiers).

Figure 5.13 shows the CDF along with the confusion matrix for the configuration providing the highest accuracy. As can be seen looking at the CDF (Figure 5.13(a)), the distance to the real position decreases as the number of levels in the hierarchy increases, obtaining an error under 9 metres for the 95^{th} percentile. On the other hand, the confusion matrix (Figure 5.13(b)) shows that, although the distance error seems to be high, most of the classification errors occur within the nearest positions. It is important to highlight that, since a topology-based indoor localization is performed, the distance error depends on the minimum distance between the topological positions (2.25 metres in this scenario).



(a) CDF.

(b) Confusion Matrix.

Figure 5.13: CDF and confusion matrix using ZCKNN with SVM. Complete scenario (133 positions).

The mean distance to the real position is 5.89 metres for the misclassified samples and 1.86 metres taking into account all (both correctly and incorrectly classified) samples, compared to 6.41 and 2.40 metres respectively obtained without applying the hierarchical approach.

It is important to remark that this scenario is larger than the previous one. In the simple scenario there are only 30 positions placed in the same floor while in the complete scenario there are 133 positions distributed over four different floors. However, even though the complexity of the problem has been increased (the number of positions is more than four times bigger), the hierarchical approach is still able to yield good results, achieving an accuracy close to 70%. This fact proves that the proposed hierarchical WiFi-based localization system works properly in large environments.

Figure 5.14 shows a comparison of the accuracy and mean error variation with the number of positions using the hierarchical approach versus using a single classifier (no environment division).

5.4 Conclusions

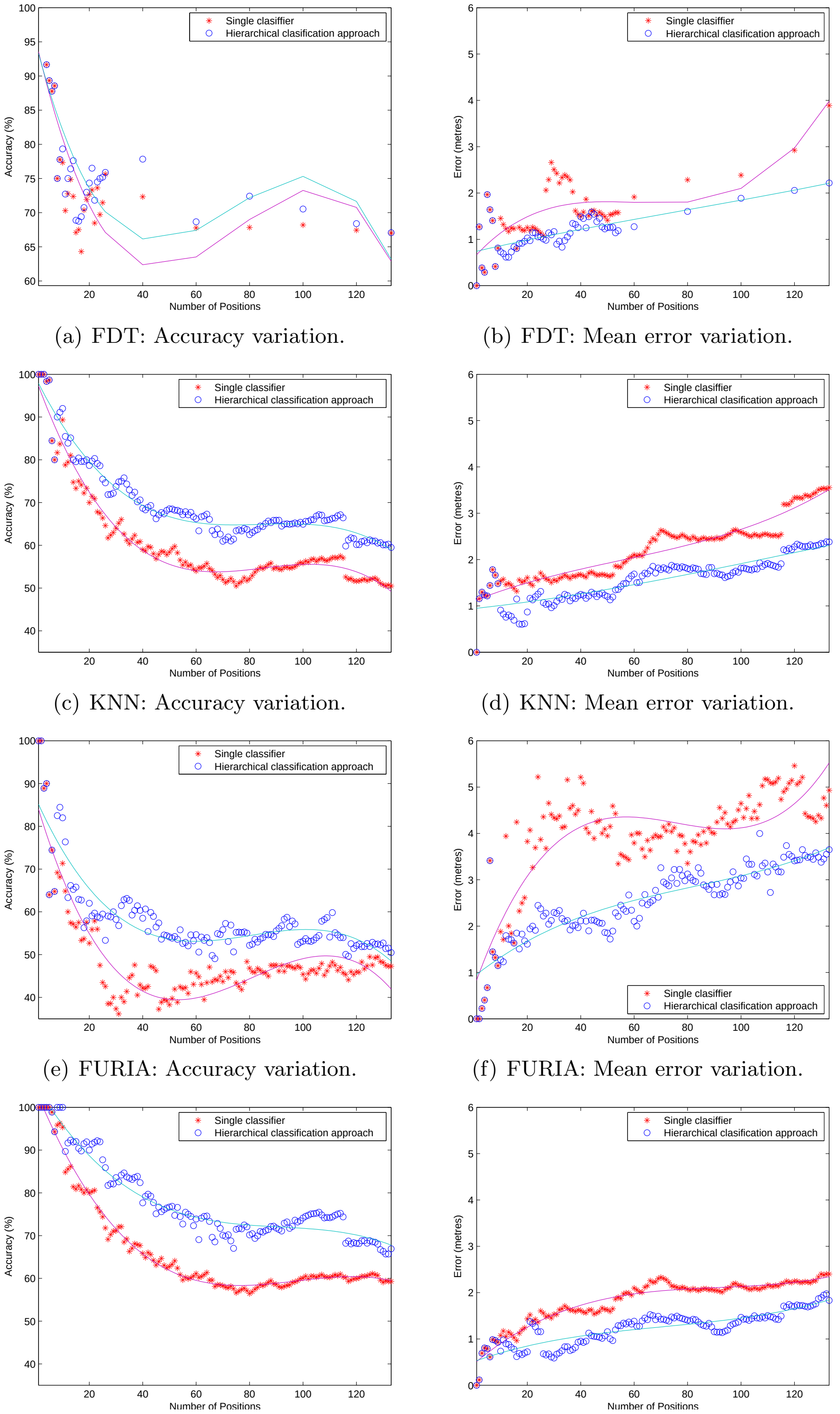
In this chapter, the effect of applying a WiFi localization system in a large environment crowded with APs has been analysed. A hierarchical division of the environment has been proposed to tackle this problem. The following conclusions can be drawn:

- The hierarchical approach simplifies the classification task, reducing the number of outputs in the first level and the number of inputs and outputs in the following ones. With this approach the loss of accuracy when the number of positions in the environment increases is reduced. As a result, the mean error of the system is also reduced.

- The proposal was tested in a real large environment considering two different scenarios. The first one was a relatively small sized but highly illustrative scenario, while the complete scenario was a much larger environment. The aim of using two different scenarios was to show how thanks to the proposed hierarchical approach the localization system was able to improve the results no matter the size of the test-bed environment under consideration.

- The best results are achieved using the KNN algorithm in the zone classifiers. This behaviour is expected since the environment was divided into zones using K-Means, the "equivalent" clustering algorithm to KNN. The highest accuracy and lowest mean error is achieved using the SVM algorithm in the position classifiers (70.98%, 1.86 metres). FDT (67.07%, 2.01 metres) and KNN (63.01%, 2.42 metres) algorithms also provide high accuracy and low error, clearly overcoming the FURIA classifier results (55.79%, 3.66 metres).

- These results were obtained at static positions of the environment. Further conclusions can be extracted by using the localization system during an online stage when the device's position is obtained while it is in motion. The final system will have to estimate the position of the device for samples measured during motion, at positions that might be different from the ones on the training set.



(a) FDT: Accuracy variation.

(b) FDT: Mean error variation.

(c) KNN: Accuracy variation.

(d) KNN: Mean error variation.

(e) FURIA: Accuracy variation.

(f) FURIA: Mean error variation.

(g) SVM: Accuracy variation.

(h) SVM: Mean error variation.

Figure 5.14: Accuracy and mean error variation with the number of positions. Single classifier vs. hierarchical approach.

Chapter 6

Recursive Bayesian Estimation of a Moving Device Position

While some indoor LBSs, such as medical equipment location in hospitals or people location in museums, do not need to estimate the device's trajectory at short time intervals, some others, such as people guidance, require an accurate and frequent estimation of the device's position. When providing an LBS for the latter, new challenges arise. Firstly, the maximum acquisition frequency of a WiFi device varies from 0.25 to 4 Hz, being 1 Hz the most frequent frequency in today's devices. This means that if 3 samples are used to estimate a position, a new position will be get every 3 seconds. Moreover, if the device is moving at 1 m/s the first and last samples could be as far as 3 meters apart. Using a topological approach this could mean that the first and last samples are taken at two different positions.

In Chapter 4, an FDT algorithm was proposed to deal with the small scale variations on static positions of an small environment non crowded with APs. Then, in Chapter 5 a hierarchical division of the environment was proposed to tackle with the associated problems of applying WiFi localization systems to large environments, with their APs not deployed for localization purposes. This proposal could be directly applied to LBSs where the localization is performed on static positions of the environment.

In this chapter, a new approach to track the position of a device in motion using a topological radio-map is proposed. This approach uses a Bayes filter that will continuously estimate the most likely position of the device. This filter will have to deal with the low working frequency of the device and the uncertainty of the observation to provide an accurate and fast estimation.

The remaining of the chapter is organised as follows: First, the proposal to track and filter the device's position will be explained in Section 6.1. Then, Section 6.2 will describe the experimental results. Section 6.3 will show a comparison of the performance of the proposed system with a commercial one. Finally, in Section 6.4 a critical discussion about the results will be provided.

6.1 Filtering and tracking the device's position

Recursive Bayesian estimation, also known as Bayes filters, is a general probabilistic approach for estimating an unknown probability density function recursively over time using incoming measurements and a mathematical process model. Essentially, Bayes filters allow to continuously update the most likely position of a device based on the most recently acquired data.

The selection of one of the existent Bayes filters depends on the characteristics and restrictions of the designed localization system [Thrun et al., 2005]. *Kalman Filters (KFs)* implement belief computation for continuous states and they are not applicable to discrete spaces. Particle Filters (PFs) and Grid Localization require the motion model of the device which is not available in the designed WiFi localization system. Finally, Markov localization is the straight forward application of Bayes filters and provides four main models depending of the characteristics of the system. On the one hand, Markov chains and *Markov Decision Processes (MDPs)* are designed for systems were the states are fully observable. On the other hand, *Partially Observable Markov Decision Processes (POMDPs)* and *Hidden Markov Models (HMMs)* are designed for systems were the states are partially observable. In the designed WiFi localization system, the states (positions) are partially observable since they are estimated by the hierarchical localization system. Thus, POMDPs or HMMs should be used to perform the filtering and tracking tasks. POMDPs are used in robotics when a motion model is available and therefore, different known actions can be chosen by the system. HMMs are used when there are no actions, which means that the transitions between positions are triggered when new information is acquired from the available sensors.

Taking into account the previously described characteristics of the filters, an HMM will be used to keep a probability distribution over the positions. The main objective is to filter unlikely transitions between positions when locating a device in motion. To do so, the hierarchical localization approach described in the previous chapter will be used to infer the device's position. A single-sample approximation will be applied due to the low working frequency of WiFi devices. This way, the device's position in the environment will be estimated as the most likely position filtering punctual missclassifications of the hierarchical localization system.

6.1.1 Hidden Markov Models

An HMM [Baum and Petrie, 1966] is a statistical Markov model in which the system being modelled is assumed to be a Markov process with non-observable (hidden) states.

In simpler Markov models (such as Markov chains), the state is directly visible and therefore the probabilities of transitions between the states are the only needed parameters. In an HMM, the states are not directly visible, instead each state produces an output observation with a certain probability. Thus, a probability distribution of the observations over the possible states is needed in order to infer the sequence of states using the known sequence of observations.

The formal definition of an HMM is as follows:

Let $S = S_1, S_2, \ldots, S_{|S|}$ be the state space and $O = O_1, O_2, \ldots, O_{|O|}$ the observation space of the system. The sequence of states is defined as $s = s_1, \dots, s_T$ with $s_t \in S$ and the corresponding observations as $o = o_1, \dots, o_T$ with $o_t \in O$ during a time sequence $t = 1, \dots, T$.

The model is characterized by the complete set of parameters: $\Lambda = \{A, B, \pi\}$.

A is the transition matrix, storing the probability of being at state $S_j \in S$ after being at state $S_i \in S$. The state transition probabilities are independent of time:

$$A = [a_{ij}]\,, \quad a_{ij} = P(s_t = S_j \mid s_{t-1} = S_i)$$

B is the observation vector, storing the probability of observation $O_k \in O$ being produced from the state $S_j \in S$:

$$B = [b_j(O_k)]\,, \quad b_j(O_k) = P(o_t = O_k \mid s_t = S_j)$$

π is the initial probability array:

$$\pi = [\pi_i]\,, \quad \pi_i = P(s_1 = S_i)$$

Two assumptions are made in HMMs. First, the Markov assumption that states that the probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it (the probability of being in a state at time t depends only on the state at time $t-1$).

$$P(z_t \mid s_1 \dots s_{t-1}) = P(s_t \mid s_{t-1})$$

Second, the independence assumption states that the output observation at a time t is dependent only on the current state, so independent of previous observations and states:

$$P(o_t \mid s_1 \dots s_t, o_1 \dots o_{t-1}) = P(o_t \mid s_t)$$

The filtering task using HMMs can be handled using the Viterbi algorithm [Viterbi, 1967]. The Viterbi algorithm is used to calculate a belief of being at each state at a certain time given the history of observations. The Viterbi algorithm is as follows:

The belief for every state S_j in $t = 1$ is initialised as:

$$Bel_j(1) = \pi_j \cdot b_j(o_1)$$

Then, the belief for every state S_j is updated as:

$$Bel_j(t) = Bel_i(t-1) \cdot a_{ij} \cdot b_j(o_t)$$

The state s_T is estimated as the $MAX_{s_j \in S}(Bel(T))$.

6.1.2 Applying HMMs to WiFi localization

In order to apply an HMM during the localization stage, the parameters that characterise the model have to be calculated. In the WiFi localization problem the states S are the defined topological positions in the environment and the observations O are the RSSs from the APs.

The transition matrix A can be approximated by assuming a person can not move from one position to another without go trough the positions in between. This way, the probabilities of transit from one position to the neighbour ones or stay in place are uniformly distributed and the rest of transitions are assumed to be 0.

The observation vector B will be obtained from the results provided by the classifiers. Three different algorithms have been tested: The FDT algorithm proposed in Chapter 4 to avoid small scale variations, the SVM classifier which has been proved as the most reliable classifier when localizing at static positions of the environment and the well-known KNN which is still one of the most common classifiers to perform WiFi indoor localization. The three classifiers have been tested as position classifiers in combination with KNN as zone classifier, following the hierarchical approach explained in Chapter 5.

Since the localization is performed using the previously described hierarchical approach, the probability of being at a certain position, needed to build the observation vector B, is calculated by propagating the probability through the hierarchical tree (Figure 6.1).

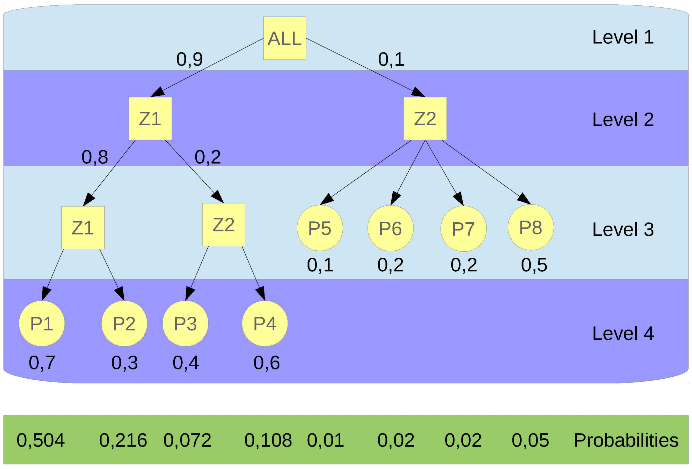


Figure 6.1: Example of the observation vector calculation using the hierarchical approach.

Different classifiers provide different information about the belief of being at a certain position: the FDT algorithm provides an activation degree, the SVM classifier provides a probability, while the KNN algorithm provides a distance. This information has been approximated to probabilities as follows:

- **FDT:** The activation degree of each class is taken as the probability of belonging to that class.
- **SVM:** SVM directly provides probability estimations. To obtain this probability values, the SVM algorithm uses Hastie and Tibshirani's pairwise coupling method [Hastie and Tibshirani, 1998].
- **KNN:** Using the distances from the sample to the radio-map data, the probability of the sample belonging to each class can be calculated as the normalized inverse of the distance.

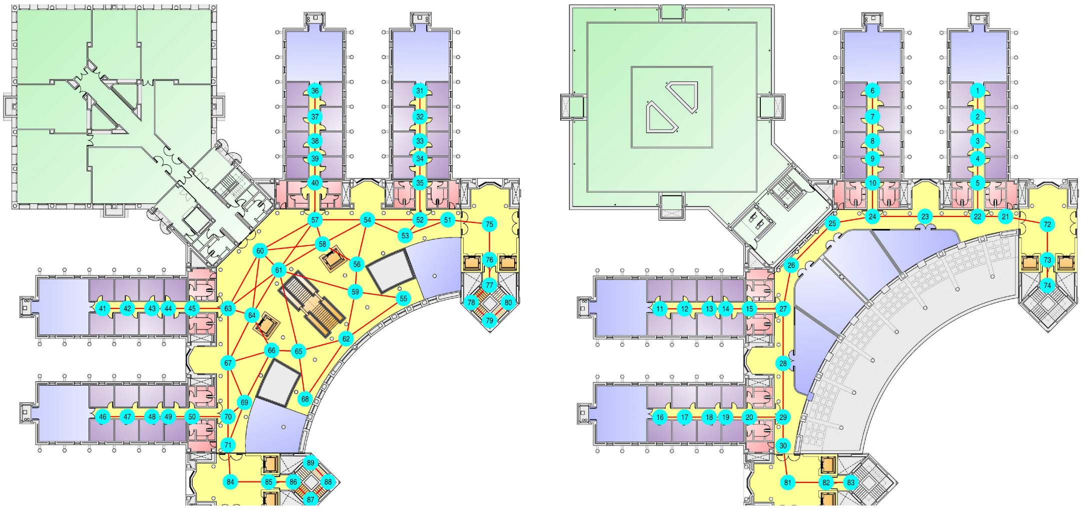
The initial probability vector π is assumed uniformly distributed since the initial position is unknown. To prevent the filter from getting trapped at low probabilities areas, a restarting mechanism was introduced. Due to the low working frequency of the device, if the classifier delivers a few wrong estimations, the actual device position can be further apart from the filter prediction. Even if the classifier starts to deliver correct estimations again, the transition probabilities to the real positions are so low that the filter gets trapped in a low probabilities area. To prevent these situations, if the probability of being at the predicted area remains low for 8 consecutive samples, the filter is restarted to its initial state. Eight samples (equivalent to eight seconds at 1 Hz) were chosen because it is the time required to advance the 2 or 3 positions necessary for the user to move to non-reachable positions.

6.2 Experimental analysis

This section describes the experimental results obtained using the HMM filter. In addition, a critical discussion of the obtained results is provided.

6.2.1 Experimental set-up

The experiments have been performed on the UAH environment described in Section 5.3.1. To allow transitions between different floors new measurements were collected at the stairs areas. Moreover, the physically connected positions are linked to indicate the allowed transitions (Figure 6.2). These connections are used to create the HMM transition matrix.



(a) Second Floor (53 Positions).

(b) Third Floor (36 Positions).

Figure 6.2: UAH test-bed environment: Allowed transitions between positions.

The tests have been carried out with a laptop computer using its internal Wireless device acquiring at its maximum allowed rate (1 sample per second). The RSSs have been collected while a person holding the laptop was following different trajectories over the environment.

Assuming that a person walks at a constant speed of 1 m/s, each trajectory will be composed of samples measured 1 metre apart from each other. This means that most of the samples are measured in positions not covered in the radio-map. It also means that if the positions are 3 metres apart, the filter will have 3 samples to decide. The groundtruth for the trajectories has been manually tagged: when the person steps over an existing position on the radio-map it is marked with the corresponding number of position. Then, the first half of untagged samples between tagged positions are labelled as the previous tagged position, while the second half is labelled as the next one (Figure 6.3). The resulting groundtruth indicates the closest position to the place where the sample is collected.

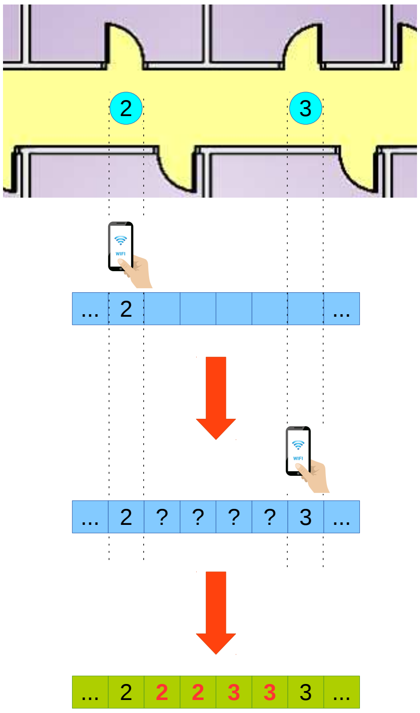


Figure 6.3: Groundtruth generation.

This groundtruth is used to calculate the mean distance error during the trajectories. The distance error is computed as the distance between the estimated position and the target position in the groundtruth. Using this estimation of the distance error, there will be samples between positions with an imprecise estimated distance error. For instance, if a sample was collected between positions 2 and 3 but closer to 3, the estimated error will be 0 metres if the localization system locate the device in position 3 and it will be the distance between positions if the system locates the device in position 2, while the real error should be a fraction of the distance between them. Nevertheless, these variations in the estimated error should be cancelled out during a long trajectory.

6.2.2 Experimental results

This section describes the results of different experiments locating a WiFi device in motion. Two kind of trajectories have been considered: Trajectories in the same floor (one-floor trajectories) and trajectories covering two floors (multi-floor trajectories).

6.2.2.1 One-floor trajectories

The results shown in this section were obtained in the third floor of the UAH environment. Two different illustrative trajectories have been collected covering all the positions in the third floor to test the behaviour of the system using the hierarchical localization approach in combination with the HMM.

On the first trajectory, a user was walking along the main, second and fourth corridors in a path of approximately 120 m at a mean speed of 0.73 m/s. Figure 6.4 shows the trajectory followed during the first experiment. This trajectory starts in the position marked with a blue circle, and continues along the green and yellow dots path to the final position marked with a red circle. Green circles indicate the groundtruth positions, while the yellow dots are places were there are available measurements not corresponding to a position in the radio-map. Finally, the semi-transparent blue circle over the starting position represents the mean distance error of the best classifier for the trajectory.

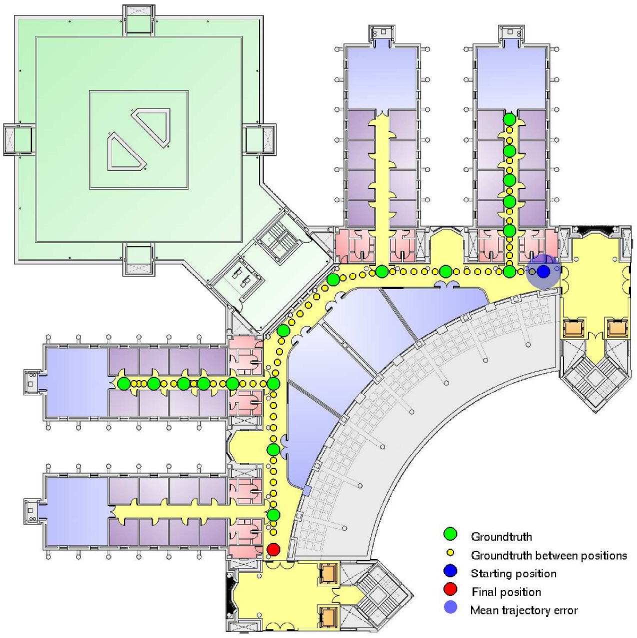


Figure 6.4: One-floor trajectory 1.

Table 6.1 summarizes the results obtained with the different classifiers for the first trajectory with and without using the HMM filter. As can be seen, the mean distance error decreases using the HMM for all the classifiers.

The following conclusions can be drawn from the information in Table 6.1:

- The best results are obtained using the FDT classifier. Using this classifier the mean error is reduced more than a 50%.
- HMM does not significantly improve results using SVM.

Table 6.1: Summary of results during the one-floor trajectory 1.

	Mean distance error		Error reduction
	No filter	HMM filter	
FDT	4.11 m	**1.91 m**	53.53%
SVM	2.72 m	2.62 m	3.68%
KNN	4.35 m	2.30 m	47.13%

On the second trajectory, the user was walking along the main, first and third corridors in a path of approximately 100 m at a mean speed of 0.73 m/s. This trajectory followed the opposite direction to the previous one along the main corridor. Figure 6.5 shows the second trajectory with the same format as in Figure 6.5.

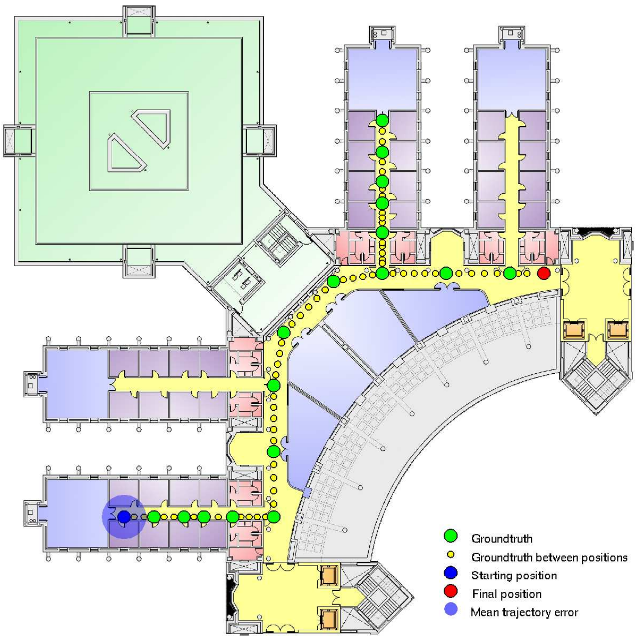


Figure 6.5: One-floor trajectory 2.

Table 6.2 summarizes the results obtained with the different classifiers during the second trajectory. As can be seen, results are slightly worse than the ones obtained for the first trajectory. This could be related to the fact that most of the measurements to build the radio-map were collected while standing on the opposite direction to this trajectory. In this way, the FDT classifier seems to adapt better to differences in orientation between the trajectory collected samples and the radio-map than KNN or SVM.

The following conclusions can be drawn from the information in Table 6.1:

- Again, the best results are obtained using the FDT classifier. Using this classifier the mean error is reduced about 40%.
- Using the KNN algorithm the mean error is slightly worse, especially since the results without using the filter are much worse than the ones provided by FDT.
- Again, HMM does not significantly improve results using SVM.

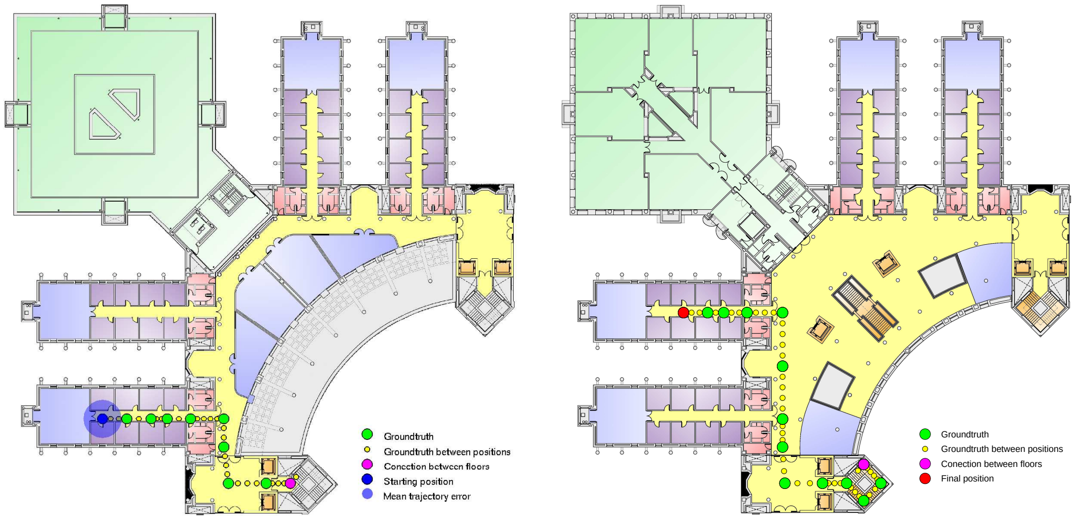
Table 6.2: Summary of results during the one-floor trajectory 2.

	Mean distance error		Error reduction
	No filter	HMM filter	
FDT	3.85 m	**2.38 m**	38.18%
SVM	3.83 m	3.77 m	1.57%
KNN	5.91 m	3.04 m	48.56%

6.2.2.2 Multi-floor trajectories

The results shown in this section were obtained in the second and third floors of the UAH environment. Two different illustrative trajectories including floor transitions have been collected to test the performance of the system. The mean error in the position estimation is expected to be higher in the second floor because the topological positions are further apart than in the third floor. In addition, the central hall of the second floor is a wide open space in which missclassifications between neighbouring positions are more likely to happen. The transitions between floors are also challenging because the vertically aligned positions at the beginning and end of the stairs have very similar RSSs. Despite of all these challenges, the system is still able to correctly follow the trajectories and differentiate between floors.

The first experiment shows the localization during a multi-floor trajectory starting in the third floor. In this experiment the user walked for approximately 90 m with a floor change. The average speed during this trajectory was 0.83 m/s. Figure 6.6 shows the multi-floor trajectory with the third floor on the left and the second floor on the right. The positions that connect the second and the third floors are indicated with magenta circles.



(a) Third Floor.

(b) Second Floor.

Figure 6.6: Multi-floor trajectory 1.

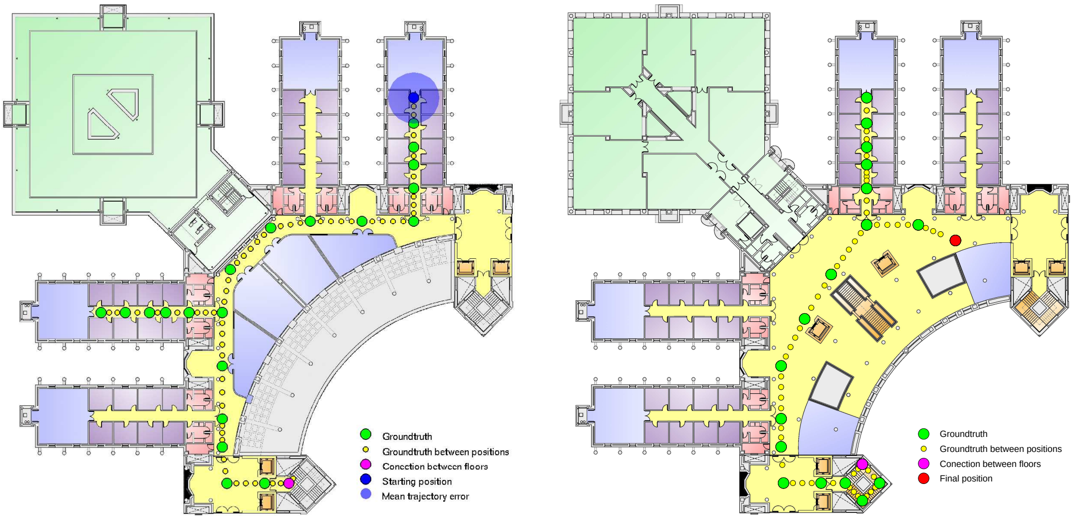
Table 6.3 summarizes the results obtained with the different classifiers during the first multi-floor trajectory. As can be seen, the filter is able to reduce the mean error for the FDT classifier, while it remains almost the same with SVM and it is even increased using the KNN. In this experiment, the filter was not able to improve the KNN algorithm results because during a few consecutive samples the classifier estimated non-reachable positions and the filter got lost. Then, it took a few samples for the filter to recover.

Table 6.3: Summary of results during the multi-floor trajectory 1.

	Mean distance error		Error reduction
	No filter	HMM filter	
FDT	4.06 m	**2.52 m**	37.93%
SVM	5.49 m	4.80 m	12.57%
KNN	4.47 m	5.57 m	-24.61%

On the second experiment, the localization was also performed for a multi-floor trajectory starting in the third floor. In this trajectory the user walked along an approximately 220 m path with one floor change. The average speed during this trajectory was 0.83 m/s. Figure 6.7 shows the second multi-floor trajectory with the same format as in the previous figures.

Table 6.4 summarizes the results obtained with the different classifiers during the



(a) Third Floor.

(b) Second Floor.

Figure 6.7: Multi-floor trajectory 2.

multi-floor trajectory using the HMM versus the hierarchical approach without using the filter. In this experiment, the best results are achieved by the KNN algorithm but results obtained by FDT and SVM are just slightly worse. Even though in this experiment the lower mean error is achieved by using the KNN algorithm, in general the FDT hierarchical system seems to be the most reliable classifier. Again, the mean distance error is reduced using the HMM filter no matter the selected classifier.

Table 6.4: Summary of results during the multi-floor trajectory 2.

	Mean distance error		Error reduction
	No filter	HMM filter	
FDT	5.64 m	3.82 m	32.27%
SVM	3.89 m	3.83 m	1.54%
KNN	3.94 m	**3.38 m**	14.21%

During both multi-floor trajectories no between-floor missclassifications were produced outside the stairs areas. Finally, as in the one-floor trajectories, SVM algorithm mean error is slightly improved using the HMM, while the largest improvements are always provided by FDT.

6.3 Comparative with a commercial system

The performance of the proposed system has been compared with indoor Google Maps localization application. Indoor Google Maps localization has been used to locate the device during the multi-floor trajectory 1. Indoor Google Maps [Indoor Google Maps, 2014] performs localization using GPS, WiFi and GSM information collected inside buildings, but the highest accuracy is obtained when the localization is performed using only WiFi and GSM information. Indoor Google Maps localization is available in more than 10000 buildings around the world, being the Polytechnic School of the UAH one of them. There is no available information about how Google performs the localization, but they have a public application to upload new maps and information collected while site-surveying the environment following their instructions.

To compare the performance of both systems the multi-floor trajectory 1 described in the previous section was followed. Figure 6.8 shows some screenshots of the indoor Google Maps localization application showing the predicted position with a blue circle (using only the WiFi and GSM information). The sequence starts in the top left image and continues row by row from left to right ending in the bottom right image. Figure 6.9 shows the screenshots of the WiFi indoor localization system proposed in this thesis obtained at the same time as the previous ones. The images of this sequence also include the groundtruth position at each time. In this sequence, the predicted position is represented with a red circle and the groundtruth position is represented with a green circle. The images where there is only a green circle represent an estimated position equal to the groundtruth.

As can be seen, the biggest problem using indoor Google Maps localization is the differentiation between floors. The estimated floor is represented by a blue circle under the number of the floor while the groundtruth floor is the one with the grey background. It can be seen that most of the times the floor is wrongly predicted. Forgetting about the floor prediction, and considering it is always correctly predicted, the groundtruth position would always be inside the uncertainty area represented by the transparent blue circle. However, even in this situation, the mean error distance is higher than the obtained with the system proposed in this thesis. Both systems seems to perform better in the corridors while the error increases in the big open area corresponding to the central hall of the second floor. This experiment was repeated using the indoor Google Maps localization including the GPS information along with the WiFi and GSM information previously used. In this case the predicted position tends to be in places outside the building worsening the localization accuracy.

6.4 Conclusions

In this chapter, an HMM algorithm which performs robust global localization using the hierarchical localization system proposed in Chapter 5 was presented. Three different algorithms have been tested as classifiers for the hierarchical localization: The FDT algorithm proposed in Chapter 4 to avoid small scale variations, the well-known KNN which is still one of the most common classifiers to perform WiFi indoor localization, and the SVM classifier which has been proved as the most reliable classifier when localizing at static positions of the environment. The main conclusions that can be drawn from this chapter are as follows:

- As expected, the FDT algorithm is the best adapting to the conditions of a device in motion, dealing with the associated noise and with measurements collected in positions not covered in the radio-map. In consequence, the FDT algorithm improved by the HMM filter provides the best and more reliable localization getting mean errors from 1.91 to 3.82 metres.

- The HMM filtering reduces the mean error in most of the experiments. However, due to the low measurement frequency (1 sample/second) when the localization system wrongly estimates a position during a few consecutive samples, the filter is not able to maintain the position over the trajectory and it gets lost. Moreover, also caused by the low measurement frequency, the device only remains at the same position for 2-4 samples providing the filter with a very low number of samples to correct the position.

- The mean error is higher in open areas (such as the central hall on the second floor) because the localization system is less accurate in these areas, and because the topological positions are further apart increasing the minimum error in missclassifications. Moreover, in these open areas there are usually more allowed transitions than in narrow areas, making the estimation more complex. Nevertheless, the system is able to estimate the position of the device with enough precision to locate and guide a person moving around the environment.

- The system is able to correctly perform floor transitions using the stairs, being the multi-floor localization one of the most challenging problems to deal in WiFi indoor localization.

- The obtained results are encouraging. Although there is not enough information about Google localization system to carry out a proper comparison, the proposed localization system outperforms the commercial localization system provided by Google.

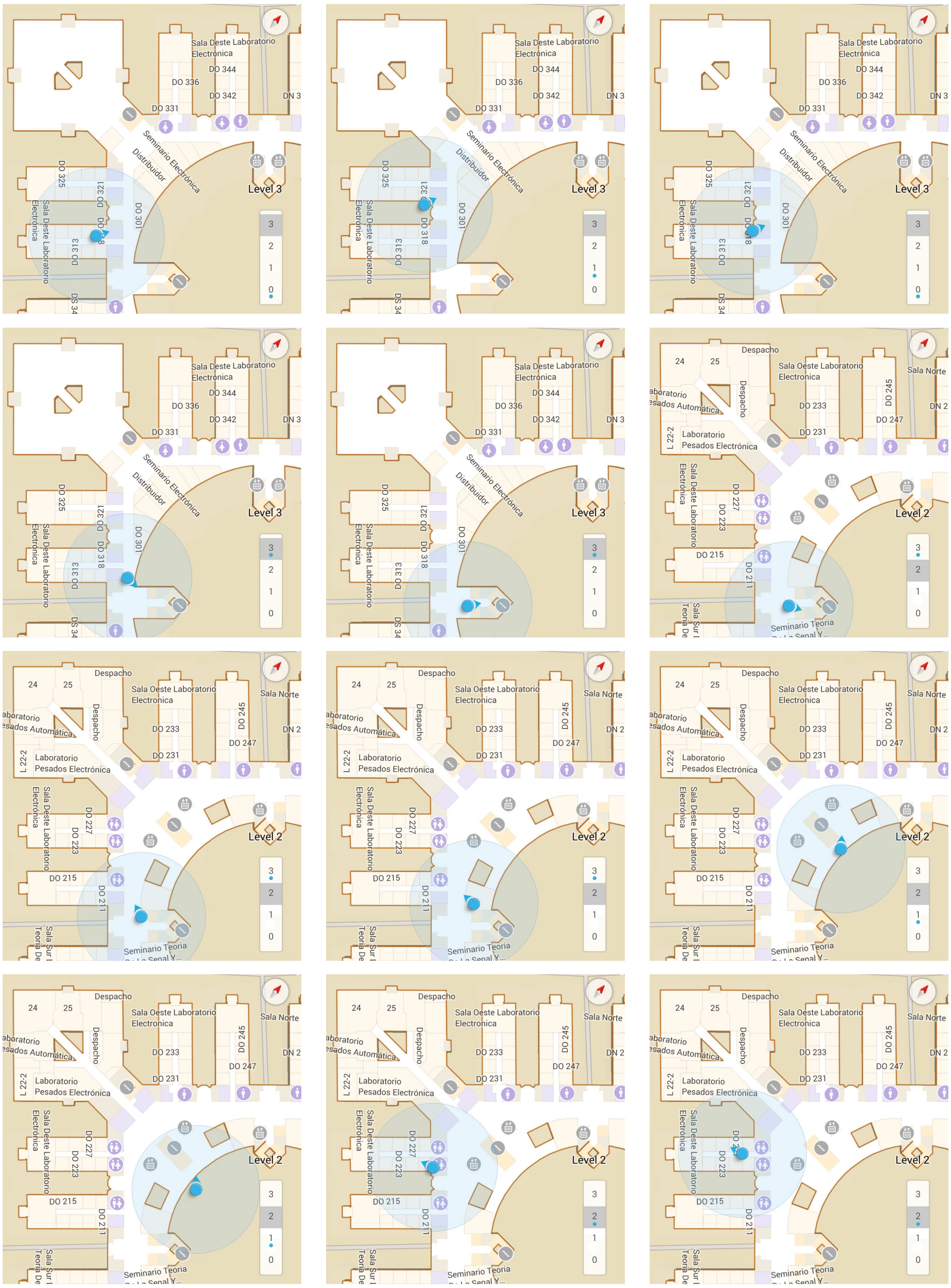


Figure 6.8: Indoor Google Maps localization sequence during the multi-floor trajectory 1. The sequence starts on the top left image and continues from left to right.

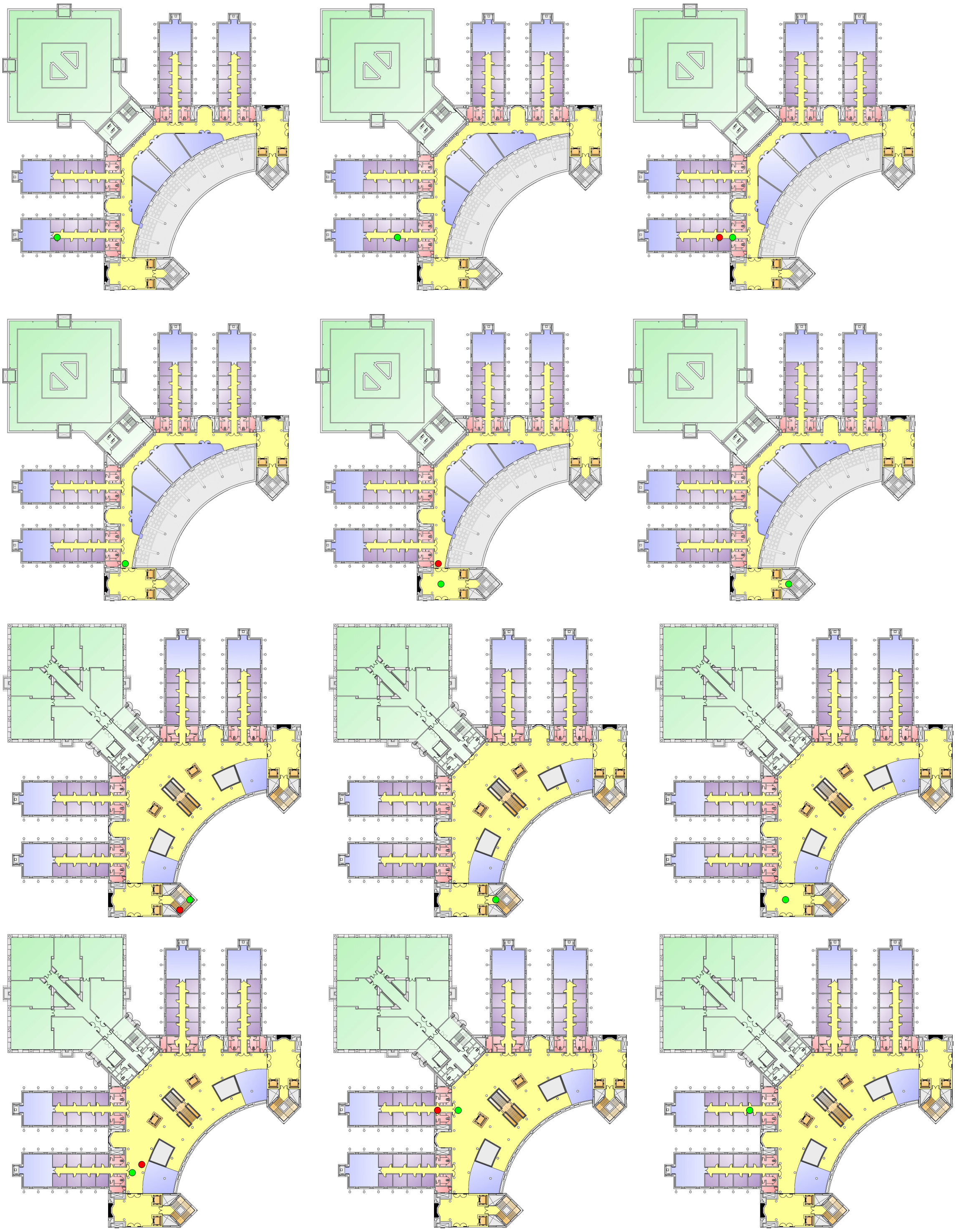


Figure 6.9: Proposed hierarchical WiFi indoor localization sequence during the multi-floor trajectory 1. The sequence starts on the top left image and continues from left to right.

Chapter 7

Conclusions and Future Work

The goal of this thesis was the localization of mobile devices in indoor environments using only the RSS from the already existing APs in the environment. Since WiFi is pre-installed in most buildings, there is no need to either modify the environment or add new devices to it. The final research objective was to develop a robust WiFi real-time localization for mobile devices, available to be deployed in any environment and to be used by any device equipped with a WiFi interface.

A method for performing WiFi indoor localization has been presented. To do so, different techniques have been applied:

1. An FRBC was designed using an FDT algorithm to reduce the uncertainty generated by the small scale variations on static positions of the environment. Using this proposal the accuracy was improved around a 7% in comparison with the well-known Nearest Neighbour algorithm. The mean error was reduced a 42% (from 1.49 to 0.85 metres).

2. A hierarchical approach to simplify the localization task was proposed. This method performs an automatic environment division into hierarchical zones with the aim of improving the accuracy of topology-based WiFi localization systems in large environments. This proposal was tested in a multi-floor real environment in two scenarios of growing complexity. On the light of the results it can be concluded that this proposal emerges as a powerful tool. The highest accuracy was close to 85% in the simple scenario and it was slightly reduced in the complete scenario where it was close to 71%. In both cases, the best results were reported when using the SVM as position classifier at the lowest classification level of the hierarchy. The accuracy improvement due to the hierarchical approach was close to 12% and the mean error was reduced from 2.20 to 1.97 metres in the complete scenario. The results obtained using the FDT classifier were very close to the ones achieved by the SVM algorithm. The accuracy was close 67% and the error was reduced to 2.01 metres in the complete scenario.

3. An approach to track the position of a device in motion using a topological radio-map was proposed. This approach uses a Bayes filter that continuously estimates the most likely position of the device. The filter was able to deal with the low

working frequency of the device and the uncertainty of the observations providing an accurate and fast estimation. The best results were obtained when combining the hierarchical FDT algorithm with the HMM achieving a mean error under 3 metres.

It is important to highlight that it is not necessary to know where the APs are located to deploy the localization system. This aspect is especially interesting regarding its deployment in new unknown environments. Moreover, since the localization is performed directly on the device, the system can be safely used without dealing with privacy issues.

The remaining of the chapter presents the main contributions introduced and developed along this thesis. Finally, the future lines of research left open by this thesis will be drawn.

7.1 Main Contributions

From the results obtained in the previous chapters, the main contributions of this thesis are as follows:

1. **WiFi signal analysis.** The influence of different effects on the WiFi RSS have been analysed, identifying the small scale variations as the main source of localization error. Some other effects, such as the temporal variations or co-channel interferences, were also identified and analysed. With the study of all these effects and sources of error some important decisions on the WiFi localization design were made. First, the use of a propagation model was discarded since all the effects that affect the RSS indoors make very difficult to adjust a model for an environment, and unlikely that the designed model fits well to different environments. Second, that it was necessary to use some techniques to cope with the small scale variations to be able to implement a realistic WiFi localization system. Finally, that some other error sources, such as the orientation of the device, are masked by more critical error sources like the small scale variations.

2. **WiFi topological datasets.** RSS from all the visible APs were recorded in two different real environments: ECSC and UAH. The database for each environment contains two datasets (train and test datasets) composed of the measurements at each topological position. Moreover, for the UAH environment, different trajectories to test the performance of the localization systems while locating a device in motion are included. All the trajectories include the topological groundtruth for each sample. The databases contain data collected under real conditions, including most sources of noise such as people wandering around. The databases are to be made available to the research community.

3. **WiFi indoor localization system.** Three approaches, that complements each other, were proposed to solve the problem of locating a device in a real large indoor environment. First, a fuzzy-based algorithm was proposed to reduce the effect of the small scale variations. Then, a hierarchical division of the environment was

designed to avoid the reduction of accuracy of WiFi localization systems when working in large environments crowded with APs. Finally, a Bayes filter to smooth the estimated trajectory followed by a device while moving over the environment was developed. The designed system complies with the pre-imposed restrictions of real-time execution, robustness to signal interferences and proper functioning using different devices. The system was tested in a challenging real multi-floor environment, proving that the proposed method is capable of tracking the global position of a WiFi device.

4. **Final application.** A desktop application (Appendix A) that allows to perform user-friendly WiFi indoor localization using the designed system was developed. Using this tool, the tedious task of site-survey new environments is simplified. Once an environment is built and the localization system is trained it can be used to perform real-time localization. Moreover, the classifiers to perform localization in an environment are built thinking of being exportable to the Android application developed as part of the ABSYNTHE project. This way, the WiFi indoor localization system object of this thesis will be available for every Android device.

7.2 Future work

From the results and conclusions of the present work, several research lines can be faced:

1. **Use the information provided by additional sensors.** The system performance can be highly improved by using the information provided by the compass and the accelerometers of a mobile phone or tablet. By knowing the direction of movement or even if the device is moving or standing at a position, the number of allowed transitions between positions would be reduced and the filtering task would be simplified.

2. **Classifiers aggregation.** The overall accuracy of the system can be improved by using an aggregation of the different classifiers results. The results of the developed approaches have shown that the different classifiers perform better under different circumstances. For instance, the SVM algorithm is the best when locating a device standing at a fixed position, while the FDT algorithm is the best locating a device in motion. Knowing the current state of the device using the information provided by the accelerometers, the system could give priority to the SVM predictions when the device is at a fixed position, and to the FDT predictions when it is in movement.

3. **Testing other classification techniques.** More advanced classification techniques could be explored. For example, the multiclassifier previously designed in [Trawinski et al., 2013] in combination with the designed hierarchical approach could be tested.

4. **Testing other techniques to divide the environment.** Alternative and more advanced methods for finding out an optimal partition of the environment will be further analysed. For instance, the so-called Wifigrams previously described in [Alonso et al., 2013] could be used.

5. **Access points selection.** A procedure for selecting some of the APs will be tested. This AP selection may be made according to the visibility criteria introduced in this thesis.

6. **Test the system in the 5 GHz band.** With the appearance of the 5 GHz WiFi APs, which work in a less noisy band, some of the main problems related to the use of this technology could be reduced. So, some experimentation could be done to test the performance of the designed system using the 5 GHz frequency band.

7. **Automatic radio-map generation process.** One of the main problems of the fingerprint-based methods is the radio-map generation. So, an interesting research area could be focused on designing an automated radio-map generation process.

Appendices

Appendix A

A New Software for WiFi Indoor Localization

This appendix presents the software developed for topology-based localization that implements the work described in this thesis. Two different applications have been designed:

- A desktop software for research purposes which allows to create new environments and to train and test different localization algorithms. Thanks to the software modularity, different localization algorithms can be evaluated on the same environments and compared to each other. The research software allows to choose the best performing localization algorithm that will be available to be loaded into the Android application.

- An open-access Android application that allows users with a smartphone or a tablet obtain their position inside the environment and be guided to any place.

A.1 Desktop software

This application was developed using C++ under Qt. Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface.

In this application, an environment is a map, or a group of maps, with user defined topological positions. The environments have to be created before the training stage, in order for the software to have all the necessary information to be able to train the localization system. The localization system must be trained for each environment to allow localization of the user in it. All these 3 steps, creation of the environment, training of the system for the environment and localization in the environment are performed into the desktop software. The training and localization stages allow the selection of different algorithms that can be executed for comparative purposes.

Figure A.1 shows a flow diagram of the software. This will be thoroughly explained in the next sections where the process for new environments creation and both training and localization stages will described.

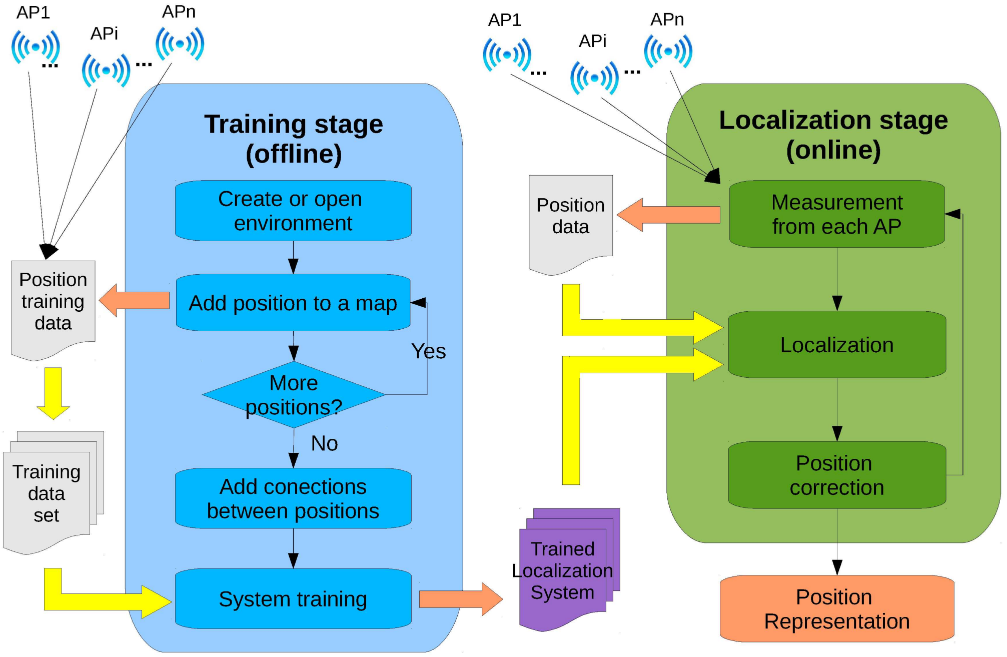


Figure A.1: Flow diagram of the software.

A.1.1 Environment handling

This tool allows for the creation of new environments. An environment stores all the necessary information to build the localization system. An environment can be composed of several maps (i.e. the maps of the different floors of a building) and the transitions between maps (i.e. stairs). The maps are composed of topological positions and the connections between them. New positions can be added to the different maps. When a new position is added, the user is asked to go to it, then the system measures the WiFi RSS from all the visible APs and stores it as the fingerprint data for the current position. A unique identifier is given to each position, but the user can also give a name to them facilitating their identification. This way, the user can ask for guidance to the "Entrance of Laboratory 1" instead of guidance to "Position 1", for instance. Finally, the positions can be linked to each other to allow or forbid transitions between them. This connections can be also defined between maps.

The environment creation is only available in the desktop software (Figure A.2). All the data collected during the environment creation is stored to be used by the training and localization modules. Once the environment has been created and stored, it can be opened for modifications, adding or removing maps, positions and connections between them.

The environment creation process can be summarized in the following steps:

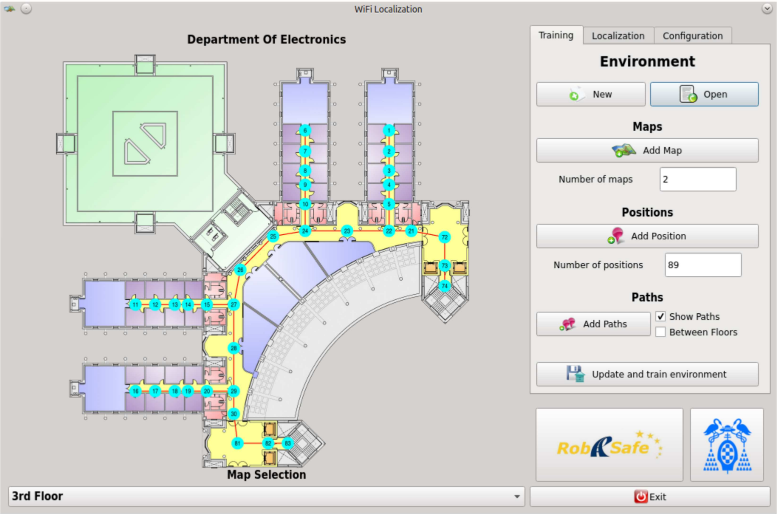
- Create or open an environment: New environments can be created using the "New" button. Already created environments can be opened to be modified or used for training using the "Open" button.

- Add maps to an environment: New maps can be added to an environment using the “Add Map” button.

- Add positions to a map: New locations can be added to the selected map using the “Add Position” button. The user will be asked to move to the position to start the RSS measurements. While the system measures the WiFi signal, the user will be asked to click on the location of the position on the map, and to give, if desired, a name to it.

- Add connections (paths) between positions: The positions that are physically connected can be linked to indicate to the system the feasible transitions to improve the localization. Connections between positions from different maps are also allowed to permit movements between different maps.

- Save the environment: All the information collected in the previous steps are stored for future uses by pressing the “Update and train environment” button.



Number of maps 2

Number of positions 89

☑ Show Paths

□ Between Floors

3rd Floor

Figure A.2: Screenshot of the training stage in the desktop software.

A.1.2 Training stage

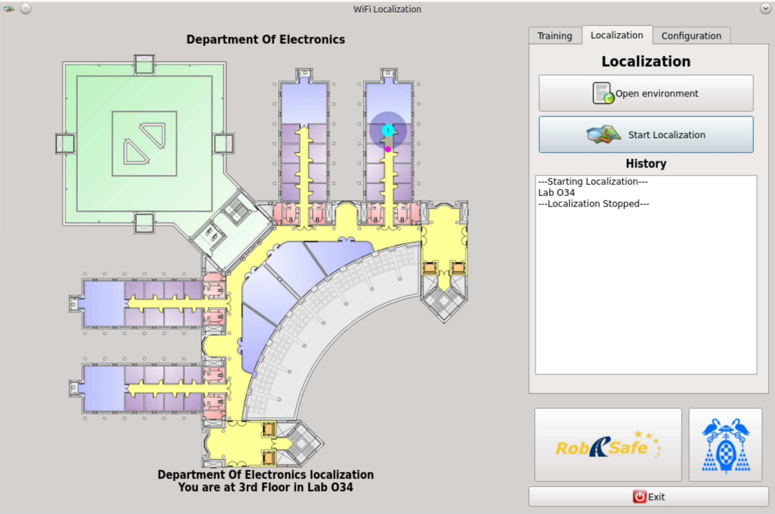
This is an offline stage and it is only available in the desktop software (Figure A.2). Its goal is to train the localization system using all the information collected during the creation of the environment.

The localization system for the environment is trained by clicking the “Update and train environment” button. The training module is called and the system is trained using all the available algorithms, creating all the models that can be selected in the localization stage to obtain the device position.

Thanks to the modularity of the software new localization algorithms can be easily added. The system will be trained using the new algorithms for the selected environment and a new model will be created.

A.1.3 Localization stage

In this stage, the WiFi device will obtain its current position using the RSS from all visible APs on an online process. The set of classifiers trained in the previous stage can be now used to locate the device. This stage is available in both the desktop (Figure A.3) and Android (Figure A.6) applications.



---Starting Localization---
Lab O34
---Localization Stopped---

Department Of Electronics localization
You are at 3rd Floor in Lab O34

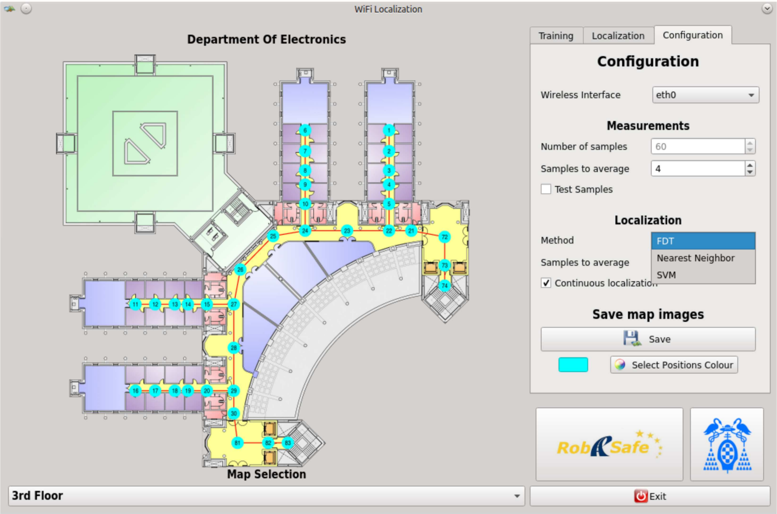
Figure A.3: Screenshot of the localization stage in the desktop software.

The localization stage comprises three steps as showed in Figure A.1:

- Measurement: The device measures the RSS from every AP and a sample in the necessary format is created to be used by the localization system.
- Localization: The previously created sample is classified using the selected localization method (Figure A.4) and the estimated position is provided.

  Currently, three different algorithms are available:

  - Hierarchical FDT: As the one explained in Chapters 4 and 5.
  - Hierarchical KNN: As the one explained in Chapter 5.
  - Hierarchical SVM: As the one explained in Chapter 5.
- Position correction: The position provided by the localization module can be corrected by ticking the “Continuous localization” check box, if it is an unreachable position from the previous one, using the information about the connections between the positions as explained in Chapter 6.



Configuration

Wireless Interface: eth0

Measurements

Number of samples: 60

Samples to average: 4

□ Test Samples

Localization

Method: FDT

Nearest Neighbor

SVM

Samples to average

☑ Continuous localization

Save map images

Save

Select Positions Colour

3rd Floor

Figure A.4: Localization configuration in the desktop software.

A.2 Open-access Android application

The open-access localization application [ABSYNTHE Application, 2013] [Humanes et al., 2013] allows to locate an Android device using the system trained by the desktop software following the same structure as the one described in Section A.1.3.

The position of the device can also be obtained by scanning a QR code as shown in the flow diagram in Figure A.5. These QR codes are distributed over the environment and contain the information related to the position where they are located.

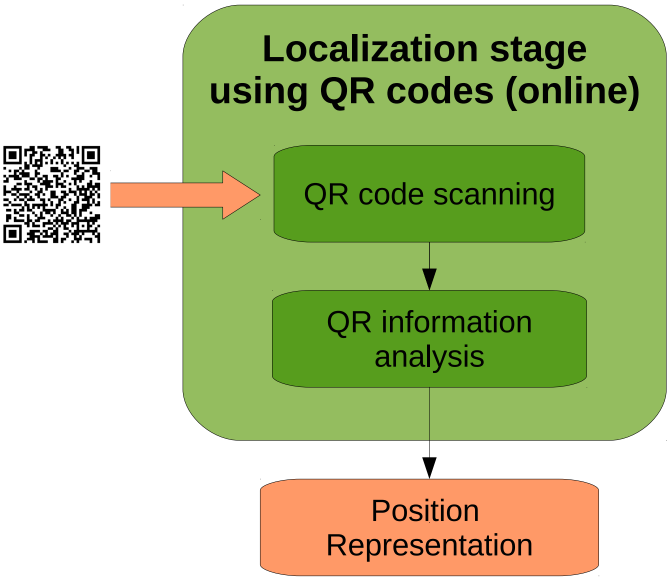
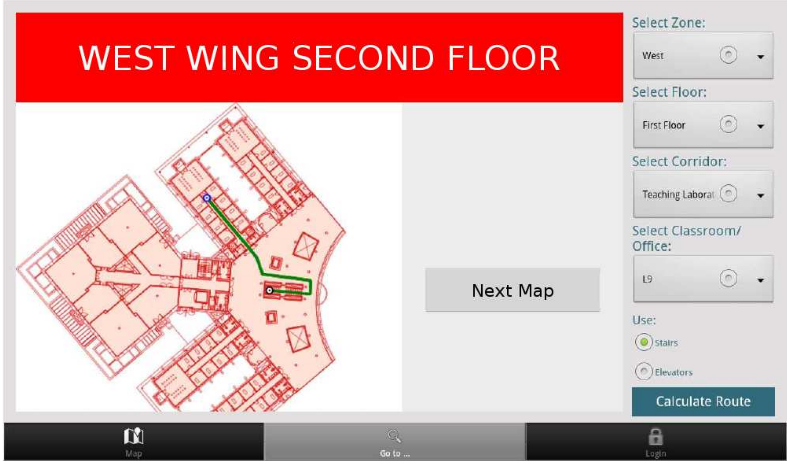


Figure A.5: Flow diagram of the localization stage using QR codes.

Once the device location is obtained, using one of the methods previously described, the application provides guidance to a selected destination (Figure A.6) using the Dijkstra algorithm [Dijkstra, 1959]. The application also allows the users to ask for a robot to guide them in certain environments. Currently, the localization and guidance application (using only the QR localization module) is available for the Polytechnic School of the UAH and at the ECSC, and can be downloaded using the QR codes located at the entrances of the buildings.



WEST WING SECOND FLOOR

Select Zone: West

Select Floor: First Floor

Select Corridor: Teaching Laborat

Select Classroom/ Office: L9

Use:
● Stairs
○ Elevators

Figure A.6: Screenshot of the open-access localization app.

Appendix B

Publications Derived from this PhD Dissertation

B.1 Journal Publications

2014 **Hierarchical Approach to Enhancing Topology-based WiFi Indoor Localization in Large Environments**, *N. Hernández, J.M. Alonso, M. Ocaña*, Multiple Valued Logic and Soft Computing (Under second round review - Minor revision).

2013 **A multiclassifier approach for topology-based WiFi indoor localization**, *K. Trawiński, J.M. Alonso, N. Hernández*, Soft Computing (ISSN: 1432-7643), Vol. 17(10), pages 1817-1831.

2013 **Impact of Signal Representations on the Performance of Hierarchical WiFi Localization Systems**, *N. Hernández, J.M. Alonso, M. Ocaña, M. K. Marina*, Lecture Notes in Computer Science (ISSN: 0302-9743), Vol. 8112, pages 17-24.

2013 **Wifigrams: Design of Hierarchical Wi-Fi Indoor Localization Systems Guided by Social Network Analysis**, *J.M. Alonso, N. Hernández, M. Ocaña*, Lecture Notes in Computer Science (ISSN: 0302-9743), Vol. 8112, pages 9-16.

2011 **Enhanced WiFi Localization System Based on Soft Computing Techniques to deal with Small-scale Variations in Wireless Sensors**, *J.M. Alonso, M. Ocaña, N. Hernández, F. Herranz, A. Llamazares, M.A. Sotelo, L.M. Bergasa, L. Magdalena*, Applied Soft Computing (ISSN: 1568-4946), Vol. 11(8), pages 4677-4691.

B.2 Conference Publications

2014 **A WiFi-based Software for Indoor Localization**, *N. Hernández, M. Ocaña, S. Humanes, P. Revenga, D.P. Pancho, L. Magdalena*, IEEE World Congress on Computational Intelligence (FUZZ-IEEE 2014), Beijing (China).

2013 **Parameter Evaluation in a Hierarchical WiFi Localization System**, *N. Hernández, J.M. Alonso, M. Ocaña*, Fourteenth International Conference on Computer Aided Systems Theory (EUROCAST 2013), pages 32-35, Universidad de Las Palmas de Gran Canaria (Spain).

2013 **Wifigrams: Social Network Analysis for Supporting the Design of Hierarchical Wi-Fi Indoor Location Systems**, *J.M. Alonso, N. Hernández, M. Ocaña*, Fourteenth International Conference on Computer Aided Systems Theory (EUROCAST 2013), pages 28-31, Universidad de Las Palmas de Gran Canaria (Spain).

2012 **Hierarchical WiFi Localization System**, *N. Hernández, J.M. Alonso, M. Magro, M. Ocaña*, IEEE Intelligent Vehicles Symposium. Workshop Perception in Robotics, pages 1-6, Alcalá de Henares (Spain).

2010 **Studying of WiFi Range-only Sensor and its Aplication to Localization and Mapping Systems**, *F. Herranz, M. Ocaña, L.M. Bergasa, M.A. Sotelo, D.F. Llorca, N. Hernández, A. Llamazares, C. Fernández*, IEEE International Conference on Robotics and Automation (IEEE ICRA 2010), pages 115-120, Anchorange (USA).

2010 **Towards people indoor localization combining WiFi and human motion recognition**, *J.M. Alonso, A. Álvarez, G. Trivino, N. Hernández, F. Herranz, M. Ocaña*, Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF 2010), pages 7-12, Huelva (Spain).

2009 **WiFi Localization System based on Fuzzy Logic to deal with Signal Variations**, *N. Hernández, F. Herranz, M. Ocaña, L.M. Bergasa, J.M. Alonso, L. Magdalena*, IEEE International Conference on Emerging Technologies and Factory Automation (IEEE ETFA 2009), pages 001295-1 - 001295-6 , Palma de Mallorca (Spain).

2009 **Intelligent Techniques applied to WiFi Localization Systems**, *F. Herranz, N. Hernández, M. Ocaña, L.M. Bergasa, D. Alonso*, Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI09), Madrid (Spain).

Bibliography

[ABSYNTHE Application, 2013] ABSYNTHE Application, "Android open-access localization application for the ABSYNTHE proyect" (2013), accessed on June 2014. http://absynthe.depeca.uah.es/www/GPS-UAH/download-ciceroneUAH.html

[Alappanavar et al., 2013] P. B. Alappanavar, A. Bhujbal and S. Deshmukh, "Location based services using augmented reality". *International Journal of Computer Engineering and Technology*, volume 4(2), pages 237–240 (2013).

[Alonso et al., 2013] J. M. Alonso, N. Hernández and M. Ocaña, "Wifigrams: Design of hierarchical WiFi indoor localization systems guided by social network analysis". In *Computer Aided Systems Theory - EUROCAST 2013*, volume 8112 of *Lecture Notes in Computer Science*, pages 9–16 (2013).

[Alonso et al., 2012] J. M. Alonso, K. LeBlanc, M. Ocaña and E. Ruspini, "ABSYNTHE: Abstraction, synthesis, and integration of information for human-robot teams". In *Proceedings of the International Workshop on Perception in Robotics, IEEE Intelligent Vehicles Symposium*, pages P14.1–P14.6 (2012).

[Alonso and Magdalena, 2011] J. M. Alonso and L. Magdalena, "Generating understandable and accurate fuzzy rule-based systems in a java environment". In *Fuzzy Logic and Applications*, volume 6857 of *Lecture Notes in Computer Science*, pages 212–219 (2011).

[Alonso et al., 2008] J. M. Alonso, L. Magdalena and S. Guillaume, "HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism". *International Journal of Intelligent Systems*, volume 23(7), pages 761–794 (2008).

[Alonso et al., 2011] J. M. Alonso, M. Ocaña, N. Hernández, F. Herranz, A. Llamazares, M. A. Sotelo, L. M. Bergasa and L. Magdalena, "Enhanced WiFi localization system based on soft computing techniques to deal with small-scale variations in wireless sensors". *Applied Soft Computing*, volume 11(8), pages 4677–4691 (2011).

[Alonso et al., 2009] J. M. Alonso, M. Ocaña, M. A. Sotelo, L. M. Bergasa and L. Magdalena, "WiFi localization system using fuzzy rule-based classification". In *Computer Aided Systems Theory - EUROCAST 2009*, volume 5717 of *Lecture Notes in Computer Science*, pages 383–390 (2009).

[Astrain et al., 2006] J. J. Astrain, J. Villadangos, J. R. Garitagoitia, J. R. G. de Mendívil and V. Cholvi, "Fuzzy location and tracking on wireless networks". In *Proceedings of the ACM International Workshop on Mobility Management and Wireless Access*, pages 84–91 (2006).

[Bahillo et al., 2009] A. Bahillo, R. M. Lorenzo, S. Mazuelas, P. Fernández and E. J. Abril, "Assessment of the shadow caused by the human body on the personal RF dosimeters reading in multipath environments". *Biomedical Engineering*, pages 133–144 (2009).

[Bahl and Padmanabhan, 2000] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system". In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 775–784 (2000).

[Bahl et al., 2000] P. Bahl, V. N. Padmanabhan and A. Balachandran, "Enhancements to the RADAR user location and tracking system". Technical Report MSR-TR-2000-12 (2000).

[Barber et al., 2002] R. Barber, M. Mata, M. J. L. Boada, J. M. Armingol and M. A. Salichs, "A perception system based on laser information for mobile robot topologic navigation". In *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society*, volume 4, pages 2779–2784 (2002).

[Baum and Petrie, 1966] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains". *The Annals of Mathematical Statistics*, volume 37(6), pages 1554–1563 (1966).

[BI Intelligence, 2013] BI Intelligence, "Cumulative app downloads since 2008" (2013), accessed on June 2014.
http://www.businessinsider.com/comparing-the-growth-of-top-app-markets-2013-7

[Biswas and Veloso, 2010] J. Biswas and M. Veloso, "WiFi localization and navigation for autonomous indoor mobile robots". In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4379–4384 (2010).

[Bose and Foh, 2007] A. Bose and C. H. Foh, "A practical path loss model for indoor WiFi positioning enhancement". In *Proceedings of the International Conference on Information, Communications Signal Processing*, pages 1–5 (2007).

[Brunato and Battiti, 2005] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs". *Computer Networks*, volume 47(6), pages 825–845 (2005).

[Caballero et al., 2008] F. Caballero, L. Merino, I. Maza and A. Ollero, "A particle filtering method for wireless sensor network localization with an aerial robot beacon". In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 596–601 (2008).

[Caliński and Harabasz, 1974] T. Caliński and J. Harabasz, “A dendrite method for cluster analysis”. *Communications in Statistics-Simulation and Computation*, volume 3(1), pages 1–27 (1974).

[Chang, 2011] J.-R. Chang, “The integration of location-based service and augmented reality techniques to a road management system”. In *Proceedings of the International Association for Automation and Robotics in Construction*, pages 1407–1408 (2011).

[Chen et al., 2008] Q. Chen, D.-L. Lee and W.-C. Lee, “Rule-based WiFi localization methods”. In *Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, volume 1, pages 252–258 (2008).

[Cheng et al., 2013] H. Cheng, M. S. Arefin, Z. Chen and Y. Morimoto, “Place recommendation based on users check-in history for location-based services”. *International Journal of Networking and Computing*, volume 3(2), pages 228–243 (2013).

[Chintalapudi et al., 2010] K. Chintalapudi, A. Padmanabha Iyer and V. N. Padmanabhan, “Indoor localization without the pain”. In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pages 173–184 (2010).

[Choey et al., 2003] M. Choey, D. Schreiber and D. Schutzer, “Method and system of transaction card fraud mitigation utilizing location based services”. US Patent US20030182194 A1 (2003).

[Christensen et al., 2007] C. M. Christensen, J. Kjeldskov and K. K. Rasmussen, “Geohealth: A location-based service for nomadic home healthcare workers”. In *Proceedings of the Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*, pages 273–281 (2007).

[Cohen, 1995] W. W. Cohen, “Fast effective rule induction”. In *Proceedings of the International Conference on Machine Learning*, pages 115–123 (1995).

[Cortes and Vapnik, 1995] C. Cortes and V. Vapnik, “Support-vector networks”. *Machine Learning*, volume 20(3), pages 273–297 (1995).

[Dharne et al., 2006] A. G. Dharne, J. Lee and S. Jayasuriya, “Using fuzzy logic for localization in mobile sensor networks: Simulations and experiments”. In *Proceedings of the American Control Conference*, pages 2066–2071 (2006).

[Dijkstra, 1959] E. W. Dijkstra, “A note on two problems in connexion with graphs”. *Numerische Mathematik*, volume 1(1), pages 269–271 (1959).

[eCall, 2007] eCall (2007), accessed on June 2014.
http://ec.europa.eu/digital-agenda/ecall-time-saved-lives-saved

[Ekahau, 2014] Ekahau, “Wi-Fi tracking systems, RTLS and WLAN site survey” (2014), accessed on June 2014.
http://www.ekahau.com

[Elnahrawy et al., 2004] E. Elnahrawy, X. Li and R. P. Martin, “The limits of localization using signal strength: A comparative study”. In *Proceedings of the Annual IEEE Communications Society Conference on Sensor Ad Hoc Communications and Networks*, pages 406–414 (2004).

[Enge and Misra, 1999] P. Enge and P. Misra, “Special issue on GPS: The global positioning system”. *Proceedings of the IEEE*, volume 87(1), pages 3–172 (1999).

[Fang and Lin, 2008] S.-H. Fang and T.-N. Lin, “Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments”. *IEEE Transactions on Neural Networks*, volume 19(11), pages 1973–1978 (2008).

[Fang and Lin, 2010] S.-H. Fang and T.-N. Lin, “A dynamic system approach for radio location fingerprinting in wireless local area networks”. *IEEE Transactions on Communications*, volume 58(4), pages 1020–1025 (2010).

[Ferris et al., 2006] B. Ferris, D. Hähnel and D. Fox, “Gaussian processes for signal strength-based location estimation”. In *Proceedings of Robotics: Science and Systems*, pages 1–8 (2006).

[Figuera et al., 2012] C. Figuera, J. L. Rojo-Álvarez, M. Wilby, I. Mora-Jiménez and A. J. Caamaño, “Advanced support vector machines for 802.11 indoor location”. *Signal Processing*, volume 92(9), pages 2126–2136 (2012).

[Fink and Kumar, 2010] J. Fink and V. Kumar, “Online methods for radio signal mapping with mobile robots”. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1940–1945 (2010).

[Fox et al., 2003] D. Fox, J. Hightower, L. Liao, D. Schulz and G. Borriello, “Bayesian filtering for location estimation”. *IEEE Pervasive Computing*, volume 2(3), pages 24–33 (2003).

[García-Valverde et al., 2012] T. García-Valverde, A. García-Sola, A. Gómez-Skarmeta, J. Botía, H. Hagras, J. Dooley and V. Callaghan, “An adaptive learning fuzzy logic system for indoor localisation using Wi-Fi in ambient intelligent environments”. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1–8 (2012).

[García-Valverde et al., 2013] T. García-Valverde, A. García-Sola, H. Hagras, J. Dooley, V. Callaghan and J. A. Botía, “A fuzzy logic-based system for indoor localization using WiFi in ambient intelligent environments”. *IEEE Transactions on Fuzzy Systems*, volume 21(4), pages 702–718 (2013).

[Google Glass, 2014] Google Glass (2014), accessed on June 2014.
http://www.google.com/glass/start/

[Guo et al., 2012] B. Guo, R. Fujimura, D. Zhang and M. Imai, “Design-in-play: Improving the variability of indoor pervasive games”. *Multimedia Tools and Applications*, volume 59(1), pages 259–277 (2012).

[Haeberlen et al., 2004] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks". In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pages 70–84 (2004).

[Hammadi et al., 2012] O. A. Hammadi, A. A. Hebsi, M. J. Zemerly and J. W. P. Ng, "Indoor localization and guidance using portable smartphones". In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 337–341 (2012).

[Hassan-Ali and Pahlavan, 1998] M. Hassan-Ali and K. Pahlavan, "Site-specific wideband and narrowband modeling of indoor radio channel using ray-tracing". In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 1, pages 65–68 (1998).

[Hastie and Tibshirani, 1998] T. Hastie and R. Tibshirani, "Classification by pairwise coupling". *The Annals of Statistics*, volume 26(2), pages 451–471 (1998).

[Hellendoorn and Driankov, 1997] H. Hellendoorn and D. Driankov, *Fuzzy model identification: Selected Approaches*. Springer, 1st edition (1997).

[Herranz, 2013] F. Herranz, *Simultaneous Localization and Mapping using Range Only Sensors*. Ph.D. thesis, University of Alcalá (2013).

[Hightower and Borriello, 2001] J. Hightower and G. Borriello, "Location sensing techniques". Technical Report UW-CSE-01-07-01 (2001).

[Hightower and Borriello, 2004] J. Hightower and G. Borriello, "Particle filters for location estimation in ubiquitous computing: A case study". In *Proceedings of International Conference on Ubiquitous Computing*, pages 88–106 (2004).

[Howard et al., 2006] A. Howard, S. Siddiqi and G. S. Sukhatme, "An experimental study of localization using wireless ethernet". In *Field and Service Robotics*, volume 24 of *Springer Tracts in Advanced Robotics*, pages 145–153 (2006).

[Hühn and Hüllermeier, 2009] C. Hühn and E. Hüllermeier, "FURIA: An algorithm for unordered fuzzy rule induction". *Data Mining and Knowledge Discovery*, volume 19(3), pages 293–319 (2009).

[Hühn and Hüllermeier, 2010] J. Hühn and E. Hüllermeier, "An analysis of the FURIA algorithm for fuzzy rule induction". In *Advances in Machine Learning I*, volume 262 of *Studies in Computational Intelligence*, pages 321–344 (2010).

[Hüllermeier, 2005] E. Hüllermeier, "Fuzzy methods in machine learning and data mining: Status and prospects". *Fuzzy Sets and Systems*, volume 156(3), pages 387–406 (2005).

[Humanes et al., 2013] S. Humanes, P. Revenga, M. Ocaña and J. M. Alonso, "Interfaz hombre-máquina del proyecto Absynthe". In *Proceedings of the workshop RoboCity2030 "Robots Sociales"*, pages 59–69 (2013).

[Ichihashi et al., 1996] H. Ichihashi, T. Shirai, K. Nagasaka and T. Miyoshi, “Neuro-fuzzy ID3: A method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning”. *Fuzzy Sets and Systems*, volume 81(1), pages 157–167 (1996).

[Indoor Google Maps, 2014] Indoor Google Maps (2014), accessed on June 2014. http://maps.google.com/help/maps/indoormaps/

[Jeon and Kim, 2013] B. Jeon and R. Y. C. Kim, “A system for detecting the stray of objects within user-defined region using location-based services”. *International Journal of Software Engineering and Its Applications*, volume 7(5), pages 355–362 (2013).

[Kaemarungsi and Krishnamurthy, 2004] K. Kaemarungsi and P. Krishnamurthy, “Pre-processing of fingerprints to improve the positioning accuracy of 802.11-based positioning systems”. In *Proceedings of the Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 14–23 (2004).

[Kibler and Aha, 1987] D. Kibler and D. Aha, “Learning representative exemplars of concepts: An initial case study”. In *Proceedings of the International Workshop on Machine Learning*, pages 24–30 (1987).

[Kortenkamp, 1993] D. M. Kortenkamp, *Cognitive maps for mobile robots: A representation for mapping and navigation.* Ph.D. thesis, University of Michigan (1993).

[Kotanen et al., 2003] A. Kotanen, M. Hännikäinen, H. Leppäkoski and T. D. Hämäläinen, “Experiments on local positioning with Bluetooth”. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, pages 297–303 (2003).

[Krumm et al., 2000] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer, “Multi-camera multi-person tracking for easyliving”. In *Proceedings of the IEEE International Workshop on Visual Surveillance*, pages 3–10 (2000).

[Kuipers and Byun, 1988] B. J. Kuipers and Y.-T. Byun, “A robust, qualitative method for robot spatial learning”. In *Proceedings of the National Conference on Artificial Intelligence*, pages 774–779 (1988).

[Ladd et al., 2005] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki and D. S. Wallach, “Robotics-based location sensing using wireless ethernet”. *Wireless Networks*, volume 11(1-2), pages 189–204 (2005).

[Liu et al., 2007] H. Liu, H. Darabi, P. Banerjee and J. Liu, “Survey of wireless indoor positioning techniques and systems”. *IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews*, volume 37(6), pages 1067–1080 (2007).

[MacQueen, 1967] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations”. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297 (1967).

[Mamdani, 1977] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems". *IEEE Transactions on Computers*, volume 26(12), pages 1182–1191 (1977).

[Mautz, 2012] R. Mautz, "Indoor positioning technologies". Habilitation thesis, ETH Zurich (2012).

[Mazuelas et al., 2009] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas and E. J. Abril, "Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks". *IEEE Journal of Selected Topics in Signal Processing*, volume 3(5), pages 821–831 (2009).

[Mencar and Fanelli, 2008] C. Mencar and A. M. Fanelli, "Interpretability constraints for fuzzy information granulation". *Information Sciences*, volume 178(24), pages 4585–4618 (2008).

[Mengual et al., 2010] L. Mengual, O. Marbán and S. Eibe, "Clustering-based location in wireless networks". *Expert Systems with Applications*, volume 37(9), pages 6165–6175 (2010).

[Molina and Alba, 2011] G. Molina and E. Alba, "Location discovery in wireless sensor networks using metaheuristics". *Applied Soft Computing*, volume 11(1), pages 1223–1240 (2011).

[Ocaña, 2005] M. Ocaña, *Sistema de localización global WiFi aplicado a la navegación de un robot semiautónomo*. Ph.D. thesis, Universidad de Alcalá (2005).

[Ocaña et al., 2005] M. Ocaña, L. M. Bergasa, M. A. Sotelo and R. Flores, "Indoor robot localization system using WiFi signal measure and minimizing calibration effort". In *Proceedings of the IEEE International Symposium on Industrial Electronics*, pages 1545–1550 (2005).

[Parodi et al., 2006] B. Parodi, H. Lenz, A. Szabo, H. Wang, J. Horn, J. Bamberger and D. Obradovic, "Initialization and online-learning of RSS maps for indoor / campus localization". In *Proceedings of the IEEE/ION Position, Location, And Navigation Symposium*, pages 164–172 (2006).

[Parwekar and Reddy, 2013] P. Parwekar and R. Reddy, "An efficient fuzzy localization approach in wireless sensor networks". In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1–6 (2013).

[Paul and Wan, 2009] A. S. Paul and E. A. Wan, "RSSI-based indoor localization and tracking using sigma-point Kalman smoothers". *IEEE Journal of Selected Topics in Signal Processing*, volume 3(5), pages 860–873 (2009).

[Pourhomayoun et al., 2012] M. Pourhomayoun, Z. Jin and M. Fowler, "Spatial sparsity based indoor localization in wireless sensor network for assistive healthcare". In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3696–3699 (2012).

[Prasithsangaree et al., 2002] P. Prasithsangaree, P. Krishnamurthy and P. K. Chrysanthis, “On indoor position location with wireless LANs”. In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 720–724 (2002).

[Priyantha et al., 2000] N. B. Priyantha, A. Chakraborty and H. Balakrishnan, “The cricket location-support system”. In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pages 32–43 (2000).

[Quercia et al., 2010] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo and J. Crowcroft, “Recommending social events from mobile phone location data”. In *Proceedings of the IEEE International Conference on Data Mining*, pages 971–976 (2010).

[Quinlan, 1986] J. R. Quinlan, “Induction of decision trees”. *Machine Learning*, volume 1(1), pages 81–106 (1986).

[Quinlan, 1996] J. R. Quinlan, “Improved use of continuous attributes in C4.5”. *Journal of Artificial Intelligence Research*, volume 4(1), pages 77–90 (1996).

[Rai et al., 2012] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan and R. Sen, “Zee: Zero-effort crowdsourcing for indoor localization”. In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pages 293–304 (2012).

[Rappaport, 1996] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 2nd edition (1996).

[Ruspini, 1969] E. H. Ruspini, “A new approach to clustering”. *Information and Control*, volume 15(1), pages 22–32 (1969).

[Saiph Savage et al., 2012] N. Saiph Savage, M. Baranski, N. Elva Chavez and T. Höllerer, “I’m feeling LoCo: A location based context aware recommendation system”. In *Advances in Location-Based Services*, Lecture Notes in Geoinformation and Cartography, pages 37–54 (2012).

[Satelise, 2010] Satelise (2010), accessed on June 2014.
https://www.cintra.es/en/Commitment/Satelise-platform

[Sichitiu and Ramadurai, 2004] M. Sichitiu and V. Ramadurai, “Localization of wireless sensor networks with a mobile beacon”. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 174–183 (2004).

[Stamoulakatos and Sykas, 2003] T. Stamoulakatos and E. Sykas, “A network architecture to obtain traffic information for location based service applications”. In *Proceedings of the International Conference on Telecommunications*, pages 197–203 (2003).

[Thrun et al., 2005] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*. Intelligent robotics and autonomous agents series, MIT Press, 4th edition (2005).

[TNS, Kantar Group, 2012] TNS, Kantar Group, “Mobile life study” (2012), accessed on June 2014.
http://www.tnsglobal.com/press-release/two-thirds-world%E2%80%99s-mobile-users-signal-they-want-be-found

[Trawinski et al., 2013] K. Trawinski, J. M. Alonso and N. Hernández, “A multiclassifier approach for topology-based wifi indoor localization”. *Soft Computing*, volume 17(10), pages 1817–1831 (2013).

[Varshavsky et al., 2007] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca and V. Otsason, “GSM indoor localization”. *Pervasive and Mobile Computing*, volume 3(6), pages 698–720 (2007).

[Viterbi, 1967] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. *IEEE Transactions on Information Theory*, volume 13(2), pages 260–269 (1967).

[Wang et al., 2012] 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 International Conference on Mobile Systems, Applications, and Services*, pages 197–210 (2012).

[Wang and Mendel, 1992] L.-X. Wang and J. M. Mendel, “Generating fuzzy rules by learning from examples”. *IEEE Transactions on Systems, Man and Cybernetics*, volume 22(6), pages 1414–1427 (1992).

[Want et al., 1992] R. Want, A. Hopper, V. Falcão and J. Gibbons, “The active badge location system”. *ACM Transactions on Information Systems*, volume 10(1), pages 91–102 (1992).

[Widyawan et al., 2008] Widyawan, M. Klepal and S. Beauregard, “A novel backtracking particle filter for pattern matching indoor localization”. In *Proceedings of the ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pages 79–84 (2008).

[Woodman and Harle, 2008] O. Woodman and R. Harle, “Pedestrian localisation for indoor environments”. In *Proceedings of the International Conference on Ubiquitous Computing*, pages 114–123 (2008).

[Wu et al., 2007] B.-F. Wu, C.-L. Jen and K.-C. Chang, “Neural fuzzy based indoor localization by Kalman filtering with propagation channel modeling”. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 812–817 (2007).

[Wu et al., 2012] C. Wu, Z. Yang, Y. Liu and W. Xi, “WILL: Wireless indoor localization without site survey”. In *Proceedings of the IEEE International Conference on Computer Communications*, pages 64–72 (2012).

[Yang and Chen, 2009] J. Yang and Y. Chen, "Indoor localization using improved RSS-based lateration methods". In *Proceedings of the IEEE conference on Global telecommunications*, pages 4506–4511 (2009).

[Yang et al., 2012] Z. Yang, C. Wu and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention". In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pages 269–280 (2012).

[Yim, 2008] J. Yim, "Introducing a decision tree-based indoor positioning technique". *Expert Systems with Applications*, volume 34(2), pages 1296–1302 (2008).

[Yim et al., 2008] J. Yim, C. Park, J. Joo and S. Jeong, "Extended Kalman filter for wireless LAN based indoor positioning". *Decision Support Systems*, volume 45(4), pages 960–971 (2008).

[Youssef and Agrawala, 2003] M. Youssef and A. Agrawala, "Small-scale compensation for WLAN location determination systems". In *Proceedings of the IEEE Wireless Communications and Networking*, volume 3, pages 1974–1978 (2003).

[Youssef and Agrawala, 2008] M. Youssef and A. Agrawala, "The Horus location determination system". *Wireless Networks*, volume 14(3), pages 357–374 (2008).

[Youssef et al., 2003] M. Youssef, A. Agrawala and A. U. Shankar, "WLAN location determination via clustering and probability distributions". In *Proceedings of the IEEE Pervasive Computing and Communication*, pages 143–150 (2003).

[Zadeh, 1965] L. A. Zadeh, "Fuzzy sets". *Information and Control*, volume 8, pages 338–353 (1965).

[Zadeh, 1973] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes". *IEEE Transactions on Systems, Man and Cybernetics*, volume SMC-3(1), pages 28–44 (1973).

[Zàruba et al., 2007] G. Zàruba, M. Huber, F. Kamangar and I. Chlamtac, "Indoor location tracking using RSSI readings from a single Wi-Fi access point". *Wireless Networks*, volume 13(2), pages 221–235 (2007).

[Zhang et al., 2011] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao and H. Zheng, "I am the antenna: Accurate outdoor AP location using smartphones". In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pages 109–120 (2011).

[Zhao and Zhang, 2011] Z. Zhao and X. Zhang, "An RFID-based localization algorithm for shelves and pallets in warehouse". In *Proceedings of the International Conference on Transportation Engineering*, pages 2157–2162 (2011).