Localization and Path Tracking Using the IEEE 802.11 Infrastructure

(M.Sc. Thesis)

Antonia Papakonstantinou

Paris

May 2007

Department of Computer Science
Ecole Polytechnique, INRIA (Project HIPERCOM), University of Paris
Sud 11 and University of Crete

Localization and Path Tracking Using the IEEE 802.11 Infrastructure

Submitted to the
Department of Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science

May, 2007

Author: _____
Antonia PAPAKONSTANTINOU

Supervisor: _____
Philippe JACQUET
Professor

Member: _____
Khaldoun AL AGHA
Professor

Member: _____
Yannis MANOUSSAKIS
Professor

Paris, May 2007

Localization and Path Tracking Using the IEEE 802.11 Infrastructure

Antonia Papakonstantinou

M.Sc. Thesis

Department of Computer Science

Ecole Polytechnique, INRIA (Project HIPERCOM), University of Paris Sud 11 and University of Crete

Abstract

The deployment of mobile networks is combined with special services such as geolocalization inside public buildings. This geolocalization cannot be established by the classic ways of GPS, which does not work indoors. The geolocalization from GSM triangulation is not precise enough. This thesis propose the development of a geolocalization's protocol based on the mesh structure of the network and the application of the statistic methods known as fuzzy logic and particle filters. It is an improvement of the Kalman Filter method, without any constraint or assumption about the node mobility. The localization algorithm is based on an analytical drift evaluation made with signal strength measurements of the 802.11 infrastructure. At each step, the localization algorithm deduces the optimal correction in the mobile node's position based on the current estimated location, the signal measurements and the known signal distribution database. In fact, the algorithm relies on an analytical evaluation of the perturbation of the signal strength distribution with respect to the most likely path, which corresponds to the discrepancy between the real and estimated positions. At each step, the algorithm aims to correct this drift in the node position. Consequently, the corrections performed result in a random walk, which on average follows the real path taken by the mobile node, i.e. the expectation of the error in the estimated position is 0. The performance of our approach is evaluated by numerical simulations, as well as, by using real measurements.

Localisation et "path tracking" en utilisant l'infrastructure IEEE 802.11

Antonia Papakonstantinou

M.Sc. Thèse

Département de l'informatique

Ecole Polytechnique, INRIA (Project HIPERCOM), Université de Paris Sud 11 et Université de Crete

Résumé

Le déploiement de réseaux mobiles est combiné avec des services spéciaux, tels que la geólocalisation dans les bâtiments publics. Cette geólocalisation ne peut être établie par la méthode classique du GPS, qui ne fonctionne pas à l'intérieur de bâtiments. Par ailleurs, la géolocalisation effectuée en utilisant le réseau GSM n'est pas assez précise. Cette thèse propose le développement d'un protocole de géolocalisation basé sur la structure maillée du réseau sans fil local et l'application des méthodes statistiques de logique floue et de filtres à particules. Il s'agit d'une adaptation de la méthode du filtre de Kalman, sans aucune contrainte ou hypothèse sur la mobilité du noeud. L'algorithme de localisation utilise des mesures de puissance du signal de l'infrastructure du réseau 802.11. À chaque étape, l'algorithme déduit la correction optimale de la position du noeud mobile en se basant sur les mesures de signal et une base de données de la distribution du signal. Les corrections effectuées correspondent à une marche aléatoire, qui en moyenne suit le chemin pris par le noeud mobile. Par conséquent, l'espérance de l'erreur de la position estimée par l'algorithme de localisation est 0. Nous évaluons la performance de notre approche par des simulations numériques, ainsi qu'avec de mesures de signal.

Εντοπισμός θέσης και ιχνηλάτηση μονοπατιού με χρήση της υποδομής IEEE 802.11

Αντωνία Παπακωνσταντίνου

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών

Ecole Polytechnique, INRIA (Project HIPERCOM), Πανεπιστήμιο Paris Sud 11 και Πανεπιστήμιο Κρήτης

Περίληψη

Η επέκταση των κινητών δικτύων συνδυάζεται με ειδικές υπηρεσίες όπως το geolocalization μέσα σε δημόσια κτίρια. Αυτό το geolocalization δεν μπορεί να επιτευχθεί με τον κλασικό τρόπο του GPS, το οποίο δεν λειτουργεί στο εσωτερικό κτιρίων. Το geolocalization από triangulation GSM δεν είναι αρκετά ακριβές. Αυτή η διατριβή προτείνει την ανάπτυξη ενός πρωτοκόλλου geolocalization βασισμένου στη δομή του ασύρματου δικτύου και στην εφαρμογή των μεθόδων στατιστικής, γνωστών ως φίλτρα συγκεχυμένης λογικής και μορίων. Προτείνουμε μια βελτίωση της μεθόδου των φίλτρων του Kalman, χωρίς οποιαδήποτε περιορισμό ή υπόθεση για την κινητικότητα των κόμβων. Ο αλγόριθμος εντοπισμού στηρίζεται σε έναν αναλυτικό υπολογισμό της μέσης μετακίνησης του κινητού κόμβου, με βάση τις μετρήσεις ισχύως σημάτων της υποδομής του δικτύου 802.11. Σε κάθε βήμα, ο αλγόριθμος εντοπισμού συνάγει τη βέλτιστη διόρθωση στη θέση του κινητού κόμβου, βασιζόμενος στη τρέχουσα κατ΄ εκτίμηση θέση, τις μετρήσεις και τη βάση δεδομένων της κατανομής των σημάτων στο χώρο, που θεωρείται γνωστή. Συνεπώς, οι διορθώσεις που γίνονται οδηγούν σε έναν τυχαίο περίπατο, ο οποίος ακολουθεί κατά μέσον όρο την πραγματική πορεία του κινητού κόμβου, δηλ. η προσδοκία του λάθους στην κατ΄ εκτίμηση θέση είναι 0. Η απόδοση της προσέγγισής μας αξιολογείται με αριθμητικές προσομοιώσεις, καθώς επίσης και, με τη χρησιμοποίηση πραγματικών μετρήσεων.

Contents

List of Figures

Chapter 1

Introduction

During the last years, the progress of wireless communications and wireless networking has led to a widespread proliferation in mobile computing systems. Mobility creates a need for location aware applications, which aim to determine the physical position of a mobile device. With the recent thrive in Ubiquitous Computing, dozens of applications demand such location information. For example, asset tracking in warehouses and hospitals, locating elderly persons or children in public areas, emergency management, vehicles with navigation tools and location-based services, guiding visitors in museums, and virtual reality applications are becoming widely spread.

In outdoors, GPS, which uses the location of satellites orbiting Earth to triangulate longitude and latitude, has been used in many commercial applications. This approach is limited, however, to environments in which clear line of sight to the satellites is available. Inside buildings and even in outdoor environments, structure and foliage can affect the ability to communicate with GPS satellites.

Developments in the wireless technology have enabled creating applications that are aware of the user's location. These applications use location to provide relevant information or use it otherwise for the benefit of the user. Different location aware applications are meant for different environments and require different accuracy. While many outdoors applications, such as friend-finder, can successively work with accuracy of hundreds of meters, indoor applications, like guiding people indoors, usually require granularity of a few meters. The accurate localization of objects and people in indoor environments has long been considered an important building block for ubiquitous computing applications. Most research on indoor localization systems has been based on the use of short-range

signals, such as Wi-Fi [3, 4, 5], Bluetooth [6], ultrasound [7, 8], infrared [9], or RFID [10, 11].

Early localization systems were based exclusively on signal strength measurements and simple triangulation methods. However, the use of only signal strength data and simple triangulation methods for positioning can be limited, especially due to the interference and transient characteristics of the radio propagation. The dynamic characteristics of the environment impose important challenges, e.g. in submarines that use low radio frequency, for the design of a scalable, easily deployed, and computationally inexpensive localization system.

Due to environmental constrains, cost and maintenance, the deployment of a specialized infrastructure for a localization system is not feasible. We aim to design a localization system, that does not depend on a specialized hardware and also on an extensive training. Given the widely deployment of the IEEE 802.11 communication infrastructure, its use for both communication and positioning becomes a very attractive choice. So based on that signal strength measurements, an analytical drift evaluation is made. Our approach comprise an improvement of the Kalman Filter method, without any constraint or assumption about the node mobility.

The present thesis is organized as follows. Chapter 2 presents a definition of the localization systems and the most important localization techniques. In Chapter 3, we describe the methodology used in our approach. In Chapter 4, it is demonstrated the performance analysis of our system via simulations and real measurements. Finally, in Chapter 5, we discuss our main conclusions and future work.

Chapter 2

Background and Related Work

Location-sensing systems can be classified according to their dependency and use of an infrastructure, specialized hardware, signal modalities, training, methodology and models for estimating distances and position, coordination system (absolute or relative), location description (physical or symbolic), scale, device identification, classification, recognition, cost, privacy, and accuracy and precision requirements [12]. The distance can be estimated using *time of arrival* or *signal strength* measurements, if the velocity of the signal and a signal attenuation model for the given environment, respectively, are known. A result is considered *accurate*, if it is consistent with the *true* or *accepted* value for that result. Unlike accuracy, precision refers to the *repeatability* of measurement.

Several location sensing systems devices of different modality, such as radio (Radar [3], Ubisense), infrared (Active Badge [9]), ultrasonic (Cricket [8], Active Bat [13]), or vision (EasyLiving [14]) while others physical contact with pressure (Smart Floor), touch sensors or capacitive detectors to infer the position. If a training phase is feasible, they may create a signature of the geographic space with readings acquired at different positions and compare those readings with the reference ones using statistical analysis and pattern matching techniques.

2.1 Localization methods

The most common techniques in location-sensing are triangulation, proximity and scene analysis [12].

2.1.1 Triangulation

Triangulation-based techniques use the geometric properties of triangles to compute the location.

Lateration

Lateration-based techniques take advantage of the fact that the coverage areas of base stations usually overlap and mobile station can hear many base stations simultaneously. Knowing that the mobile station is located in the intersection of the areas of multiple base stations increases the accuracy of localization. The distance between the base and mobile station can be estimated by measuring the time it takes the signal to travel between them or the amount of signal attenuation along the way. The intensity of an emitted signal decreases as the distance from the emission source increases. The decrease relative to the original intensity is the attenuation. For example, at the free space model expressed in Fig. 2.1, $P_r = P_0(d/d_0)^{-n}$. Where n is the path-loss exponent, P_0 is the power at reference distance d_0 and P_r is the power at distance d.

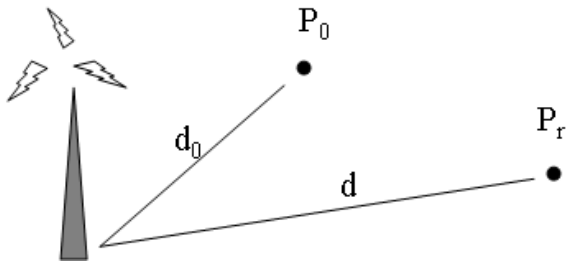


Figure 2.1: Attenuation.

Radio signals travel at the speed of light, so by knowing the time, the distance from the base station could be easily calculated. Knowing distances between the mobile station and at least three base stations, the actual location can be calculated. Each distance forms a circle around a base station. The intersection of three circles is the position of the mobile station. Popular methods based on this approach are Time of Arrival (ToA), Time Difference of Arrival (TDoA), Enhanced Observed Time Difference (E-OTD).

In case of ToA, the distance is derived from the absolute time for a radio signal to travel. The distance between a mobile object and a base station is measured by finding the Time of Arrival (ToA) which is the one-way propagation time between them, assuming that the transmission time is known. Geometrically, this is depicted with a circle, centered

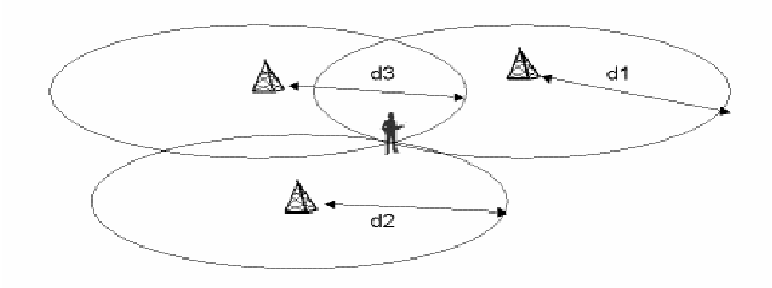


Figure 2.2: Localization based on Time of Arrival (ToA), source [1].

at the base station, on which the mobile object must lie. By using at least three base stations to resolve ambiguities, the position of the mobile object is at the intersection of the circles (Fig. 2.2).

ToA, however requires that the receiver knows the exact time of transmission. To overcome this requirement, round-trip time can be measured instead. TDoA measures time difference of the same signal at different cells. A basic precondition for the system to

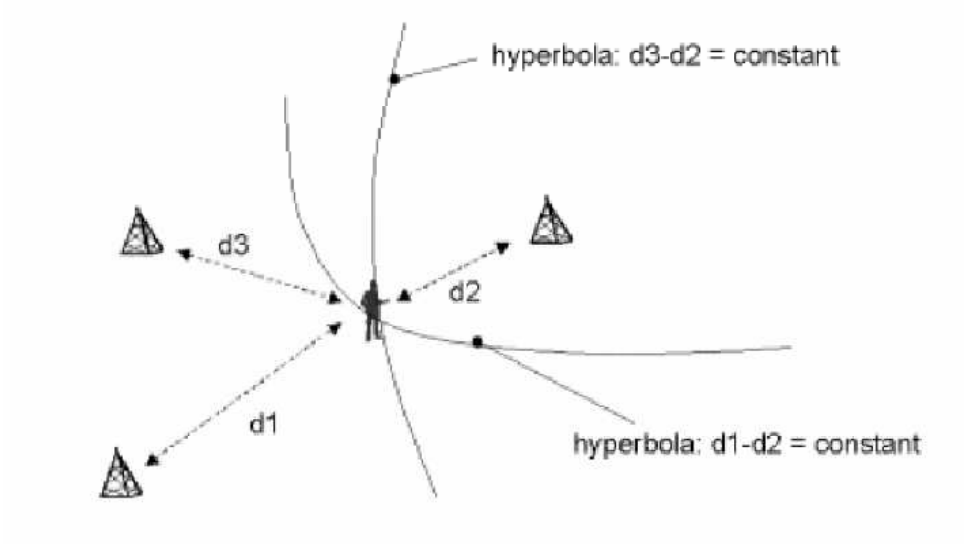


Figure 2.3: Localization based on Time Difference of Arrival(TDoA), source [1].

function is the absolute synchronization of the mobile object and the base station. When this is not true, instead of the absolute times, the Time Difference of Arrival (TDoA) is used, that is, the differences of the times of arrival in every base station, since it is much easier for the base stations to be synchronized. Since the hyperbola is a curve of constant time difference of arrival for two base stations, the time differences define hyperbolae, on which the mobile objects must lie. Hence, the location of the mobile object is at the intersection of the hyperbolae, as it is depicted in Fig. 2.3. nFurthermore, the motion is

supposed to be non relativistic which is not the case with satelittes. In the latter case, a relativistic correction of the geometry is required.

In E-OTD, base stations broadcast messages to mobile stations, which then compare the relative times of arrival to estimate its distance from stations.

Angulation

In addition to that, angulation tries to estimate the angle between the mobile station and the base station, increasing the accuracy even more.

The angle can be measured by base stations with directional antennas that detect the direction of the signal transmitted by the mobile station. If at least two base stations detect the angle (Fig. 2.4), the intersection of the lines formed by the angles identifies the two-dimensional location of the mobile station. This method is also referred to as Angle of Arrival (AoA).

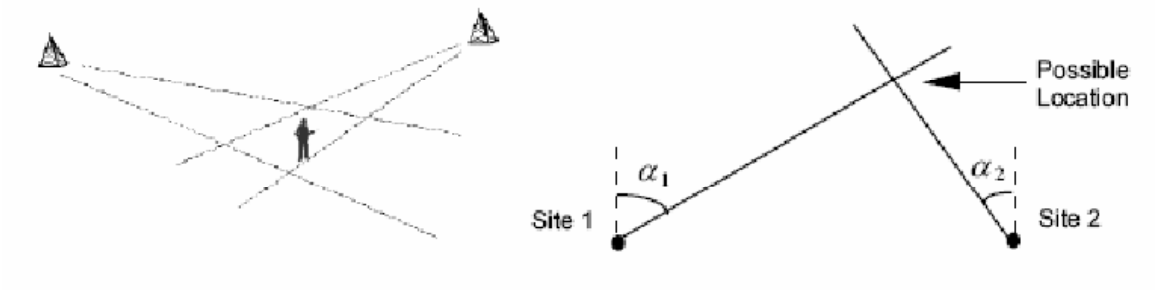


Figure 2.4: Localization based on Angle of Arrival (AoA), source [1].

In the absence of a line-of-sight (LoS) signal component, the antenna array will take a reflected signal that may not be coming from the direction of the mobile object. Even if a LoS component is present, multipath will still interfere with the angle measurement. The accuracy of the AoA method diminishes with increasing distance between the mobile object and the base station due to fundamental limitations of the devices used to measure the arrival angles. In some measurements that have taken place, this technique gives good results when it is used in macrocells, since the signals arrive with a relatively narrow AoA spread at the base stations. For microcells, this technique gives worse results.

Unfortunately, these methods are not very accurate in real life. In reality, radio signals are corrupted by unwanted random effects such as noise, interference from other sources, and interference between different radio channels. Signal propagation indoors is even more

complex. Indoor environments cause harsh multi-path effects, interference and dead-spots [5]. Thus, these methods work ideally only in line-of-sight conditions, where no obstructions are on the way of the signal, which is rarely the case indoors. They do not take into account complicated radio signal propagation and therefore lack the accuracy required for indoor positioning.

GPS

The most widespread and reliable system of localization is the Global Positioning System (GPS), which is constituted by a constellation of 24 satellites [2] (Fig. 2.1.1). It uses these satellites as reference points to calculate the position of an object with precision of meters. Substantially it gives in each cell-meter of planet a unique address. It is included in the satellite systems of auto-localization, since each user must have a special appliance GPS which is needed in the localization. Moreover, the optical contact with at least 3 satellites is essential. Its precision is better than 10 meters for military use and 100 meters for commercial one. The last years has been developed the differential GPS (Differential GPS) that improves considerably the precision in 2 with 5 meters. Due to variations in atmospheric conditions and clock deviation in satellites, errors are introduced in position calculations, resulting in a difference between the actual position and the calculated GPS position. Fixed reference stations knowing their exact position and equipped with GPS-receivers can constantly compare their real position with the position given by the GPS-system. The concept of Differential GPS is based on the principle that this considered error is valid for the position of the station, but also for positions in a wide range (up to hundreds of kilometers) around the station. Other GPS users within this range can use the error-information for improved accuracy. Note that error-information is losing its validity when time passes, because the error constantly changes as satellites are moving through space with varying atmospherically circumstances. Updating the error-information frequently is vital. A solution is to design special stations for transmitting error-information to specialized DGPS receivers. An alternative is to use the existing Long Range Navigation system (LoranC) which is installed world-wide, to transport the correctional data.

One of the basic disadvantages of GPS concerning the other systems of localization is that the appliance that is needed is quite expensive, bulky and consumes lot of energy. Consequently, it is disadvantageous in order to be incorporated in other appliances. To

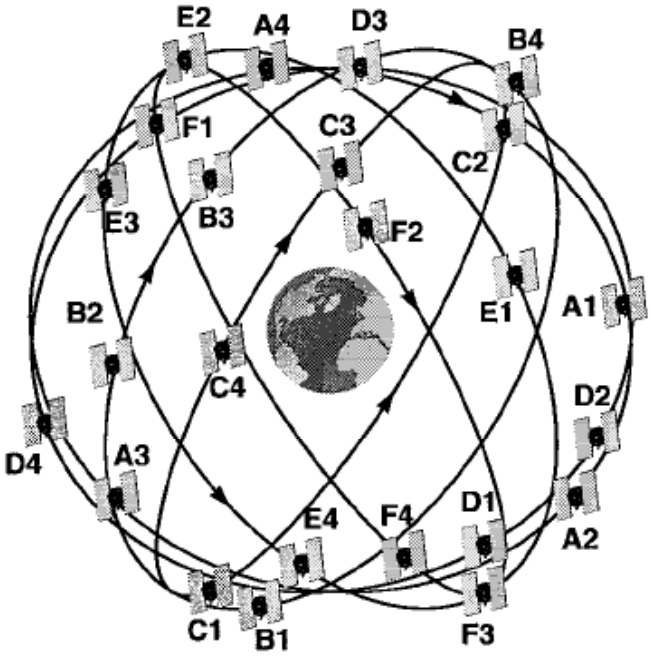


Figure 2.5: The GPS constellation consists of six orbital planes with four satellites in each plane. Each satellite is identified with a two-character code: a letter identifies the orbital plane (A through F) and a number identifies the satellite number in the plane (1 through 4), source [2]

obtain less than 10 meters accuracy, it needs to take precise measurements from at least 3 satellites, which is difficult to be achieved in big cities with the abundance of buildings. Consequently, an important disadvantage is that it cannot work in internal spaces, such as buildings.

2.1.2 Proximity

Detecting an object when it is near a known location through observed changes at that location. There are three general approaches to sense proximity:

- **Detecting physical contact**. Technologies for sensing physical contact include pressure sensors, touch sensors and capacitive field detectors [15].

- **Measuring wireless cellular access points**. Another implementation of the proximity location technique is monitoring when a mobile device is in range of one or more access points in a cellular network [9].

- **Observing automatic ID systems**. A third implementation of the proximity

technique uses automatic identification systems, such as credit card point-of-sale terminals,RFID badges, UPC scanning, computer login histories, land-line telephone records and electronic card lock logs. As a result the location of scanner, badge, computer, phone, identifies the location of an object.

2.1.3 Scene analysis

The scene analysis technique uses features of a scene observed from a particular vantage point used to infer the location of the observer. In *static* scene analysis, observed features are looked up in a predefined dataset that maps them to object locations. In contrast, *differential* scene analysis tracks the difference between successive scenes to estimate location. Differences in the scenes will correspond to movements of the observer. The scene itself can consist of visual images or any other measurable physical phenomena, such as the electromagnetic characteristics that occur when an object is at a particular position and orientation.

The advantage of scene analysis is that the location of objects can be inferred using passive observation and features that do not correspond to geometric angles or distances. The disadvantage of scene analysis is that the observer needs to have access to the features of the environment against which it will compare its observed scenes. Furthermore, changes to the environment in a way that alters the perceived features of the scenes may necessitate reconstruction of the predefined dataset or retrieval of an entirely new dataset.

Fingerprinting

One approach to overcome the problem of signal propagation peculiarities is to teach them to the system. The varying signal strengths, propagation times or angles can be measured at different known locations and recorded. When a new point needs to be localized, these quantities can be compared to the ones encountered before. The new location can then be assumed to be close to the previously collected points that have similar signal characteristics. This technique is called fingerprinting and the collected signal characteristics are called fingerprints. Therefore, to localize a mobile device using fingerprinting, the current signal fingerprint has to be compared to the fingerprints collected during a training period whose locations are known.

Two factors account for the good performance of radio fingerprinting. The first is that

the signal characteristics observed by mobile devices exhibit considerable spatial variability. For example, a given radio source may be heard stronger or not at all a few meters away. The second factor is that these characteristics are consistent in time; for example a medium-weak signal from a given source at a given location is likely to be similar tomorrow and next week. In combination, this means that there is a radio profile that is feature-rich in space and reasonably consistent in time. Fingerprinting-based location techniques take advantage of this by capturing this radio profile for later reference.

The advantage of a fingerprinting based localization system is that it allows determining the location very accurately as all the signal propagation oddities can be taken into account. However, the more details are learned, the more vulnerable is this radio map to changes in the environment, such as moving furniture, construction of new buildings, weather conditions or even people and cars moving inside or outside the buildings. Therefore, this approach requires recalibration time after time to adapt to the changes in the environment. However, the parts of the environment that affect the signal propagation the most (buildings, walls) are usually stable, so recalibration is not needed often.

Fingerprinting approach can be used with different technologies (e.g., GSM, 802.11), and with different types of input data. Most common is to use signal strength measurements, times or angles of arrival, or combinations of these. Another important part of fingerprinting based localization method is the predictive algorithm. The role of this algorithm is to calculate the locations of new points by building a generalizing model that matches the training samples, but more importantly, is able to predict the location of the yet unseen samples with high accuracy. In other words, determining the location if the fingerprint is identical to one of the training points is trivial; the algorithm has to be able to estimate the location in all the other cases as well, for example if the user is in between the training locations. Possible predictive algorithms include k-Nearest Neighbors, Support Vector Machines, Neural Networks, or other machine learning algorithms for supervised learning [16, 17, 18].

Cell Identification

Most of the wireless radio networks make use of cellular architecture. This means that instead of one wide-range radio transmitter, many stations with smaller ranges are used. This allows more effective bandwidth and energy use. The general idea behind

cell identification method is that such small stations (or cells) transmit unique location information that is only heard by mobile stations in the radio range of the particular cell. Reading this information, the stations can extract their own location. The transmitted information can be in a form of explicit location, or as a unique identifier, which requires further matching of identifiers and explicit locations to make the information useful.

Obviously, the accuracy of this approach depends on the size of cells. Unfortunately the optimal cell size of many technologies is quite large. In case of cellular telephony systems such as GSM, the cell size can be up to tens of kilometers [19]. Wireless local area networks such as 802.11 also use cellular organization, but cells are much smaller, usually up to hundreds of meters [20]. However, the usefulness of this approach also depends on the area that is covered by the base technology, i.e. the geographical area where any of the cells can be heard and thus location determined.

In the case of GSM, the Cell Identification (CI) method relies on the fact that a cell phone is constantly aware of the cell ID it is currently using. The size of cells is usually smaller (up to tens of meters) in urban areas and much larger (up to tens of kilometers) in rural environments [19], as Fig. 2.6 shows. CI's accuracy can be improved by TA (Timing Advance) [21]. TA is a delay time used to adjust the transmission timing to the propagation delay between cell phone and cell station that are farther away. In practice, TA is a discrete parameter, each unit of which represents about 500 meters.

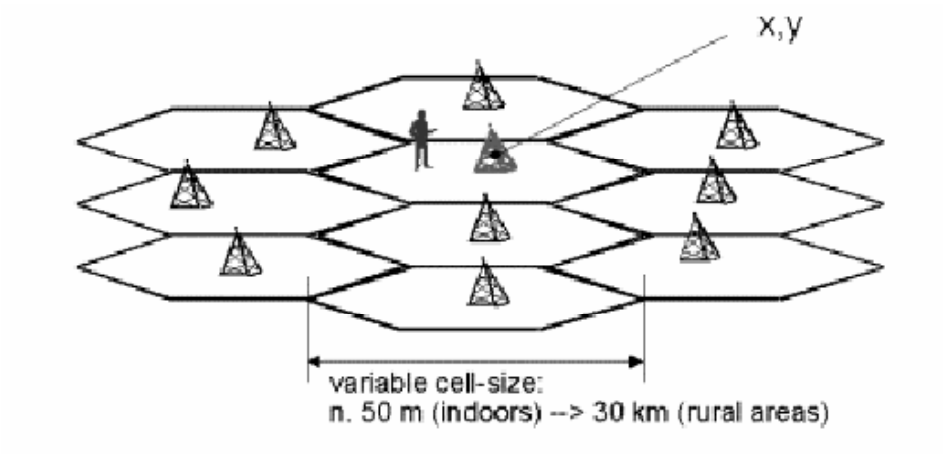


Figure 2.6: Cell Identification, source [1].

Filters

There are several methods for filtering the data. The Kalman filter and particle filter have been used to filter data for robot localization.

Since first introduced as a method for robot localization, Kalman filtering has been employed by many systems to localize [22, 23]. Originally introduced in 1960, the Kalman filter assumes a multivariate Gaussian distribution [24]. The Kalman filter has the advantage that the representation of the distribution is compact; a Gaussian distribution can be represented by a mean and a covariance matrix. The robots pose estimation is maintained as a Gaussian distribution and sensor data from dead reckoning and landmark observations is fused to obtain a new position distribution.

Recent extensions of Kalman filtering allow for non-Gaussian, multimodal probability distributions through multiple hypothesis tracking. The result is a more versatile estimation technique that still preserves many of the computational advantages of the Kalman filter. Monte Carlo localization, or particle filtering, provides a method of representing multimodal distributions for position estimation [25, 26], with the advantage that the computational intensity remains the same regardless of the localization area. The main advantage here is the ability to recover robustly from a poor initial condition.

2.2 Related Work

2.2.1 Indoor Localization

Indoor location systems have been successfully built using a variety of technologies. The Active Badge [9], PARCTab [27] and follow on commercial systems like Versus [28] used infrared emitters and detectors to achieve 5-10 meter accuracy. Both the Cricket [7, 8] and Active Bat [13] used ultrasonic signals to estimate location. Depending on the density of infrastructure and degree of calibration, ultrasonic systems have accuracies from a few meters to a few centimeters. Radio Frequency ID (RFID) technology has been used in research systems, such as SpotON [10] and Landmark [11] and commercial solutions like PinPoint [29, 30] to perform three-dimensional localization using signal strength measurements.

Active Badge

Active Badge [9] was proposed in 1992. It's solution to the problem of automatically determining the location of an individual was to design a wearable tag that emits diffuse infrared pulses every 10 seconds. These signals were then picked up by ceiling-mounted sensors around the building. A master station polled the sensors for badge sighting and processed the data, making it available for client applications. The emitted signals operated in approximately 6 meters range, and did not travel through the walls. Thus, sensors needed to be installed at least in every room, more than one to bigger or more complicated rooms. People had to wear special badges. Because the signals were transmitted through an optical path, the badges had to be worn outside of the clothing, preferably clipped to a shirt or a blouse. Sensors needed to be placed high up on walls or ceiling tiles of offices and on the entrances and exits of corridors and other public areas. The total cost of sensors, badges, cabling and installation was high, especially when large buildings had to be covered. It was not expected that sensors had overlapping coverage areas. Even if multiple sensors received signals from the same badge, this information was not used to increase accuracy.

Active Bat

Active Bat [13] uses an ultrasound time-of-flight technique to provide accurate physical positioning. Now, users and objects must carry Bat tags, which emit an ultrasonic pulse. The Active Bat has location accuracy of nine centimeters in 95% of the measurements. It needs one base station in every 10 square meters. The Bat makes 25 computations in every sec for every room. However, it requires large infrastructure (ceiling sensor grids) and maintenance cost.

Cricket

Cricket [7, 8], developed in MIT in 2000, is a location-support system for indoor location aware applications. It allows mobile and static nodes to learn their physical location by using listeners that hear and analyze information from beacons located throughout the building. On the contrary to the Active Badge, devices carried by users infer the location, not the central server. Thus the device controls the location information and can determine to whom it actually publishes it and to what extent. This alleviates privacy concerns, but

on the other hand, makes it impossible to create applications that require guaranteed location (e.g., location based billing).

Cricket uses beacons that send location information to listeners. Every device has a listener attached to it. Devices measure the one-way propagation time of the ultrasonic signals emitted by a beacon. A beacon sends information over radio frequency, together with an ultrasonic pulse. The additional radio frequency signals are used for synchronization of the time measurement and to distinguish ultrasound signals that stem from multi-path effects and ignore them. Cricket has simpler hardware and infrastructure than the aforementioned systems, but worse accuracy.

Spot-On

In Spot-On system [10], special tags use radio signal attenuation to estimate distance between tags. In this system, there are multiple base stations, which provide signal strength measurements and a central server which aggregates this data. The aim is to localize wireless devices relative to one another, rather than to fixed base stations, allowing for ad-hoc localization. Its accuracy depends on the cluster size, and every cluster must have at least two tags.

2.2.2 Indoor Localization Using 802.11 Fingerprinting

RADAR

RADAR [3] was the first attempt to use fingerprinting and an existing 802.11 infrastructure for localization. Instead of utilizing special equipment like infrared or ultrasound sensors and badges, RADAR used wireless networks already deployed in the building to localize hosts. The hosts periodically broadcast packets to the network and access points measured the signal strengths of these packets. The collected signal strength data was used to train the system and to determine later the location of a mobile host.Specifically, RADAR maintains a database of (x, y, z, SS_i) values, where x, y, and z are the physical coordinates of the training sample and SS_i is the signal strength measurements from the i-th access point (AP) for $i = 1, \quad \ldots, n$. Each measured signal strength vector is then compared against the database and the coordinates of the best matches are averaged to give the solution. The RADAR approach offers two advantages: it requires only a few

base stations and uses the same infrastructure that provides the buildings general-purpose wireless networking.

Improvements to 802.11 Fingerprinting

In this section we describe further works closer to our own. In their subsequent report [31] Bahl et al. proposed a number of enhancements to the RADAR system. Specifically, they describe a Viterbi-like algorithm for continuous user tracking. This algorithm takes into account the mobility pattern of the user to disambiguate between candidate user locations guessed by the system. They also describe access point-based environmental profiling scheme, where they automatically switch between two sets of fingerprints, taken in different environmental conditions (busy hour, non-busy hour).

Ladd et al. [5] have increased the accuracy of 802.11 fingerprinting by applying standard approaches from robotics-based localization, notably the explicit manipulation of noise distributions and the modelling of position as a probability distribution.

Zaruba *et al.* [32] performed localization based on the Received Signal Strength Indication (RSSI) as the only sensor reading. Based on the SS data gathered from 2 APs at various predefined points and orientations of an indoor environment, they built a grid-based signal strength map. They used the SIR algorithm of the particle filters approach. In particularly, they used 3000 particles in an area of 88 cells, where they had at most 2.1 meters mean location error. In their real time tests, the mean location error was at most 6 meters using 5000 particles. Moreover, their training phase is quite extensive and time-consuming since it takes place at each cell of their map and for every orientation.

Evennou and Marx [33] employed Kalman and particle filters. To build the SS map, the Motley-Keenan propagation model was considered, acquiring one measurement in each room and one measurement in every 2 meters in the corridor in a 35 x 35 m^2 area with 4 APs placed at each corner. Kalman and particle filters reported a mean positioning accuracy of 2.29 meters and 1.86 meters, respectively. However, sometimes after a re-sampling step all the particles were trapped in a part of the building and always bumped into a wall after then. In [34] they used additional information from Inertial Navigation Systems (INS) and they accomplished a mean location error equal to 1.53 meters using 10000 particles.

Hightower and Borriello [35] also applied particle filters to location estimation for an

indoor environment. Their implementation is based on the Sequential Importance Sample with Resampling (SISR) algorithm. The appropriate number of samples is determined at each step using a procedure called KLD adaptation. This work belongs to the field of robotics. The authors used a robot to take their measurements. Their observation step was consisted not only from the values from a WiFi client device, but also from an ultrasound badge, two types of infrared badges and RFID tags. It was tested in a 900 m^2 office building and the speed of the robot was ranged from [0-2] m/sec. The estimated position was computed as the weighted mean of its samples. The 80% location error was at most 1.8 meters. They also presented performance results showing that it is practical to run particle filters on devices ranging from high-end servers to handhelds.

In [36], the authors study the use of Wi-Fi as a localization sensor for mobile robots. Armed only with a SS map, a Wi-Fi adapter and odometry, robots can localize themselves using a variant of the standard Monte Carlo Localization algorithm. They encode the SS map using a regular grid and they compare the localization results achieved using three different combinations of sensors: Wi-Fi, contact and Wi-Fi plus contact sensing.

The [37] studies the use of Wi-Fi as a localization sensor both indoors and outdoors. They have introduced a hierarchical Bayesian technique for learning local Gaussian likelihood models of signal strength. Their sensor model is then integrated into a graph-based location estimation system.

In [38], they studied the WLAN user location estimation problem in the machine learning framework in which the physical properties of the wireless networks are not considered directly, but the problem is solved by using an inductive inference procedure based on a set of training data, a database of signal measurements in known locations. They tried three different machine learning approaches, two probabilistic approaches, in which they used the Bayes rule to estimate the posterior distribution and the nearest neighbor method.

In [39], they use a small number of seed nodes that know their location whereby other nodes estimate their location from the messages they receive. In particular, they tested different scenarios including the case of mobile seeds. They adapt the Sequential Monte Carlo localization method and especially the particle filters' method.

2.2.3 Localization Using GSM Fingerprinting

Place Lab

Place Lab [40] provides wide scale localization by listening for the transmissions of wireless networking sources like 802.11 access points, fixed Bluetooth devices, and GSM cell towers. However, instead of relying on extensive training phase, they use a public database of measurements collected by people in volunteering basis. Many of these beacon databases can come from institutions that own a large number of wireless networking beacons. Companies, universities and departments often know the locations of their 802.11 access points since this information is commonly recorded as part of a deployment and maintenance strategy. Other sources of Place Lab mapping data are the large databases produced by the war-driving community. War-driving is the act of driving around with a mobile computer equipped with a GPS device and a radio (typically an 802.11 card but sometimes a GSM phone or Bluetooth device) in order to collect a trace of network availability.

They have used three methods for location calculation. A simple Centroid calculates the average coordinates of the beacons in range and uses this as estimation. Fingerprinting method takes also signal strength information into account. More complicated Bayesian Particle Filter method uses the information about previous locations of the user to pinpoint the location.

SkyLoc Floor

SkyLoc [41], a GSM-fingerprinting-based localization system that runs on a mobile phone and identifies the current floor of a user in tall multi-floor buildings. The innovative point in this system is the use of feature selection techniques for matching signal strength fingerprints. In addition, the authors showed that it works for different network operators and with different hardware. The SkyLoc system has two components, the PlaceLogger and the SkyLoc. The PlaceLogger performs the collection of the fingerprints, as the user visits several places and the SkyLoc matches the GSM measurements that it takes with them to the training collected by the PlaceLogger. The Euclidean distance was used to determine the floor. Firstly, the naive fingerprinting algorithm is presented, which in practice showed that not all the sources help in localization, because either they are too noisy, too stable

across the floors, or simply inappropriate. Thus, a machine learning technique, feature selection, was used.

2.2.4 Indoor Localization and Global Positioning System

Additionally, many cell phone manufacturers have integrated GPS [2] units into the phones. A technique called Assisted GPS (A-GPS) [42] is used to shorten the time it takes for MS to localize themselves. As GPS-antennae and chipsets get smaller and cheaper, it will soon be feasible to integrate GPS-hardware into mobile phones. AGPS is a DGPS implementation where the handset is mobile phone with integrated GPS-receiver and where the telecom network of the operator is used to communicate correctional DGPS data to the handset. Although accurate outdoors, these solutions are not very useful indoors or in urban canyons, because of the lack of line of sight (LoS) between phone and multiple satellites. Indoor GPS [43] installs expensive GPS repeaters inside buildings to make the GPS devices work. However, the technique is still based on trilateration, which does not consider complicated signal propagation inside buildings, and thus requires large empty rooms or huge number of repeaters to provide high accuracy.

Chapter 3

Methodology

In this section is extendedly described the methodology we used, which is inspired by P. Faurre's filtering algorithm[44]. We assume k fixed wireless nodes called anchored point (AP). We consider a mobile node A that moves in a plan. Our aim is to track this node's movement. We denote $z(t) = (x(t), y(t))$ the position of node A at time t. We denote $\mathbf{v}(t)$ the signal level sampling vector at time t: $\mathbf{v}(t) = (u_1(t), u_2(t), \ldots, u_k(t))$ where $u_i(t)$ is the signal level (in dB) by which node A receives from AP_i at time t. We assume that there is a database established prior to the measurement that provides the probability distribution of the signal level vector at any position $z = (x, y)$. For example and in order to simplify computations we will assume that at point z signal level vector is distributed according to a normal distribution with mean vector $m(z)$ and inverse covariance matrix $\mathbf{Q}_z$, that is the probability density of the distribution of $\mathbf{v}$ on vector $\mathbf{u}$ is $p(z, \mathbf{u}) = \sqrt{\frac{det(\mathbf{Q}_z)}{\pi^k}} \exp(-\frac{1}{2} \langle (\mathbf{u} - m(z))\mathbf{Q}_z(\mathbf{u} - m(z)) \rangle)$.

The main objective is to find a path $(z(0), z(1), \ldots, z(T))$ such that the product $\prod_{i=T}^{i=0} p(z(i), \mathbf{v}(i))$ is maximal, assuming sampling at integer multiple of time unit. If we denote $\ell(z, \mathbf{u}) = -\log p(z, \mathbf{u})$ this is equivalent to minimize the sum $\sum_{i=T}^{i=0} \ell(z(i), \mathbf{v}(i))$, or the integral $\int_0^T \ell(z(t), \mathbf{v}(t))dt$, assuming a continuous approximation of sampling times.

3.1 Path Optimization

In this section we explain how to proceed in the particular case where the signal characteristics are known only at specific locations.

3.1.1 Path optimization without constraint

If we want to optimize $\int_0^T \ell(z(t), \mathbf{v}(t))dt$ without constraint, by a small perturbation $z(t) + \delta(t)$ we should have a $\delta(t)^2$ perturbation:
$\int_0^T \ell(z(t) + \delta(t), \mathbf{v}(t))dt = \int_0^T \ell(z(t), \mathbf{v}(t))dt + \int_0^T \langle \delta(t).\nabla_z \ell(z(t), \mathbf{v}(t)) \rangle \, dt$,
therefore at minimum we should have $\langle \delta(t).\nabla_z \ell(z(t), \mathbf{v}(t)) \rangle = 0$ pointwise for all possible $\delta(t)$, that is $\nabla_z \ell(z(t), \mathbf{v}(t)) = 0$ or $m(z(t)) = \mathbf{v}(t)$. This last condition would imply that $z(t)$ will follow locations where $m(z(t)) = \mathbf{v}(t)$ which may imply unbounded speed $\dot{z}(t)$ or unrealistic teleportation (when $\mathbf{v}(t)$ is discontinuous).

3.1.2 Path optimization with speed constraint

Therefore we should introduce a constraint based on speed bound. For example we can assume that the speed at any given point x must be constant with only direction as a tunable parameter. In this case the condition $\delta(t)$ must be orthogonal to speed $\dot{z}(t)$ adds to condition $\langle \delta(t).\nabla_z \ell(z(t), \mathbf{v}(t)) \rangle = 0$: we should have $\dot{z}(t)$ proportional to $\nabla_z \ell(z(t), \mathbf{v}(t))$.

Therefore a path such that $\dot{z}(t)$ is always proportional to $\nabla_z \ell(z(t), \mathbf{v}(t))$ may be a good candidate path. But if $\mathbf{v}(t)$ has lot of variation that may pose a problem. For example one can assume that $z(i+1) = z(i) + \nabla_z \ell(z(i), \mathbf{v}(i))$. If $\mathbf{v}(t)$ is random, then $z(i)$ will evolve like a random walk.

The expression of the gradient is the following

$$\nabla_z \ell(z, \mathbf{v}) = \frac{1}{2}(\langle (\mathbf{v} - m(z)) \nabla_z \mathbf{Q}_z (\mathbf{v} - m(z)) \rangle - 2 \langle \nabla_z m(z) \mathbf{Q}_z (\mathbf{v} - m(z)) \rangle - \nabla_z \log\det(\mathbf{Q}_z) \tag{3.1}$$

Notice that $\log\det(\mathbf{Q}_z) = \mathrm{tr}(\log \mathbf{Q}_z)$ and therefore $\nabla_z \log\det(\mathbf{Q}_z) = \mathrm{tr}(\nabla_z \mathbf{Q}_z . \mathbf{Q}_z^{-1})$. We also have $\langle u \mathbf{Q}_z u \rangle = \mathrm{tr}(\mathbf{Q}_z u \otimes u)$ by simple equivalence of lecture.

3.1.3 Random walk optimization

In this section we will suppose that the algorithm for tracking the mobile position is the following $z(i+1) = z(i) + \nabla_z \ell(z(i), \mathbf{v}(i))$ or, with an arbitrary time step δt: $z(t+\delta t) = z(t) + \delta t \nabla_z \ell(z(t), \mathbf{v}(t))$. At the end many path will be candidate, the one that minimizes $\int_0^T \ell(z, \mathbf{v})dt$ will be the best approximation.

Since the evolution of a candidate path will look like a random walk, therefore it will

be interesting to know the average drift of this random walk. Assume that $\mathbf{v}$ is distributed according a normal distribution of inverse covariance matrix $\mathbf{Q}$ and mean m, i.e. $E(\mathbf{v}) = m$ and $E((\mathbf{v} - m) \otimes (\mathbf{v} - m)) = \mathbf{Q}^{-1}$. Therefore

$$E(\nabla_z \ell(z, \mathbf{v})) = \frac{1}{2}(\mathrm{tr}(\nabla_z \mathbf{Q}_z \mathbf{Q}^{-1}) + \langle (m - m(z)) \nabla_z \mathbf{Q}_z (m - m(z)) \rangle \\ -2 \langle \nabla_z m(z) \mathbf{Q}_z (m - m(z)) \rangle - \mathrm{tr}(\nabla_z \mathbf{Q}_z . \mathbf{Q}_z^{-1})) \tag{3.2}$$

First of all we notice that if $\mathbf{Q} = \mathbf{Q}_z$ and $m = m(z)$ that is $\mathbf{v}$ is distributed like the sampled signal at position z, then we have $E(\nabla_z \ell(z, \mathbf{v})) = 0$ (thanks to $\mathrm{tr}(\nabla_z \mathbf{Q}_z . \mathbf{Q}_z^{-1})$ which is cancelled by $\nabla_z \log\det(\mathbf{Q}_z)$). Now if $m = m(z + \delta t \dot{z}) = m(z) + \langle \dot{z} . \nabla_z m(z) \rangle \delta t$ and $\mathbf{Q} = \mathbf{Q}_{z + \dot{z}\delta t} = \mathbf{Q}_z + \langle \dot{z} . \nabla_z \mathbf{Q}_z \rangle \delta t$ we have at first order:

$$E(\nabla_z \ell(z, \mathbf{v})) = \frac{\delta t}{2}(\mathrm{tr}(\nabla_z \mathbf{Q}_z \mathbf{Q}_z^{-1} \langle \dot{z} . \nabla_z \mathbf{Q}_z \rangle \mathbf{Q}_z^{-1} - 2(\nabla_z m(z) \mathbf{Q}_z \langle \dot{z} . \nabla_z m(z) \rangle) = \mathbf{R}_z(\dot{z}\delta t) \tag{3.3}$$

with $\mathbf{R}_z$ a 2x2 matrix with expression:

$$\mathbf{R}_z = \frac{1}{2}(\mathrm{tr}(\nabla_z \mathbf{Q}_z . \mathbf{Q}_z^{-1} \nabla_z \mathbf{Q}_z . \mathbf{Q}_z^{-1}) - 2\mathrm{tr}(\mathbf{Q}_z \nabla_z m(z) \otimes \nabla_z m(z))) \tag{3.4}$$

Therefore in order to have the correct drift a good algorithm consists in taking at each step $z(t + \delta t) = z(t) + \mathbf{R}_z^{-1} \nabla_z \ell(z(t), \mathbf{v}(t))$, since $E(\mathbf{R}_z^{-1} \nabla_z \ell(z(t), \mathbf{v}(t))) = \dot{z}\delta t$. This implies that the expectation of the difference between the estimated and the real path is zero. Consequently, in order to perform localization of the mobile node, we need to evaluate $\mathbf{R}_z$ and $\nabla_z \ell(z(t), \mathbf{v}(t))$.

3.2 Applications and tries

3.2.1 Constant variance

We consider the simplest case that $\lambda(z) = \lambda$ is a constant. Therefore $\mathbf{R}_z^{-1} \nabla_z \ell(z, \mathbf{v}) = \Pi_{\nabla_z m(z)}(\mathbf{v} - m(z))$ where $\Pi_{\nabla_z m(z)}$ is the projection on the vector space generated by the pair $(\frac{\partial m(z)}{\partial x}, \frac{\partial m(z)}{\partial y})$. In this case we just have

$$\begin{aligned}\nabla_z \ell(z,\mathbf{v}) &= -\lambda \sum_{i=1}^{i=k} \nabla_z m_i(z) \otimes \nabla_z m_i(z) \\ &= -\lambda \begin{bmatrix} \sum_{i=1}^{i=k} (u_i - m_i) \frac{\partial m_i}{\partial x} \\ \sum_{i=1}^{i=k} (u_i - m_i) \frac{\partial m_i}{\partial y} \end{bmatrix} \end{aligned} \tag{3.5}$$

and

$$\begin{aligned}\mathbf{R}_z &= -\lambda \sum_{i=1}^{i=k} \nabla_z m_i(z).(\mathbf{v}_i - m_i(z)) \\ &= -\lambda \begin{bmatrix} \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial x} & \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} \\ \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} & \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial y}\frac{\partial m_i}{\partial y} \end{bmatrix} \end{aligned} \tag{3.6}$$

So the drift will be:

$$\mathbf{R}_z^{-1} \nabla_z \ell(z,\mathbf{v}) = \begin{bmatrix} \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial x} & \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} \\ \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} & \sum_{i=1}^{i=k} \frac{\partial m_i}{\partial y}\frac{\partial m_i}{\partial y} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{i=k} (u_i - m_i) \frac{\partial m_i}{\partial x} \\ \sum_{i=1}^{i=k} (u_i - m_i) \frac{\partial m_i}{\partial y} \end{bmatrix} \tag{3.7}$$

3.2.2 Non constant variance

We consider the simple case where $\mathbf{Q}_z = \lambda(z)\mathbf{I}$, that means that there is no covariance between the different APs and the variance is the same function of the position for every AP. In this case

$$\begin{aligned}\nabla_z \ell(z,\mathbf{v}) &= \frac{1}{2}\left(\nabla_z \lambda(z) \sum_{i=1}^{i=k} (\mathbf{v}_i - m_i(z))^2 - 2\lambda(z) \sum_{i=1}^{i=k} \nabla_z m_i(z).(\mathbf{v}_i - m_i(z)) - k\frac{\nabla_z \lambda(z)}{\lambda(z)} \right) \\ &= \frac{1}{2}\left(\sum_{i=1}^{i=k} (u_i - m_i)^2 \begin{bmatrix} \frac{\partial \lambda(z)}{\partial x} \\ \frac{\partial \lambda(z)}{\partial y} \end{bmatrix} - 2\lambda(z) \begin{bmatrix} \sum_{i=1}^{i=k} (u_i - m_i) \frac{\partial m_i}{\partial x} \\ \sum_{i=1}^{i=k} (u_i - m_i) \frac{\partial m_i}{\partial y} \end{bmatrix} - \frac{k}{\lambda(z)} \begin{bmatrix} \frac{\partial \lambda(z)}{\partial x} \\ \frac{\partial \lambda(z)}{\partial y} \end{bmatrix} \right) \end{aligned} \tag{3.8}$$

and

$$\begin{aligned}\mathbf{R}_z &= \frac{1}{2}\left(\frac{\nabla_z\lambda(z)}{\lambda(z)}\otimes\frac{\nabla_z\lambda(z)}{\lambda(z)} - 2\lambda(z)\sum_{i=1}^{i=k}\nabla_z m_i(z)\otimes\nabla_z m_i(z)\right)\\ &= \frac{1}{2}\left(\frac{k}{\lambda(z)^2}\begin{bmatrix}\frac{\partial\lambda(z)}{\partial x}\frac{\partial\lambda(z)}{\partial x} & \frac{\partial\lambda(z)}{\partial x}\frac{\partial\lambda(z)}{\partial y}\\ \frac{\partial\lambda(z)}{\partial x}\frac{\partial\lambda(z)}{\partial y} & \frac{\partial\lambda(z)}{\partial y}\frac{\partial\lambda(z)}{\partial y}\end{bmatrix} - 2\lambda(z)\begin{bmatrix}\sum_{i=1}^{i=k}\frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial x} & \sum_{i=1}^{i=k}\frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y}\\ \sum_{i=1}^{i=k}\frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} & \sum_{i=1}^{i=k}\frac{\partial m_i}{\partial y}\frac{\partial m_i}{\partial y}\end{bmatrix}\right)\end{aligned} \tag{3.9}$$

If we had different functions of the variance for every AP, the $\mathbf{Q}_z$ would be a diagonal array of the form:

$$\mathbf{Q}_z = \begin{bmatrix}\lambda_1(z) & 0 & \dots & 0\\ 0 & \lambda_2(z) & \dots & 0\\ \dots & \dots & \dots & \dots\\ 0 & 0 & \dots & \lambda_k(z)\end{bmatrix} \tag{3.10}$$

In that case, which is the most realistic assumption,

$$\begin{aligned}\nabla_z\ell(z,\mathbf{v}) &= \frac{1}{2}\left(\sum_{i=1}^{i=k}\nabla_z\lambda_i(z)(\mathbf{v}_i - m_i(z))^2 - 2\sum_{i=1}^{i=k}\lambda_i(z)\nabla_z m_i(z).(\mathbf{v}_i - m_i(z)) - \sum_{i=1}^{i=k}\frac{\nabla_z\lambda_i(z)}{\lambda_i(z)}\right)\\ &= \frac{1}{2}\left(\begin{bmatrix}\sum_{i=1}^{i=k}\frac{\partial\lambda_i(z)}{\partial x}(u_i - m_i)^2\\ \sum_{i=1}^{i=k}\frac{\partial\lambda_i(z)}{\partial y}(u_i - m_i)^2\end{bmatrix} - 2\begin{bmatrix}\sum_{i=1}^{i=k}\lambda_i(z)\frac{\partial m_i}{\partial x}(u_i - m_i)\\ \sum_{i=1}^{i=k}\lambda_i(z)\frac{\partial m_i}{\partial y}(u_i - m_i)\end{bmatrix} - \begin{bmatrix}\sum_{i=1}^{i=k}\frac{1}{\lambda_i(z)}\frac{\partial\lambda_i(z)}{\partial x}\\ \sum_{i=1}^{i=k}\frac{1}{\lambda_i(z)}\frac{\partial\lambda_i(z)}{\partial y}\end{bmatrix}\right)\end{aligned} \tag{3.11}$$

and

$$\mathbf{R}_z = \frac{1}{2}\left(\frac{\nabla_z\lambda_i(z)}{\lambda_i(z)}\otimes\frac{\nabla_z\lambda_i(z)}{\lambda_i(z)} - 2\sum_{i=1}^{i=k}\lambda_i(z)\nabla_z m_i(z)\otimes\nabla_z m_i(z)\right)$$

$$= \frac{1}{2}\begin{bmatrix}\sum_{i=1}^{i=k}\frac{1}{\lambda_i(z)^2}\frac{\partial\lambda_i(z)}{\partial x}\frac{\partial\lambda_i(z)}{\partial x} & \sum_{i=1}^{i=k}\frac{1}{\lambda_i(z)^2}\frac{\partial\lambda_i(z)}{\partial x}\frac{\partial\lambda_i(z)}{\partial y}\\ \sum_{i=1}^{i=k}\frac{1}{\lambda_i(z)^2}\frac{\partial\lambda_i(z)}{\partial x}\frac{\partial\lambda_i(z)}{\partial y} & \sum_{i=1}^{i=k}\frac{1}{\lambda_i(z)^2}\frac{\partial\lambda_i(z)}{\partial y}\frac{\partial\lambda_i(z)}{\partial y}\end{bmatrix} - \begin{bmatrix}\sum_{i=1}^{i=k}\lambda_i(z)\frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial x} & \sum_{i=1}^{i=k}\lambda_i(z)\frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y}\\ \sum_{i=1}^{i=k}\lambda_i(z)\frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} & \sum_{i=1}^{i=k}\lambda_i(z)\frac{\partial m_i}{\partial y}\frac{\partial m_i}{\partial y}\end{bmatrix} \tag{3.12}$$

Chapter 4

Evaluation

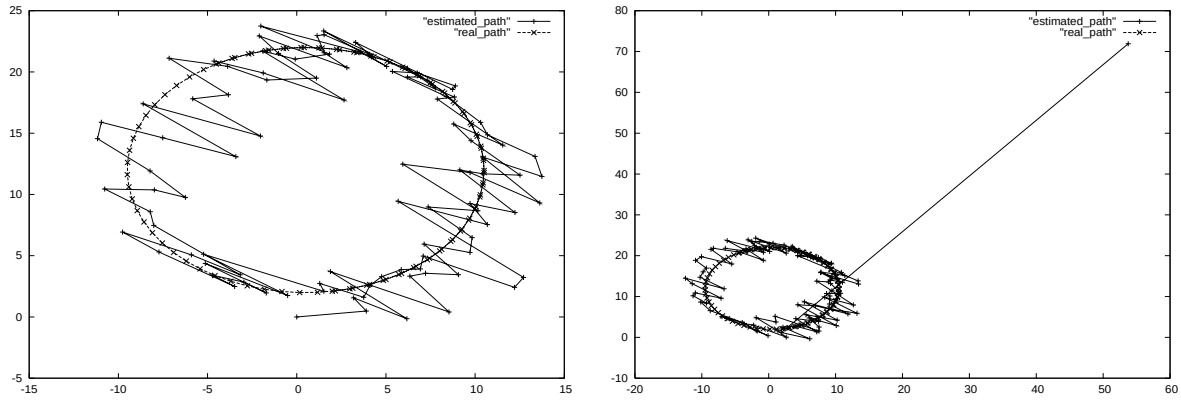
In this chapter we study the performance of our approach in simulations, when we have at first a constant variance of signal strength, λ and then when it is a function of the position, either the same for all APs, or different for every AP. In the last section of the chapter we test our approach with real environment measurements. We assume that there is a zero covariance between the signal strength of the different APs.

4.1 Constant variance

In this section we perform numerical simulations when the variance is a constant λ. We consider that we have three APs, so we have three different functions for the mean of every AP. At the beggining we tested linear functions, for instance $m_1(x,y) = x+y$, $m_2(x,y) = x+2y$ and $m_3(x,y) = 2x+3y$. More particularly, from Eq. (3.5) $\mathbf{R}_z = -\lambda \begin{bmatrix} 6 & 9 \\ 9 & 14 \end{bmatrix}$ and from Eq. (3.6) $\nabla_z \ell(z, \mathbf{v}) = -\lambda \begin{bmatrix} (u_1 - m_1) + (u_2 - m_2) + 2(u_3 - m_3) \\ (u_1 - m_1) + 2(u_2 - m_2) + 3(u_3 - m_3) \end{bmatrix}$.
So finally from Eq. (3.7) the drift would be

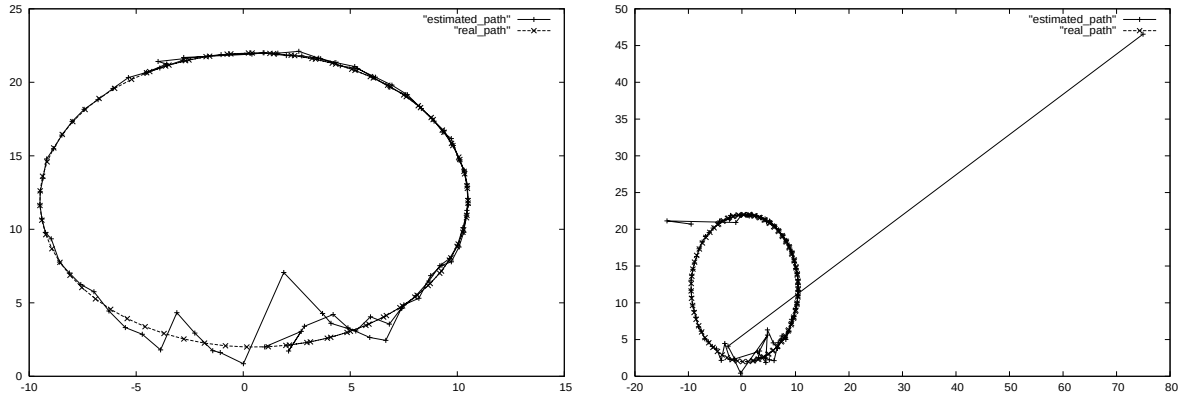
$$\begin{aligned} \mathbf{R}_z^{-1} \nabla_z \ell(z, \mathbf{v}) &= \begin{bmatrix} \frac{14}{3} & -3 \\ -3 & 2 \end{bmatrix} \begin{bmatrix} (u_1 - m_1) + (u_2 - m_2) + 2(u_3 - m_3) \\ (u_1 - m_1) + 2(u_2 - m_2) + 3(u_3 - m_3) \end{bmatrix} \\ &= \begin{bmatrix} \frac{5}{3}(u_1 - m_1) - \frac{4}{3}(u_2 - m_2) + \frac{1}{3}(u_3 - m_3) \\ -(u_1 - m_1) + (u_2 - m_2) \end{bmatrix} \end{aligned}$$

We also tried non linear, like $m_1(x, y) = x^3 + y$, $m_2(x, y) = \sin(x) + \cos(y)$ and $m_3(x, y) = x^3 + y^2$. For a circular path, Fig. 4.1 and Fig. 4.2 show that the estimated path follows sufficient enough the real path and even when we do not give the real initial position, but a random one, the estimated path converges the real path into one step.



(a) given initial position (b) not given initial position

Figure 4.1: Real path vs. estimated path with linear functions, when the variance is constant.



(a) given initial position (b) not given initial position

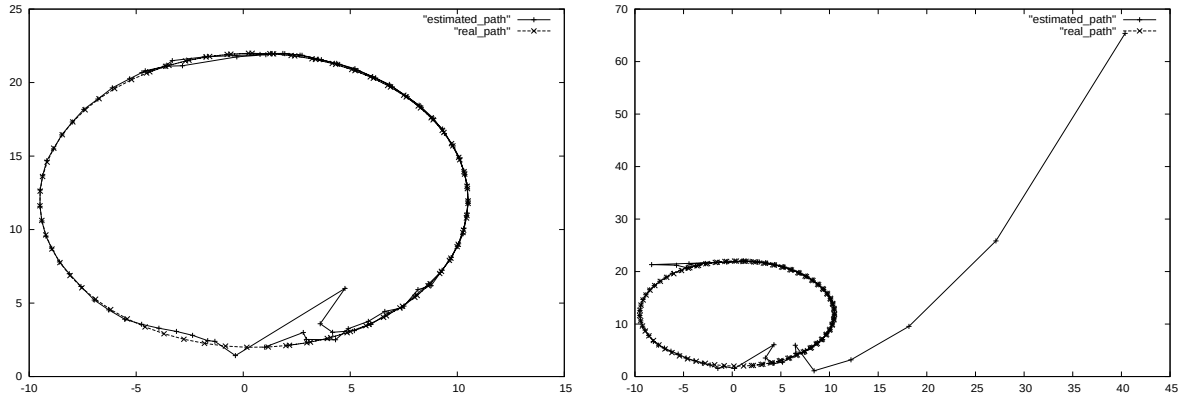
Figure 4.2: Real path vs. estimated path with non linear functions , when the variance is constant.

4.2 Non constant variance

4.2.1 Same variance for all APs

In this section the variance is not constant but it is the same for all three APs. We used the non linear functions of Section 4.1 for the mean, and for the variance the function

$\lambda(x, y) = 1000/(1000 + x)$. Fig 4.3 shows that even with a non constant variance the estimation is quite good. Even, when the initial position is chosen at random, the algorithm still converges after five steps.



(a) given initial position (b) not given initial position

Figure 4.3: Real path vs. estimated path with non linear functions with a variance $\lambda(x, y) = 1000/(1000 + x)$.

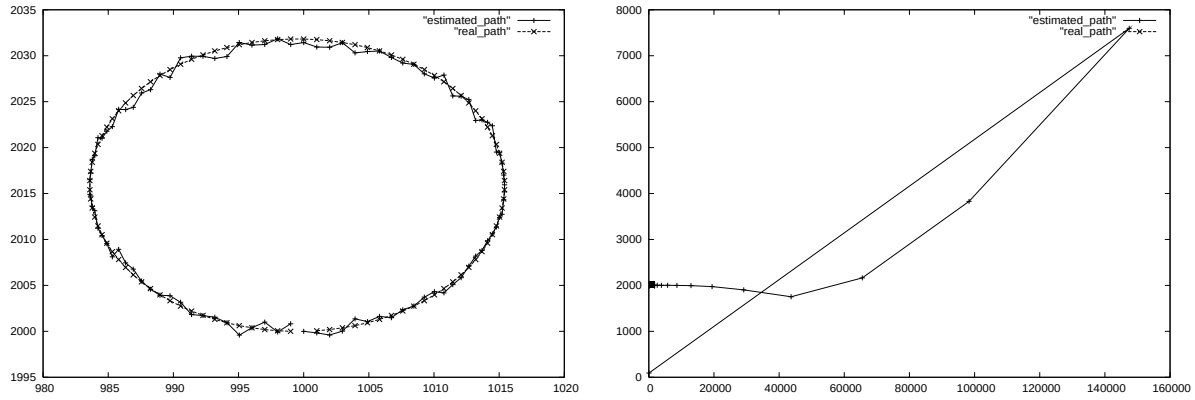
4.2.2 Different variance for every AP

In this section the variance is different for every AP. We used again the non linear functions of Section 4.1 for the mean, and for the different variances $\lambda_1(x, y) = 4000000$,$\lambda_2(x, y) = 1000 + 100y^{0.5}$ and $\lambda_3(x, y) = 1000/(1 + x/100)$. Fig 4.4 shows that the estimation is quite well when we give the initial position. However, when the initial position is random, Fig 4.4can give us any impression whether the algorithm still converges to the real path or not. So, we zoom to see finally in Fig 4.5 that it only takes 12 steps to converge.

4.3 Use of triangulation

In this section we consider that we have a database of signal strengths in some positions, so we have the mean and the variance of these selected positions only. Therefore, we discuss to use delaunay triangulation for the estimation of the mean and variance of every other position hosed on linear extrapolation.

The Computational Geometry Algorithms Library (CGAL), offers data structures and algorithms like triangulations (2D constrained triangulations and Delaunay triangulations



(a) given initial position (b) not given initial position

Figure 4.4: Real path vs. estimated path with non linear functions and different variance for each AP.

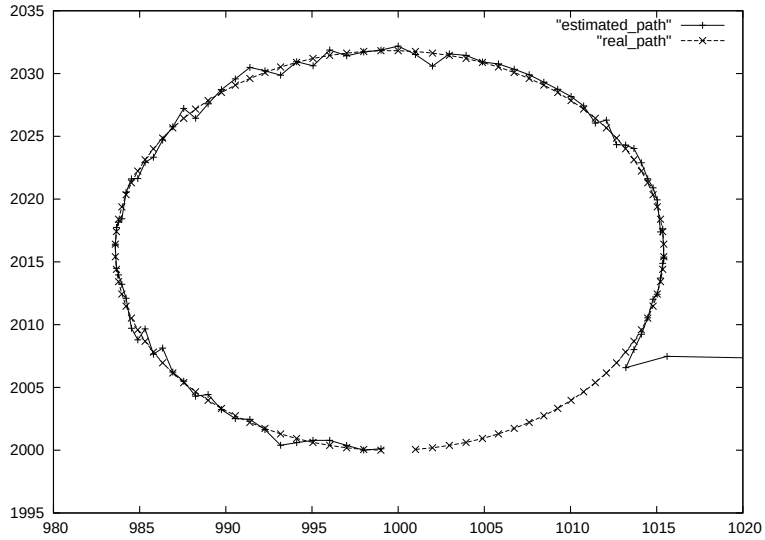
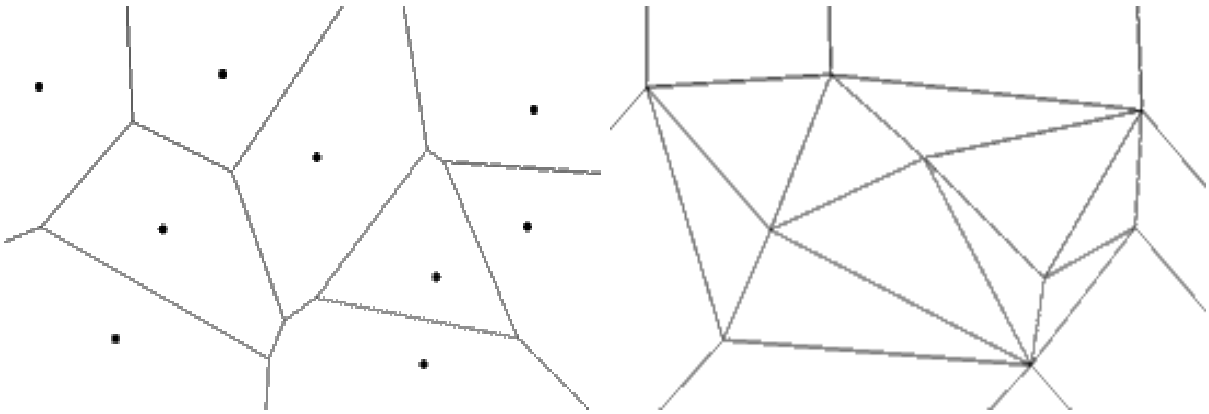


Figure 4.5: Zoom of the Fig 4.4(b).

in 2D and 3D), Voronoi diagrams (for 2D and 3D points, 2D additively weighted Voronoi diagrams, and segment Voronoi diagrams), Boolean operations on polygons and polyhedra, arrangements of curves, mesh algorithms (2D Delaunay mesh generation and 3D surface mesh generation, surface mesh subdivision and parameterization), alpha shapes, convex hull algorithms (in 2D, 3D and dD), operations on polygons (straight skeleton and offset polygon), search structures (kd trees for nearest neighbor search, and range and segment trees), interpolation (natural neighbor interpolation and placement of streamlines), optimization algorithms (smallest enclosing sphere of points or spheres, smallest enclosing ellipsoid of points, principal component analysis), and kinetic data structures. All these

data structures and algorithms operate on geometric objects like points and segments, and perform geometric tests on them. These objects and predicates are regrouped in CGAL Kernels. For that reason we used, CGAL Open Source Project, which provides easy access to efficient and reliable geometric algorithms in the form of a C++ library.

In mathematics, and computational geometry, a Delaunay triangulation for a set P of points in the plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation. In the general n-dimensional case it is stated as follows: For a set P of points in the (n-dimensional) Euclidean space, a Delaunay triangulation is a triangulation DT(P) such that no point in P is inside the circum-hypersphere of any simplex in DT(P). It is known that there exists a unique Delaunay triangulation for P, if P is a set of points in general position; that is, no three points are on the same line and no four are on the same circle, for a two dimensional set of points, or no n + 1 points are on the same hyperplane and no n + 2 points are on the same hypersphere, for an n-dimensional set of points. The Delaunay triangulation of a discrete point set P corresponds to the dual graph of the Voronoi tessellation for P. A Voronoi diagram, named after Georgy Voronoi, also called a Voronoi tessellation, a Voronoi decomposition, or a Dirichlet tessellation (after Lejeune Dirichlet), is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in the space. More particular, for any discrete set S of points in Euclidean space and for almost any point x, there is one point of S closest to x. The word "almost" is used to indicate exceptions where a point x may be equally close to two or more points of S.



(a) Voronoi diagram (b) Delaunay diagram

Figure 4.6: Both diagrams for the same set of points.

The Delaunay triangulation, although it is not optimal, gives always satisfactory (very close to the optimal) extrapolating estimations. Since we have the signal strength measurements of the edges of every triangle, triangulation will be used to extrapolate the values of every other point inside the triangle. A triangle can determined by the three points $\mathbf{p}_1 = (x_1, y_1, z_1)$, $\mathbf{p}_2 = (x_2, y_2, z_2)$ and $\mathbf{p}_3 = (x_3, y_3, z_3)$, where z_1, z_2, z_3 are the either the mean SS value or the variance of each point. If we have a point $P(x, y, z)$ inside the triangle, the plane passing through these three points can be determined by the following determinant equations:

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = \begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x - x_2 & y - y_2 & z - z_2 \\ x - x_3 & y - y_3 & z - z_3 \end{vmatrix} = 0. \tag{4.1}$$

From this equation, the value of z (mean or variance) of the point inside the triangle can be easily found.

$$z = z_1 - \frac{\begin{vmatrix} y_2 - y_1 & z_2 - z_1 \\ y_3 - y_1 & z_3 - z_2 \end{vmatrix}}{\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}} * (x - x_1) - \frac{\begin{vmatrix} z_2 - z_1 & x_2 - x_1 \\ z_3 - z_2 & x_3 - x_1 \end{vmatrix}}{\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}} * (y - y_1) \tag{4.2}$$

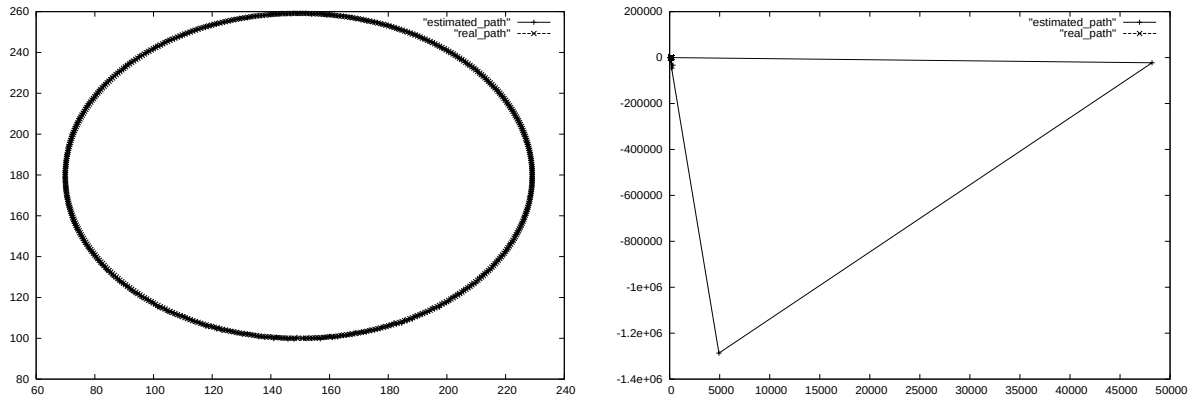
Similarly, we have for the gradient of z the following expression:

$$\nabla_z = -\left(\frac{\begin{vmatrix} y_2 - y_1 & z_2 - z_1 \\ y_3 - y_1 & z_3 - z_2 \end{vmatrix}}{\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}}, \frac{\begin{vmatrix} z_2 - z_1 & x_2 - x_1 \\ z_3 - z_2 & x_3 - x_1 \end{vmatrix}}{\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}} \right)$$

Simulation

The same functions of Section 4.2.2 were used for the mean and the variance, but only for the points used to perform the triangulation. The points form isosceles triangles and the grain of the triangulation is 25. Every time we need the mean or the variance of a point, we find in which triangle it is in, and we perform the previous equation. Fig. 4.7 demonstrates the results, which in the case where the initial position is not given, becomes

clear in Fig. 4.8.



(a) given initial position (b) not given initial position

Figure 4.7: Real path vs. estimated path with delaunay triangulation.

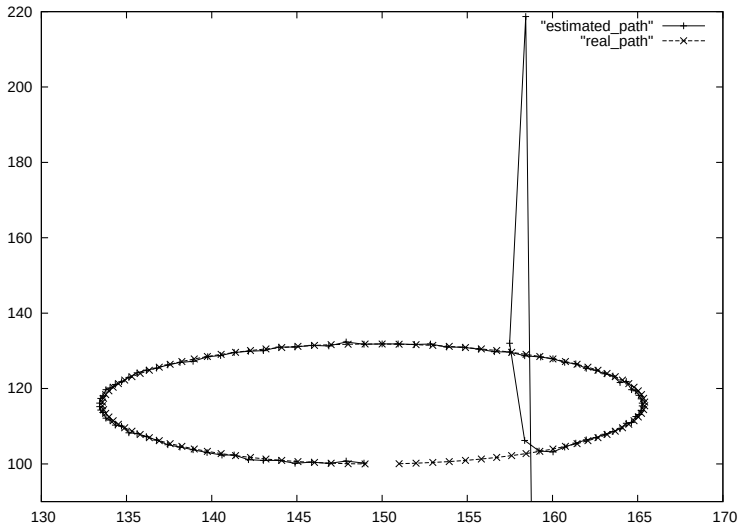


Figure 4.8: Zoom of the Fig 4.7(b).

Although, the previous case is optimal as the triangles are isosceles. So we also tried the case where the given points do not form such kind of triangles. Fig 4.9 shows the distribution of the points with given mean and variance. Those steps are placed randomly and their density are not equal everywhere.

In Fig 4.10(a) we have the estimated path when the initial position is not given and in Fig 4.10(b) when it is not given. It is easily observed that after four steps it follows the real step.

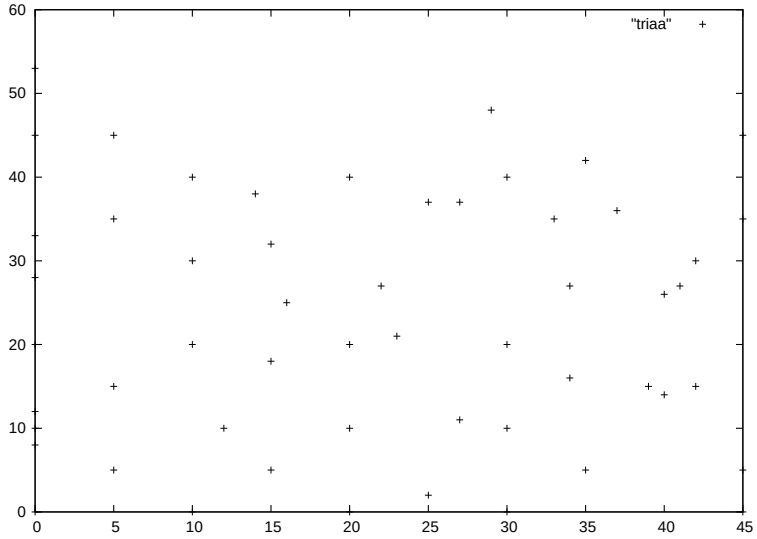
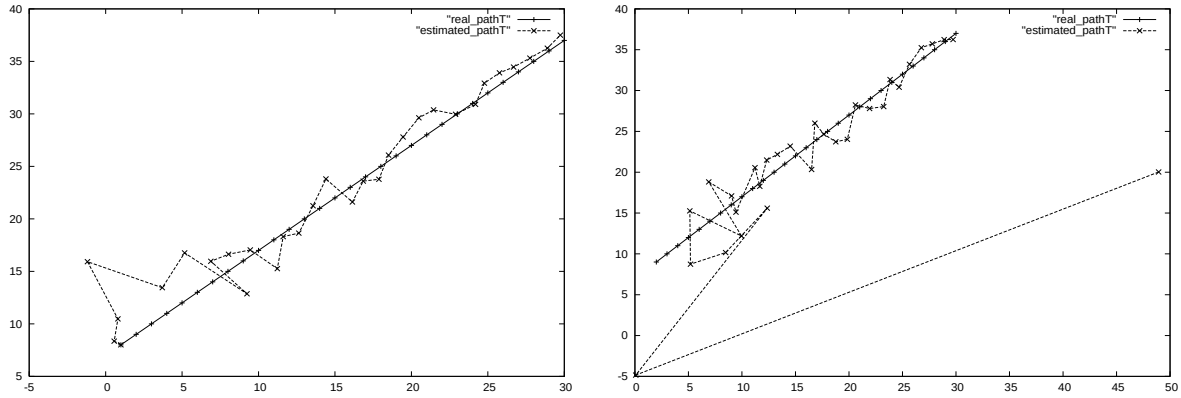


Figure 4.9: Distribution of the points.



(a) given initial position

(b) not given initial position

Figure 4.10: Real path vs. estimated path with delaunay triangulation when the triangles are not isosceles.

4.4 With real measurements

In this section we evaluate our approach by using real signal strength measurements. Firstly, we test it in a laboratory by giving the initial position. And then we used a combination of the particle filters and our approach, so as to have a better estimation of the initial position.

4.4.1 Training phase

Our testbed resides in the Networks and Telecommunications Laboratory of ICS-FORTH encompassing two corridors, one office and a common area with desks. The

floorplan, describing the layout of the testbed is shown in Fig. 4.11, in which dots mark the locations of the testbed. During the *training phase*, signal strength measurements were collected from the three APs using NetStumbler [45], a wireless network monitoring tool that measures and records the signal strength values from APs. NetStumbler is a freeware for Microsoft Windows used extensively in wardriving, verifying network configurations, and detecting areas with poor coverage and unauthorized access points. For each received packet broadcast from an AP, netstumbler records the signal strength measured at its reception. Such signal strength measurements were acquired from all the five APs at 38 different locations of our floorplan. 50 signal strength values were collected at each location.

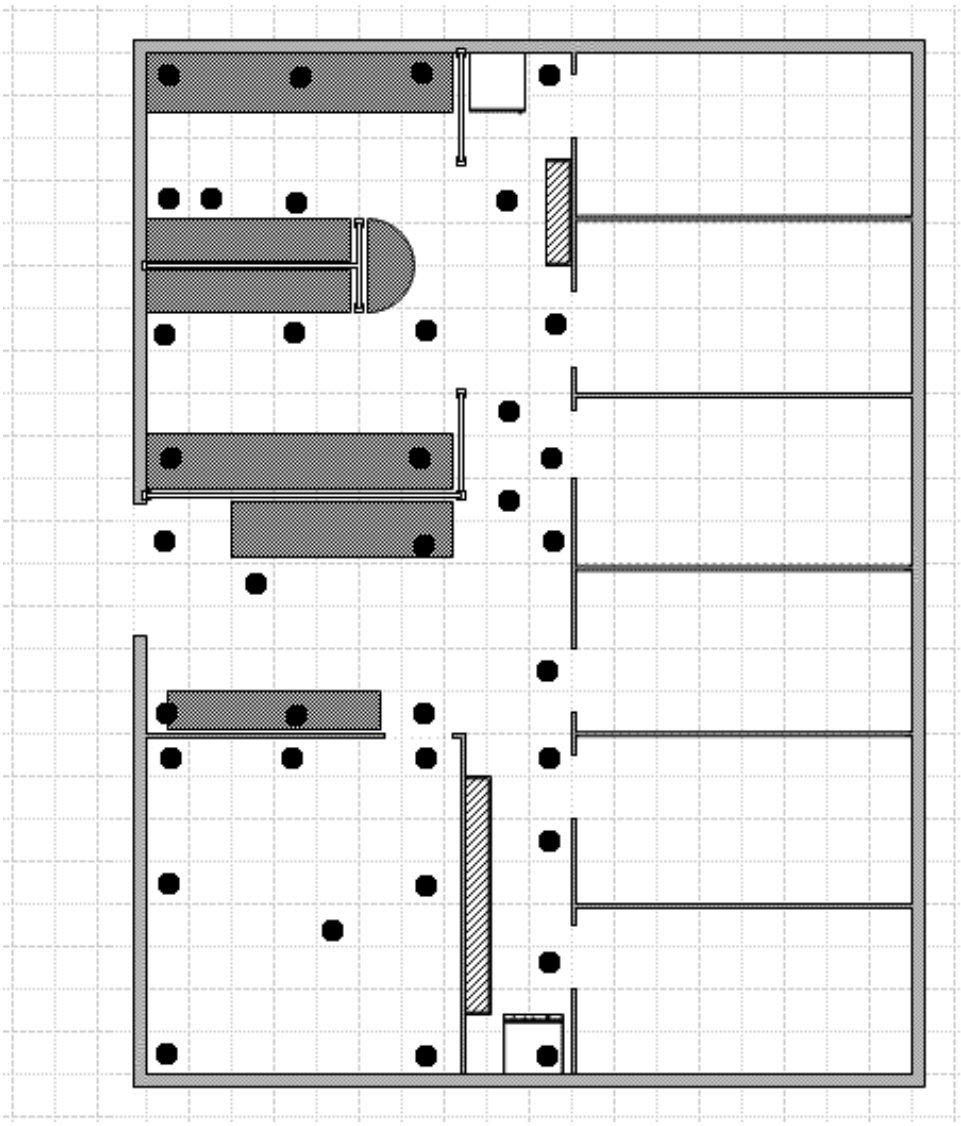


Figure 4.11: Networks and Telecommunications Laboratory of ICS-FORTH.

4.4.2 Path estimation when initial position is given

The path given and the estimated are depicted in Fig. 4.12. It is obvious that the estimated path follows the movement although there is a countable drift. We did not further make a statistic analysis on the results of different paths because the space our paths were tested is so small to change the aspect of the results. However, the right direction, that the estimated path gives, is efficient enough. On the other hand, we tried not to give the initial position and we did not get anything that can be considered as path. So, we tried to find the first position with the help of the particle filters.

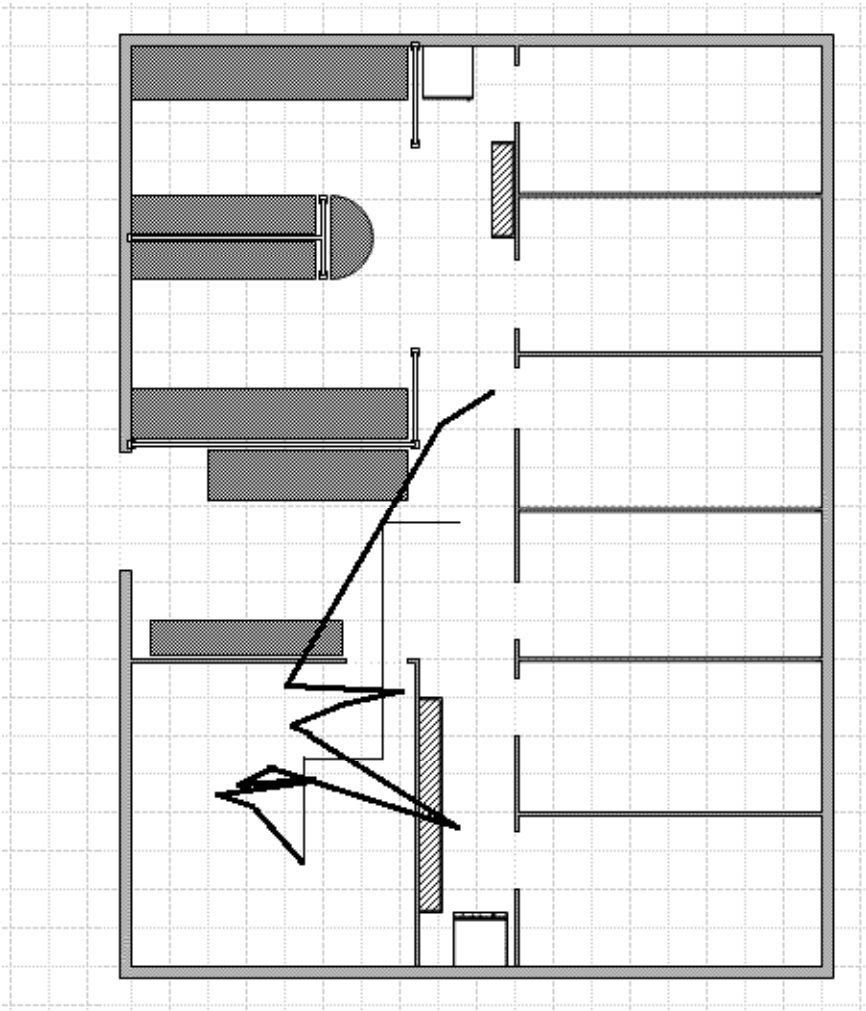


Figure 4.12: Real path vs. estimated when initial position is given. The real path is the fine line and the other is the estimated.

4.4.3 Theoretical framework of particle filters

In probabilistic terms, the location sensing problem can be considered as the problem of determining the probability of a node being at a certain state (location) x_k, at time instant $t = k$, given a sequence of observations (signal strength) $y_1..y_k$. Assuming first order Markov dynamics, the above problem can be formulated with the utilization of the network graph depicted in Fig. 4.13.

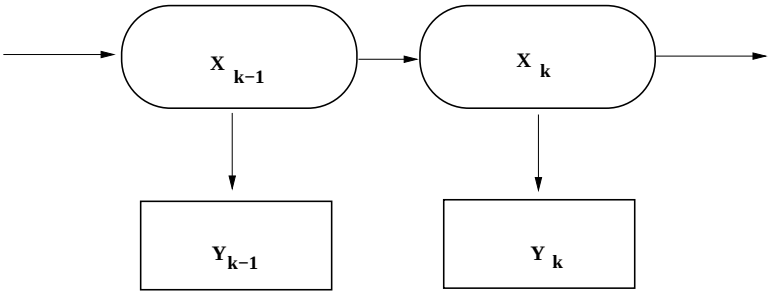


Figure 4.13: Markov chain. The squares indicate the state (location) x_k at time instant $t = k$ and the circles the observation (signal strength) y_k.

The node locations x_k cannot be observed directly (they are "hidden"). Yet, for each location x_k, a measurement vector y_k is available that depends on the hidden variable according to a known observation function. As can be observed from Fig. 4.13 and because of the Markov assumption, the true state is conditionally independent of all earlier states given the immediately previous state, that is $p(x_k|x_0, x_1, \ldots, x_{k-1})$. Similarly the measurement at the $k-th$ timestep is dependent only upon the current state and is conditionally independent of all other states given the current state, $p(y_k|x_0, x_1, \ldots, x_k) = p(y_k|x_k)$. In other words, each y_k only depends on x_k [26].

The location sensing task can then be formulated as: Compute the location x_k of the node at time instant k given the sequence of observations $y1, y2, ..., y_k$, up to time k (inference problem). That is, compute the posterior distribution $P(x_k|y_1, ..., y_k)$.

In order to estimate the probability of the posterior distribution which is actually a density over the whole state space of the model of figure, in this paper we utilize particle filters [46]. Particle filters is a technique for implementing a recursive Bayesian filter by Monte Carlo sampling, i.e., a technique that relies on the generation of a sample mass (particle) representation of the filtering distribution of $P(x_k|y_1, ..., y_k)$.

An inherent characteristic of particle filters that makes them very well suited for the location sensing task is that they impose no requirements on the format of the involved

distributions and noise models and the linearity of the involved functions. This is very significant in location sensing problems where the linearity and gaussian constraints of other popular, kalman filter based methods, would be very limiting.

In order to generate and maintain the samples (particles), we utilize of the sampling/importance resampling (SIR) algorithm introduced by Rubin in 1988 [47]. According to SIR the filtering distribution is approximated by a weighted set of particles, $(w_k^{(L)}, x_k^{(L)}) \ : \ L = 1, \ldots, P$, where P is the total number of particles, L their index and k the time. The importance weights $w_k^{(L)}$ are approximations to the relative posterior probabilities of the particles such that $\sum_{L=1}^{P} w_k^{(L)} = 1$.

The following pseudocode describes the implementation of the SIR algorithm for the specific task at hand.

The algorithm performance is dependent on the choice of the importance distribution $\pi(x_k|x_{0:k-1}, y_{0:k})$. The *Optimal importance distribution* is given as $\pi(x_k|x_{0:k-1}, y_{0:k}) = p(x_k|x_{k-1}, y_k)$.

However, the transition prior is often used as importance function, since it is easier to calculate, and also simplifies the subsequent importance weight calculations $\pi(x_k|x_{0:k-1}, y_{0:k}) = p(x_k|x_{k-1})$ and also $w_k^{(L)} = w_{k-1}^{(L)} p(y_k|x_k^{(L)})$. The method is often inefficient in simulations as the state space is explored without any knowledge of the observations. It is especially sensitive to outliers. However, it does have the advantage that the importance weights are easily evaluated.

(SIR) filters with transition prior as importance function are commonly known as bootstrap filter and condensation algorithm. Resampling is used to avoid the problem of degeneracy of the algorithm, that is, avoiding the situation that all but one of the importance weights are close to zero. The performance of the algorithm can be also affected by proper choice of resampling method.

The SIR implementation for the specific task at hand is described by the following algorithm.

In the *training phase*, the collected signal strength measurements were used. Each mobile node has a set of P particles. The particle filters' algorithm tends to gather the particles in an area that indicates the final estimated position of the node. Initially, the particles are randomly placed to the floorplan using the uniform distribution. In the observation phase, the particles receive SS values (either from an AP or from a peer). The

Algorithm 1 SIR algorithm

1: **For** $L = 1, \ldots, P$
2: Draw $x_k^{(L)} \sim p(x_k^{(L)}|x_{k-1}^{(L)})$ (transition step)
3: Calculate $w_k^{(L)} = w_{k-1}^{(L)} \cdot p(y_k|x_k^{(L)})$ (observation step)
4: **End For**
5: Normalize weights so that $\sum_{J=1}^{P} w_k^{(J)}$=1
6: Resample according to the normalized weights.
7: Set the new weights $w_k^{(L)} = \frac{1}{P}$

SS measurements from different APs are independent, thus the marginal distribution was calculated as $p(y_k|x_k) = p((y_{k1}, y_{k2}, \ldots, y_{kn})|x_k) = p(y_{k1}|x_k) \cdot p(y_{k2}|x_k) \cdot \ldots \cdot p(y_{kn}|x_k)$, where n the number of APs. For each AP i the $p((y_{ki}, x_k)$ was computed using the Gaussian probability function, as the SS measurements in a specific position follow the Gaussian distribution.

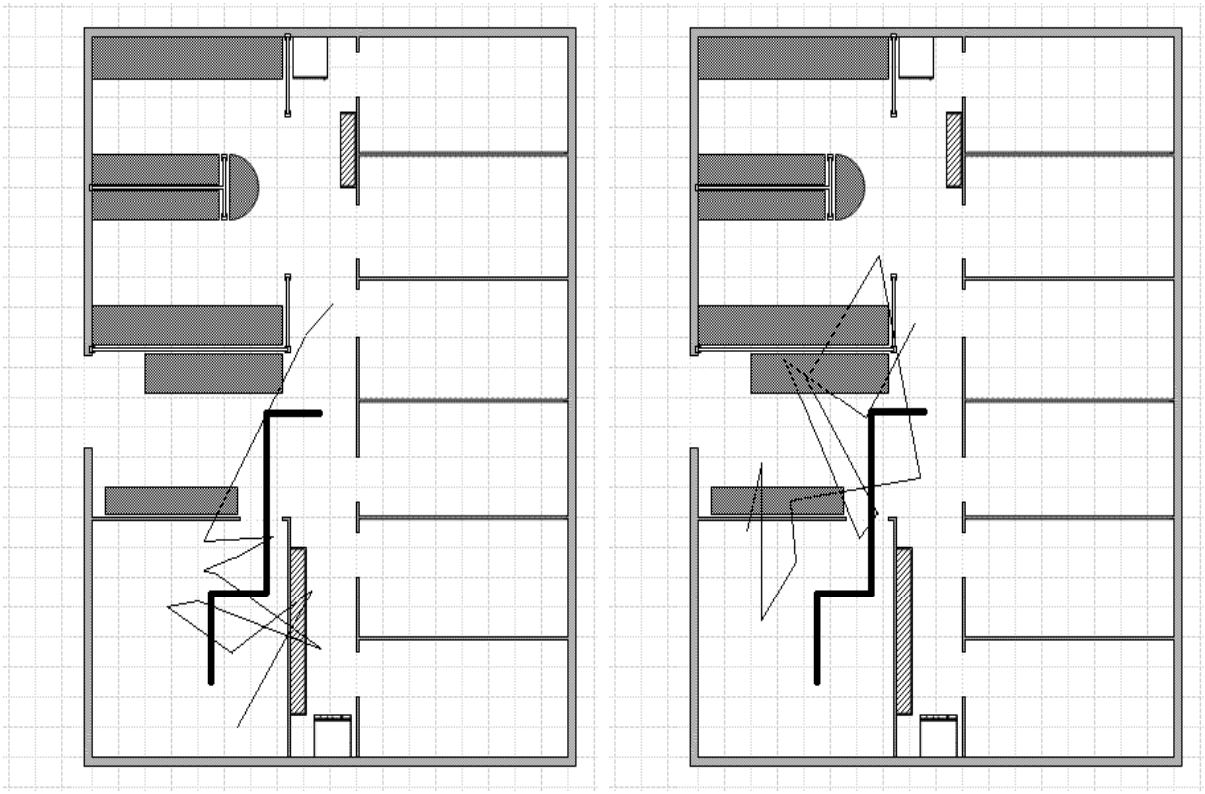
Since there is no training set for the position of each particle,the mean μ and standard deviation σ for that position are computed at run time from the neighbouring positions that there is a training set, using the *Inverse distance weighting* method.

In the transition phase, the particles move according to a motion model, used to predict the new position of the node. Specifically, each particle moves by sampling from the transition prior $P(x_k|x_{k-1})$. To implement the transition model, we considered the case with a particle inside an office and another one in which it is in a corridor. In the former case, the prior position corresponds to the mean value μ of a Gaussian distribution and with standard deviation σ equal to 1 meter. Hence, the probability of being stationary is much higher than moving in a neighbouring position. In the latter case, the probability of moving in a neighbouring position is much higher than being stationary. Therefore, the posterior position x_k corresponds to an area that is (0, 1] meters away from the prior position.

The estimated position is extracted from the final distribution using the *Highest particle density in a circle with given radius* algorithm (the particle with the highest population of particles in a circle with center its coordinates is at first chosen. In that circle, the centroid is calculated. With that centroid as the center of the circle, the centroid is calculated. This process is repeated until the same centroid is found again.)

4.4.4 Path estimation when initial position is not known

Fig. 4.14(a) depicts the results for the same path but with the initial position estimated by the particle filters. In that figure it is easily observed that after some steps the estimated path follows the same path as in the previous Section. In the other hand, in Fig. 4.14(b) it is demonstrated the estimated path with the use only of the particle filters. Comparing those two figures we can see that in our approach the estimated path converges more to the real path, than in the case of particle filters. Such a result, can only be extracted after a quantitative analysis, where we will compare the cumulative distribution function (cdf) of both approaches. However, both cases are quite optimal. The particle filters use the random function to initialize the particles as we already saw in the previous section. That makes their performance less stable.



(a) when initial position is given from the particle filters

(b) only using the particle filters

Figure 4.14: Real path vs. estimated. The estimated path is the fine line and the other is the real.

Chapter 5

Conclusion and Future Work

In this thesis, we analyzed an improvement of the Kalman Filter method for localization, without any constraint or assumption about the node mobility. Our approach relies on an analytical evaluation of the perturbation of the signal strength distribution with respect to the the most likely path, which corresponds to the discrepancy between the real and estimated positions. At each step, the algorithm aims to correct this drift in the node position. Consequently, the corrections performed result in a random walk, which on average follows the real path taken by the mobile node.

In addition, we evaluated the performance of our algorithm either by simulations, or using real measurements. The results were hearteningly, especially in the case of the simulations. In particular, even in the case where the initial position was not given the estimated path converged the real path after few steps. In the case of the real measurements, when we gave the initial position the results were also good. However, using the particle filters to estimate the initial position we did not get always the results we expected. Considering the impact of randomness in particle filters, we can easily understand why that makes our performance less stable. Finally, we can see that in our approach, with the initial position given by the particle filters, the estimated path converges more to the real path, than in the case of particle filters.

From the later results, we are motivated to perform a quantitative analysis, where we will compare the cumulative distribution function (cdf) of our approach and that of particle filters. We also plan to model it and to extend it, so that data collected during the system operation can dynamically refine the initial training and also take under consideration the orientation of the anntenna. We would like to test our system in case we have an ad hoc

network. Another extension is to incorporate in the system heterogeneous devices (e.g, RF tags, sensors) and investigate their performance.

Bibliography

[1] N. Boertien, and E. Middelkoop, "Location Based Services." Virtuele Haven Consortium, May 2002.

[2] P. Enge, P. Misra, "Special Issue on Global Positioning System," in *Proceedings of the IEEE*, January 1999.

[3] P. V. Bahl and Padmanabhan, "Radar: An In-Building RF-based User Location and Tracking System," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Tel Aviv, Israel), March 2000.

[4] Eiman Elnahrawy, Xiaoyan Li, and Richard Martin, "The Limits of Localization Using Signal Strength: A Comparative Study," in *SECON*, (Rutgers), October 7th 2004.

[5] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach, "Robotics-based location sensing using wireless Ethernet," in *Proceedings of the Eight ACM International Conference on Mobile Computing and Networking (MOBICOM 2002)*, (Atlanta, GE), September 2002. ACM Press.

[6] L. Aalto, N. Gothlin, J. Korhonen, and T. Ojala, "Bluetooth and wap push based location-aware mobile advertising system," in *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pp. 49–58, ACM Press, 2004.

[7] Nissanka Bodhi Priyantha and Anit Chakraborty and Hari Balakrishnan, "The Cricket Location-Support System," in *6th ACM MOBICOM*, (Boston, MA), August 2000.

[8] N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," in *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, (Rome, Italy), pp. 1–14, July 2001.

[9] Roy Want and Andy Hopper and Veronica Falcao and Jon Gibbons, "The Active Badge Location System," 1992. ACM Transactions on Information Systems.

[10] J. Hightower, R.Want, and G. Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," February 2000. UW CSE.

[11] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: indoor location sensing using active rfid," *Wirel. Netw.*, vol. 10, no. 6, pp. 701–710, 2004.

[12] J. Hightower and G. Borriello, "A Survey and Taxonomy of Location Sensing Systems for Ubiquitous Computing," 2001. UW CSE.

[13] R. Harle, A. Ward, and A. Hopper, "Single reflection spatial voting," in *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, (San Francisco), May 2003.

[14] John Krumm and Steve Harris and Brian Meyers and Barry Brumitt and Michael Hale and Steve Shafer, "Multi-Camera Multi-Person Tracking for EasyLiving," in *VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, (Washington, DC, USA), IEEE Computer Society, 2000.

[15] R. J. Orr and G. D. Abowd, "The smart floor: a mechanism for natural user identification and tracking," in *CHI '00: extended abstracts on Human factors in computing systems*, pp. 275–276, ACM Press, 2000.

[16] T. Mitchell, *Machine Learning*. 1977.

[17] M. Hearst, "Support vector machines," *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 18–28, July 1998.

[18] R. Battiti, M. Brunato, and A. Villani, "Statistical Learning Theory for Location Fingerprinting in Wireless LANs." Tech. Report, University of Trento, Oct. 2002.

[19] M. Mouly and M. Pautet, *The GSM System for Mobile Communications*. 1992.

[20] J. Ross, *The Book of Wi-Fi: Install, Configure, and Use 802.11b Wirelless Networking*. No Starch Press, 2003.

[21] E. Trevisani and A. Vitaletti, "Cell-ID location technique, limits and benefits: an experimental study," in *Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2004)*, 2004.

[22] J. F. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," June 1991.

[23] J. Z. Sasiadek and P. Hartana, "Sensor data fusion using kalman filter," in *In Proceedings of the Third International Conference on Information Fusion*, vol. 2, pp. 19–25, 2000.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the AMSE, Part D, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[25] Mike Klaas and Nando de Freitas and Arnaud Doucet, "Toward Practical N2 Monte Carlo: the Marginal Particle Filter," in *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, 2005.

[26] A. Doucet, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering." Cambridge University Department of Engineering, 1998.

[27] Roy Want and Bill Schilit and Norman Adams and Rich Gold and Karin Petersen and John Ellis and David Goldberg and Mark Weiser, "The PARCTAB Ubiquitous Computing Experiment," Tech. Rep. CSL-95-1, Xerox Palo Alto Research Center, mar 1995.

[28] "Versus Technology Inc." http://www.versustech.com.

[29] J. Webr and C. Lanzl, "Designing a positioning systems for finding things and people indoors," *IEEE Spectr.*, vol. 35, no. 9, pp. 71–78, 1998.

[30] "Ekahau." http://www.ekahau.com.

[31] P. V. Bahl and Padmanabhan, "Enhancements to the RADAR user location and tracking system," February 2000. MSR TR, Microsoft Research.

[32] Vinay Seshadri and Gergely V. Zaruba and Manfred Huber, "A Bayesian Sampling Approach to In-Door Localization of Wireless Devices Using Received Signal Strength

Indication," in *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, 2005.

[33] Frédéric Evennou and Francois Marx, "Improving positioning capabilities for indoor environments with WiFi," 2005. IST05.

[34] Frédéric Evennou and Francois Marx, "Advanced Integration of WiFi and Inertial Navigation Systems for Indoor Mobile Positioning," 2006. EURASIP Journal on Applied Signal Processing.

[35] Hightower, J. and Borriello, G., "Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study," in *Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp '04)*, 2004.

[36] A. Howard and S. Siddiqi and G. Sukhatme, "An experimental study of localization using wireless ethernet." International Conference on Field and Service Robotics, 2003.

[37] Julie Letchner and Dieter Fox and Anthony LaMarca, "Large-Scale Localization from Wireless Signal Strength.," in *AAAI*, 2005.

[38] Roos Teemu, Myllymaki Petri, Tirri Henry, Misikangas Pauli and Sievanen Juha, "A Probabilistic Approach to WLAN User Location Estimation," 2002. International Journal of Wireless Information Networks.

[39] Lingxuan Hu and David Evans, "Localization for mobile sensor networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004.

[40] A. LaMarca and Y. Chawathe and S. Consolvo and J. Hightower and I. Smith and J. Scott and T. Sohn and J. Howard and J. Hughes and F. Potter and J. Tabert and P. Powledge and G. Borriello and B. Schilit, "Place Lab: Device Positioning Using Radio Beacons in the Wild," in *Proceedings of PERVASIVE 2005, Third International Conference on Pervasive Computing*, (Munich, Germany), 2005.

[41] Alex Varshavsky, Anthony Lamarca, Jeffrey Hightower and Eyal de Lara, "The SkyLoc Floor Localization System," in *PerCom '07: Fifth Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 125–134, 2007.

[42] G. M. Djuknic and R. E. Richton, "Geolocation and assisted gps," *Computer*, vol. 34, no. 2, pp. 123–125, 2001.

[43] "Global Locate Inc - Indoor GPS." http://www.globallocate.com.

[44] P. Faurre, *Navigation inertielle optimale et filtrage statistique*. 1971.

[45] "NetStumbler." http://www.netstumbler.com/.

[46] S. Arulampalam and S. Maskell and N. Gordon and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking." IEEE Transactions of Signal Processing, Vol. 50(2), pages 174-188, February 2002.

[47] Rubin, D. B., "Using the SIR algorithm to simulate posterior distributions." in Bayesian Statistics 3, eds. J.M. Bernardo, M.H. DeGroot, D.V. Lindley, and A.F.M. Smith, Cambridge, MA: Oxford University Press, pp. 395–402.