UNIVERSITY OF DEUSTO

FACULTY OF ENGINEERING

PHD THESIS

# CONDITIONAL RANDOM FIELD-BASED OFFLINE MAP MATCHING FOR INDOOR POSITIONING

Author

**SAFAA BATAINEH**

Supervisors

**ALFONSO BAHILLO MARTÍNEZ**

**ENRIQUE ONIEVA CARACUEL**

BILBAO, MAY 2017

UNIVERSITY OF DEUSTO

FACULTY OF ENGINEERING

# CONDITIONAL RANDOM FIELD-BASED OFFLINE MAP MATCHING FOR INDOOR POSITIONING

PhD thesis submitted by Safaa Ahmed Hijris Bataineh enrolled in the Doctoral Program in Engineering for the Information Society and Sustainable Development at the University of Deusto, Bilbao, Spain

STUDENT

Safaa Ahmed Hijris Bataineh

SUPERVISORS

Alfonso Bahillo Martínez

Enrique Onieva Caracuel

BILBAO, MAY 2017

*To my parents and grandparents*

# Abstract

The indoor localization is of great importance for many context-aware applications. Although many indoor localization systems have been proposed, this area is still open for research; new methods are needed to overcome the tradeoffs between cost, accuracy, robustness and complexity. Using the map information is one of the methods to improve the feasibility and the accuracy of the localization systems. Unlike many systems that fuse the map information with other measurements in the same algorithm, the proposed system separates the map matching step from the basic localization process which makes it compatible with various different localization systems. The proposed map matching algorithm is independent of the positioning system implementation and it can be applied offline and coupled to different indoor positioning systems that use the coordinates to represent their output trajectory. For this purpose a platform for indoor map matching was created using Matlab. The platform uses the building plans presented in CAD files to automatically generate semantic maps suitable for map matching. A map matching algorithm based on the Conditional Random Field (CRF) probabilistic model is developed to match trajectories obtained by indoor positioning systems with the map. A ground truth was constructed and actual input trajectories were obtained by real experiments and by simulations; the output map-matched trajectories were tested using the ground truths and the accuracy of the output corrected trajectories were measured. The behavioral information of pedestrians has been used to enhance the results by smoothing the output trajectories. The parameters were adjusted in order to obtain the best possible results. The algorithm can be developed in the future by using different map models and more features.

# Acknowledgment

I would like to thank my supervisors, Alfonso Bahillo Martínez and Enrique Onieva Caracuel, for their guidance throughout my research. Thanks to the PhD program coordinator Diego López-de-Ipiña, and to the heads of the mobility unit in DeustoTech Asier Perallos Ruiz and Pilar Elejoste Larrucea for their help and administration.

Thanks to Luis Enrique Diez for being a supportive colleague in our small positioning system team, who helped me in developing my project and in reviewing my written works. I am grateful to Inaki Garcia Zuazola for his continuous help and support during my first months of research. To Hugo Landaluce for being always close when I had any question or needed help that he always gave with pleasure, to Pedro Lopez, Ashwani Sharma, Abo Baker, Laura Arjona Aguilera, idoia de la Iglesia, Ashwani Sharma, Abo Baker, Antonio David Masegosa Arredondo, Juan Alejo Vázquez Pardavila, Itziar Salaberria, Leire Serrano and every member of the mobility team as a whole has provided a friendly place to work. They helped me with their knowledge and experience whenever I needed help.

My PhD studies were generously sponsored by the Erasmus Mundus Avempace 2 scholarship, I am also very grateful to them and to the International office in the University of Deusto, especially, Eider Inunciaga Serna, Ana Mariscal Mozo de Rosales and Olga Benítez. Thanks to the officials of Yarmouk University, in the president office, the department of Computer Engineering and the deanship of research and graduate studies.

Thanks to family for their encouraging and support. And to all my friends in Bilbao who were with me during this time away from my family and country.

# List of Acronyms

| | |
|---|---|
| 2D | Two Dimentional |
| 3D | Three Dimentional |
| AoA | Angle of Arrival |
| BPF | Backtracking PF |
| CAD | Computer Aided Design |
| CDF | Cumulative Distribution Function |
| CRF | Conditional Random Fields |
| DXF | Drawing Interchange file format |
| EKF | Extended Kalman Filter |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| GVD | Generalized Voronoi diagram |
| HDR | Heuristic Heading Reduction |
| HMM | Hidden Marcov Models |
| IFC | Industry Foundation Classes |
| iHDE | improved Heuristic Drift Elimination |
| IMU | Inertial Measurement Unit |
| INS | Inertial Navigation System |
| IPS | Indoor Positioning System |
| KF | Kalman Filter |
| LPS | Local Positioning Systems |
| PDR | Pedestrian Dead-Reckoning |
| CDF | Cumulative Distribution Function |
| RFID | Radio Frequency Identification |
| RSS | Received Signal Strength |
| RSSI | Received Signal Strength Indicator |
| TDoA | Time Difference of Arrival |
| ToA | Time of Arrival |
| ToF | Time of Flight |
| VD | Voronoi diagram |
| VLC | Visible Light Communication |
| ZARU | Zero angular rate update |
| ZUPT | Zero Velocity Update |

# Table of Contents

# List of Tables

# List of Algorithms

# List of Figures

# 1. Introduction

Man has always felt the need to position himself using the sun and the stars as the only reliable reference system for many centuries. Nowadays, location became a fundamental for context-aware applications that need determining position, navigation, tracking, routing, logistics, or monitoring. Among other information that might be used like time, identity and activity, location is the most important and the most used context information (López-de-Ipiña, 2002). Position tracking and monitoring systems are needed in hospitals to guide and monitor patients; tourists in museums need navigation service; firemen need location detection in buildings on fire; these examples are among many other real world applications that require location information systems.

Today, with the advent of Global Navigation Satellite Systems (GNSS) such as the Global Positioning System (GPS), global positioning in the earth's surface is a successfully overcome problem. Nonetheless, local positioning in indoor environments is still a matter of active research. The importance of indoor navigation arises from the necessity of context-aware services that work inside buildings, given that people spend most of their time indoors (Klepeis et al., 2001; Harle, 2013).

Although outdoor navigation appeared prior to indoor navigation, the former techniques cannot be applied directly to the indoor environment. This is firstly associated with the fact that outdoor navigation systems mostly depend on the signals coming from the satellite that function poorly indoors due to multipath and attenuation losses, and secondly because pedestrian indoor movement is limited by building structure and details that are totally different from those found on road maps. In addition to that, the noise from wired and wireless networks in the building affects the accuracy of positioning. (Shang, Hu, Gu, Wang, &Yu, 2015; Gu, Lo, & Niemegeers, 2009).

There is a wide variety of positioning systems. Some are based on geometrical and topological techniques, other are more advanced and use complex probabilistic models. This applies for both outdoor and indoor positioning systems, taking into account the different characteristics for each environment. While outdoor systems depend mainly on GNSS, the indoor positioning systems depend mainly on local sensors available in the building or carried by the user. The indoor localization systems can be generally

classified to beacon-based systems and beacon-free systems; in the beacon-based systems an infrastructure of beacons is installed in the building to transmit/receive signals that are used to estimate user location. In beacon-free systems no infrastructure is needed and all the measurements and calculations are performed by devices carried by the user to be located. There are also hybrid systems that combine both. Positioning systems use the information obtained by the sensors to determine the position of the user (can be human, robot or any other tracking asset). In addition to the sensor information, the spatial information of the environment can also be fused to the positioning system to improve the accuracy of the results. The indoor positioning systems are further discussed in Chapter 2.

As indoor environments possess spatial constraints that can be used to rule out certain incorrect localization results, indoor maps have become an additional data source with which to improve the accuracy and reliability of indoor localization systems. For instance, it is unreasonable that an estimated walking trajectory passes through an area occupied by an obstacle such as a wall. The process of using map information in the localisation processes of objects or persons is known as map matching. Map matching means the use of the environment map to aid the localization process of people and objects. Map matching can be used to find the correspondence between a sequence of points representing the walking trajectory obtained via a localization system and a given map. Indoor map matching is the map matching done for localization inside buildings where building plans are used to refine and correct the positioning results.

In outdoor positioning systems, map matching techniques vary from simple geometric search methods to advanced complex probabilistic models (Quddus, Ochieng, & Noland, 2007). Similarly, existing indoor map matching techniques can be geometric, topological or probabilistic. The differences between indoor and outdoor map matching problems lie in the complex structure of indoor maps compared to road maps, as well as in the randomness of pedestrian behaviour compared to that of vehicles (Shang, 2015).

Indoor map matching techniques that use advanced probabilistic models are usually based on recursive Bayesian models such as Hidden Marcov Models (HMMs) and Particle Filters (Arulampalam, Maskell, Gordon, & Clapp 2002; Davidson, Collin, & Takala, 2010; Klepal & Pesch, 2007; Widyawan, Klepal, & Beauregard, 2008). However, recursive Bayesian models involve computing the joint probability

distribution between all states and observation variables, which makes them computationally expensive. The Conditional Random Fields (CRFs) (Klinger & Tomanek, 2007; Lafferty, McCallum, & Pereira, 2001), a probabilistic model that is widely employed in speech recognition, is also used in map matching (Xiao, Wen, Markham, & Trigoni, 2014; Xu, Du, Wu, & Zhou, 2015).

The CRF is a sequence-labelling algorithm that employs the observations and context information as features to evaluate the conditional probability of state transitions. CRF is based on calculating conditional probability which makes it computationally more efficient than Bayesian models, besides being more flexible in handling arbitrary dependencies between observations (Ramos, Fox, & Durrant-Whyte, 2007). However, CRF requires a backward phase in order to evaluate different path alternatives, which delays the process and makes it less suitable for real-time applications. On the other hand, offline[1] (i.e. not real time) map matching can allow for reduced performance in favour of accuracy (Pereira, Costa, & Pereira, 2009). Offline map matching results can be used for both post-processing analysis and modelling purposes. Many indoor applications that need behavioural analysis and data mining can make use of offline map matched trajectories. More information about the map matching techniques can be found in Chapter 4, and more about the probabilistic models can be found in Chapter 5.

Indoor localization and map matching requires that the location map be represented in a data model that suits the localization algorithm. Map information should be extracted from the available map files such as images or CAD (Computer Aided Design) files and then this information should be represented in another format usable by the map matching algorithm. Some map models are grid-based and boundary-based which use geometrical information of the space (Bohn & Vogt, 2003; Crowley, 1989); other models such as graph models and set-based models use topological or semantical

[1]**Offline localization systems:** In localization systems, the expression "offline" has been used in the literature with two different meanings: (1) those localization systems that are not real time, also known as global systems, in which measurements are taken for the whole walk and an offline algorithm is then used to estimate the track; and (2) those localization systems that use local resources and sensors without having been connected to any global system or central server. In this thesis the term is used to refer to the first meaning.

information (Becker & Dürr, 2005; Choset & Burdick, 2000). More about map modelling can be found in Chapter 3.

## 1.1 Research Objectives

The main objective of this thesis is to improve the feasibility and accuracy of indoor localization systems using the map information in the offline stage. To achieve this objective the following specific objectives will be addressed:

- To create a tool for automatically extracting the map information from building plans in CAD format, and to model it in a format useful to be used by the map matching algorithm.
- To design an offline map matching algorithm to use the map information for indoor positioning of pedestrians that can be used to improve the result of different localization systems.
- To investigate the efficiency of the CRF technique in indoor map matching using different map information including behavioural information improving the localization feasibility and accuracy.

## 1.2 Hypotheses

Probabilistic models have been used for map matching; one of these models is the CRF which is known to be a computationally efficient model. CRF can be used efficiently as an offline map matching algorithm to refine the positioning trajectories of pedestrian in indoor environment and by creating feasible and accurate walking paths. An Offline CRF map matching algorithm can be applied to the results obtained by various types of positioning systems using map information presented in a suitable semantic map model. Generating digital map that contains various and semantically presented information can help in developing matching algorithm with better accuracy. Semantic maps of buildings can be automatically generated from information extracted from existing CAD plans. Behavioural information and behavioral maps that describe the behaviour of pedestrians can also be a used to improve the positioning results and can be integrated into the CRF algorithm.

## 1.3 Methodology

In order to achieve the research objectives, the following methodology (shown in Figure 1) has been followed:

1. Defining the problem: different map matching techniques have been proposed for indoor positioning but still there is no solution that fulfils the requirements of high performance, high accuracy, low computational complexity and robustness. More solutions need to be explored. Coupling the map matching algorithms to different types of positioning systems is another issue, where the map matching algorithm is less dependent on the implementation of the localization system.
2. Reviewing the state of the art: as the goal is to design a platform for map matching, literature review had to be done on a number of topics, including map matching algorithms, map modeling, mathematical modeling, and positioning systems.
3. Designing and developing an algorithm for map generation: this is the first part of the system; the generation of a semantic map that is suitable for map matching applications given a normal CAD building plan. A Matlab[2] tool was developed. A heuristic approach was used to extract map data from CAD files, the extracted data were used to generate the map semantic model which can be used in the map matching algorithm.
4. Testing the map extraction process and the map model using different CAD files.
5. Designing and developing an algorithm for map matching: this comprises the main part of our system and the main contribution of this thesis. A CRF algorithm that uses the semantic map information has been developed as an offline map matching algorithm. It has been designed to be independent of the implementation of the basic positioning system so it can be loosely coupled to any available positioning system.
6. Testing the map matching algorithm using different trajectories and maps.
7. Analyzing the results via empirical trials: different scenarios of different trajectories have been used to test the system; some trajectories were simulations

[2] https://www.mathworks.com/products/matlab.html

other where obtained from actual measurements. The results were analyzed and evaluated by comparison with the actual path (the ground truth).

8. Drafting the conclusions and future work: conclusions have been drawn based on the results, and future work tasks have been planned.

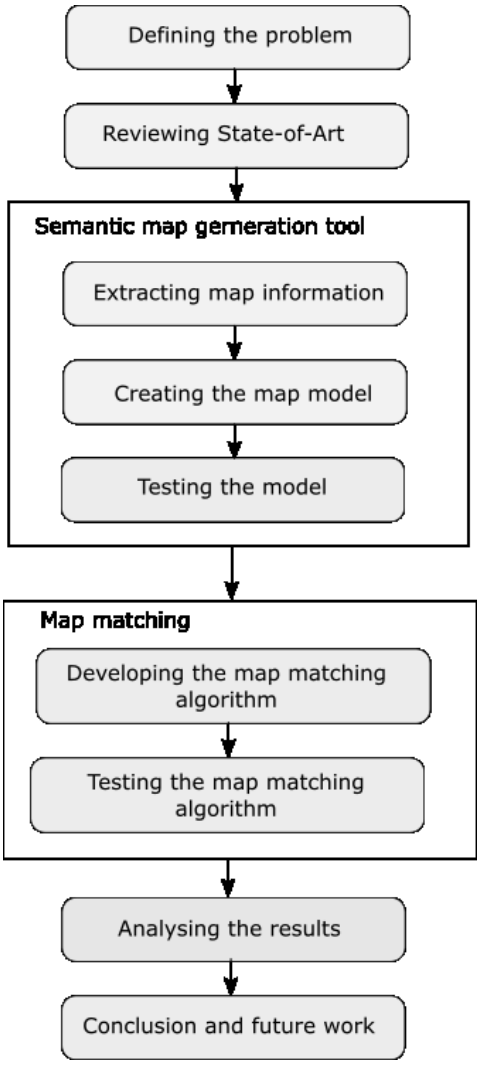



*Figure 1.* Research Methodology.

## 1.4 Thesis Organization

A literature review of related work and theoretical background for topics related to this thesis are presented in Chapters 2 to 5. In Chapter 2, a literature review of positioning system is presented. A brief description of different indoor positioning technologies has

been reviewed including systems that depend on wireless beacon technologies and systems that use inertial sensors as well as some common positioning methods such as triangulation and fingerprinting.

Chapter 3 reviews the map modeling techniques; there are different ways to represent the map in order to be usable by the map matching algorithm such as the cell-based models, boundary-based models, graph-based models and set-based symbolic models. Some of these models contains mainly geometrical information other use topological or semantical information. Detailed description of these model types with different examples from the literature is presented.

Chapter 4 is about map matching algorithms, different map matching approaches both for outdoor and indoor environments have been described along with comparisons of the different methods in addition to examples from the literature. The presented approaches include simple geometrical methods, some are topological methods, and others use complex mathematical models that are based on probability theory.

Chapter 5 includes a mathematical background about the probabilistic models. The probabilistic models can be either generative models or discriminative models; both have been explained using examples for single variable prediction and multiple variable prediction. For generative models; the Naïve Bayes model that is used to predict a single variable is explained; the HMM model is an example of a generative model for predicting multiple variables. For discriminative models, the maximum entropy model that is used to predict a single variable is explained and the CRF model is explained as an example for models that are capable of predicting multiple variables.

The proposed map matching system is presented in Chapters 6 to 8; Chapter 6 describes the system architecture and components, as well as the proposed CRF model; the system is composed of (1) a map generating module that generate the map model from the cad files, and (2) a CRF map matching algorithm that uses the map information represented by the map model to correct an estimated trajectory obtained by some positioning system by matching it with the map. The process of extracting the map information from the CAD files and the map model are described and a detailed explanation of the use of CRF model in the map matching process is presented.

The experimental results are presented and discussed in Chapter 7; Actual walking trajectories to be corrected are obtained using a PDR system through a real experiment in the DeustoTech building at the University of Deusto. Other input trajectories are generated using simulations. The CRF algorithm is applied to both the real data and the simulation inputs and the output map matched paths were compared to the ground truth to evaluate the system. Two problems of the system are discussed and solutions are proposed; these problems are the non-smooth output paths, and the divergence problem that happens when the output path sticks in semi-closed areas.

Finally, the conclusions drawn from this research, the contribution and published papers, and the planned future work are presented in Chapter 8.

# Part 1:

# Theoretical Background and Literature Review

# 2. Indoor Positioning Systems

"An Indoor Positioning System (IPS) considers only indoor environments such as inside a building" (Gu et al., 2009, p. 13). In such environment the outdoor positioning systems that depend mainly on the GNSS cannot be used as they do not work efficiently due to multipath and attenuation losses. Instead, other various technologies are used for indoor positioning of pedestrian localization which is the focus of this research.

Two main research approaches are used in the indoor positioning problem: (1) Solutions that rely on the existence of a network of receivers or emitters placed at known locations (beacon-based solutions), and (2) Solutions that mainly rely on dead-reckoning methods with sensors installed on the person or object to locate (beacon-free solutions). The first type beacon-based approaches can use different technologies for the estimation of range or angle between the mobile object and the beacons; it is typical to use ultrasound, radio (WiFi, UWB, RFID, Zigbee, etc.) or vision-based technologies. These beacon-based solutions are normally termed as Local Positioning Systems (LPS). The second beacon-free approaches are sometimes preferable since they do not depend on a pre-installed infrastructure. During the last decade several beacon-free methodologies based on inertial sensors have been proposed for person's location. These methodologies, often called Pedestrian Dead-Reckoning (PDR) solutions, integrate step lengths and orientation estimations at each detected step, or alternatively directly integrate accelerometer and gyroscope readings to compute the position and attitude of the moving person (Jiménez, Seco, Prieto, & Guevara, 2010; Zampella, Bahillo, Prieto, Jiménez, & Seco, 2013; Folxin, 2005).

Each of both approaches has the problem of accuracy, the beacon-based systems estimation usually have an absolute and relatively high error, the beacon-free systems have relative errors proportional to the walked distance so the error accumulates by time. To achieve more accuracy different solutions can be used: hybrid systems of beacon-free and beacon-based combination, the use of building map and benefiting from Signals of Opportunity (SoOP). SoOP are those signals which are freely available most of the time in the environment and not originally meant to be used for localization purposes but can be exploited for that. Examples on SoOP include radio stations and TV

transmission towers, mobile phones base stations, WiFi access points, Bluetooth, magnetic fingerprints, illumination, pressure, and temperature (Jiménez, Zampella, & Seco, 2013).

The location information can be represented in different ways depending on the applications: physical location that is represented in the form of 2D or 3D coordinates, symbolic location such as office or bedroom, absolute location that uses reference grid, and relative locations that depends on its own frame of reference points. (Liu, Darabi, Banerjee, & Liu, 2007).

Different systems vary in their accuracy, cost, complexity and robustness; evaluations of different positioning systems can be found in the surveys of Liu et al. (2007), Gu et al. (2009) and Harle (2013). For example, systems may give accuracy in centimeters but are expensive, or may have less accuracy but at the same time have less complexity. Therefore, trade-offs usually have to be considered when designing positioning systems. Lymberopoulos et al. (2015) brought multiple indoor location technologies of 22 teams for a real experiment competition. The technical approaches included a wide variety of technologies including WiFi, ultrasound, IMU, magnetic signals, fingerprinting, and other technologies. The accuracy and overhead requirements were compared directly and "the average location error achieved varied between 0:72m and 10:22m. Only 3 teams were able to achieve less than 2m accuracy, while half of the teams achieved less than 3m error"( Lymberopoulos et al., 2015, p. 82).

## 2.1 Beacon-based Systems

Beacon based systems usually require a network of on receivers or emitters, different technologies have been used in such networks and different methods to calculate the location have been developed. An overview of wireless technologies (WiFi, Bluetooth, RFID, etc.) that have been used in indoor location sensing are presented in Subsection 2.1.1, the methods for estimating the position associated to these technologies are mainly the triangulation, the fingerprinting (scene analysis) and proximity which are presented in Subsections 2.1.2, 2.1.3, and 2.1.4, respectively.

### 2.1.1 Wireless Technologies

Different wireless technologies have been developed for indoor localization, comprehensive surveys of such technologies are provided by Liu et al. (2007) and Gu et al. (2009). These technologies are usually built as transmitter receiver systems where the user to be located hold a transmitter or receiver device and make signal measurements with different base stations. The wireless technologies can be classified to sound-based technologies, optical technologies and radio technologies.

Ultrasound is a sound-based technology that is used in positioning systems (Ward, Jones, & Hopper, 1997; Priyantha, Chakraborty, & Balakrishnan, 2000; Hazas &Hopper, 2006; Schweinzer & Syafrudin, 2010; Yazici, Yayan, & Yücel, 2011; Saad, Bleakley, Ballal, & Dobson, 2012; Medina, Segura, & De la Torre, 2013). The infrastructure can be designed so that it contains ultrasound sources at known locations; in this case the user to be located carries a microphone to pick the signal. Another way to design the system is to have the receiver fixed in the infrastructure and the transmitter carried by the user. Ultrasound systems have shown high accuracy but are expensive to deploy and maintain (Harle, 2013).

Optical technologies include Visible Light Communication (VLC) and Infrared technology (IR). VLC technology (Liu, Makino, & Maeda, 2008; Lou, Zhang, Zhang, Yao, & Xu, 2012) is used in indoor positioning systems through using lamps in fixed known locations in the infrastructure and using different flicker encoding for each lamp. The user to be located carries a light sensor (photo diode or photocell) to capture the light and the lamp is determined according to a known encoding scheme; the sensor location is determined in the vicinity of the corresponding lamp. IR technology (Want, Hopper, Falcao, & Gibbons, 1992; Aitenbichler, & Muhlhauser, 2003; Sayeef, Madawala, Handley, & Santoso, 2004; Tas & Altiparmak, 2009) uses electromagnetic radiation with frequencies less than the frequencies radio waves and higher than of the visible light frequencies the carrier frequency of the infrared signals is typically around 38 kHz. The infrared system is composed to a light emitter diode as a transmitter and an infrared photodiode to capture the light pulses (Rogalski, 2002).

Several radio technologies have been used in indoor positioning, these technologies include:

1- Wireless Local Area Network (WLAN); WiFi is the term used for the products that work in WLAN. Due to WiFi wide availability, it is widely used in indoor positioning systems.
2- Bluetooth, an accurate and low cost technology for short range communications that is used in positioning systems (Feldmann, Kyamakya, Zapater, & Lue, 2003; Subhan, Hasbullah, Rozyyev, & Bakhsh 2011; Frost, Jensen, Luckow, Thomsen, & Hansen, 2011).
3- ZigBee, a wireless communication standard that is low-cost and low-power consumption that has also been used in indoor positioning systems (Alhmiedat, & Yang, 2008).
4- Radio Frequency Identification system (RFID) that uses a small electronic device (called tag) as identifier to objects or people when attached to them. It can be for commercial purpose where it serves like the barcode, but also can be used in localization systems for tracking purposes like baggage and for access control of people and vehicles (Ni, Liu, Lau, & Patil, 2004). The RFID tag consists of microchip and an antenna which can receive radio waves, and can store a small amount of data (Weinstein, 2005). To retrieve the data, the RFID tag must be scanned by an RFID reader. There are different types of RFID tags; they can be passive or active. The passive does not have a battery, it just use a small energy amount emitted by reader to answer it. On the other hand, the active RFID tags have their own power supply and they transmit there ID periodically, they have a longer read range and their size is bigger than the passive RFID as well as they have a greater cost and a shorter operational lifetime (Symonds, 2009).

### 2.1.2 Triangulation

The triangulation technique computes the location of a target by using the properties of triangles and depending on the distances (trilateration) or angles (angulation) between the target and the beacons (Liu et al., 2007; Hightower & Borriello, 2001). In trilateration methods the distances are usually deduced from measurements such as ToA, TDoA, and RSS, then position is calculated. In angulation, the AoA measure is usually used, where the position is determined on the intersection between the lines that forms the angles between the user and the beacons. ToA, TDoA, RSS, and AoA are described below.

### *2.1.2.1 Time of Arrival (ToA)*

ToA (also called Time of Flight ToF); is the time for the signal to travel from the transmitter to the receiver. To calculate the user location in two dimensions, three ToA measurements should be done between the user and three beacons (points A, B and C in Figure 2); three circles are created around each of the receivers with diameters equal the estimated distance between the receiver and the transmitter, the intersections point of the three circles $E_1$ constituted by distance estimates is supposed to be the user location. But due to measurement errors a region of uncertainty is created. ToA is considered as the most accurate technique because it can filter out multi-path effects. However, it is complex to implement (Gu et al., 2009); it requires that all transmitters and receivers to be synchronized, and a timestamp to be labeled in the signal (Liu et al., 2007).

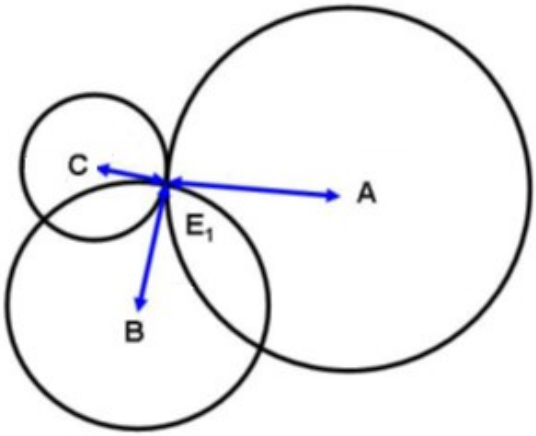



*Figure 2.* The triangulation method, time measurements. "*Note.* Adapted from (Gu, et al., 2009)".

### *2.1.2.2 Time Difference of Arrival (TDoA)*

TDoA it is related to ToA, But instead of using the travel time between the transmitter and receiver, the difference of travel times of the signal at multiple measuring units to find the distance.it examines the difference in time at which the signal arrives at multiple measuring units (Liu et al., 2007).

### *2.1.2.3 Angle of Arrival (AoA)*

AoA (also called direction of arrival DOA) is measured either using array of antennas (Widrow, Mantey, Griffiths, & Goode, 1967; Dobson, 1988) that enables the estimation of direction of the signal (Zeytinci, Sari, Harmanci, Anarim, & Akar, 2013). At least two angles ($\theta 1$ and $\theta 2$ in Figure 3) with two reference points (A and B in Figure 3) are needed to find the location of the user on the intersection between the lines that forms the angles with the beacons (Kawakami & Ohtsuki, 2010). This method has a low accuracy problem when the target object is far away (Chen, Wang, Maa, & Chen, 2006).

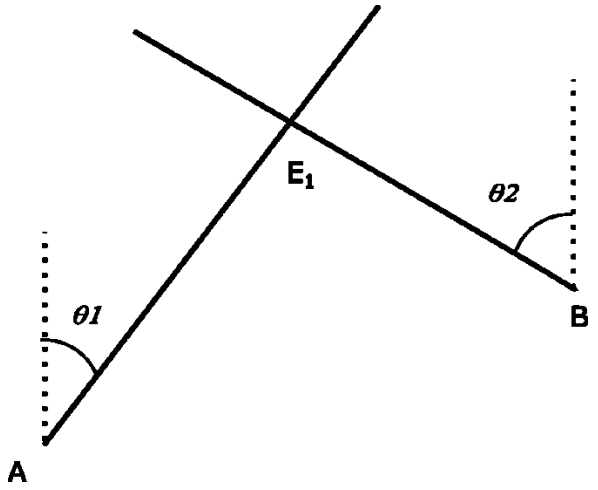



*Figure 3.* AoA Measure.

### *2.1.2.4 Received Signal Strength (RSS)*

The RSS is the intensity of field of a signal at the reception point knowing the transmission power. It is measured in order to calculate a distance between a transmitter and a receiver, which can be done using the propagation characteristics of the channel (Kawakami & Ohtsuki, 2010). Two RSS measurements are used in positioning process, the Received Signal Strength Indicator (RSSI), which measures the power of the received signal, and the Link Quality Indicator (LQI) which measures the quality of the connection between the transmitter and the receiver (Alhmiedat & Yang, 2008). Three beacons are needed to calculate the location of a target in 2D as shown in Figure 4. "Signal attenuation-based methods attempt to calculate the signal path loss due to

propagation. Theoretical and empirical models are used to translate the difference between the transmitted signal strength and the received signal strength into a range estimate" (Liu et al., 2007, p. 1069).

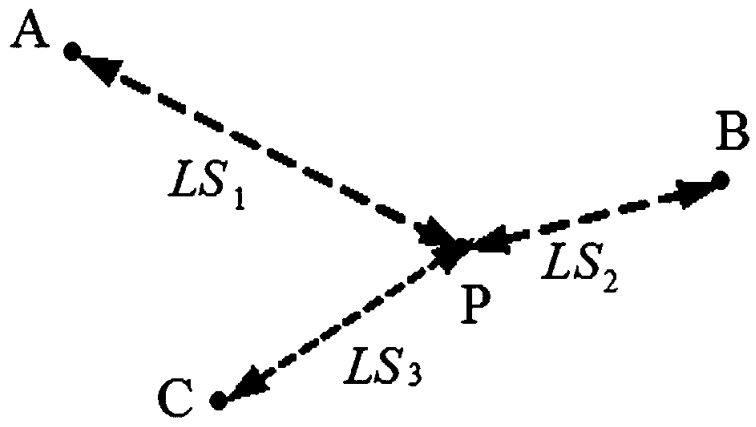



*Figure 4*. Positioning based on RSS, where *LS*1*, LS*2*,* and *LS*3 denote the measured path loss. "*Note.* From (Liu et al., 2007).

### 2.1.3 Fingerprinting

Fingerprinting (Kaemarungsi & Krishnamurthy, 2004) is a popular approach for indoor positioning systems in which a database of measurements is constructed to be used later in the localization process. The fingerprinting process has two phases, in the first phase, also called the calibration phase, a radio map is created by observing and recording the RSS of several access points in the environment from different locations, and this is done once as offline and should be redone when necessary like when the distribution or number of access point is changed. The used signal sources used should be stable so that their characteristic does not vary with time as that will affect the performance of the fingerprinting (Shang, Hu, Gu, Wang, & Yu, 2015). In the second phase which is supposed to be online a matching algorithm is performed and the current measurement is compared to the value in the database of the radio map obtained in the first phase to determine the location in approximation with pre-calibrated known location; A proximity matching algorithm (such as k-NN (Cover & Hart, 1967)) is used to infer the

location, the propagation model can also be used to infer locations between two pre-calibrated points. Using more observations in the calibration phase will result in better resolution and higher accuracy (Taheri, Singh, & Emmanuel, 2004).

Problems of fingerprinting include the fluctuation of the signal strength measured value changes over time, it is also affected by the direction of user and the brand of the measuring device (e.g. smart phones) (Kjærgaard & Munk, 2008; Shang et al., 2015)

In fingerprinting different kinds of signals can be used, including WLAN (Kaemarungsi & Krishnamurthy, 2004), Bluetooth, FM (Chen, Lymberopoulos, Liu, & Priyantha, 2012; Frost et al., 2011), lights, sounds (Elhamshary & Youssef, 2014; Azizyan, Constandache, & Roy Choudhury, 2009) and earth magnetic fields (Vildjiounaite, Malm, Kaartinen, & Alahuhta 2002).

### 2.1.4 Proximity

The proximity localization technique does not provide the location as coordinates but as a course location with respect to some known reference points. Detector are places in fixed known positions and when the target is detected by a detector the location is considered to be in the proximity area of that detector For example, if the sensing area is a room the proximity sensing can specify whether a target is in the room or not; it does not give absolute position estimations (Gu, et al., 2009). When more than one detector detects the target, it is assigned to the proximity area of the one that receives the strongest signal. Systems using infrared radiation (IR) and radio frequency identification (RFID) are often based on the proximity method (Liu et al., 2007).

## 2.2 PDR Systems

The dead-reckoning (which was historically used by the ancient sailors) is a navigation method that estimates of the future position given the initial position plus the speed and direction. The dead-reckoning is also used to locate pedestrians in indoor and outdoor environments. Displacement and direction of the pedestrian are used to calculate the position (see Figure 5). There are two types of PDR systems: Inertial Navigation Systems (INSs) and Step-and-Heading Systems (SHSs). The INSs track position by estimating the full trajectory on time basis, while SHSs estimate position by accruing (distance, heading) vectors of either steps or strides (Harle, 2013). The PDR systems

"typically compute their own positions and their key advantage is that they require very little, if any, physical infrastructure to function. They also offer a degree of location privacy since the user can choose not to share the information with any third party" (Harle, 2013, p. 1281).

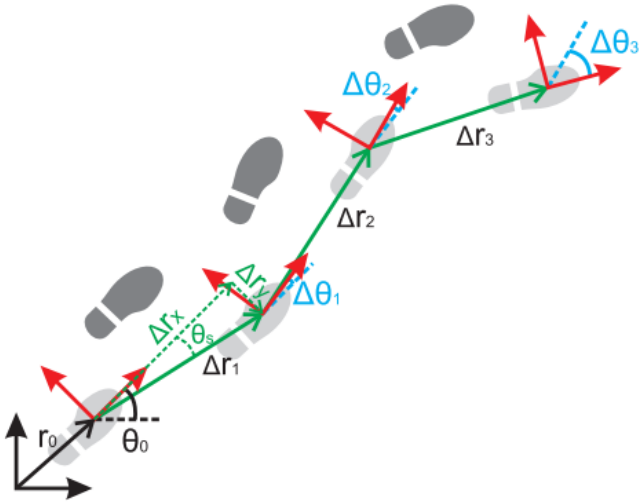



*Figure 5.* Reconstruction of the position in PDR systems, using the step displacements (green) and heading changes (blue) and starting from the initial position $r_0$and heading $\theta_0$. "*Note*, From (Jiménez et al., 2013)".

In the aim of solving the problem of needed infrastructure that exist in the pedestrian tracking solutions, a tracking method based on dead-reckoning using inertial sensors (accelerometer and gyroscope) which can be used indoor and outdoor using only inertial sensors carried by the user have been proposed (Foxlin, 2005). Digital inertial sensors are used to provide the acceleration (obtained from accelerometer) and direction (obtained from gyroscope); from acceleration the speed then the distance is calculated using integration. The distance and the orientation are used to calculate the current position given the initial position. Figure 6 shows the algorithm of obtaining the position from the accelerometer in the INS.

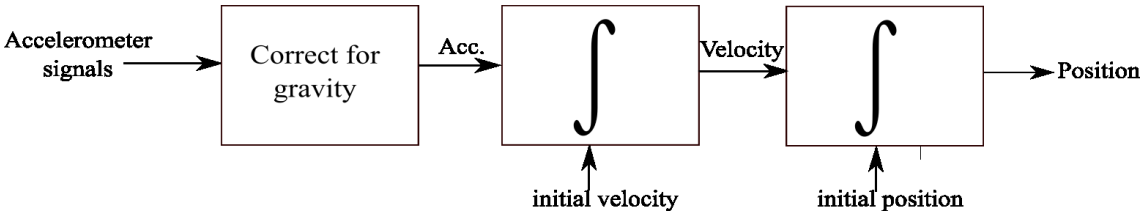



*Figure 6.* Obtaining position using the accelerometer in the INS.

The problem of the PDR systems is the drift; a small error in direction will result in a big error after walking a long distance, the error accumulates with the distance.

> In practice it is the accuracy of the gyroscopes and not the accelerometers which limit the overall accuracy of most INSs. An error in the computed orientation causes the body-frame acceleration signals obtained from the INS to be projected incorrectly onto the global axes. This results in a rapidly accumulating error in position when the signals are subsequently integrated (Woodman, 2007, p. 25).

To solve the drift problem and to obtain a robust pedestrian tracking system, Foxlin introduces the ZUPT (Zero Velocity Update) solution; this solution is based on the fact that when the pedestrian put her/his foot on the ground the velocity becomes zero. ZUPT asserts that the sensor is static provided that the sensor is placed on the foot; this is used to reset the velocity and reinitialize the calculations so that the error will not accumulate with distance. But the position drifts by 0.3% of the traveled distance.

The inertial sensors are usually available in the mobile phones, but more accurate sensors can be integrated in an Inertial Measurement Unit (IMU) which can be carried on foot like the one used in Foxlin experiment (see Figure 7) or can be worn in wrist (Díez, Bahillo, Bataineh, Masegosa, & Perallos, 2016) as in Figure 8. The IMU that is carried on the foot is more accurate as the ZUPT method can be applied when the foot touches the ground.

*Figure 7.* IMU mounted on foot. "*Note*. From (Widyawan et al., 2008)".

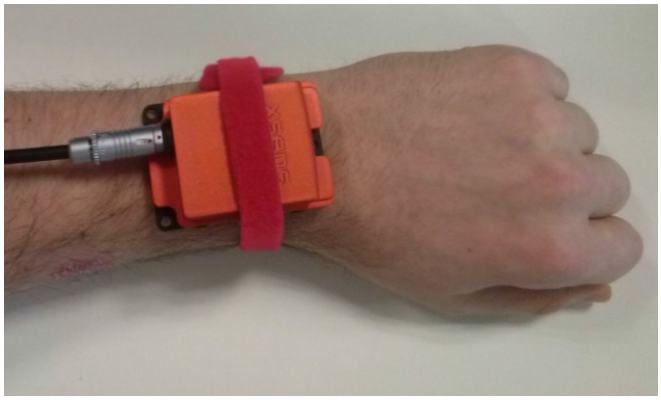


*Figure 8.* IMU worn on the wrist. "*Note*. From (Díez et al., 2016)".

In general, drift can be reduced by fusing additional sensors information like the absolute position obtained by the GPS, which is efficient only if the pedestrian is walking outside, or by the measurements of the vector magnetometer, which measures the strength and direction of the magnetic field. An example of indoor positioning system that fuses magnetic measurements is presented in (Li et al., 2012); this system combines the accelerometer, gyroscope and compass. Different methods can be used for sensor fusion, the most popular are Kalman filter (Grewal, 2012) and Particle filter (Gustafsson, 2010; Jiménez et al., 2010) fused different sensor information in their PDR system. Jiménez et al. implemented a Kalman based framework called INS_EKF_ZUPT (IEZ) to estimate the position and attitude of a pedestrian. The platform used the Extended Kalman Filter (EKF) (Fujii, 2013) with INS algorithm with application of the ZUPT methodology as well as a stance detection algorithm. For heading drift reduction, several methods (ZARU (Zero angular rate update) propose by Rajagopal (2008), HDR

(Heuristic Heading Reduction) proposed by (Borenstein, Ojeda, & Kwanmuang, 2009), and the magnetometer as a compass) were integrated with the platform and the positioning error is about 1% of the traveled distance; which is a good ratio given that the PDR positioning accuracy which, according to Jiménez et al. (2010), normally ranges from 0.3% to 10% of the traveled distance.

An SHS works as follows: it identifies the data of individual steps or strides, then estimates the length of the step and the step heading and generate as output a series of (step length, step heading) or (step length, step heading change) vectors. A step detection process is needed to identify the sensor data associated with each step. Step length is estimated using the duration without integration. Step heading change is estimated through single integration of gyroscope signals. Magnetometer can also be fused with the gyroscope signals to estimate the heading (Harle, 2013). The SHS can use the IMU integrated in the smart phone to detect steps, but this is complex due to unconstrained location and orientation of the smartphone carried by the user (Brajdic & Harle, 2013); Díez et al., (2015) and Díez et al., (2016) proposed an SHS-based system that uses a wrist-worn IMU which means a fixed location on the body but more convenient for the user than the foot-mounted IMU.

# 3. Map Modeling

Indoor localization and map matching requires that the location map be represented in a map model that suits the localization algorithm. Some map models such as grid-based and boundary-based models use geometrical information of the space (Afyouni, Cyril, & Christophe, 2012). Other map models such as graph-based models and set-based symbolic models use topological or semantical information (Dürr & Rothermel, 2003; Hu & Lee, 2004); they use space character characteristics like connectivity and reachability between spatial units (Franz, Mallot, Wiener, & Neurowissenschaft, 2005; Jensen, Lu, & Yang, 2009; Van Treeck & Rank, 2004), closeness and overlapping relationships (Dürr & Rothermel, 2003; Hu & Lee, 2004; Jensen et al., 2009), as well as containment from which hierarchical information can be derived (room, roof, building) (Becker, Nagel, & Kolbe, 2010; Choset & Burdick, 2000; Stoffel, Schoder & Ohlbach, 2008).

The most commonly used map models in indoor localization are the grid-based models and the graph-based models (Afyouni et al., 2012). In the grid-based models the space is partitioned into regular cells with semantic. The graph models (Becker, Nagel, & Kolbe, 2009) reconstruct the space as a graph where nodes represent entities or places of interest in the building. In (Xiao et al., 2014) for example they obtained from maps in image formats and a standard edge detecting algorithms to extract edges from the image then a grid-based map model was used to model the map information.

## 3.1 Cell–based Models

The cell-based geometrical models divide the space into adjacent polygonal cells that can be regular or irregular. The regular tessellations or the grid-based models cells have the same shape and size, mostly squares and hexagonal shapes are used. The irregular tessellations are used when adaptive cells are required to suit the environment complexity (Afyouni et al., 2012).

### 3.1.1 Grid-based Models

In the grid-based models the space is divided into similar regular cells with semantic; that means every cell has information about its contents, for example: wall, door, or free space (Afyouni et al., 2012). All the cells have the same shape and size; the cell size depends on the application, but normally it has not to be very big which decreases the accuracy, and not too small because smaller cells means more cells and more computations and more memory usage. There are two common types of regular cells, square cells (Figure 9) and hexagonal cells (Figure 10), the later have the advantage of equal distances between the cell and each of its neighbors. The relation between the cell and its neighbor is very important as it is used in the localization algorithm to determine the transitions between steps.

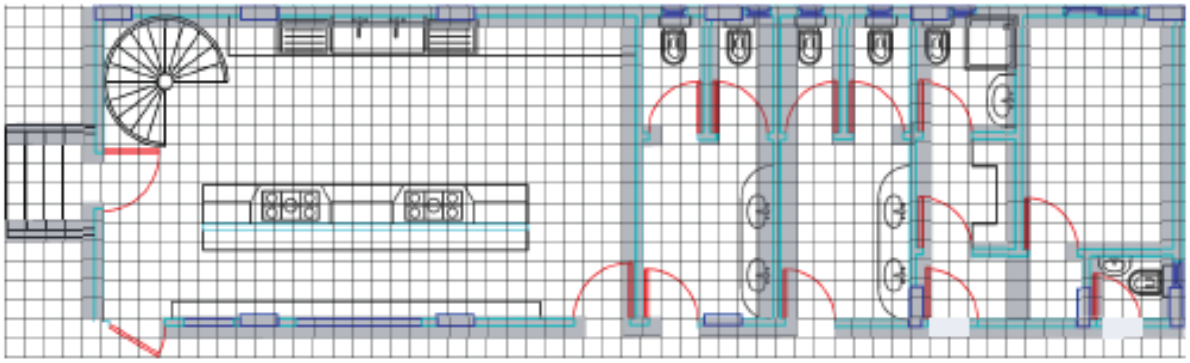


*Figure 9.* Uniform square cells "*Note*. From (Afyouni et al., 2012)".

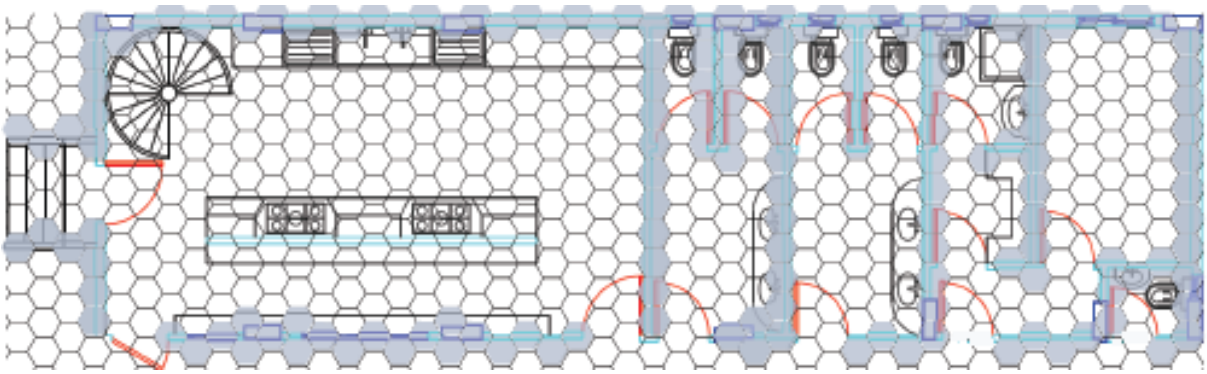


*Figure 10.* Uniform hexagonal cells. "*Note*. From (Afyouni et al., 2012)".

The grid-based models are widely used in indoor positioning systems. In (Xiao et al., 2014) for example they use map information obtained from maps in image formats using a standard edge detecting algorithms to extract edges from the image, then a grid-based map model was used to model the map information. And they construct a transition graph between cells to be used by their CRF map matching algorithm. Bohn and Vogt (2003) employ a 2D grid model with a fixed cell size in their localization algorithm that is based on high-level sensor fusion architecture. In their model, depending on the existence of obstacles each cell contains three probabilities: the probability that the target was located within the cell; probabilities for movement into the adjacent cells; probability for staying in the cell. The cell with the maximum probability would be selected as the target's location (Shang et al., 2015). The choice of the cell size depends on the application and the required degree of accuracy, for example, as their system is designed for pedestrian localization, Xiao et al. a cell size of 0.8 m was chosen as it is equal to the average step length of a pedestrian; while for mobile robots localization, Fox et al. (1999) used a regular fine-grained 3D grid model (10 cm–40 cm cells); Bohn and Vogt used fixed cell size but the size was chosen individually for each map.

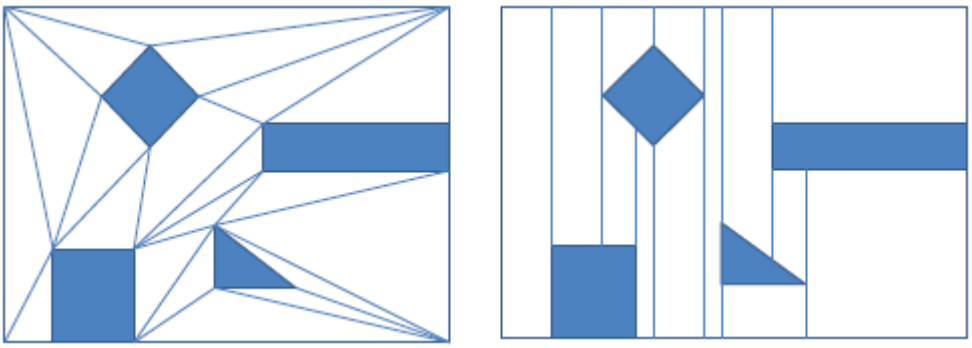
The map matching models use the spatial constraints to narrow the probability space of the user location. One of the typical spatial constraints represented in the grid-based model are: the cell occupancy; cells contains obstacles such as furniture and walls are forbidden for users and there probability of occupancy by the user should be equal to zero, only free cells are available for user motion. Another constraint is the buffer, which represent the area around the cell to can move to from that cell directly. The model can also limit the direction of movement and forbid some directions in some areas. The grid cell model have the flexibility to employ different constraints in the model though the cell characteristics which can help to improve location estimation (Shang et al., 2015).

### 3.1.2 Irregular Tessellation Models

The irregular tessellation models are adaptive to the environment and to the distribution of obstacles. The cells here have different shapes and sizes (Afyouni et al., 2012). There are two categories of irregular tessellation: free-space tessellations (Demyen & Buro,

2006; Mekni, 2010), and Voronoi tessellations (Aurenhammer, 1991; Choset & Burdick, 2000).

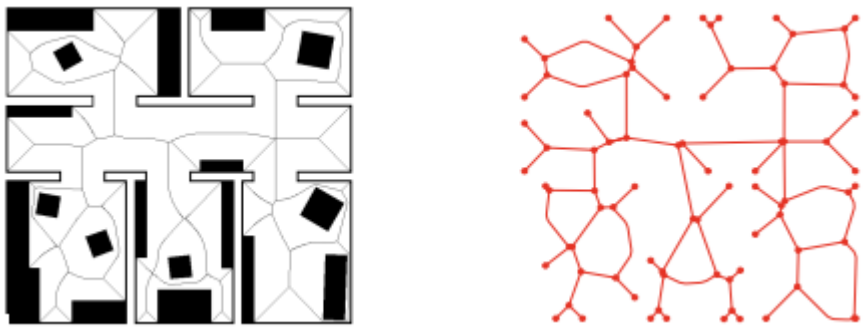
In the free-space tessellations the obstacles are taken into account while decomposing the free spaces into convex polygonal cells (Afyouni et al., 2012). These polygonal cells are usually triangles (Demyen & Buro, 2006) or trapezoids. In Figure 11 we can see triangular tessellations (a) and trapezoidal tessellations (b); the blue areas represent the obstacles.



(a) Triangulation tessellation. (b) Trapezoidal-based tessellation.

*Figure 11.* Examples of free-space tessellation techniques. "*Note*. From (Afyouni et al., 2012)".

Voronoi tessellations subdivide space into a set of special cells called *Voronoi cells.* One approach to construct the model is by using the Voronoi diagram (VD) of a set of points so that the environment is represented as a network of curves that forms pathways across these points forming an irregular tessellation of space (Afyouni et al., 2012; Aurenhammer, 1991). In the Generalized Voronoi diagram (GVD), the convex objects can be line segments or obstacles rather than points, see Figure 12.



(a) A generalized Voronoi diagram (b) The corresponding generalized Voronoi graph

*Figure 12.* Generalized Voroni diagram. "*Note.* From (Wallgrün, 2010)".

## 3.2 Boundary-based Models

Boundary-based geometric models form the most direct way to represent an indoor space. Boundary-based models represent the obstacles' boundaries of an indoor space with sequences of primitive geometries, such as points, lines, and curves. Usually, a boundary-based map constructed by concatenating line segments designed using designed using a CAD system or extracted from sensor data; the obstacles are represented as polygons (Chatila & Laumond 1985; & Crowley, 1989), or designed using a CAD system (Afyouni et al., 2012).

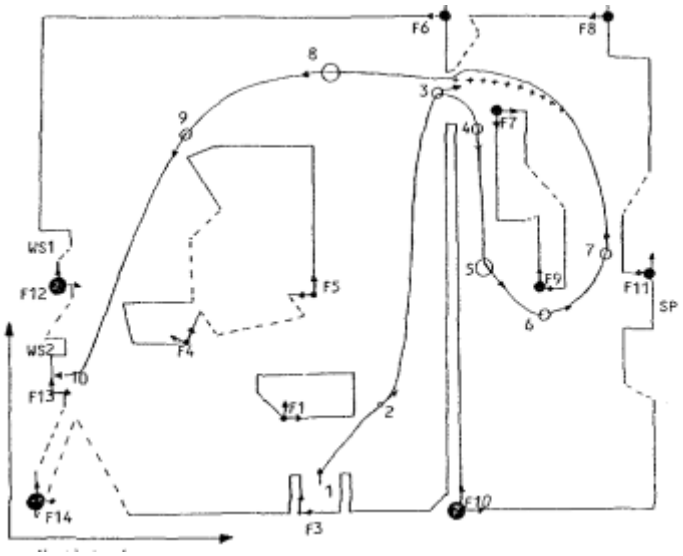



*Figure 13.* An example of boundary-based model. "*Note*. From (Chatila & Laumond, 1985)".

## 3.3 Graph-based Models

Grid-based models are of high accuracy especially when using small cell size. But that means high computational complexity which online positioning systems try to reduce. Graph-based models on the other hand can help in reducing the computational complexity; they represent an indoor space as a graph of nodes and links between the nodes; each node represent predefined locations like rooms, doors, attractions, etc., (Choset & Burdick 2000; Franz et al., 2005; Remolina & Kuipers 2004; Remolina, Fernandez, Kuipers, & Gonzalez, 1999; Werner, Krieg-Brückner, & Herrmann, 2000) which could be extracted either manually or automatically from the building plans. The links or edges represent the possible walking paths between the nodes, in other words they represent the connectivity between nodes. Using the graph model it is very likely that the whole room can be represented as one node, and the moving between two nodes take the shape of line or curve. Unlike grid-based models, there is no space for random walk in this model which makes sense knowing that people tend to walk on straight lines in their natural motion from one place to another (Jiménez Seco, Zampella, Prieto, & Guevara, 2012). The motion limited to specific lines also helps avoiding the effect of the sensors' noise. Figure 14 shows an example of the graph-based map model.

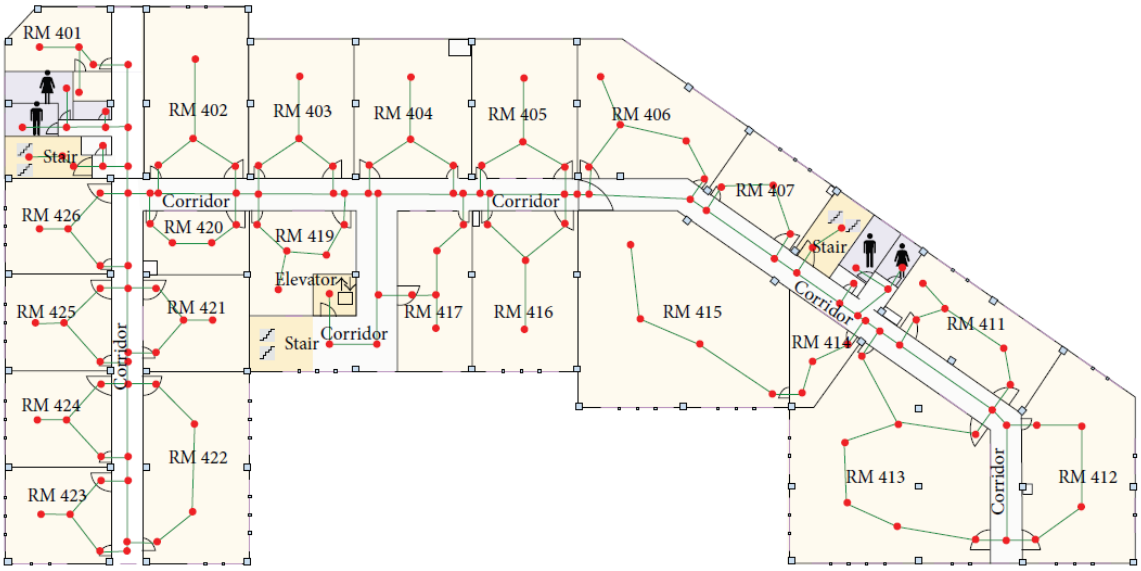



*Figure 14.* The walkable graph model of indoor space: path links (green) and nodes (red). "*Note.* From (Shang et al., 2015)".

Figure 15 shows another example of the graph-based model, where the graph to the right shows the nodes that represent the rooms of the building shown in the left side of the figure and the links of the graph represent the connectivity between the rooms. The graphs can be used to infer topological and semantical relationships such as the connectivity, adjacency and containment properties by annotating nodes and edges and supporting a graph with multiple levels of granularity.

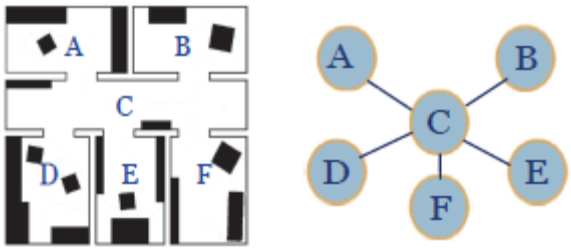



*Figure 15.* A place graph of an indoor space. "*Note.* From (Afyouni et al., 2012)".

Graph-based model are efficient because they represent distributions over a small discrete state space (Shang et al., 2015). And that make them suitable for online pedestrian tracking and navigation systems. But the price is the accuracy; graph models do not provide precise (fine-grained) locations, therefore they might be not suitable for some applications that require high precision (in meters or centimeters).

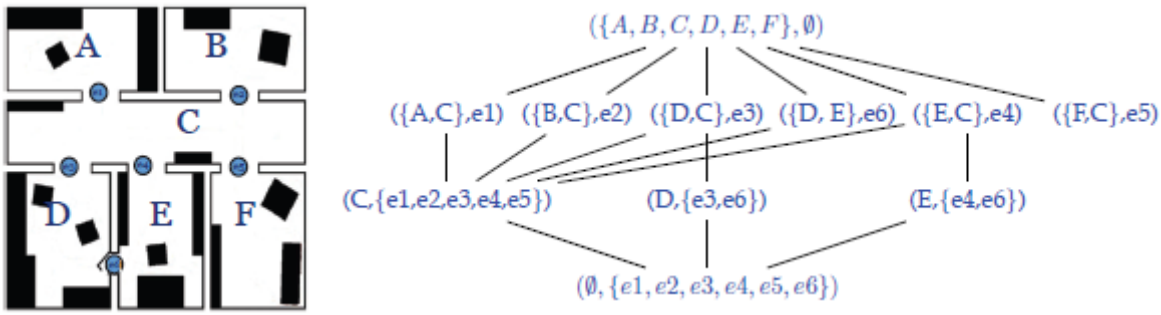
## 3.4 Set-based Symbolic Models

Set-based models symbolically identify places or objects of interest and then gather them into sets. According to Afyouni et al. (2012), Two types of set-based models exist: place-based (Becker & Dürr, 2005; Li & Lee, 2008), and object-oriented models (Bhatt, Dylla, & Hois, 2009). The place-based models form the sets in hierarchical structure depending on the architectural properties of the indoor space. The object oriented models deals with the entities that may contribute in building a smart environment and consider not only places, but all other objects of interest such as walls, doors, and sensors. The symbolic models are complex to create and usually they are application dependent. The accuracy of symbolic model depends on the level of abstraction, that's why they are appropriate for applications that does not require high accuracy or for

indoor positioning systems that requires measuring the Euclidean distance (Afyouni et al.).

### 3.4.1 Place-based Models

The place-based approaches model the space in a hierarchical manner (Dürr & Rothermel, 2003; Kainz, Egenhofer, & Greasley, 1993). It creates sets and subsets based on the architectural properties. For example, the building contains floors; the floor contains rooms, and so on. The model is usually represented as a tree, which is capable of representing the 'contain' relationship between the symbolic locations. But as the conventional tree structure does not help including some relationships such as overlapping, a lattice can be used instead (Kainz et al., 1993; Li & Lee, 2008). Figure 16 shows an example of place-based set models; the sets are organized in a lattice Figure 16 (b), the rooms and the doors assigned to rooms and connectivity between rooms through doors are all presented in the lattice. The overlapping between subset is addressed by allowing for multiple inheritance.




(a) (b)

*Figure 16***.** Example of place-based set model: (a) An exit-location area in a building, (b) the corresponding lattice model. "*Note.* From (Afyouni et al., 2012)".

### 3.4.2 Object-oriented Models

In the object-oriented models, entities are represented as objects, and semantic is added to them using attributes and relations. This semantic depends on the properties of the environment. The Industry Foundation Classes (IFC) for example provide a

specification of an object-oriented data model with a specifications that serves as a data exchange format reflecting building information (Froese et al., 1999; Liebich et al., 2012; Bhatt et al. 2009). In IFC, all entities like walls and doors are represented as objects. Figure 17 shows an example of object-oriented model of indoor space.

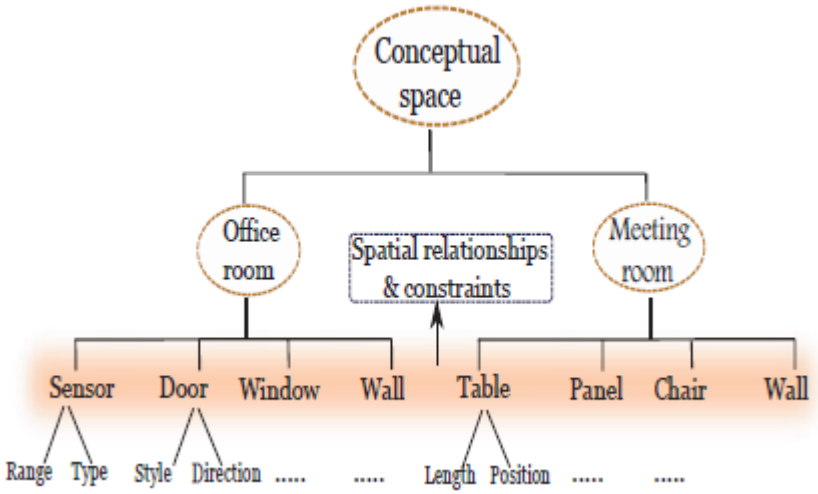



*Figure 17.* An example of object-oriented model of indoor space. "*Note.* From (Afyouni et al., 2012)".

## 3.5 Hybrid Models

Hybrid models combine different modeling techniques. The model proposed in (Li, Claramunt, & Ray, 2010) is a good example of such approach where the model nodes represents grid cells and edges represent the connections between cells (Figure 18 and Figure 19). Nodes and edges are labeled according to spatial unit (e.g. room) that they belong to. This model has performance and scalability issues as for large indoor space to model, a huge number of nodes and edges are required which makes it less suitable for real-time applications (Afyouni et al., 2012). Combining the graph-based approach with grid-based approach helps avoiding the graph-based approach disadvantage of not allowing precise location, and in the same time it keeps the structural-based properties of the space as any other graph-based model.

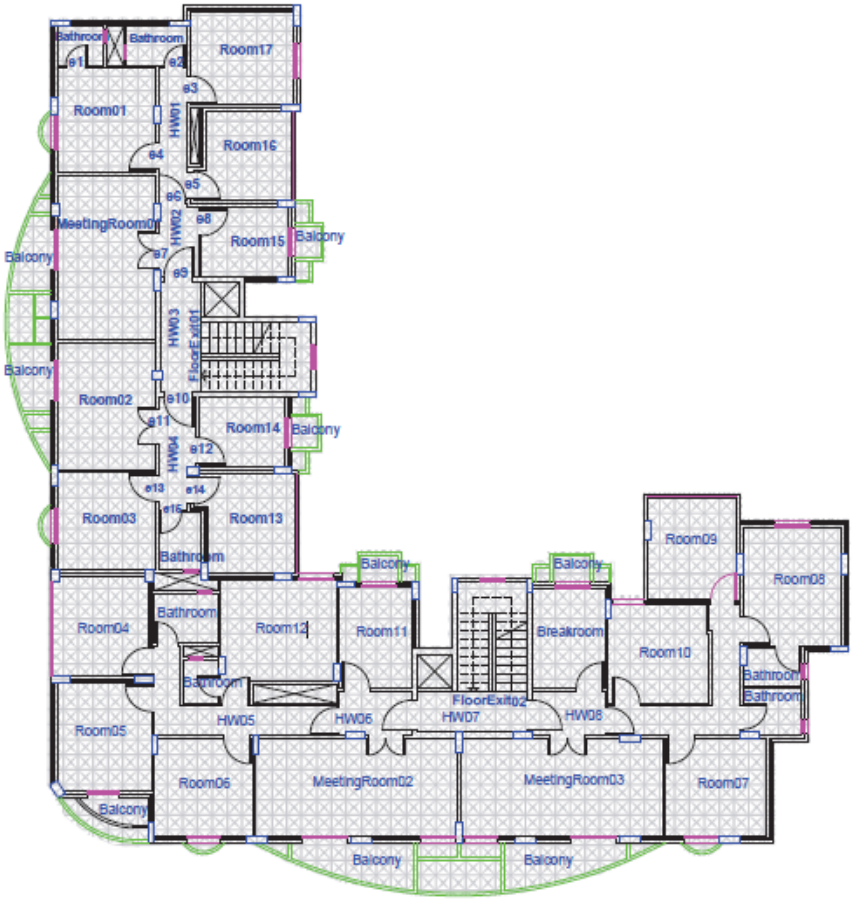



*Figure 18.* Example of hybrid model. "*Note.* From (Afyouni et al., 2012)".

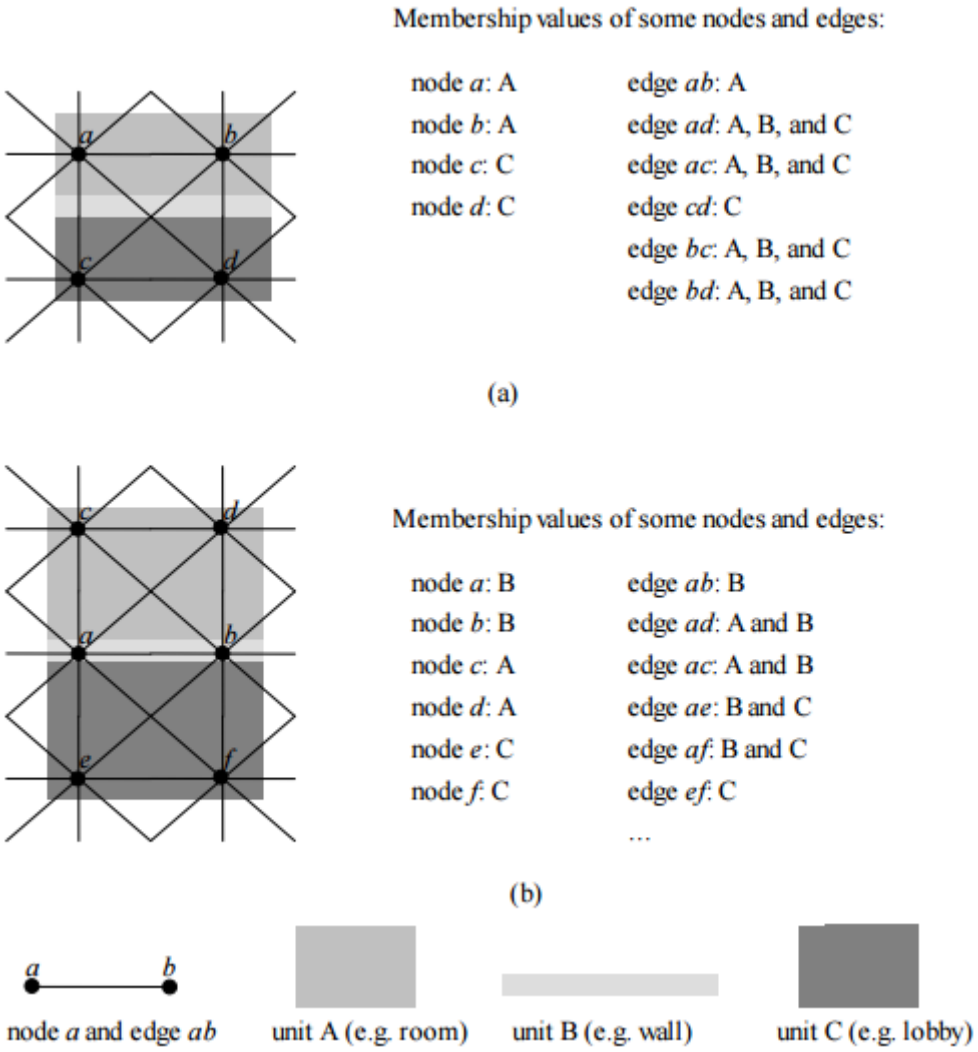



*Figure 19.* Intersections between cellular units and graph components (nodes and edges). "*Note.* From (Li et al., 2010)".

In (Becker & Dürr, 2005) a set-based symbolic locations model was used to combine the benefits of two types of models: the graph-based and the set-based. The set-based part of the model consists of symbolic locations (e.g., sensor, room, and stairs); locations are subsets of this set of locations. As the set-based symbolic models does not support relationships such as connectivity and adjacency between locations graph-based modeling is used to define these relationships; weighted edges between locations are used to define distance in addition to connectivity.

In (Shao & Terzopoulos, 2005) a comprehensive model of pedestrians is proposed in which different map models where used. First, a topological map in which nodes

correspond to environmental regions and edges represent accessibility between 3D regions (e.g., corridor, room, stairs, and floor). Second, a perception maps that includes local uniform grid maps to represent the environment per region basis, and a global uniform grid map that keeps track of the pedestrian the where each grid cells store information about the objects that occupy that cell. And finally, the path maps that include a quadtree map for long-range path planning and a grid map for short-range path planning. The model is presented in Figure 20.

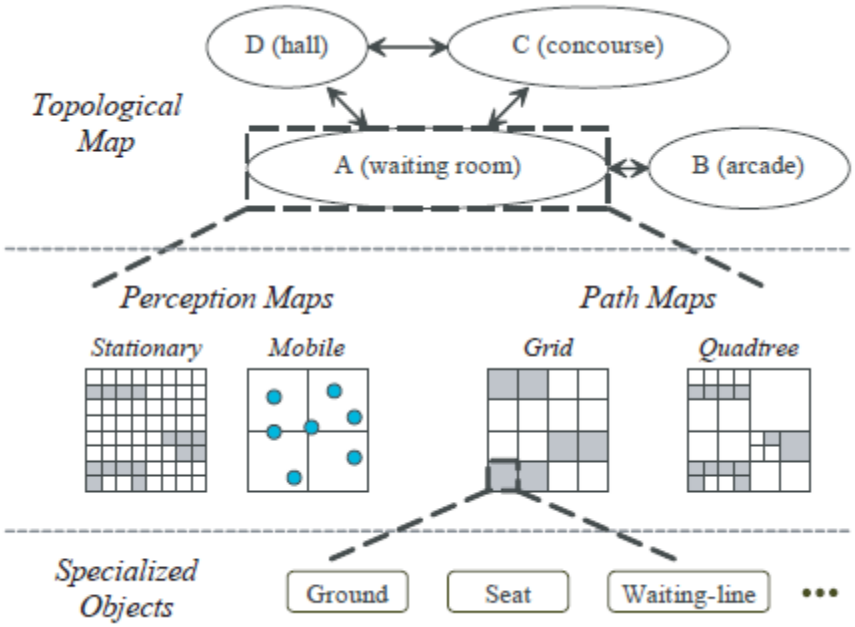



*Figure 20.* Hierarchical environmental model. "*Note.* From (Shao & Terzopoulos, 2005)".

# 4. Map Matching Techniques

The information given by a building plan can be used in local positioning system to improve the localization process. The process of using map information in the localization process is called map matching. Map matching was first used for outdoor localization on roads after obtaining an approximate coarse location using GPS or PDR (Shang et al., 2015). Map matching was then used in indoor environments, but the map matching techniques are different from those of the outdoor environments because building structures are different from road networks.

## 4.1 Outdoor Map Matching

In outdoor environment, the road network map is integrated to the positioning system in order to determine the road segment on which a vehicle travels and on what location on that segment the vehicle is. Different methods have been used for outdoor map matching. These methods can be classified to four categories: geometrical methods, topological methods, simple probabilistic methods and probabilistic methods. In the first category there are three geometric fundamental map matching techniques: point to point, point to curve map curve to curve. Topological map matching uses geometrical information in addition to relationships between geometrical entities such as the connectivity and contiguity of between points, lines and polygons. Simple probabilistic methods use an error region within which the refined location lies. Advanced map matching use geometrical and/or topological information and in addition applies models, such as Kalman filter, particle filter and fuzzy logic. Using advanced methods gives more accurate results than the geometrical and topological methods but they have the drawback of high computational complexity (Quddus, et al., 2007).

### 4.1.1 Geometrical Methods

The simplest geometrical method is point to point method; it is a simple search algorithm in which the estimated position obtained by the navigation system is matched to the closest node on a road segment (Bentley & Maurer, 1980; & Fuchs, Kedem, & Naylor, 1980). Point to point map matching is very sensitive to the way the road network is represented; the case shown in Figure 21 represents an example of such

problem. In the figure it can be see that even the point $P_t$ is much closer to $B_1$ than it is to either $A_0$ or $A_1$, it will be matched to arc B while it is intuitively clear that it should be matched to arc A (White, Bernstein, & Kornhauser, 2000). To reduce such a problem more nodes should be used; the more nodes used, the more accurate the results, although not guaranteed.

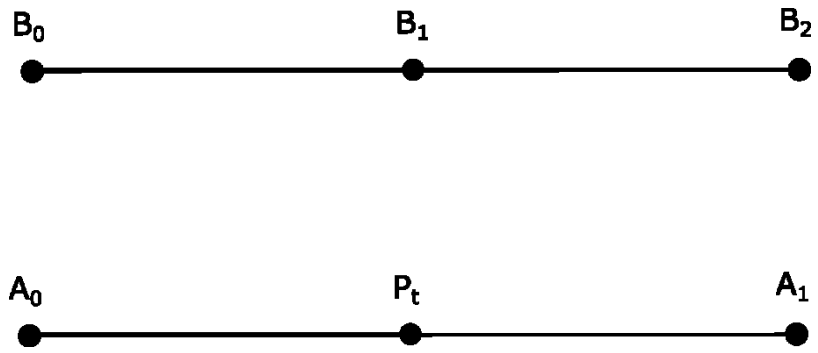



*Figure 21.* A problem with point to point method. "*Note.* Adapted from (white et al., 2000)".

In point to curve method each estimated position is assigned to the closest road segment (curve) i.e. to the segment that has shorter distance to the estimated position (Quddus et al., 2007). An example of point to curve methods can be found in (Bernstein & Kornhauser, 1998; White et al., 2000). The point to curve map matching is more accurate than the point to point map matching, but still it is not guaranteed for accurate solution, especially in dense road network of urban area.

Curve to curve map matching compares the vehicle's trajectory against known roads (Bernstein & Kornhauser, 1998; Quddus et al., 2007; White et al., 2000). This approach finds a matched road segment in three steps. In the first step, it uses point to point map matching to construct piecewise linear curves. Next, it constructs piecewise linear curves using the vehicle's trajectory. And finally, it calculates the curve to curve distance between vehicle's trajectory and the curves corresponding to road segments and selects the closest segment to the vehicle's trajectory as the solution. As it relies on point to point method, curve to curve map matching may adopt its shortcomings. Besides, its sensitivity to outliers may also negatively affect the results (Ochieng, Quddus & Noland, 2003; Ren, 2012).

### 4.1.2 Topological Methods

Topological map matching makes use of the geometric entities (points, lines, and polygons) of the links as well as the relationships (e.g., adjacency, connectivity, contiguity and containment) between the entities (Ochieng et al. 2003; Quddus et al., 2007; Yu, 2006). It takes into account both geometrical and topological information of the road network, as well as, the history of GNSS data. In weight-based topological map matching, which are the most robust and widely used topological map matching algorithms weights are assigned to the links (Yang, Cheng, Jiang, & An, 2013). The performance of topological map matching can be improved by modifying the weighting scheme with additional criteria and parameters such as "vehicle's speed, position relative to candidate road segments, heading information (directly from GPS data), or position data obtained by integrating GPS and DR" (Ren, 2012, p. 40).

### 4.1.3 Simple Probabilistic Methods

In the simple probabilistic methods (Honey et al., 1989) an elliptical or rectangular region of confidence is defined around the position obtained by the navigation sensors. The road segments that are located within the error region represent the candidate segments to be evaluated using the criteria of heading, connectivity and closeness (Quddus et al., 2007). Some algorithms, such as algorithm proposed by Ochieng et al. (2003), restrict the use of the error region to the case of vehicle travel through junctions and not while the vehicle is traveling along within a road segment which makes the algorithm faster.

### 4.1.4 Advanced Methods

Both geometrical and topological map matching algorithms are used as the basis for developing other advanced map matching algorithms. These advanced algorithms employ additional techniques to improve performance, those includes applying probabilistic models. For example, Bayesian models such as Markov models (Rabiner & Juang, 1986), Kalman filter (Grewal, 2011), particle filters (Arulampalam et al., 2002), as well as probabilistic models that are not Bayesian such as Conditional Random Fields models (Lafferty, 2001; Xu et al., 2015). It also includes methods such as an interacting multiple model (Cui & Ge, 2003), a fuzzy logic model (Kim and Kim, 2001; Quddus et al., 2006; Syed and Cannon, 2004). Although they give better results

than the geometrical and topological methods, the advanced methods have high computational complexity and need more resources.

## 4.2 Indoor Map Matching

The indoor map matching is different from outdoor map matching due to the difference between indoor environment and outdoor environment. While outdoor maps have information about roads, the indoor maps have information about walls, doors, rooms, corridors, stairs, etc. The constraints given by the indoor maps are more complex and need special methods to be used in the localization process. In addition, most of the indoor localization applications are pedestrian-oriented that assumes slow motion and random behaviour (Shang et al., 2015).

There are different map matching techniques that have been proposed for indoor localization, some of them are adopted from the outdoor map matching but adapted to the indoor environment and constraints. Indoor map matching methods include geometrical methods, topological methods and advanced probabilistic methods.

### 4.2.1 Geometrical Methods

One of the indoor geometrical methods is the landmark matching (Wang et al., 2012) which represents point to point matching method. The landmarks can be visible such as elevators and stairs, or virtual such as a spot of unusual magnetic measurement (Shang et al. 2015).

The point to point matching is simple fast, but in this context it is very sensitive to the correct recognition of landmarks, which may lead to a big localization error.

### 4.2.2 Topological Methods

Indoor topological map matching uses the geometry as well as topological indoor information such as corridors, corners, rooms and stairs. This information can be used to calibrate errors of the positioning systems such as PDR by using the similarities between the trajectory and the map such as what have been done by Lan and Shih (2013) who have used the geometrical similarity of angles and relative link lengths to infer the last-visited corner using a line-node map model (Figure 22).

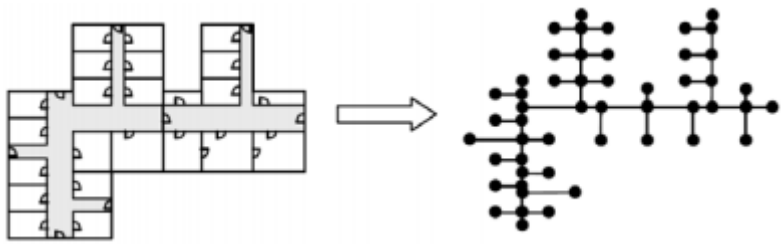


*Figure 22.* Link-node model. "*Note.* From (Lan & Shih, 2013)".

Topological matching gives better results and more robust than geometrical methods but it has higher complexity. It also has poor online capabilities as the matching process does not start until the trajectory becomes long enough (Shang et al, 2015).

### 4.2.3 Advanced Probabilistic Methods

Advanced probabilistic models are used to reduce uncertainty of the location estimations that violate the space constraints such as crossing walls (Shang et al., 2015). Bayesian models such as HMM and particle filters are the most commonly used methods in probabilistic map matching; non-Bayesian probabilistic models such as CRF are also used in some solutions such as the map matching algorithm proposed by Xiao et al. (2014).

Particle filters are efficient and easy to implement methods in solving complex nonlinear and non-Gaussian estimation problems in an online manner. They were developed based on the idea of a selection step that multiplies and/or discards particles at each time, an idea which had been proposed by Gordon et al. (1993) algorithm (the bootstrap filter) (Doucet, Gordon, & Krishnamurthy, 2001).

Particle filter is a "tool for tracking the state of a dynamic system modeled by a Bayesian network" (Hsiao, Miller, & de Plinval-Salgues, 2005, p. 2). It is a very common technique for map matching (Arulampalam, et al., 2002; Widyawan et al., 2008; Li, Zhao, Ding, Gong, Liu, & Zhao, 2012; Rai, Chintalapudi, Padmanabhan, & Sen, 2012; Woodman & Harle, 2008). In a map matching problem, the particle filter represents the probability distribution as a set of particles each of which has a certain weight. The particles are given equal initial weight which will be updated in each step. Every round the locations of the particles are estimated according to a transition model

that determines the transition possibility from one location to another, and their weights are updated according to the observations; for example, if the location of a particle is invalid, the weight of the particle is usually assigned to zero. If the observations such as space constraints or sensor measurements support the location of the particle, the particle will have higher weight. Every time step the particles are re-sampled according to their weights, where the low weight particles are discarded and new particles are generated. Generating particles in invalid locations should be avoided. Overtime, the particles are supposed to converge to the most probable location.

Widyawan et al. (2008) use a Backtracking Particle Filter (BPF) to fuse map details to improve PDR performance. The BPF is a developed particle filter that uses the trajectory history to refine the estimation results. In (Ascher, Kessler, Wankerl, & Trommer 2010) the BPF algorithm was enhanced to refine the 3D estimations in two-floor building by using more map information like elevators and stairs. Figure 23 shows the transition of a particle near an obstacle; a particle with impossible movement like crossing a wall is a failed particle that will be assigned a zero weight in order to be removed from the particle set.

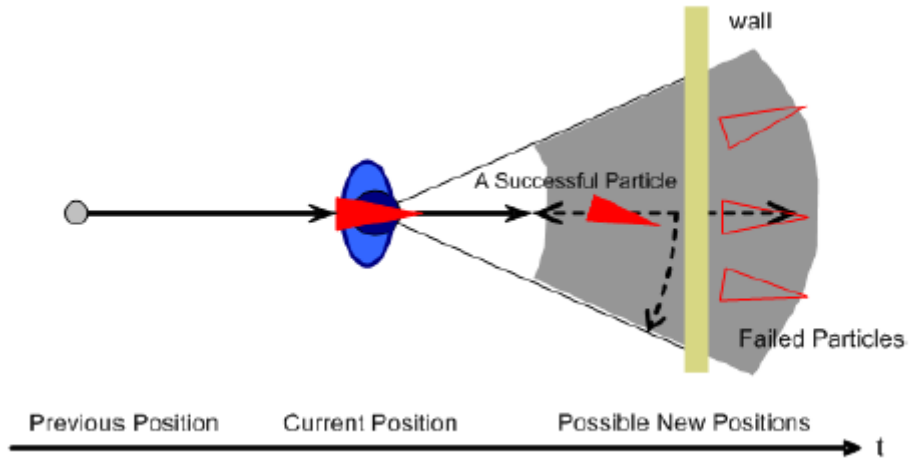



*Figure 23.* The transition of a particle near an obstacle. "*Note.* From (Widyawan et al., 2008)".

In (Ascher et al., 2010) a reduced and simplified particle filter is used in order to reduce computations in enable running on mobile phones. They reduce the calculations by reducing the particle weights to binary values {0, 1}; the particle that crosses a wall is

given zero weight and can be deleted. In the resampling step, a number of particles equal to the number of the deleted particles are generated randomly.

Figure 24 represents a flow chart of the particle filtering algorithm proposed by Spassov, Bierlaire, and Merminod (2006). At the initial state where there still no measurements available thus the pedestrian can be at anyplace in the building, a uniform probability distribution is defined with equal weights for all particles. Then measurements are taken and the weights are updated. In the prediction step a new set of samples are chosen by giving a weight 1 to the neighbour samples of the states at the last time step and a weight 0 to the rest of the samples. Then the algorithm turns to the update step again.

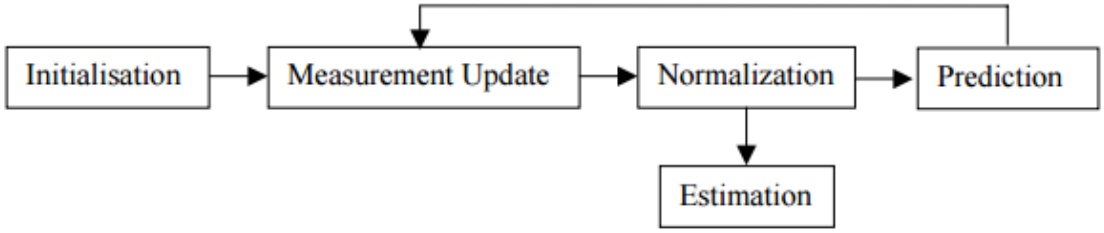



*Figure 24.* The particle filter implementation. "*Note.* From (Spassov et al., 2006)".

Very few particles may lead to improbable states, the improbable states have very few or no particles. This may cause a time delay between the event and the corresponding state; therefore the number of particles should be high enough to make sure all states are represented states (Hsiao et al., 2005); but the more particles used, the higher computational requirements. Turgut and Martin (2009) proposed an approach for automatic restart of the particle filter when the cloud diverges from the observations; in this case the sampling the latest trusted observation is sampled.

HMM has been used for indoor localization. Lamy- Perbal, Guénard, Boukallel, and Landragin-Frassati (2015) propose a HMM solution to the map matching problem adapted to infrastructure-less PDR localization. They model the space to grids with probability densities, and they link the states with transition probability; the inference of the HMM at some time step depends only on the two previous time steps.

Seitz, Vaupel, Meyer, Boronat, and Thielecke (2010) use discrete positions of Wi-Fi fingerprints as the hidden states of the Markov model and received signal strength (RSS) Wi-Fi measurements as the observable states; an input-output 1st order HMM is used in which the states represent user locations and the inertial data are used to calculate transition probability between states. The iterative estimation process is presented in Figure 25.

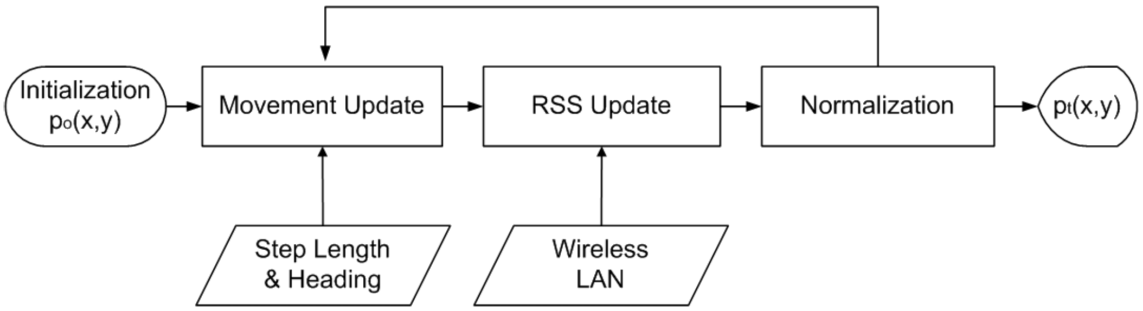



*Figure 25.* Iterative estimation process with the HMM. "*Note.* From (Seitz et al., 2010)".

The HMM "does not take into account correlations between nearby inertial observations, for example correlated magnetometer bias due to the metal disturbances in the earth's magnetic field" (Xiao et al., 2014, p. 133). This negatively affects the accuracy.

Unlike generative models such as HMMs that model the joint probability by applying Bayes' rule, CRFs are discriminative models that model the conditional distribution which make them of less computational complexity. A comparison of generative and discriminative models can be found in (Ng & Jordan, 2002). CRF was applied for the first time in pedestrian indoor localization by Xiao et al. (2014) in their localisation system that is called MapCraft. In MapCraft, the CRF map matching algorithm is tightly coupled to the localization system that takes raw measurements directly from sensors such as Wi-Fi, Bluetooth and inertial sensors as inputs to the algorithm, and fuses it together with the map information. What is meant by tight coupling is that the CRF algorithm here used to fuse all measurements and spatial map information altogether at the same time; both positioning and map matching are done together in one module.

MapCraft is aimed to be used as real time map matching system; however the delay produced by the backward phase of CRF is a real challenge that affects the efficiency of the algorithm when used for real-time tracking; In MapCraft they overcome this problem by making the compromise of converting the conditional probability discrete distribution obtained by the forward phase to a Gaussian distribution and displaying it on the map in real-time. This means less accuracy as it does not give the optimal solution which can only be obtained in the backward phase; that means that the CRF was not implemented completely in MapCraft for the online mode. It is not known exactly to what degree that step affected the accuracy of their system when used online. Nevertheless, their step can be a good example of that CRF is more suitable for offline usage. And as they mentioned clearly in their paper, the optimal path can only be calculated "in the case of delay tolerant offline tracking" (Xiao et al., 2014, p. 137).

It is possible to apply the CRF directly to the raw sensor measurements to estimate the location and match it with map; however this make the map matching dependent on the environment and highly customized, as the types of the available sensors should be known. Another way is to separate the map matching part from localization; location trajectories previously obtained by any localization system can be used and matched with the map offline; the map matching system will be independent of the sensors types and of the localization system, and can be loosely coupled with an existing localization system as an independent module which is separate from the primary localization module; this modular structures means more flexibility.

An example of using CRF as offline map matching algorithm for outdoor systems can be found in the work of Xu et al. (2015); in their work they use the GPS timestamped locations generated by the vehicle as input trajectory for the map matching algorithm and the output is a sequence of road segments traversed by the vehicle. As an outdoor map matching algorithm the map model is based on road segments.

It will be explained later in details how the CRF algorithm functions; but for the sake of comparison it can be said in brief that the CRF employs observations and context information through features functions that relates the observation with state transitions. In MapCraft (Xiao et al., 2014), they used three feature functions depending on observations from sensors: (1) the first function uses displacement and heading measurements from the inertial sensors, (2) the second function is to handle correlations

in heading errors of the inertial sensors, and (3) the third function uses the signal strength observation in conjunction with fingerprint map. In the map matching CRF algorithm proposed in this thesis, no observations from sensors were used as this supposed to be done by a pre-existing localization system; this research only make use of the spatial context provided by the map as presented by the model in addition to the coordinates generated by the primary localization systems as observations. Feature functions that make use of the map information, behavioural information and input coordinates were used.

# 5. Mathematical Background: Probabilistic Models

Predicting multiple dependent variables is a fundamental process to many applications such as image processing and natural language processing (Nadkarni, Ohno-Machado, & Chapman, 2011). In such systems, an output vector $\vec{y}$ of random variables (are called hidden variable, state variables, class variables or labels) is to be predicted given an observed feature vector $\vec{x}$. This is called the multivariable prediction problem, also called classification problem in which a new observation x is classified to the category/label in $\vec{y}$ which it belongs. The success of the classification problem is determined by the number of labels $y_s$ that are correctly classified.

To model a multivariable predicting problem, probabilistic theories are used. There are two categories of probabilistic models: the generative models use joint probability, and the discriminative models that use the conditional probability. These models will be explained through examples. The start will be with generative models; the Naïve Bayes model is an example of a generative model that is used to predict a single variable, and the HMM model is an example of a generative model for predicting multiple variables. After that the discriminative models that are used to predict a single variable through maximum entropy model and to predict multiple variables through conditional random fields will be explained.

The following explanations are inspired by the work of Klinger and Tomanek (2007).

## 5.1 Naïve Bayes

Naïve Bayes represents a family of models that are used to assign labels for an input of feature values by applying Bayes' theorem in the decision rule. Naïve Bayes assumes a strong independence between the features, without any correlation. "For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A Naïve Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features" (Shan, n.d.).

The Bayes' law is formulated as follows:

$$p(y|\vec{x}) = p(y,\vec{x})/\,p(\vec{x}) \quad (1)$$

Where $p(y|\vec{x})$ is the conditional probability distribution, $\vec{x} = (x_1, \dots, x_m)$ is the input vector where m is the number features, and $y$ is the single class variable/ label to be predicted (Klinger & Tomanek, 2007). As the values of the features are given as inputs, the denominator is effectively constant and used as a normalization factor. $p(y,\vec{x})$ is the joint probability which is complex for computation. By applying a decomposition using the chain rule and by considering the Naïve Bayes assumption of the independency between all inputs $x_i$ , $p(y,\vec{x})$ is usually simplified to the formula:

$$p(y,\vec{x}) = p(y) \cdot p(\vec{\mathrm{x}}|\mathrm{y}) \qquad (2)$$

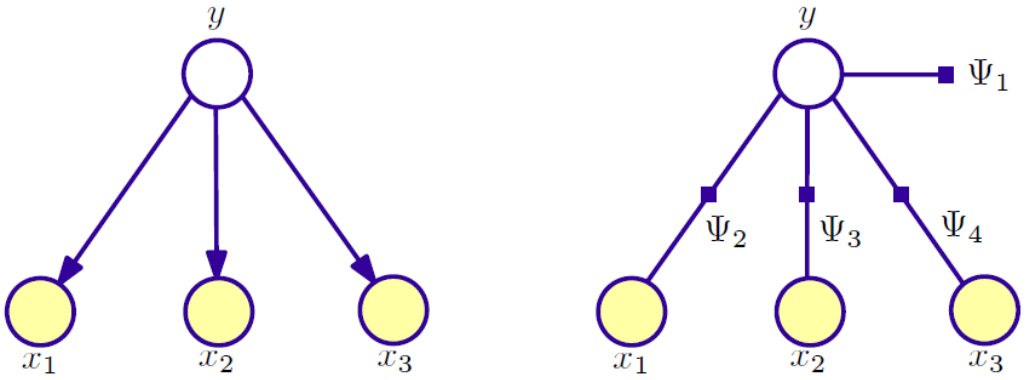
The conditional probability can then be rewritten as:

$$p(y|\vec{x}) = p(y) \cdot p(\vec{x}|y)/\,p(\vec{x}) \quad (3)$$

The probability distributions of probabilistic models can be represented graphically using two types of graphs: the independency graphs and the factor graphs. The independency graph is a graph of nodes that represent all the random variables, and edges that represent the dependency between the variables. An edge between two variable mean that they are dependent; the absence of edges is informative (Klinger & Tomanek 2007; Bishop, 2006). The edges can be directed or undirected edges, depending on the kind of graphical model it represents. See (Sutton & McCallum 2010) for the difference between directed and undirected graph models.

From graphical modeling perspective, the distribution over multiple variables can be represented as a product of local functions (called factors) each of which depends on a subset of input variables; which make it possible represent the distribution in the so-called factor graphs. A factor graph (Kschischang, Frey, & Loeliger 2001) has circular nodes that represent the random variables and small square nodes that represent the factors. The edges link the random variables nodes to the factor nodes, and these edges are always undirected. An independency graph can be transformed into factor graphs (Klinger & Tomanek, 2007). Figure 26 shows both the independency graph and the

factor graph for a Naïve Bayes model with three input variables ($x_1$, $x_2$, and $x_3$). The factorized form of the probability distribution is $p(y, x_1, x_2, x_3) = p(y) \cdot p(x_1|y) \cdot p(x_2|y) \cdot p(x_2|y)$.




(a) Independency graph (b) Factor graph

*Figure 26.* Naïve Bayes Classifier. "*Note.* From (Klinger & Tomanek, 2007)".

## 5.2 Hidden Markov Models

HMMs are directed generative graphical models that are used to predict a sequence of multiple output variables $\vec{y} = (y_1, \ldots, y_n)$. For a sequence of input observation $\vec{x} = (x_1, \ldots, x_n)$, an HMM model is formulated as a product over Naïve Bayes models with one feature at each sequence position:

$$p(\vec{y}, \vec{x}) = \prod_{i=1}^{n} p(y_i) \cdot p(x_i|y_i) \quad (4)$$

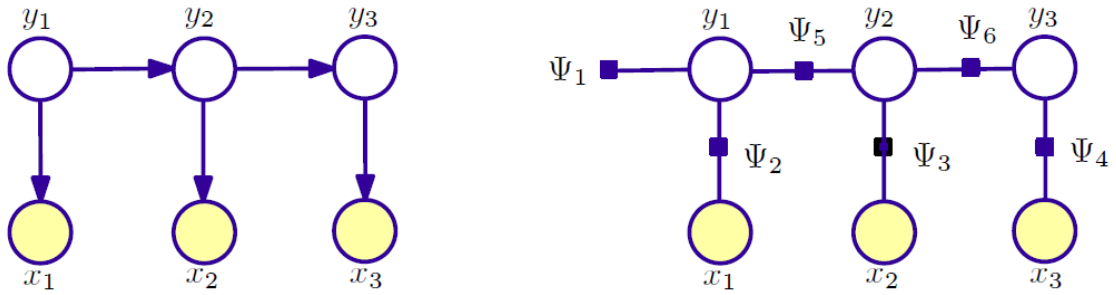
In the previous formula, observation $x_i$ depends only on $y_i$, and dependencies between sequence positions are ignored which means that transition probability between positions are ignored too; this affects the performance of the model and to avoid that dependencies between consecutive positions is added but only between output variable (state variables) $y_i$, as modeling the dependencies between input variables $x_i$ is very complex:

$$p(\vec{y}, \vec{x}) = \prod_{i=1}^{n} p(y_i|y_{i-1})\, p(x_i|y_i) \quad (5)$$

This leads to formulate the HMM model as following:

$$p(\vec{x}) = \sum_{\overrightarrow{y \epsilon Y}} \prod_{i=0}^{n} p\big((y_i|y_{i-1})\big)p(x_i|y_i) \qquad (6)$$

The HMM model can be represented graphically as shown in Figure 27; the figure shows the dependency graph and the factor graph for a sequence of three input variables ($x_1$, $x_2$, and $x_3$). The factorized form of the probability distribution is $P(x_1, x_2, x_3, y_1, y_2, y_3) = \psi_1(y) \cdot \psi_2(x_1, y_1) \cdot \psi_3(x_2, y_2) \cdot \psi_4(x_3, y_3) \cdot \psi_5(y_1, y_2) \cdot \psi_6(y_2, y_3)$. (Klinger & Tomanek, 2007).




(a) Independency graph (b) Factor graph

*Figure 27.* Independency and factor graphs for HMM. "*Note.* From (Klinger & Tomanek, 2007)".

## 5.3 Maximum Entropy Model

The Maximum Entropy Model (see Figure 28) is a conditional probability model that is based on the *Principle of Maximum Entropy* (Jaynes, 1957) which states that if an incomplete information about a probability distribution is given, the only unbiased assumption is a distribution which is as uniform as possible given the available information (i.e. the training data) (Klinger & Tomanek, 2007). That means that for a conditional model $p(y|x)$, the proper probability distribution to find is the model $p^*(y|x)$ that has the largest possible conditional entropy $H(y|x)$ (Korn & Korn, 2000; Bishop, 2006) given the training material. The conditional entropy is defined as:

$$H(y|x) = -\sum_{(x,y)\epsilon Z} p(y,x) \log p\,(y|x) \qquad (7)$$

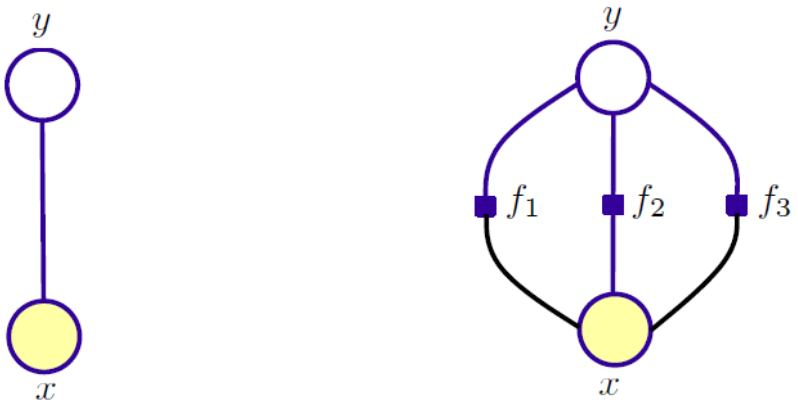
Where $Z = X \times Y$ that contains all possible $(x, y)$ combinations.Thus the objective function $p^*(y|x)$ is:

$$p^*(y|x) = \underset{p(y|x)\in P}{\operatorname{argmax}} H(y|x) \qquad (8)$$

where P is the set of all models that are consistent with the training material[3]. The training material is represented by features that are defined as feature functions, for example with 1 or 0 values. An example of a feature function is:

$$f_i(x, y) = \begin{cases} 1, & if\ distance\ between\ x\ and\ y\ is\ less\ than\ the\ threshold \\ 0, & otherwise \end{cases}$$

The expected value of each feature $f_i$ is estimated from the empirical distribution $\tilde{p}(x, y)$.




(a) Independency graph (b) Factor graph

*Figure 28.* Maximum Entropy Classifier. "*Note.* From (Klinger & Tomanek, 2007)".

The empirical distribution is obtained by simply counting how often the different values of the variables occur in the training data:

$$\tilde{E}(f_i) = \sum_{(x,y)\in Z} \tilde{p}(x, y) f_i(x, y) \qquad (9)$$

[3] "A model is consistent with the training material when the expected value of each feature in the empirical distribution (training dataset) is equal to its expected value in the model's distribution." (Xiao et al. 2014)

All possible pairs $(x, y)$ are taken into account here. As the empirical probability for a pair $(x, y)$which is not contained in the training material is 0, $\tilde{E}(f_i)$ equals the number of occurrence of the feature $f_i$ found with value 1 in the training data T divided by the size of the training set N:

$$\tilde{E}(f_i) = \frac{1}{N} \sum_{(x,y)\epsilon T} f_i(x, y) \qquad (10)$$

The expected value of a feature on the model distribution is:

$$E(f_i) = \sum_{(x,y)\epsilon Z} p(x, y) f_i(x, y) \qquad (11)$$

which can be rewritten as:

$$E(f_i) = \sum_{(x,y)\epsilon Z} p(x) p(y|x)\, f_i(x, y) \qquad (12)$$

To make the calculation of $E(f_i)$ possible an approximation is done by a substitution of $p(x)$ with the empirical distribution $\tilde{p}(x)$ (more details can be found in Lau et al. (1993)):

$$E(f_i) \approx \sum_{(x,y)\epsilon Z} \tilde{p}(x) p(y|x) f_i(x, y) \qquad (13)$$

which in analogue to equation (10) can be transformed into:

$$E(f_i) = \frac{1}{N} \sum_{x\in T} \sum_{y\in Y} p(y|x) f_i(x, y) \qquad (14)$$

The Maximum Entropy Model can then be formulated as following:

$$p^*_{\vec{\lambda}}(y|x) = \left(\frac{1}{Z_{\vec{\lambda}}(x)}\right) exp\left(\sum_{i=1}^{m} \lambda_i f_i(x, y)\right) \qquad (15)$$

and

$$Z_{\vec{\lambda}}(x) = \sum_{y \in Y} \exp\left(\sum_{i=1}^{m} \lambda_i f_i(x, y)\right) \quad (16)$$

For a more detailed discussion of Maximum Entropy Models and related approaches, the book by Pearl (1988) and the Maximum Entropy Tutorial by Berger et al. (1996) are recommended.

## 5.4 Conditional Random Fields

CRFs were first introduced by Lafferty et al. (2001). They are used to model the probability of structured outputs given an input vector. It can have an arbitrary graphical structure which can be understood as an extension to the logistic regression (Hosmer & Lemeshow, 2000; Sutton & McCallum, 2010). Linear Chain Conditional Random Field, a special type of CRFs that models the output variables as a chain and can be viewed as the sequence version of Maximum Entropy Models (Klinger & Tomanek, 2007), in the same way that HMMs is an extension of the naive Bayes classifier model (Xiao et al. 2014), See Figure 29.

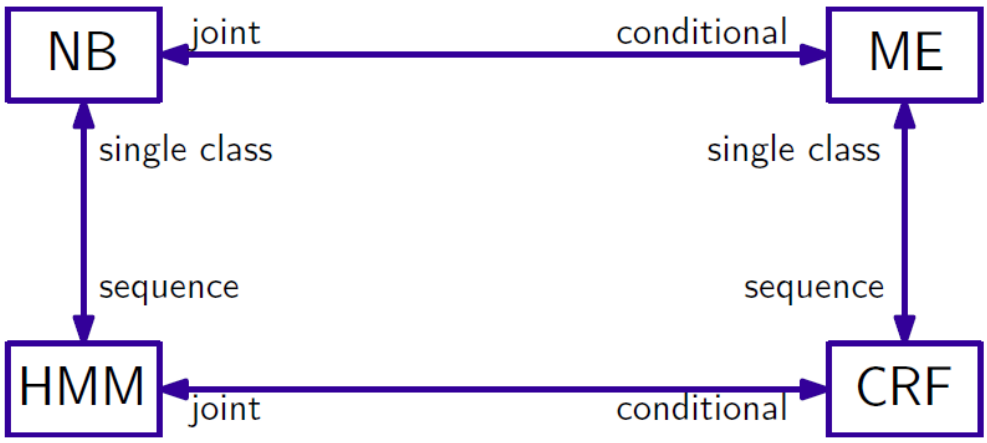
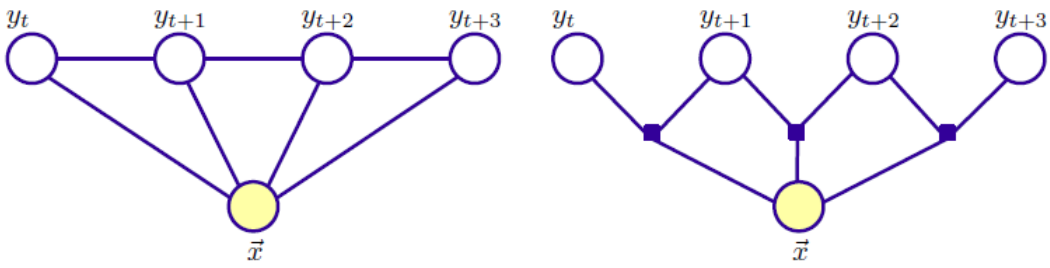



*Figure 29.* Overview of probabilistic models. NB stands for Naïve Bayes, and ME stands for Maximum Entropy. "*Note.* From (Klinger & Tomanek, 2007)".

As undirected probabilistic graphical models developed for labelling data (Klinger & Tomanek, 2007) CRF models are used for an input set of observations $\vec{x}$ to predict a vector of hidden variables $\vec{y}$. Unlike generative models such as HMMs that model the

joint probability $p(\vec{x}, \vec{y})$ by applying Bayes' rules, CRFs are discriminative models that model the conditional distribution $p(\vec{x}|\vec{y})$ over the hidden variables $\vec{y}$ given observation vector $\vec{x}$. In linear chain CRFs, a special form of CRF graphs that model the output variable as a sequence (Lafferty et al., 2001), the conditional probability $p(\vec{x}|\vec{y})$ is proportional to the product of potential functions that link observations to consecutive states. Figure 30 shows a representation of the linear chain CRF model. The hidden states $\vec{y}$ are dependent on input observation vector $\vec{x}$; each hidden state depends not on one input value but rather on the whole input vector. As a result they can be affected by input observations from different time steps; which is considered an advantage of the CRF technique.




(a) Independency graph (b) Factor graph

*Figure 30.* Linear Chain CRF model. "*Note.* From (Klinger & Tomanek, 2007)".

In linear chain CRFs, the conditional probability of states $\vec{y}$ given observations $\vec{x}$, is proportional to the product of potential functions $\psi_j(\vec{x}, \vec{y})$ that link observations to consecutive states (Xiao et al, 2014). The general formula of CRF can be written as:

$$p(\vec{y}|\vec{x}) = \left(\frac{1}{Z(\vec{x})}\right) \cdot \prod_{j=1}^{n} \psi_j(\vec{x}, \vec{y}) \quad (17)$$

Where Z is the normalization factor Z that normalizes to [0; 1]:

$$Z(\vec{x}) = \sum_{\vec{y}'} \prod_{j=1}^{n} \psi_j(\vec{x}, \vec{y}') \quad (18)$$

The potential function $\psi_j(\vec{x}, \vec{y})$ can be written in the following form:

$$\psi_j(\vec{x},\vec{y}) = \exp(\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)) \qquad (19)$$

Assuming the length of the sequence = n+1, then the Linear Chain CRF can be written as:

$$p_{\vec{\lambda}}(\vec{y}|\vec{x}) = \left(\frac{1}{Z_{\vec{\lambda}}(x)}\right) \cdot exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right) \qquad (20)$$

The index j is the position in the input sequence $\vec{x}$ , and it is needed in the equation in comparison to the Maximum Entropy Model as a label sequence is to be predicted not a single label. (Klinger & Tomanek, 2007).

The normalization factor can be written as:

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{\vec{y}\in Y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right) \qquad (21)$$

The Linear Chain CRF can be rewritten as:

$$p_{\vec{\lambda}}(\vec{y}|\vec{x}) = \left(\frac{1}{Z_{\vec{\lambda}}(\vec{x})}\right) . \prod_{j=1}^{n} exp\left(\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right) \qquad (22)$$

as well as,

$$p_{\vec{\lambda}}(\vec{y}|\vec{x}) = \left(\frac{1}{Z_{\vec{\lambda}}(\vec{x})}\right) . \prod_{i=1}^{m} exp\left(\sum_{j=1}^{n} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right) \qquad (23)$$

or

$$p_{\vec{\lambda}}(\vec{y}|\vec{x}) = \left(\frac{1}{Z_{\vec{\lambda}}(\vec{x})}\right) . \prod_{i=1}^{m}\prod_{j=1}^{n} exp\left(\lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right) \qquad (24)$$

More details about the Linear Chain CRFs and on arbitrarily structured CRFs can be found in (Klinger & Tomanek, 2007).

# Part 2:

# Conditional Random Field-Based Offline Map Matching for Indoor Positioning

# 6. System Architecture

In this research a loose coupling architecture for indoor localization and map matching is developed. A semantic map generation tool and an offline CRF map matching algorithm that can be easily loosely coupled to an existing localization system were designed such that the map matching algorithm uses the output estimated walking trajectory of the localization system as an input and refines it based on only the map information without the direct use of raw sensor data. The adoption of loose coupling means that the map matching algorithm does not need to know the implementation details of the primary localization system; this also then allows the map matching algorithm to be linked to any localization system as a separate module and to use only its output. Any change in the localization system will not affect the implementation of the map matching algorithm. The system architecture is illustrated in Figure 31. As can be seen from this figure, the system is composed of three subsystems, with some interconnections between them. The three components of the system are:

1. The primary localization system that produces the first estimated trajectory. This can be any localization system that produces as its output a walking trajectory in the form of a time sequence of estimated location coordinates. This trajectory will be the input of the map matching algorithm.
2. The semantic map generation system, a unit that models the floor plan obtained from CAD files in a semantic format that can be used by the map matching algorithm. In our case, a grid-based map model was employed in which the floor plan map is divided into square uniform-grid cells. Each cell is associated with a semantic representation of its contents, for example if it contains a wall or a free space.
3. The map matching algorithm that refines the estimated trajectory (path) using the CRF technique and the semantic floor plan information.

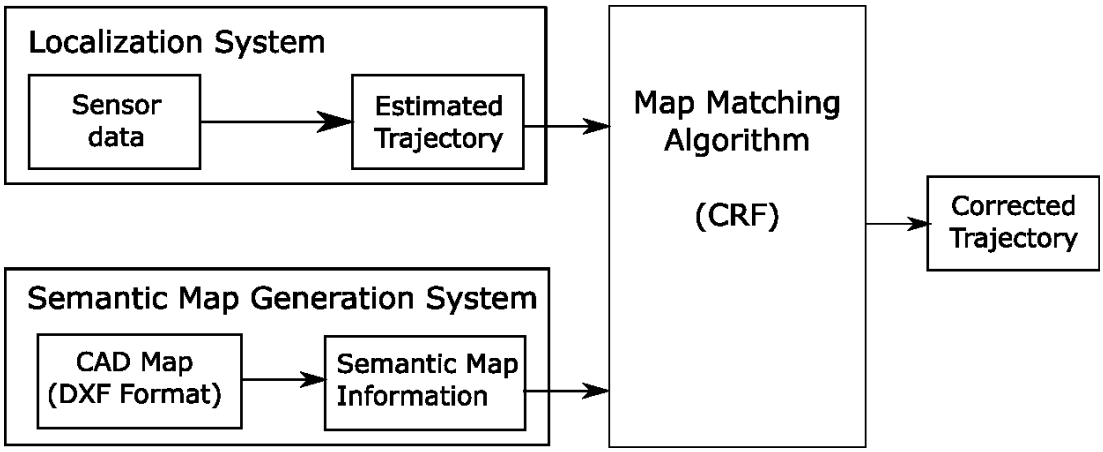



*Figure 31.* System Architecture, illustrating the coupling between the proposed map matching algorithm and the localization and semantic map systems.

System operation proceeds as follows: Once the primary estimated trajectory is obtained by the localization system, map matching is carried out by applying the CRF model that also uses the information obtained from the map to produce a refined (corrected) trajectory. The goal is to estimate the most feasible trajectory, taking into account the constraints provided by the map such as walls that are obtained using the semantic map generation system which is a separate unit that extracts map data from CAD files and represents it in a model so it can be used by the algorithm.

## 6.1 The Primary Localization System

The suitable primary localization system is a localization system that produces as output a trajectory of a pedestrian represented in (x, y) coordinates of the estimated locations which will be the input of our map matching algorithm. In the experiments of this research the used primary localization system is a step-and-heading based PDR system that used a wrist-worn inertial measurement unit composed of three accelerometers and three gyroscopes. Heading drift was reduced by applying the improved Heuristic Drift Elimination (iHDE) method (Díez et al., 2016).

## 6.2 The Semantic Map Generation System

This module contains two processes, the first is to extract the map information from its available format, and the second is to represent this information in a format suitable for

the map matching algorithm. The detailed implementation of these two processes is explained in the following subsections 6.2.1 and 6.2.2.

### 6.2.1 Extraction of Map Information

Indoor maps can be available in different formats, such as images, pdf files, or CAD files. These formats are not suitable to be directly used by the map matching algorithm. Two tasks should be performed first, the map information should be extracted from the available map files, and then this information should be represented in another format usable by the map matching algorithm. In the system developed in this research the CAD files have been chosen because CAD is very widely used in architectural design; the maps were modelled the using a grid-based model.

AutoDesk[4] has developed the drawing interchange file format called DXF to enable data interchange between AutoCAD and other programs. "The DXF format is a tagged data representation of all the information contained in an AutoCAD drawing file" ("DXF reference").

The DXF file is organized of four sections, plus the END OF FILE marker, as following ("Drawing Interchange and File Formats, Release 12", n.d.):

1. HEADER section: This section is a comment that includes general information about the drawing like version numbers, layer and grid spacing, as well as the creator of the file and creation date.
2. TABLES section - This section contains predefined constants like: Line type table (LTYPE), Layer table (LAYER), Text Style table (STYLE), View table (VIEW), User Coordinate System table (UCS), Viewport configuration table (VPORT), Dimension Style table (DIMSTYLE), Application Identification table (APPID),
3. BLOCKS section - This section contains the block definitions each of which contains the entities that make up that block as it used in the drawing.
4. ENTITIES section - This section contains description of the geometric entities used in the drawing.
5. END OF FILE

[4] www.autodesk.com

Table 1 is a representation of a DXF file that includes only the section markers and table headers. The file sections and different tables are separated using separator groups to delimiters (ENDSEC and ENDTAB). (“Drawing Interchange and File Formats, Release 12”).

*Table 1*. A representation of a DXF file with basic sections. “*Note.* Adapted from (“Drawing Interchange and File Formats, Release 12”, n.d.)”.

| | |
|---|---|
| 0 | *(Begin HEADER section)* |
| SECTION | |
| 2 | |
| HEADER | |
| *<<<<Header variable items go here>>>>* | |
| 0 | |
| ENDSEC | *(End HEADER section)* |
| 0 | *(Begin TABLES section)* |
| SECTION | |
| 2 | |
| TABLES | |
| 0 | |
| TABLE | *(First table)* |
| 2 | |
| VPORT | |
| 70 | |
| *(viewport table maximum item count)* | |
| *<<<<viewport table items go here>>>>* | |
| 0 | |
| ENDTAB | *(End of first table)* |
| 0 | |
| TABLE | *(Second table)* |
| 2 | |
| APPID, DIMSTYLE, LTYPE, LAYER, STYLE, UCS, VIEW, or VPORT | |
| 70 | |
| *(Table maximum item count)* | |
| *<<<<Table items go here>>>>* | |
| 0 | |
| ENDTAB | *End of second table)* |
| 0 | |
| ENDSEC | *(End TABLES section)* |
| 0 | *(Begin BLOCKS section)* |
| SECTION | |
| 2 | |
| BLOCKS | |
| *<<<<Block definition entities go here>>>>* | |
| 0 | |
| ENDSEC | *(End BLOCKS section)* |
| 0 | (Begin ENTITIES section) |
| SECTION | |
| 2 | |
| ENTITIES | |

| | |
|---|---|
| *<<<<Drawing entities go here>>>>* | |
| 0 | |
| ENDSEC | *(End ENTITIES section)* |
| 0 | |
| EOF | *(End of file)* |

In DXF, the geometrical shapes of CAD data are encoded as the basic entities (lines, curves, circles and polylines) distributed over different layers. Doors for example are represented as curves and lines. The information about these entities is included in the ENTITIES section of the DXF file. Table 2 represents an example of information of a line as it appears in the DXF file, this section is supposed to be a part of the ENTITIES section and represents one of the lines in the CAD file that lies in on layer 2 and is of color number 4 and has the start point (7800, 5670.6, 0) and the end point (7800, 11377.89, 0).

*Table 2.* A DXF representation of a LINE.

| | |
|---|---|
| 0 | →*entity code* |
| LINE | →*entity string value* |
| 8 | →*layer code* |
| 2 | →*layer value* |
| 62 | →*color code* |
| 4 | →*color value* |
| 10 | →*floating point code* |
| 7800 | →*start point x coordinate: x1* |
| 20 | →*floating point code* |
| 5670.6 | →*start point y coordinate: y1* |
| 30 | →*floating point code* |
| 0 | →*start point z coordinate: z1* |
| 11 | →*floating point code* |
| 7800 | →*end point x coordinate: x2* |
| 21 | →*floating point code* |
| 11377.89 | →*end point y coordinate: y2* |
| 31 | →*floating point code* |
| 0 | →*end point z coordinate: z2* |

In the DXF file, after each string value “LINE” there are 16 rows that represents eight code/values pairs, the first two values are for the layer and for the color, the next six values represents the 3D coordinates in the following order (x1, y1, z1, x2, y2, z2). If the DXF file represents a 2D plan, the third value is usually set to null.

The CIRCLE has three values that represent the 3D coordinates of the center of the circle and one value that represents the radius (see Table 3).

*Table 3.* A DXF representation of a Circle.

| | |
|---|---|
| CIRCLE | |
| 5 | |
| 5 | → *Entity handle (hexadecimal string)* |
| 223 | |
| 8 | → *layer value* |
| 0 | |
| 10 | |
| 2019.9788218124791 | → *center x coordinate* |
| 20 | |
| 1762.6794886488881 | → *center y coordinate* |
| 30 | |
| 0.0 | → *center z coordinate* |
| 40 | |
| 345.74200832680771 | → *radius* |
| 0 | |
| ENDSEC | |

The ARC has a 3D coordinate for the centre, one value for the radius and two values that represent the start angle and end angle of the ARC, a DXF representation of an ARC is shown in Table 4.

*Table 4.* A DXF representation of an ARC.

| | |
|---|---|
| 0 | |
| ARC | |
| 5 | |
| 1F72 | → *Entity handle (hexadecimal string)* |
| 8 | |
| DOORS | → *layer value (string)* |
| 10 | |
| 22.938335432871298 | → *center x coordinate* |
| 20 | |
| 6.0734978622693134 | → *center y coordinate* |
| 30 | |
| 0.0 | → *center z coordinate* |
| 40 | |
| 0.9999989070184941 | → *radius* |
| 50 | |
| 270.00000149560202 | → *start angle* |
| 51 | |
| 359.99999113523859 | → *end angle* |

The POLYLINE is a sequence of segments; it is represented in the DXF file by the coordinates of the vertices that connect the POLYLINE segments. Following is an example of DXF file contents that represents a POLYLINE; after the word POLYLINE we can see different group codes and value, like 10, 20, and 30 codes that represents the polyline elevation and the vertices-follow flag (code 66) which has a fixed value of 1 to indicate that a series of Vertex entities is expected to follow the POLYLINE,; the sequence is ended with a sequence end (SECEND). Each VERTEX word is followed by information about the entity and layer and three code/value pairs (10, 20, and 30) that represent its coordinates. Table 5 shows an example a POLYLINE that has three vertices as represented in a DXF file. More details are available in ("Drawing Interchange and File Formats, Release 12")

*Table 5.* A DXF representation of a POLYLINE.

| | |
|---|---|
| 0 | |
| POLYLINE | |
| 5 | →*Entity handle code* |
| 2165 | →*Entity handle value (hexadecimal string)* |
| 8 | → *layer code* |
| DOORS | →*layer value (string)* |
| 66 | →*code of "vertices follow flag"(fixed)* |
| 1 | →*value of "vertices follow flag"(always equal to1)* |
| 10 | → *code for polyline elevation* |
| 0.0 | |
| 20 | → *code for polyline elevation* |
| 0.0 | |
| 30 | → *code for polyline elevation - supplies elevation* |
| 0.0 | |
| 70 | →*Polyline flag (optional)* |
| 1 | →*closed polyline* |
| 0 | |
| VERTEX | →*first vertex in the polyline* |
| 5 | |
| 489C | |
| 8 | |
| DOORS | |
| 10 | |
| -29.842451282359718 | → *x coordinate of the first vertex* |
| 20 | |
| 0.1494880089802635 | →*y coordinate of the first vertex* |
| 30 | |
| 0 | → *z coordinate of the first vertex* |
| VERTEX | → *second vertex in the polyline* |
| 5 | |
| 489D | |
| 8 | |
| DOORS | |
| 10 | |
| -28.944451290878821 | → *x coordinate of the second vertex* |
| 20 | |
| 0.1494880089802635 | → *y coordinate of the second vertex* |
| 30 | |
| 0 | → *z coordinate of the second vertex* |
| VERTEX | |
| 5 | |
| 489E | |
| 8 | |
| DOORS | |
| 10 | |
| -28.944451290878821 | → *x coordinate of the third vertex* |
| 20 | |
| 0.1114931244940251 | →*y coordinate of the third vertex* |
| 30 | |
| 0 | → *z coordinate of the third vertex* |
| VERTEX | |
| 5 | |

| | |
|---|---|
| 489F | |
| 8 | |
| DOORS | |
| 10 | |
| -29.842451282359718 | → *x coordinate of the third vertex* |
| 20 | |
| 0.1114931244940251 | → *y coordinate of the third vertex* |
| 30 | |
| 0.0 | → *z coordinate of the third vertex* |
| SEQEND | |

The semantic map generation unit extracts the map data from the CAD files before modelling it in a semantic model suitable for the map matching algorithm. Figure 32 represent the plane map of the 4th floor of DeustoTech building in the University of Deusto as extracted from CAD DXF files using Matlab 14. However, map information obtained from DXF files is not suitable for localisation applications, as it is not enhanced with semantic information that allows computers to understand the architectural structure of the building and to distinguish between different architectural objects such as walls and stairs (Schäfer, Knapp, Chakraborty, 2011). Instead, DXF-derived map information comprises only line, curve, circle and polyline drawing data, which means that it requires more processing in order to be used by the proposed map matching algorithm. Therefore, the DXF information are extracted and represented using the proposed map model, which is based on dividing the map into square cells and determining the possible transitions of the pedestrian from one cell to another depending on the existence of obstacles. The simple entity information are extracted from the DXF files and located in the corresponding grid cells in the map model of the floor plan. Figure 33 shows DeustoTech floor plan in the grid-based representation.

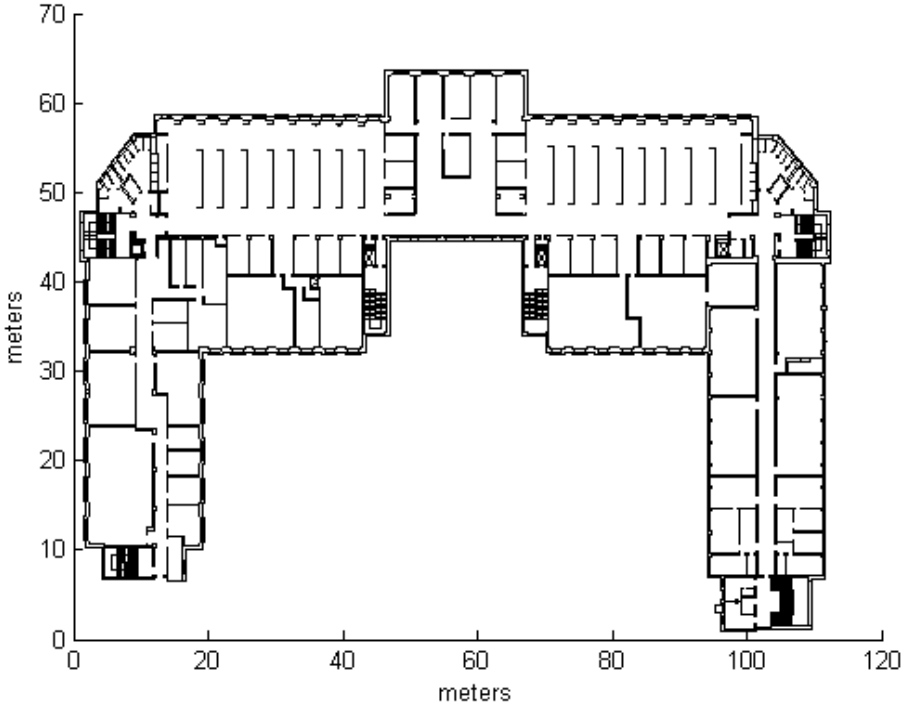



*Figure 32.* Map information extracted from DXF files and drawn using Matlab.

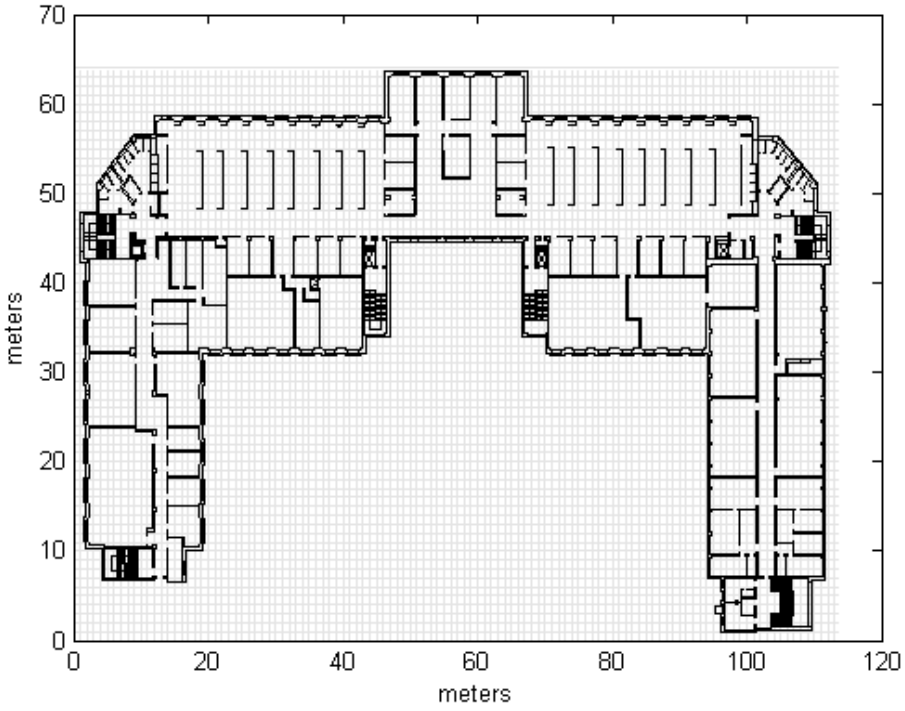



*Figure 33*. Grid-based representation of the map, cell size = 1 m.

### 6.2.2 The Map Model

As will be explained explain in the next subsection, the proposed map matching model is a Linear Chain CRF model that is a classifier that requires pre-defined states/labels. To this end, the proposed map model is designed to suit this specific purpose by representing the map as a group of square cells, with each cell representing a squared area in the building; to the algorithm this represents a state/label that can be used to specify the position of the pedestrian. In the model, each cell has its own characteristics; the cell can represent a free space in the building or can be occupied by an obstacle such as a wall or furniture. For the purpose of representing state transitions, each cell knows its neighbour cells and the neighbours of neighbour cells. A transition graph/table is generated in such a way that the transition is only possible between neighbour cells that do not contain any obstacle; for more flexibility, a transition to a neighbour of a neighbour cell is also allowed under the condition that there are no obstacles impeding that transition. Staying in the same cell is also allowed. The search region for the next location from the current location is known as the buffer (Shang et al., 2015) with the maximum transition distance allowed in each direction known as the buffer size. Allowing transitions to neighbour cells and neighbours of neighbours results in a buffer size of two cells in each direction, as shown in Figure 34 (a). Figure 34 (b) represents an example of possible transitions based on the existence of obstacles.

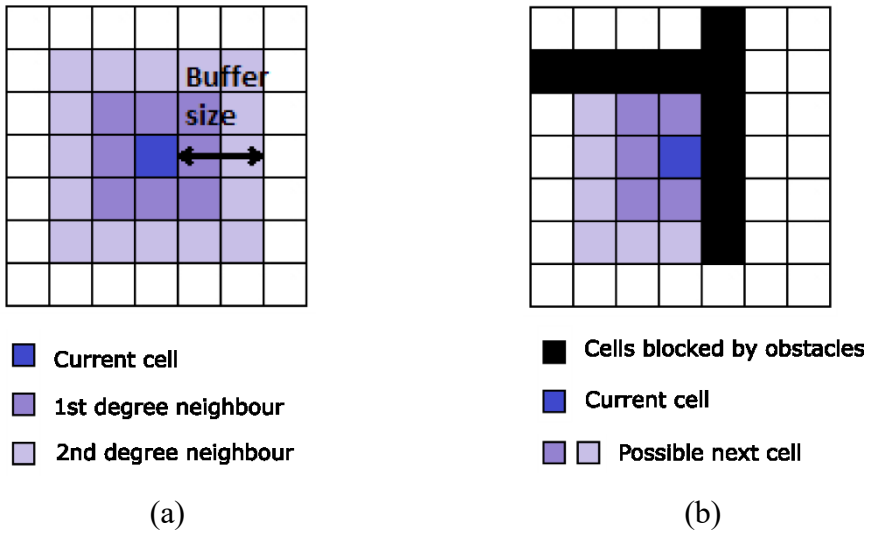



*Figure 34.* The map model; search region for the next location, including allowed transitions in the next step (a) and allowed steps in the presence of obstacles (b).

## 6.3 The Map Matching Algorithm

A CRF model is used as the base model in the developed system. In the map matching problem, the hidden state vector $\vec{y}$ represents the sequence of locations to be calculated, i.e. the corrected walking trajectory; the proposed Linear Chain CRF algorithm uses the cells in the map model as the hidden states/labels. Vector $\vec{x}$ represents the system input that is time series of coordinates of an estimated trajectory obtained by some localization system.

The CRF algorithm consists of two phases: the forward phase and the backward phase. Following the calculation of the conditional probabilities of all cells in all time steps during the forward phase, inference is carried out in the second phase (backward phase), with the optimal trajectory chosen among different candidate trajectories.

During the forward phase, the CRF algorithm evaluates the possible transitions at each time step according to the input trajectory and the transition graph obtained from the map model; this involves calculating the probabilities of transition from all cells of the current time step to all cells of the next time step. A probability value is assigned to each cell in the map at each time step; this value represents how probable it is that the pedestrian is located in that cell at that time step (for example, the probability of pedestrian movement to a cell occupied by a wall is zero). This probability is also a conditional probability calculated using so-called feature functions that compute to what degree the input observations support the choice of a cell to be on the trajectory at that time step. However, cell selection depends not only on its probability value at the current time step, but also on the previous time steps, i.e. the path history, and how this cell is related to others in the path. Hence, choosing the cell with the highest probability is not enough; the probability of a whole trajectory should be calculated.

The feature functions specify how the transition between two states is supported by the set of observations $\vec{x}$.The potential function of a cell is calculated at each time step by calculating the exponential of the summation of all feature functions that support the selection of this cell multiplied by the transition probability of moving to this cell from the current cell at the current time step. The higher the potential function value, the higher the probability of the cell to be the next cell. At each time step *j* and using feature functions $f_i$ the potential function can be written as:

$$\psi_j(\vec{x},\vec{y}) = \exp(\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)) \qquad (25)$$

where $m$ is the number of features and $\lambda_i$ is the feature weight that can be determined by training the model. The conditional probability $p(\vec{x}|\vec{y})$ of each cell is calculated by normalizing the potential function as follows:

$$p(\vec{y}|x) = \left(\frac{1}{Z(\vec{x})}\right) \cdot \prod_{j=1}^{n} \psi_j(\vec{x},\vec{y})$$

$$= \left(\frac{1}{Z(\vec{x})}\right) \cdot \exp(\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)) \qquad (26)$$

where $n$ is the number of output states/cells and $Z$ is the normalization factor, with

$$Z(\vec{x}) = \sum_{\vec{x}} \prod_{j=1}^{n} \psi_j(\vec{x},\vec{y}) \qquad (27)$$

If N is the number of cells, then at every time step the potential functions should be calculated $N^2$ times; however, knowing that transition can only happen between neighbour cells, one need only calculate the potential function of those neighbours. Thus, the calculation number at each step is O(N) and the complexity of the whole procedure is O(NM), where M is the number of time step observations.

For each observation, each cell stores only one conditional probability value (the highest), while the previous cell that gave this value is also stored as the best parent. This is necessary for the following inference step.

During the backward phase, inference is implemented in order to estimate the location over time, with the most likely sequence of hidden states calculated by maximising the sum of the conditional probability function. After the potential function for all cells is calculated for each observation, the optimal path is determined. This is carried out via a backward process using dynamic programming, particularly the Viterbi algorithm

(Forney, 1973). The optimal path is obtained by maximising the sum of conditional probabilities along the path:

$$\vec{y}^{\,*} = \underset{\vec{y}}{\operatorname{argmax}}\, p(\vec{y}|\vec{x}) \qquad (28)$$

Although this process might be assumed to be of high complexity, as it is needed to choose between all possible trajectories, it is actually linear because for each time step only one value of the conditional probability for each cell is saved. Hence, the number of candidate paths equals the number of cells (N). The backward process is O(NT), as backward process is started for each cell by determining the best parent, which is the neighbour cell with the highest conditional probability at the previous time step, before summing these conditional probabilities along the path until reaching the first parent in the path. The path with the highest sum is then chosen. The forward and backward phases of the map matching CRF algorithm are shown in Algorithm 1.

The main feature used in the proposed CRF map matching algorithm is the Euclidean distance (in meters) between the centre of the candidate cell and the estimated position derived from the primary localization system; a smaller distance means a higher value of the potential function given that the transition to this cell is possible, as determined by the transition graph obtained from the map model. The proposed feature function can thus be defined as follows:

$$f_{dis} = \frac{T(y_{j-1}, y_j)}{Distance\,(x_j, y_j)} \qquad (29)$$

where the minimum distance equals cell-size/2 and where $T$ is a function that indicates the transition possibility from the current cell to the candidate cell, depending on the transition table, and which can be either 0 or 1, $j$ is the time step and $x$ is the current observation, which is the current location estimated by the primary localization system. This feature means that cells close to the current input locations have higher probabilities than far cells, on the condition that transition to the cell is possible from its neighbour cells and no obstacle forbids this transition. Adding other features will be discussed later.

*Algorithm 1.* CRF algorithm for the map matching problem

| | |
|---|---|
| 1: | **Input :** Observation= a vector of coordinates of the input estimated trajectory |
| 2: | **Output**: CorrectedPath = a vector of coordinates of the output corrected trajectory |
| | *% Forward Phase:* |
| 3: | **For** each observation (Observation$_j$) *%Observation $_j$ = coordinates of the input at step j* |
| 4: | **For** all cells (i) |
| 5: | **For** all neighbour cells of i (k) |
| 6: | $f_{dis}$= T(i,k) / Distance (k,Observation$_j$) *%T(i,k): transition possibility from cell i to cell k {0,1}* |
| 7: | Potential(k,Observation$_j$)= exp($f_{dis}$); |
| 8: | Z =sum(Potential ( : , j ) ) *% normalization factor* |
| 9: | ConditionalProbabilty (k,Observation$_j$)= Potential(k,Observation$_j$)/Z |
| 10: | **If** (ConditionalProbabilty (i,Observation$_{j-1}$)> ConditionalProbabilty (bestParent (k)) |
| 11: | **then** CorrectedParent(k)=i |
| 12: | **End** |
| 13: | **End** |
| 14: | **End** |
| 16: | **End** |
| | *% Backward Phase:* |
| 17: | #CandidatePaths= #Cells |
| 18: | **For** p=1→ #CandidatePaths *% p is the last cell in the CandidatePaths* |
| 19: | k=p *% k is the current cell in the CandidatePath* |
| 20: | Construct each CandidatePath: |
| 21: | **For** all observations (j →1) |
| 22: | CandidatePath(j)= k; *% add cell K to the path at time step j* |
| 23: | sum(CandidatePath)= sum(CandidatePath)+ ConditionalProbabilty(k,j) |
| 24: | k=BestParent(k,j) *% choose the best parent of k to be the next cell in the path* |
| 25: | **End** |
| 26: | **End** |
| 27: | CorrectedPath= CandidatePath with highest sum of ConditionalProbabilities |

# 7. Experimental Results and Discussion

The algorithm was tested using both simulations and data obtained from actual measurements. Map information for the 4th floor of the DeustoTech building at the University of Deusto (Spain) was extracted from AutoCAD files according to the method described in the previous section and modelled as a grid-based map model. The map matching system is applied to refine the estimated trajectory by avoiding crossing obstacles; the most feasible trajectory is the sequence of positions that violates the fewest constraints (Xiao et al., 2014). Therefore, to evaluate the algorithm, in each case the number of crossed obstacles by the corrected trajectory are counted as a metric to measure of accuracy. In addition, the precision is measured the Cumulative Distribution Function (CDF) of the errors obtained by calculating the Euclidean distance between the actual position in the ground truth and the corresponding corrected position at each time step.

Accuracy and precision some times are used as if they have the same meaning but actually they measure different things. Accuracy measures how approximate the measurements to the truth and is usually represented by the average error, while precision measures the consistency and robustness of the system, and it is usually represented using the CDF of the error. However, for simplicity accuracy and precision are combined and called accuracy (Liu et al., 2007; Shang et al., 2015).

Section 7.1 presents the results of the conducted simulations, while the results obtained using the Actual measurements are presented in Section 7.2

## 7.1 Simulations

A base trajectory (shown in Figure 35) was constructed as the ground truth, with different unmatched trajectories then created randomly as the input (observations) of the map matching algorithm. In real-life applications, the input trajectory is the output of an existing localization system that represents the primary estimations (as will be shown in

Subsection 7.2). After matching the input trajectory to the map using the CRF algorithm, the matched (corrected) trajectory accuracy and precision are calculated.

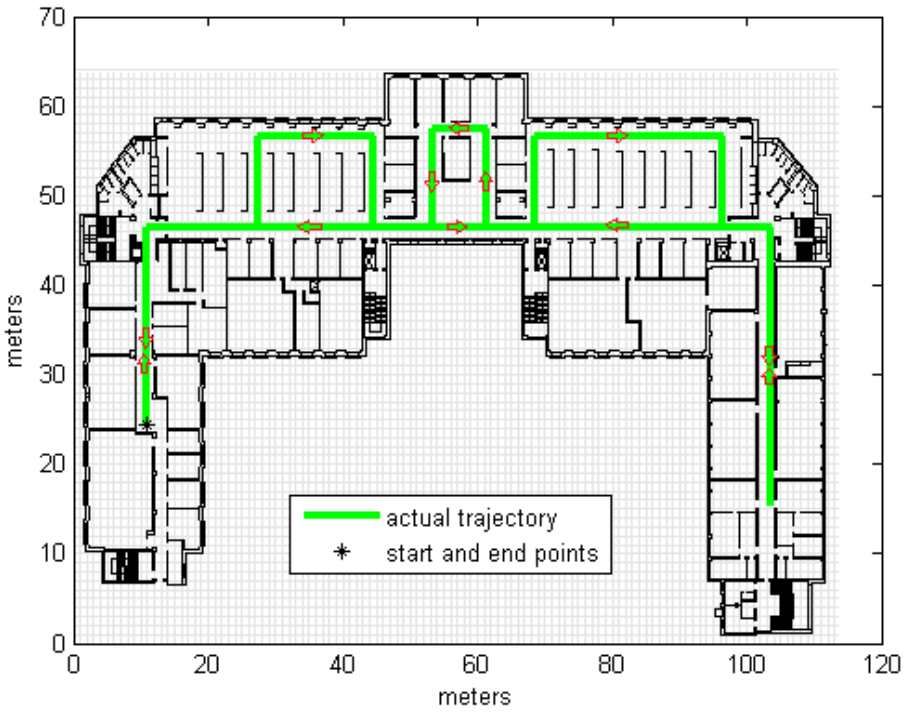



*Figure 35.* Ground truth for simulated trajectories.

To simulate different possible input trajectories with different qualities, different levels of noise were added to the base track by allowing different degrees of deviation (distances) and randomness around the ground truth. A low noise level was used to represent a primary estimation (input) trajectory with low error (mean error of about 1.3 m and standard deviation of about 1 m for noise level 1); a simulated input with a high noise level was used to represent an input trajectory with high error (mean error of about 2.5 m and standard deviation of about 1.8 m for noise level 4). Table 6 shows approximate values of the mean and standard deviations of the four different noise levels obtained by simulating 25 random samples of each noise levels. Examples of different simulated trajectories with different noise levels are shown in Figure 36.

*Table 6.* Means and standard deviations of the different levels of noise used in simulations.

| Noise Level | Mean (meters) | Standard deviation (meters) |
|---|---|---|
| 1 | 1.3 | 1 |
| 2 | 1.6 | 1.1 |
| 3 | 2.0 | 1.3 |
| 4 | 2.5 | 1.8 |


*Figure 36.* Trajectories with different noise levels.

The results of matching different input trajectories with the map are shown in Table 7 where it can be seen that in all the 300 samples the number of obstacles crossed by the map matched trajectory is zero. This is an important proof of the efficiency of the algorithm.

*Table 7.* Simulated trajectory results using the distance feature ($f_{dis}$). Path length = 352 m.

| Cell size (meters) | Noise Level | Number of crossed obstacles | Cumulative error (meters) 50% | Cumulative error (meters) 90% |
|---|---|---|---|---|
| 0.8 | 1 | 0 | 0.9867 | 2.6699 |
| | 2 | 0 | 2.682 | 19.7820 |
| | 3 | 0 | 5.7945 | 32.464 |
| | 4 | 0 | 6.5158 | 32.3861 |
| 1 | 1 | 0 | 1.1574 | 3.078 |
| | 2 | 0 | 1.2463 | 3.8172 |
| | 3 | 0 | 1.4853 | 5.1879 |
| | 4 | 0 | 2.1409 | 11.1061 |
| 1.5 | 1 | 0 | 19.4403 | 70.7003 |
| | 2 | 0 | 16.6941 | 61.6751 |
| | 3 | 0 | 17.7461 | 68.5491 |
| | 4 | 0 | 20.0473 | 71.8129 |

Mathematically the best path is the path with highest some of conditional probabilities as explained in Section 6.3. It is mathematically possible that a path has the highest sum of conditional probabilities while the conditional probability of some cell on that path is zero; one single zero probability would not affect much the whole sum. If at some point the conditional probability of the cell is equal to zero because there is a cross of obstacle it still can be in the path if the sum of all points on the path is still high. But such situation is very rare as the algorithm and features are designed to minimize the chance of crossing obstacles; what prevents crossing obstacles is that normally there are other free cells with higher conditional probabilities that the path to choose instead of occupied cells. And that's why the probability to cross obstacles is extremely low. No cross of obstacles will happen unless it is necessary and there is no other choice; in odd cases like the in the case of the divergence (stuck) problem that happens when the trajectory is trapped in a small area; the path might cross the wall to get out of the trap. To explain that, in some cases of divergence, the distance between the input location

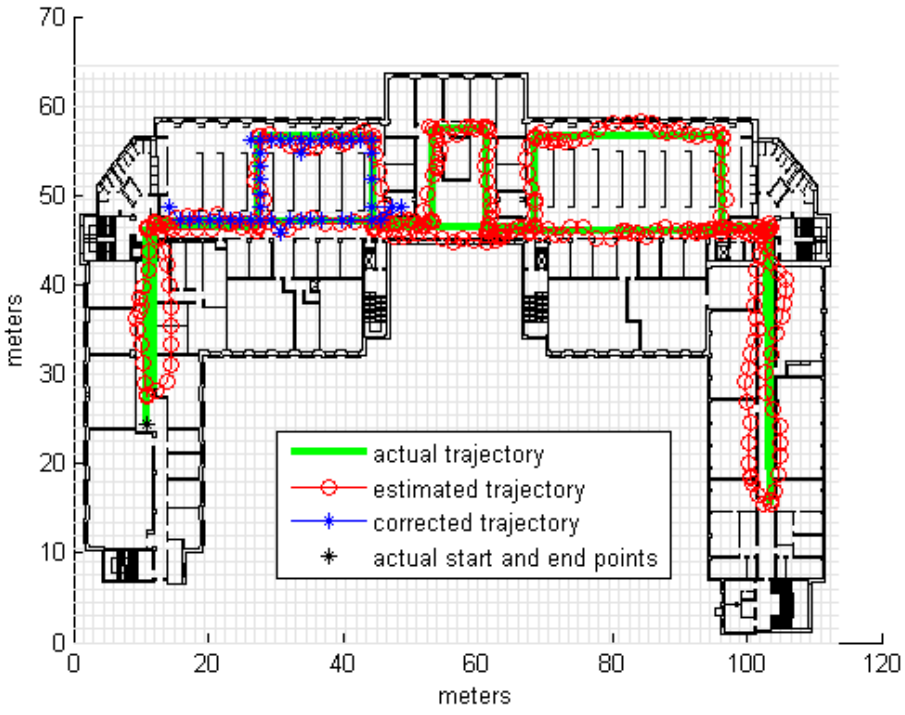
and the current cell grows until the probability (Equation (26)), which depends on the distance, will be of very small value. That means no big difference of probability between free cells and blocked cells in the stuck area; both would have very low score because all are so far in distance from the real place. In this case it is possible that the path choose to cross an obstacle to get out from the trap, as again at the end the path is evaluated with the sum of probability along it not on a single obstacle cross. According to our observations this case of crossing obstacles happens very rarely and does not happen in most of divergence problem cases, but few of them. The divergence problem itself happens rarely as will be shown below.

To study the effect of cell sizes and noise level, Table 7 is constructed with different combinations of parameters. To measure the precision, the cumulative error for 50% and 90% of positions on the trajectory was calculated for different parameter values, in this case cell size and noise level. For each parameter combination, the algorithm was run on 25 random input trajectories and the average cumulative error registered for 50% and 90% of positions.

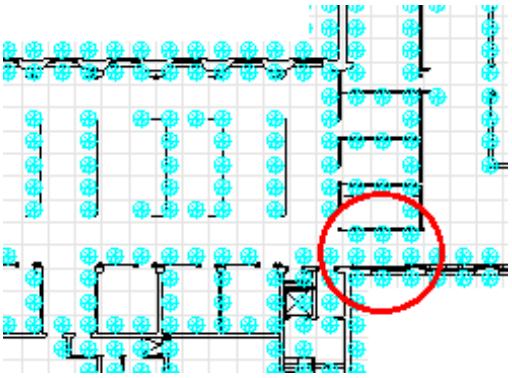
It is clear from Table 7 that a cell size of 1 m provides good results, and even when adding high levels of noise to the input the results still acceptable. At a low noise level (equal to 1), for 50% of the trajectory the average cumulative error is 1.16 m and for 90% the average cumulative error is 3.078 m, based on a total track length of 352m. Using a larger cell size of 1.5m length results in more cells occupied by obstacles and thus blocked to movement; this explains the high level of error observed in the respective results, as illustrated by the example shown in Figure 37 (a). Figure 37 (b) illustrates how the corridor is blocked because it contains cells occupied by side walls. Besides being of high error it can also be noticed that the results of this cell size, it is also totally random in relation with the noise level, higher error of low noise level is just due to the random input samples, using different samples would result in different numbers, that explains the available results of 70.7003 m using noise level 1 while 61.6751 m error is obtained using noise level 2, noise level in this specific situation is irrelevant and does not have any significant impact.

Although the use of a cell size of 0.8 m would be expected to lead to more precise results, the obtained results show that this is correct only for low noise input; at higher noise levels the error was high, likely due to what is called here the stuck or divergence

problem. This particular problem occurs when the corrected trajectory enters a room or small area and cannot leave; an example of such a situation is shown in Figure 38. Although with smaller cells it is more likely that certain cells will contain no obstacles, some of these empty cells could form entrances to small areas, allowing the trajectory to pass through them and become trapped. The stuck problem recorded in 0.8 m cell size simulations resulted in high error values in about 16% of samples, with the remainder achieving good results (average cumulative error of 4.2522m for 90% of observations at a noise level of 2). It should be noted that the values shown in Table 7 indicate the average cumulative errors in all samples, including those experiencing the stuck problem, which explains the high value of some of these averages. Figure 39 displays examples of the results obtained for different cell sizes and different noise levels, these examples show that the algorithm can refine trajectories that noises levels from 1 to 4 and using cells sizes of 1 m and 0.8 m.




(a) Trajectories, cell size = 1.5 m, noise level = 2



(b) Cell occupation by obstacles

*Figure 37.* Large cells result in more cells being occupied by obstacles and blocked to movement, and corridors can be blocked. Figure (a) shows a map matched trajectory for a cell size equal to 1.5 m; occupied cells are marked with blue circles (b).

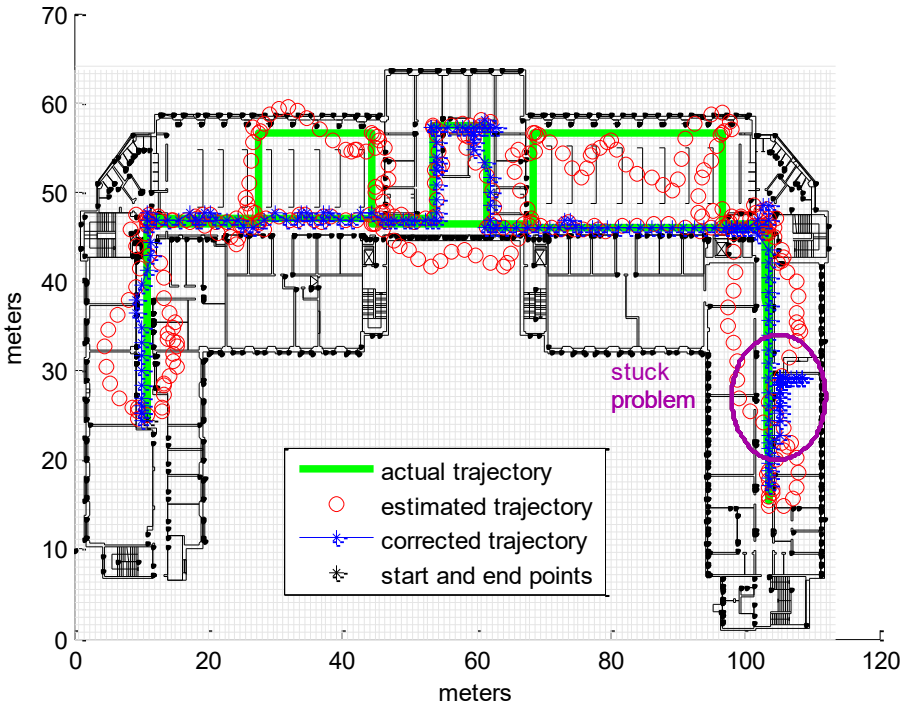
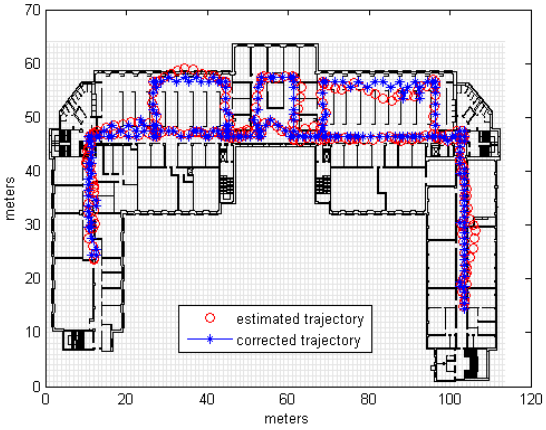
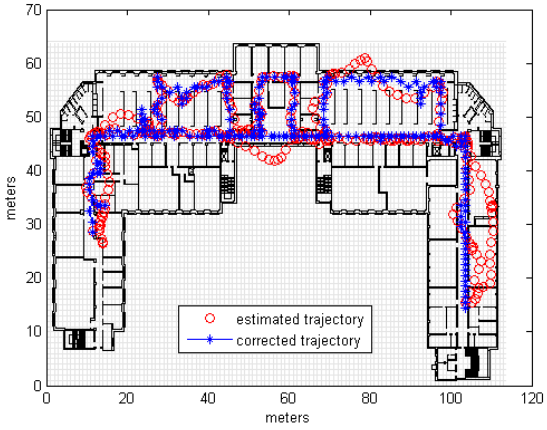


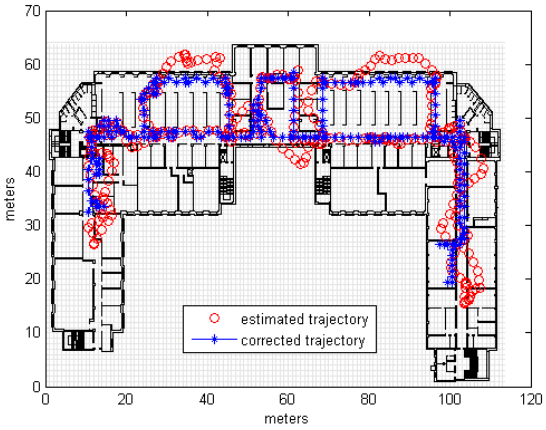

*Figure 38.* Stuck problem scenario, cell size = 0.8 m, noise level = 4.



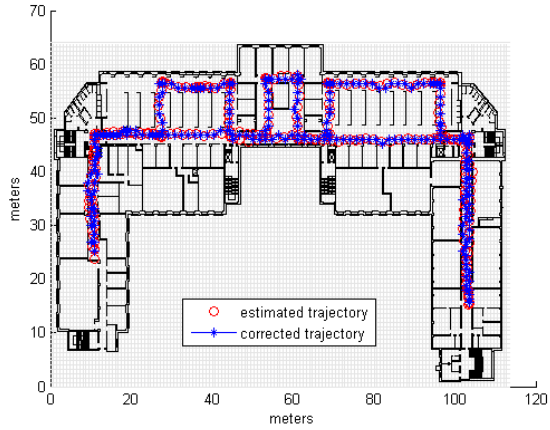

(a) Cell size = 1 m, noise level = 2,
CDF(50%) = 1.239 m, CDF(90%) = 3.470 m



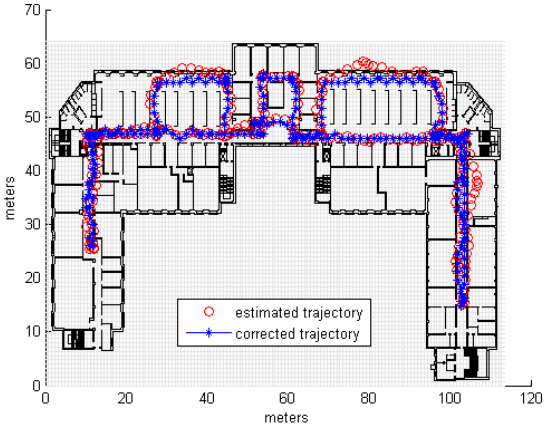

(a) Cell size = 1 m, noise level = 3,
CDF(50%) = 1.401 m, CDF(90%) = 4.161 m



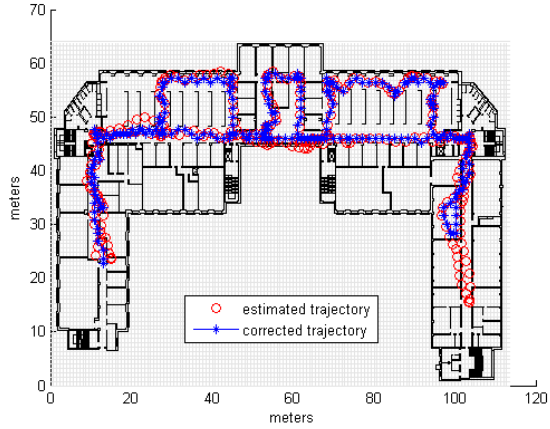

(c) Cell size = 1 m, noise level = 4,
CDF(50%) = 1.444 m, CDF(90%) = 4.679 m




(d) Cell size = 0.8 m, noise level = 1,
CDF(50%) = 0.784 m, CDF(90%) = 2.016 m




(e) Cell size = 0.8 m, noise level = 2,
CDF(50%) = 0.944 m, CDF(90%) = 2.201 m




(f) Cell size = 0.8 m, noise level = 3,
CDF(50%) = 1.032 m, CDF(90%) = 6.780 m

*Figure 39.* Ground truth, estimated and corrected trajectories obtained using different cell sizes and noise levels.

A moderate walking speed is thought to be around 1.67 steps/second (1.67 Hz) ("Moderate Intensity Walking", 2009), with average step length 0.7 m for women and 0.78 m for men ("Determine your Stride"), resulting in a movement speed of around 1.3 m/second. Assuming a frequency of input of 1 estimation/second (1 Hz), i.e. a time step equal to 1 second, the subject would move 1.3 m every time step, 2.6 m in two steps, etc.

As the proposed map model employs discrete values for movement (cell units), allowing only one cell transition, for example, in each time step (buffer size equal to 1 cell) would mean that the system would not be able to capture the movement of a pedestrian walking at moderate speed; it is therefore more suitable to have flexibility of movement by allowing 0, 1, or 2 cell moves in each time step by allowing transition to neighbour cells and to neighbours of neighbours (buffer size equal to 2), which is necessary in order to obtain a correct trajectory assuming a cell size of 1 m. If the input frequency was 2 Hz (2 estimations/second), this would mean an estimation every half-second time step. During this time period a pedestrian moving at moderate speed moves around 0.65 m in each time step, a distance smaller than the cell size (1 m). In this case if only one cell transition was allowed there should not be a problem, while allowing more possibilities (one or two cells) should also work. In some systems there is no fixed frequency for estimations, with the location registered at each step whatever the speed. In the present case, assuming a moderate walking speed the average frequency will be around 1.67 Hz, meaning that a buffer size of one cell would be suitable. However, in all cases a buffer size of two cells is still more flexible in capturing any speed variations and in dealing with other types of movement such as running.

The results shown in Table 7 are those for a buffer size of 2 cells and assuming that the estimations of the primary localization system are registered at a frequency of 1 Hz. To experimentally test the effect of buffer size and estimation frequency on the proposed algorithm, further simulations were carried out using a cell size equal to 1 m and noise level equal to 1; the results of these simulations are shown in Table 8. It is clear from this table that the results were worse when using a small buffer size (one cell) and 1 Hz

frequency, with 33.1054 m error for 90% of data compared to the 3.078 m recorded for a buffer size of two cells and the same 1 Hz frequency. This discrepancy is due to the inability of the trajectory to reach the correct destination using small steps, as the estimated trajectory was moving faster. Furthermore, the adoption of a 2 Hz measurement frequency achieved better results, even when using a small buffer. Nevertheless, a higher buffer size always provides more flexibility.

*Table 8.* Simulated trajectory results for different buffer sizes and measurement frequencies. Cell size = 1 m , noise level = 1, Path length = 352 m.

| Estimation Frequency | Buffer size (cells) | Cumulative error (meters) 50% | 90% |
|---|---|---|---|
| 1 Hz | 1 | 12.4931 | 33.1054 |
| 2 Hz | 1 | 1.1348 | 2.4989 |
| 1 Hz | 2 | 1.1574 | 3.078 |
| 2 Hz | 2 | 1.3639 | 3.0044 |

The analysis above does not mean that the system requires a specific frequency for the input to work, the purpose of this analysis is to justify the choice of the step size and the buffer size used in our model; they have been chosen in a way that enable the map matching algorithm to suit the pedestrian movement. An input of a higher frequency can also be compatible to the system using the same cell size and buffer size as the algorithm allows staying in the same cell for consecutive time steps; however, in this case, if more accuracy is required, a smaller cell sizes can be used. If the input frequency is very low, much less than 1 Hz that means bigger cell size or buffer size is needed.

Some pedestrian positioning systems estimates the trajectory on time bases, i.e. fixed frequency, others (the SHS systems) estimates the position on the step or stride. The SHS system is also suitable for our system as it is designed for pedestrians. The

localization system that has been used in the actual experiments that is discussed in subsection 7.2 is SHS system.

The proposed map matching system is tested with pedestrians in mind; however, this system can be used with any object moving with the pedestrian speed, and the same analysis applies for the input frequency. For different speeds or measurement frequencies the cell size and buffer size should be adapted to achieve the required accuracy. Even for objects moving in other speeds, it is not necessary to know the input frequency, it is enough to adapt the cell size to the average distance between consecutive coordinates which can be calculated from the input trajectory.

## 7.2 Actual Measurements

Testing was carried out on the 4$^{th}$ floor of the DeustoTech building at the University of Deusto (Spain). Primary results were estimated via the iHDE system developed by Díez et al. (2016), which is a SHS PDR system that uses a wrist-worn inertial measurement unit; the developed map matching algorithm then applied offline. Estimation frequency was not constant but depended on the step, as the employed PDR system calculated the location at each step. For the map model a cell size of 1 m and a buffer size of two cells were used. Given the ground truth is shown in Figure 40, the estimated results of the primary localization system and the results corrected by CRF map matching for one round walk of the DeustoTech area are shown in Figure 41. For the same ground truth, Figure 42 (a) displays the estimated trajectory for two rounds, with the corrected trajectory for this estimated trajectory shown in Figure 42 (b).

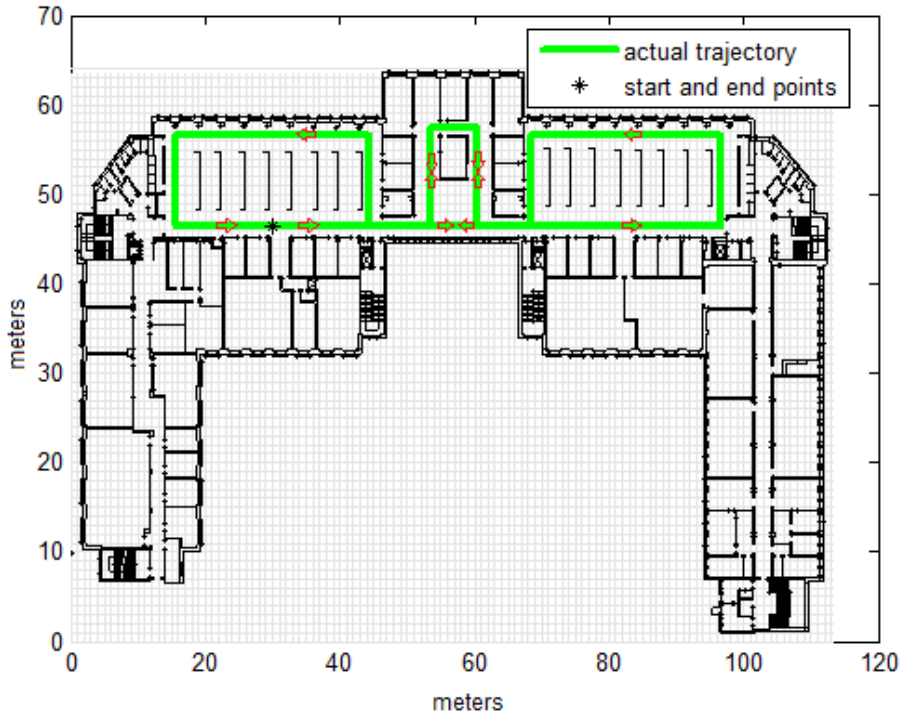



*Figure 40.* Ground truth for actual measurements.

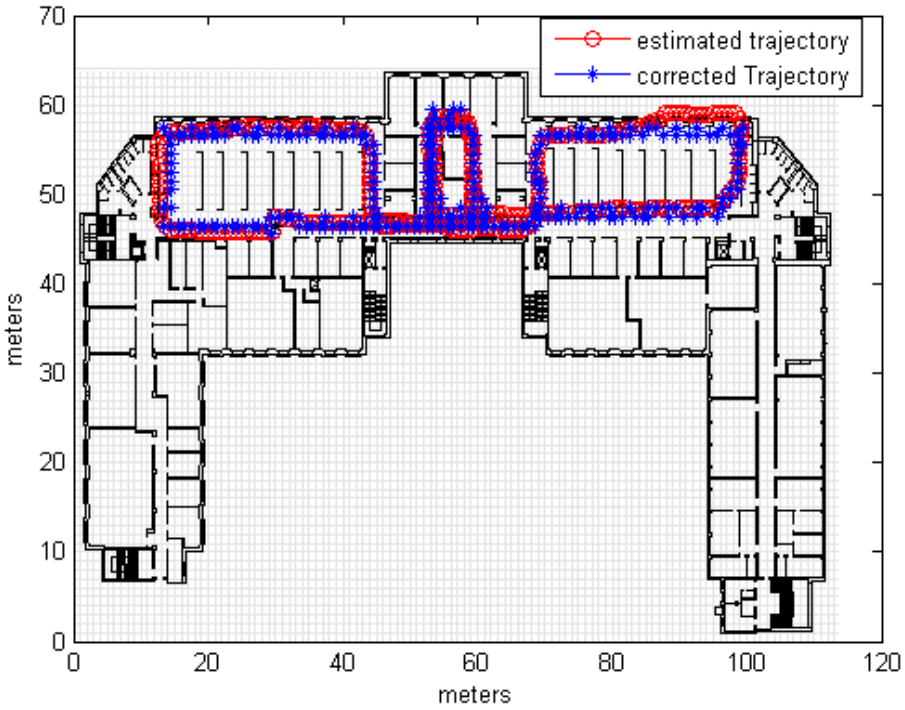
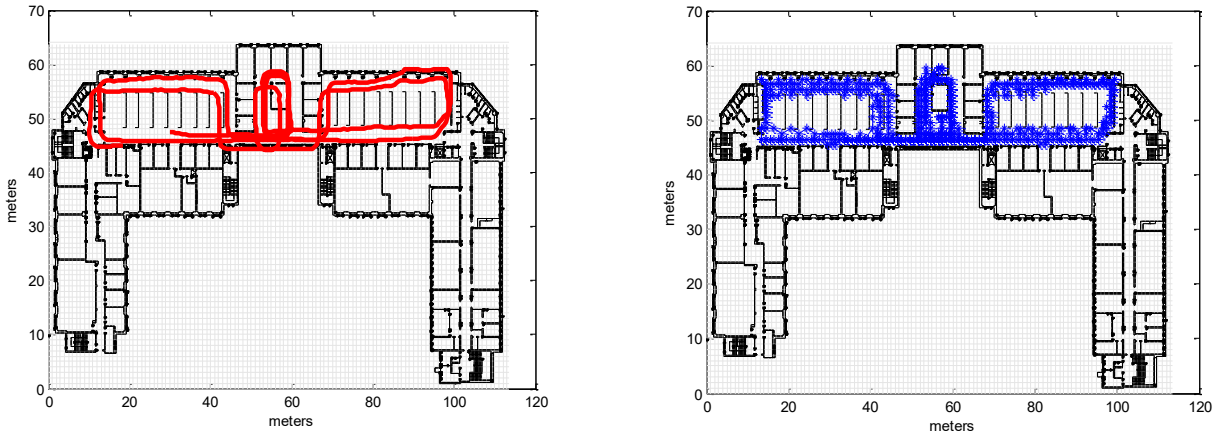



*Figure 41.* Estimated and corrected trajectories obtained from actual measurements using iHDE and trajectory corrected using CRF for one round walk.




(a) Estimated trajectory for two rounds

(b) Corrected trajectory for two rounds

*Figure 42.* Estimated and corrected trajectories obtained from actual measurements using iHDE and trajectory corrected using CRF for two rounds walk.

Analysis of Figure 42 reveals that although the primary estimated trajectory crosses a number of walls and obstacles, this was corrected successfully using the map matching algorithm and zero obstacles was crossed by the map matched trajectory, see Table 9. As a consequence of matching the trajectory with the map the accuracy was also improved; even though the accuracy of the iHDE trajectory was already good, the map matching using CRF has further improved the accuracy and the cumulative error was decrease as shown in Table 9. Before applying the map matching, using only the primary localization system (iHDE), the average cumulative error was 1.2861 m and for 50% of the trajectory 2.6858 m for 90% of the trajectory. The cumulative error was reduced by applying the CRF map matching algorithm to be 1.0634 m for 50% of the trajectory, and 2.2316 m for 90% of the trajectory; which means improvement of 17.5% and 19.8% respectively.

*Table 9.* Real measurement results using the distance feature ($f_{dis}$), cell size = 1 m. Path length = 347 m.

| Algorithm | Number of crossed obstacles | Cumulative error (meters) | |
|---|---|---|---|
| | | 50% | 90% |
| iHDE | 15 | 1.241 | 2.688 |
| iHDE+CRF | 0 | 1.056 | 2.231 |

## 7.3 Problems and Solutions

From the results that were discussed in Chapter 7, the map matching algorithm has two main problems; the first is the divergence or stuck problem that occurs when the output trajectory stuck in a small area, and the second is that the output trajectories are not always smooth and straight which does not reflect the natural way of walking. In this chapter solutions to these problems are proposed.

### 7.3.1 The Divergence Problem

The divergence (stuck) problem occurs when the corrected trajectory is trapped in a small area, a room for example (see Figure 43). To solve this problem a divergence correction process is proposed. After obtaining the corrected trajectory it is checked for divergence, if a divergence problem is found in the trajectory, the time step in which the divergence started is determined; this time step will call it the divergence point, and the CRF algorithm backward phase is redone for all time steps starting from the divergence point to the last time step. The new obtained trajectory is used to replace the part of the corrected trajectory after the divergence point.

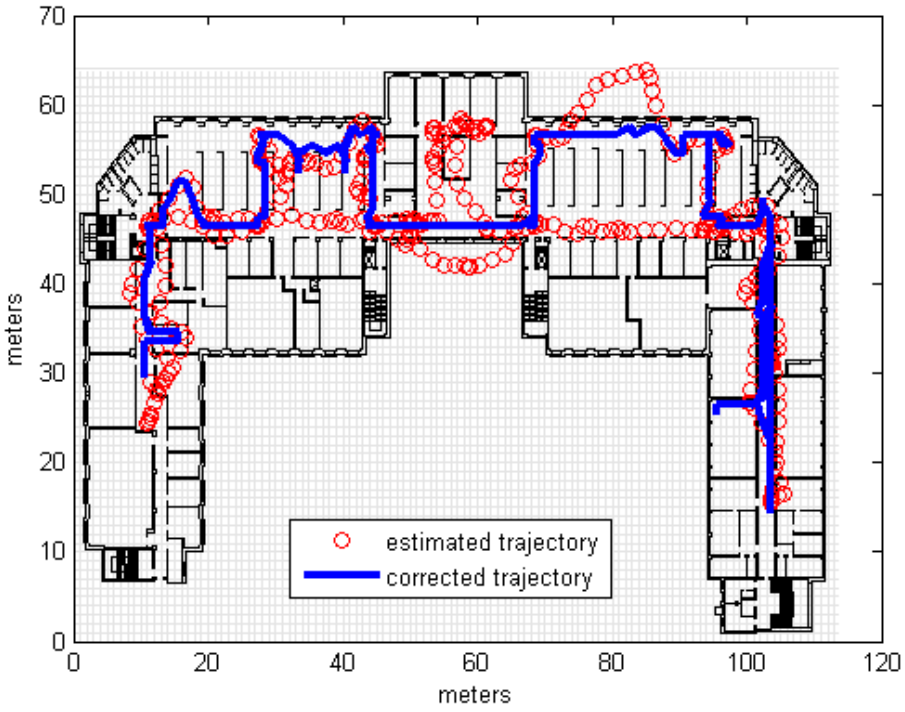



*Figure 43.* An example of a corrected trajectory that has a divergence problem.

A simple method to detect the divergence has been developed; the distances between the corrected location and the estimated location for each time step are calculates, if the distance exceeded a pre-determined threshold; we go back in the trajectory where the distance between the estimated and corrected location is expected to be decreasing. The point at which the distance stops decreasing as we are going back is chosen to be the divergence point (see Figure 44).

According to the noise levels we have applied it is possible to have a distance between the input and the output around 5 or 6 m meters and sometimes up to 10 m. The divergence occurs when the distance much exceeds that, so the threshold must be high enough to avoid the expected input noise, and the higher the better as when the divergence happens it is usually keep diverging until long distances, but without exceeding the building dimensions. In the building plan used in this research, a threshold of 30 m is logical. Small threshold should be avoided to avoid the cases where the distance is just within the normal noise. If the divergence detection algorithm leads to a divergence point more than 6 m which is normal distance as stated before (this will be a second threshold to represent minimum accepted distance difference), the

algorithm should continue tracing back the trajectory until the distance between the input and the output is less than 6 m.

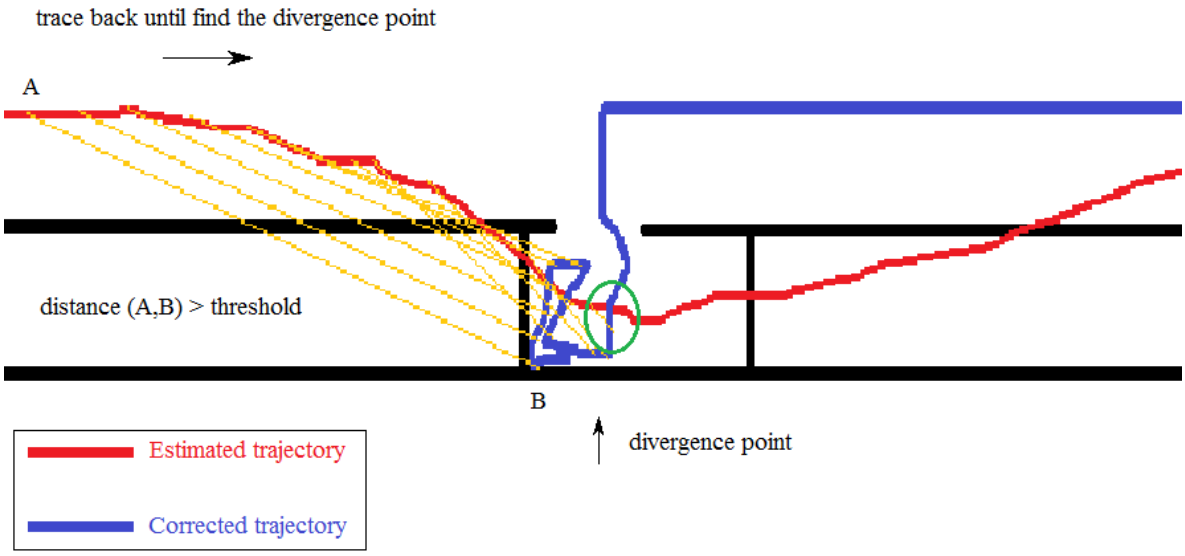



*Figure 44.* Detecting the divergence.

The divergence correction algorithm (see Algorithm 2) is iterative, if after solving the divergence in the original corrected trajectory a new divergence is found, the process will be repeated until no there is no divergence. After applying the algorithm to the example shown in Figure 43 that has a divergence problem, the trajectory was corrected as shown in Figure 45; the first part of the trajectory, before the divergence point, shown in Figure 46; the CRF algorithm was applied again to the part of the trajectory after the divergence point, this part is shown in Figure 47.

*Algorithm 2.* Solving divergence problem.

```
1:  Set threshold = 30, minDistance = 6
2:  For all time steps
3:      Calculate the distance between estimated location and corrected location (DIS)
4:  End
5   Create a Boolean vector (DecreaseVector) to store flags to indicate decrease of DIS between
    consecutive steps
6:  Initialize DecreaseVector to zeros
7:  If the distance DIS of the last time step > threshold
8:      For all time steps (last time step to  the first time step)
9:          If DIS of the previous time step is less than the current
10:             step or is exceeding the MinDistance
11:             Set DecreaseVector(time step) to 1
12:         End
13:     End
14: End
15: Divergence point = the location of the last zero in the Decreasing vector
16: Repeat the backward phase from the last time step to the divergence point
17: Replace the part of the corrected trajectory after the divergence point with the new obtained
    trajectory
18: Check the final trajectory for divergence
19: Repeat the algorithm until no divergence exists
```

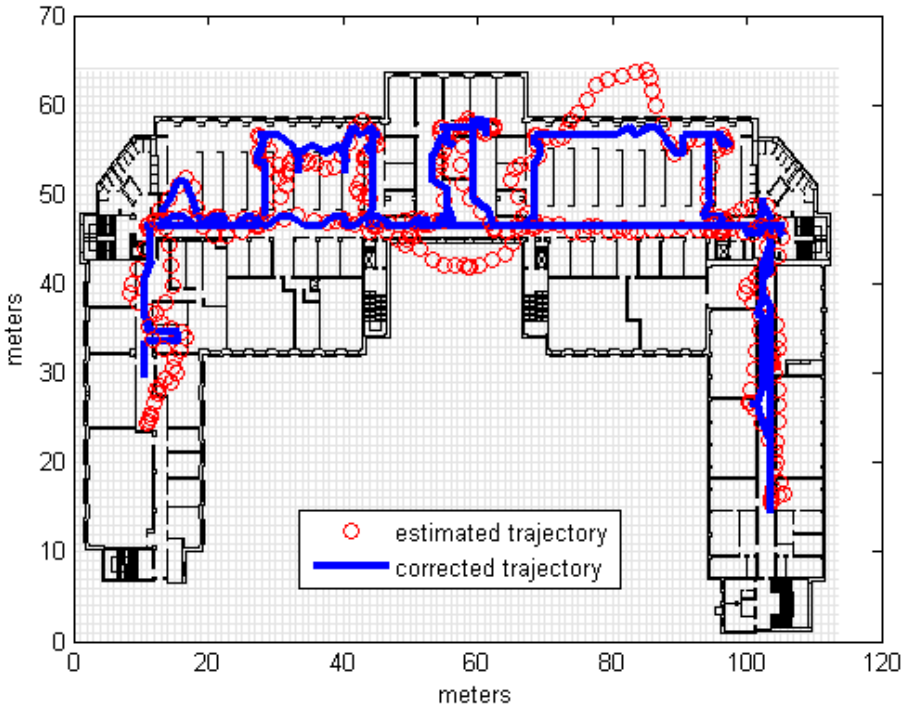



*Figure 45.* The full path after fixing the divergence.

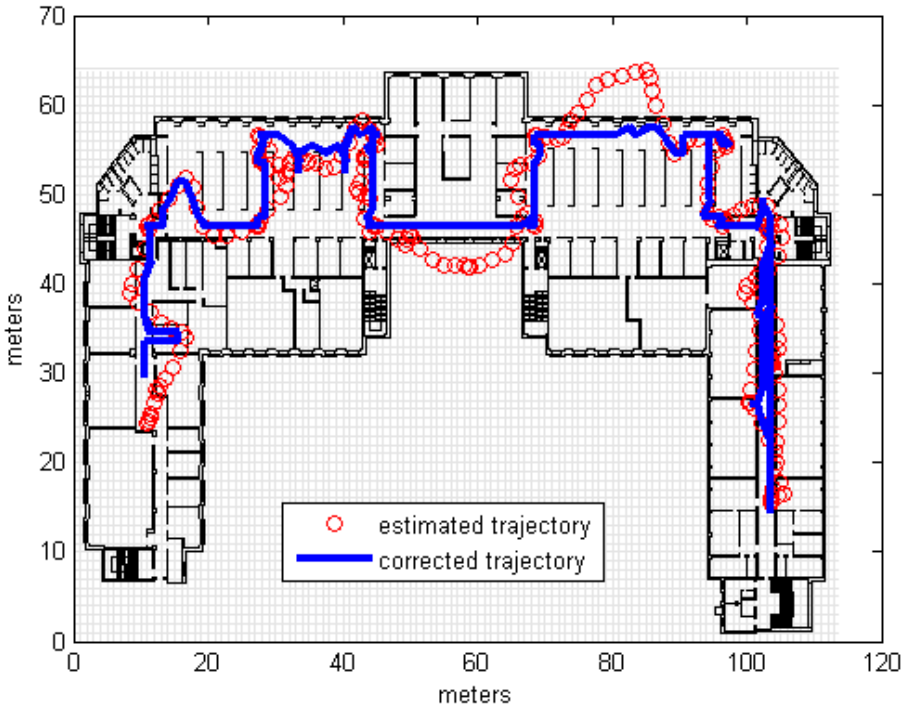



*Figure 46.* The first part of the path before the divergence point.

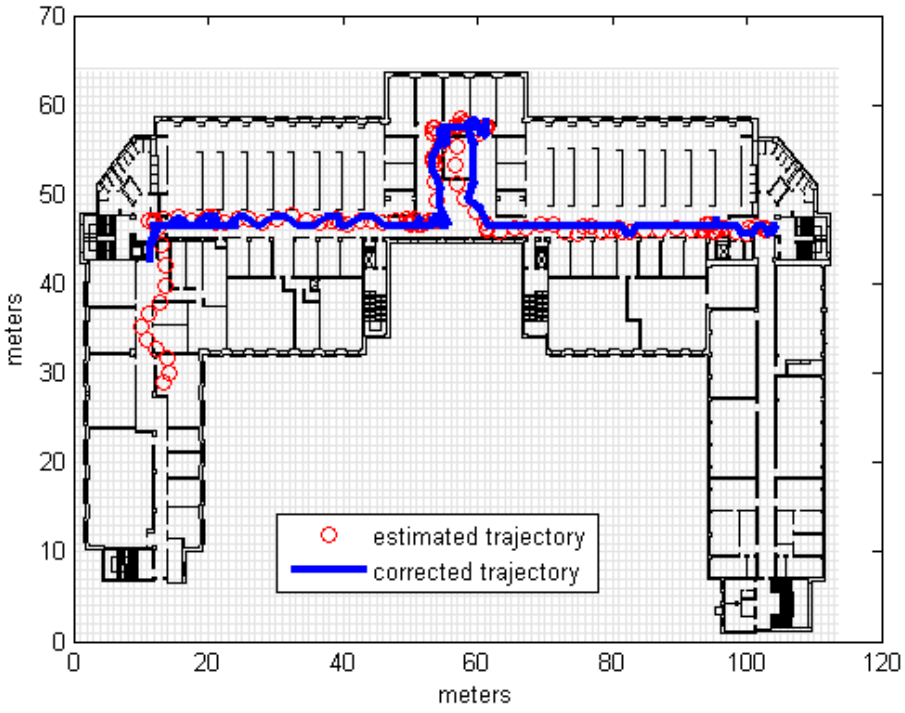



*Figure 47.* The second part of the path after the divergence point.

### 7.3.2 Smoothing TheTrajectory Using Behavioural Information

As have been noticed from the results, sometimes the trajectory corrected using the proposed CRF map matching algorithm has non-smooth part, where the trajectory sections are not straight. An example of such case is the corrected trajectory corrected is shown in Figure 48 given the ground truth shown in Figure 40. As we see the output corrected trajectory is not smooth and we can see the non-smooth zigzag sections, which is not the way how people walk, as people tend to walk in straight lines and not as a zigzag.

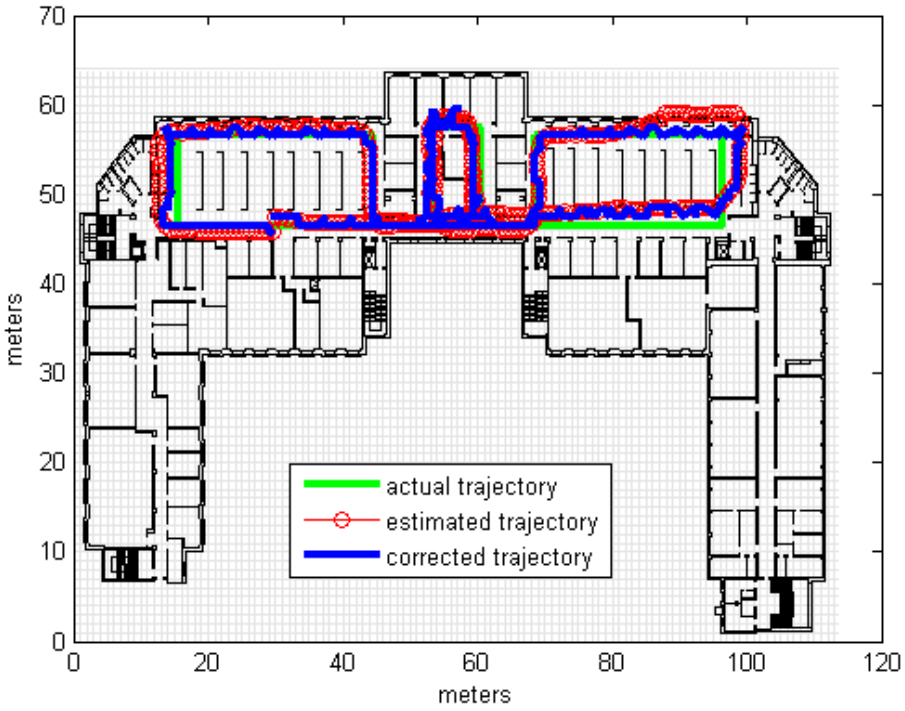



*Figure 48.* Ground truth, estimated and corrected trajectories obtained using one feature function ($f_{dis}$).

A zoom of one of the non-smooth sections of the trajectory in Figure 48 is shown in the Figure 49. This figure can help in explaining why the zigzag problem occurs. The CRF algorithm used so far uses only one feature which is the distance between the input location and the candidate cell to be on the output trajectory; in addition to the transition possibility to the cell which depends mainly on the existence or absence of obstacles and the neighborhood between consecutive cells on the trajectory, i.e. the continuity of

the trajectory. In brief it can be said that the algorithm try to keep close to the input while at the same time avoiding obstacles. In the case shown in Figure 49 we can see a line of cells alternate of being free or having obstacles; it can be seen how the output coordinate tend to be equal to the input coordinate in the absence obstacles, while it has to avoid obstacle when exist, and return back to follow the input trajectory when there is no obstacles and so on; which results in the non-smooth trajectory and the zigzag sections when there is alternation of free cells and cells occupied with obstacles.

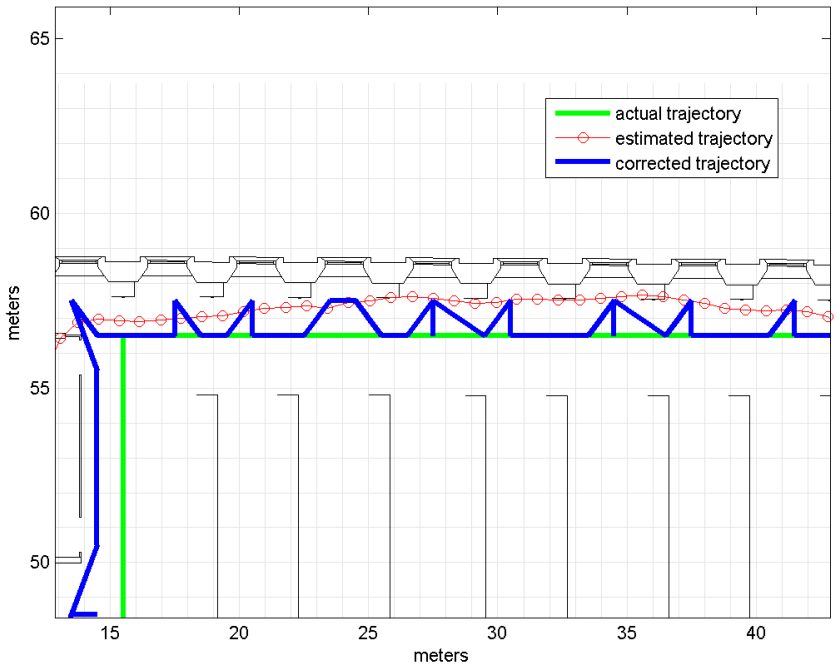



*Figure 49.* The zigzag problem.

To avoid the zigzag problem more features should be included in the algorithm, but first more map information other than the existing or absence of obstacles in the cell should be considered. So far the available map information used in the algorithm was whether the cell is occupied by some obstacle {value = 0} or it is free of obstacles {value = 1}. Now more semantic information of the map will be used; one more semantic layer will be added to the map model to deal with the behaviour of people moving indoors. People do not just walk in free areas and avoid obstacles, they move depending on the structure of the building. When they move from one place to another they walk in the corridors when they exist, they tend to walk in straight lines (Jiménez

et al., 2012), and normally they walk in the center area of the corridor and not beside the walls. If this behaviour is taken to consideration and employed in the system (the map model and the map matching algorithm), more feasible and smooth trajectories are expected.

But how to know the areas or lines where the people tend to walk more? In an area of corridors the common (or dominant) walking lines can be easily determined by eye, and can be added to the map model; or it can be done by developing an algorithm to capture the internal structure and entities of the map such as corridors from the CAD files and then determining the centers of these corridors. For open wide areas it is more appropriate to use behavioural maps.

Information about pedestrian behaviour can be obtained by observations. In the area of architecture design, behavioural maps have been proposed as a mean of place planning and design. The behavioural maps record the behaviour of people and allow determining how people use the space by recording the behaviours or tracking their movement (Bahillo, Marušić, G., & Perallos, 2015). "By associating a behaviour with a certain environment it is then possible to both ask questions and draw conclusions about the behaviour and its relationship to a place" (Bechtel et al., 1987: 23 as cited by Marušić and Marušic (2012)). The behavioural information can be a good source of information to help in the localization process by improving the map model; in this research it has been used to enhance the results through smoothing the output trajectories.

The behavioural map is a tool to record behaviour of people. The behavioural maps can be constructed using the classical way of repeated observation (by eye or by camera) or using tracking technology such as GPS for outside environments (Marušić & Marušic, 2012) or indoor positioning techniques for indoor environments. These maps help determining the common lines of walking; examples of behavioural maps are shown in Figure 50. As mentioned before, the experiments of this research were carried out in DeustoTech laboratory which is an area of corridors; Figure 51 shows the common lines of walking in DeustoTech. Here a new semantic layer that contains the lines that people use more while walking is added to the map model; after that the cells that are located on these lines are determined and flagged.

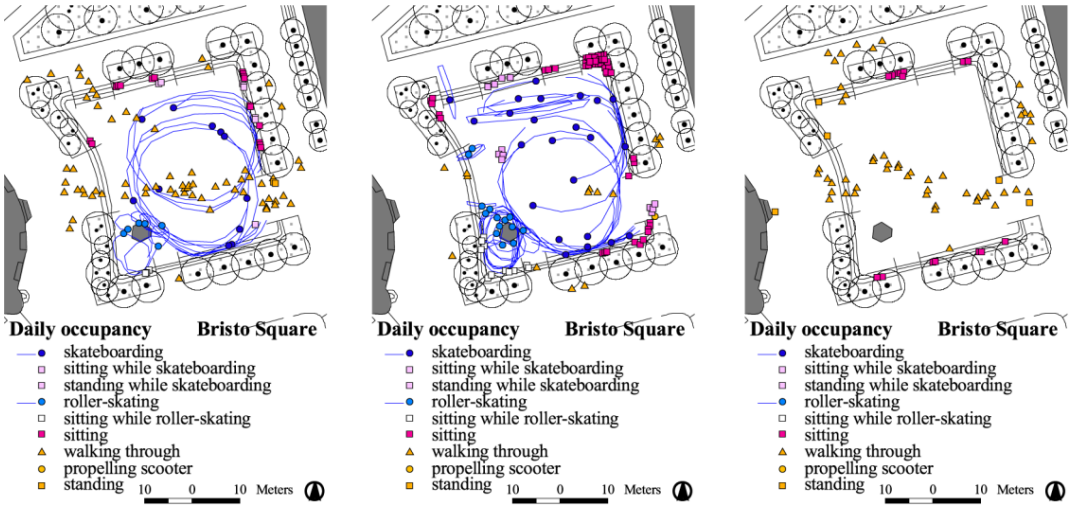



*Figure 50.* Maps of typical occupancies in Bristo Square, Edinburgh. "*Note.* From (Goličnik, 2005 as cited by (Marušić & Marušic, 2012))".

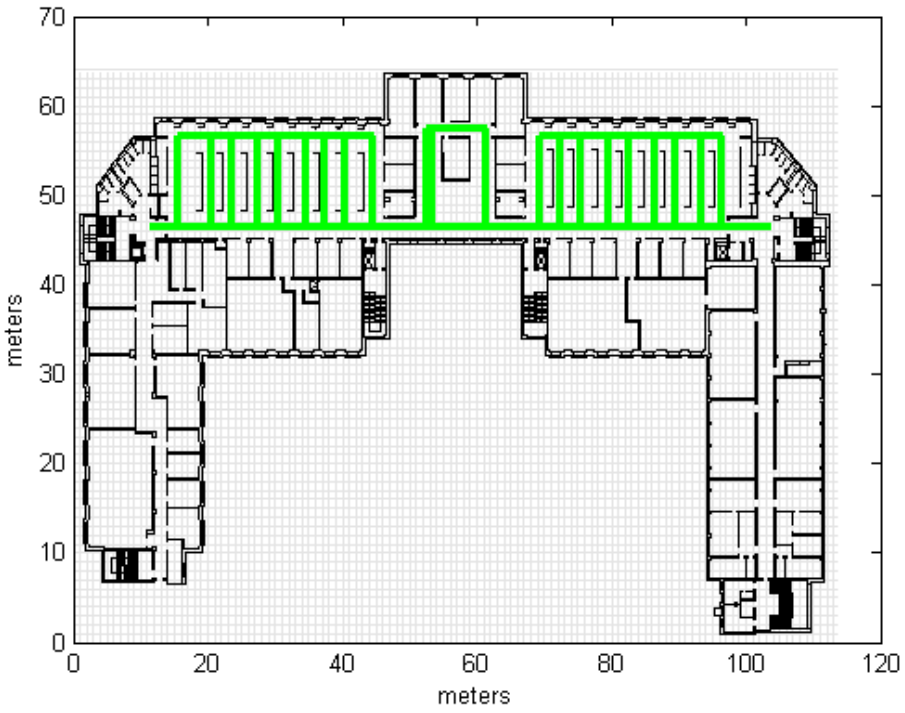



*Figure 51.* Common lines of walking, the green lines.

As a characteristic of the cell we had a value of 0 or 1 that indicates if the cell is free of obstacles or not. Now a new characteristic which indicates if that cell lies on the common lines of walking is added, cells on the common walking lines will be called the behavioural cells.

*Cell → isFree {0, 1} → isBehavioural {0, 1}*

The next step is to use this map information in the map matching process. And that can be done by modifying the CRF map matching algorithm by adding a new feature function that deals with the new feature of behavioural cells.

The proposed formula to grasp this feature is as follows:

$$f_{behavioral} = T(y_{j-1}, y_j) \cdot isBehavioral(y_j) \cdot withinRange(x_j, y_j) \qquad (30)$$

where *T* transition function, $isBehavioral\ (y_j)$ is 0 or 1 depending on whether the cell lies on the behavioural lines or not, and $withinRange\ (x_j, y_j)$ has also a 0 or 1 value depending on the distance between the cell and input observed location. And again the minimum distance equals cell-size/2. For distances more than the determined range, $withinRange\ (x_j, y_j)$will be 0 and the $f_{behavioral}$ value will be 0, because if the input coordinate is far from the input (there is no meaning to give more probability to the cell. The probability will then be the normalized exponential of the sum of the features each multiplied by its weight λ using equation (26).

$$\mathrm{P}(\vec{x}|\vec{y}) = \left(\frac{1}{Z(\vec{x})}\right) \cdot \exp\left(\sum_{j=1}^{n} \lambda_{dis} f_{dis} + \lambda_{behavioural} f_{behavioural}\right) \qquad (31)$$

The pseudo code of the CRF algorithm after adding the behavioural feature can be found below (Algorithm 3).

*Algorithm 3*. CRF algorithm for map matching after adding the behavioural feature.

```
1:  Input : Observation= a vector of coordinates of the input estimated trajectory
2:  Output: CorrectedPath = a vector of coordinates of the output corrected trajectory
    % Forward Phase:
3:  For each observation (Observation_j)   %Observation _j = coordinates of the input at step j
4:    For all cells (i)
5:      For all neighbour cells of i (k)
6:        f_dis= T(i,k) / Distance (k,Observation_j)        %T(i,k): transition possibility from
                                                            cell i to  cell k {0,1}
7:        f_behavioural= T(i,k)*isBehavioural(k)*withinRange(i,k)
8:        Potential(k,Observation_j)= exp(λ_dis*f_dis +λ_behavioural*f_behavioural);
9:        Z =sum(Potential ( : , j ) )                    %  normalization factor
10:       ConditionalProbabilty (k,Observation_j)= Potential(k,Observation_j)/Z
11:      If (ConditionalProbabilty (i,Observation_j-1)> ConditionalProbabilty (bestParent (k))
12:           then CorrectedParent(k)=i
13:       End
14:     End
16:   End
17: End
    % Backward Phase:
18: #CandidatePaths= #Cells
19: For p=1→ #CandidatePaths                % p is the last cell in the CandidatePaths
20:   k=p                                    % k is the current cell in the CandidatePath
21:   Construct each CandidatePath:
22:   For all observations (j →1)
23:     CandidatePath(j)= k;                % add cell K to the path at time step j
24:     sum(CandidatePath)= sum(CandidatePath)+ ConditionalProbabilty(k,j)
25:     k=BestParent(k,j)        % choose the best parent of k to be the next cell in the path
26:   End
27: End
28: CorrectedPath= CandidatePath with highest sum of ConditionalProbabilities
```

The matching algorithm was applied using equal weights for the features ($\lambda_{dis} = \lambda_{behavioral} = 1$). Figure 52 shows an example of the results after the new feature function was added to the algorithm; we can see that the corrected trajectory is now smoother and more feasible than when only the distance feature was used. In Figure 53 we can see a zoom of one of the sections which where non-smooth before applying the new feature function and now it is smooth. While the behavioural feature is only meant to smooth the output trajectory, it is expected that it also will have a small positive effect on the accuracy. The accuracy and precision results of and different distance ranges are shown in Table 10. Using other weight combinations gave very close results

without a significant effect of the weights on the results. For the best tuning of the weights, training of the model should be performed.

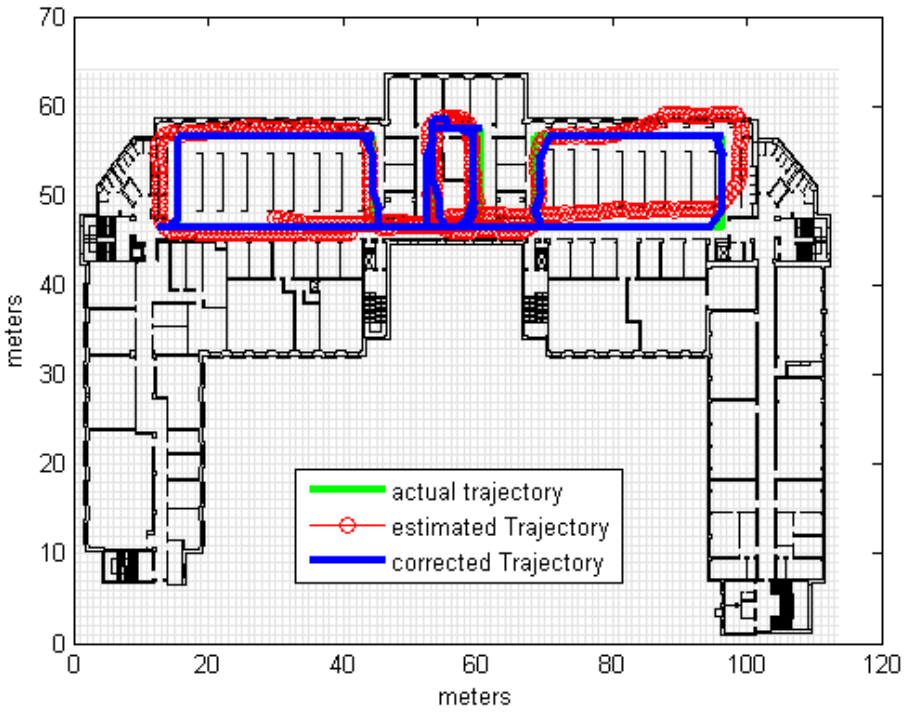



*Figure 52.* Ground truth, estimated and corrected trajectories obtained using two feature functions ($\boldsymbol{f_{dis}}$ ***and*** $\boldsymbol{f_{behavioral}}$).

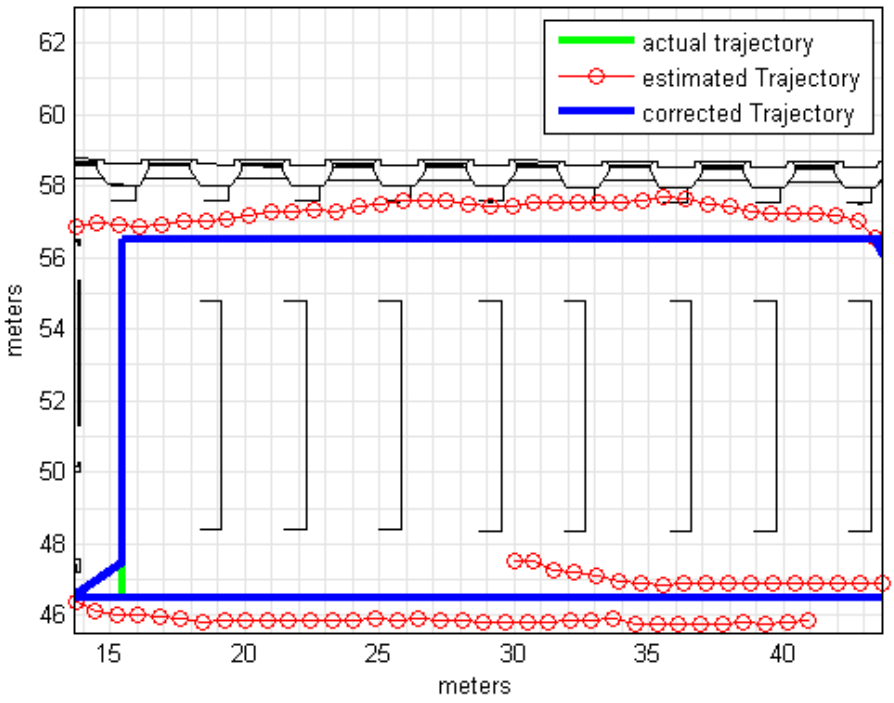



*Figure 53.* Smooth sections after applying the behavioural feature.

*Table 10.* Accuracy comparison for different ranges in the behavioural feature using equal weights ($\lambda_{dis} = \lambda_{behavioral} = 1$).

| Trajectory | Cumulative error (meters) | | | Average error |
|---|---|---|---|---|
| | 50% | 75% | 90% | (meters) |
| Input (iHDE) | 1.241 | 1.845 | 2.688 | 1.398 |
| Using $F_{dis}$ only | 1.056 | 1.673 | 2.321 | 1.255 |
| Output(Range = 2 m) | 0.946 | 1.438 | 2.247 | 1.065 |
| Output(Range = 3 m) | 0.932 | 1.499 | 2.453 | 1.131 |
| Output(Range = 4 m) | 0.985 | 1.866 | 2.999 | 1.319 |
| Output(Range = 5 m) | 0.992 | 2.041 | 3.901 | 1.539 |

Table 10 shows the effect of the used range on the results, and helps comparing the accuracy and precision after adding the behavioural feature with the accuracy of the input and with the accuracy obtained using the distance feature only. The table shows

the cumulative for 50%, 75% and 90% of positions on the trajectory in addition to the average error. The results show that for 50% of the positions the cumulative error is less than the cumulative error of the input. That means that adding the behavioural feature, in addition to smoothing, it improves the accuracy compared to the input and also compared to the output of the map matching algorithm before adding the behavioural feature. But for 75% and 90% of the positions, the accuracy start becoming worse for ranges 4m and 5m for some metrics, which can be explained by the possibility for inputs to be located on different behavioural walking lines or cells which might lead to a wrong position can be chosen especially in an area dense of corridors; for example when we the allowed range is 5 m within this range there might be different behavioural cells that can be assigned to that input, and it might be assigned to a cell that is far from the correct position even exist on the walking lines, while using a range of 3 m, less options are available and the smoothing is using close cells.

These results show that the new algorithm, after adding the behavioural feature is able to smooth the trajectory in addition to a small enhancement of the accuracy or at least maintain close same accuracy when using range less than 4m. For ranges greater than 4 m, the smoothing is still achievable, and more sections can be smoothed as more farther input positions are also smoothed but on the price of accuracy.

Figure 54 shows the example shown in Figure 43 after solving both the divergence problem and the zigzag problem.

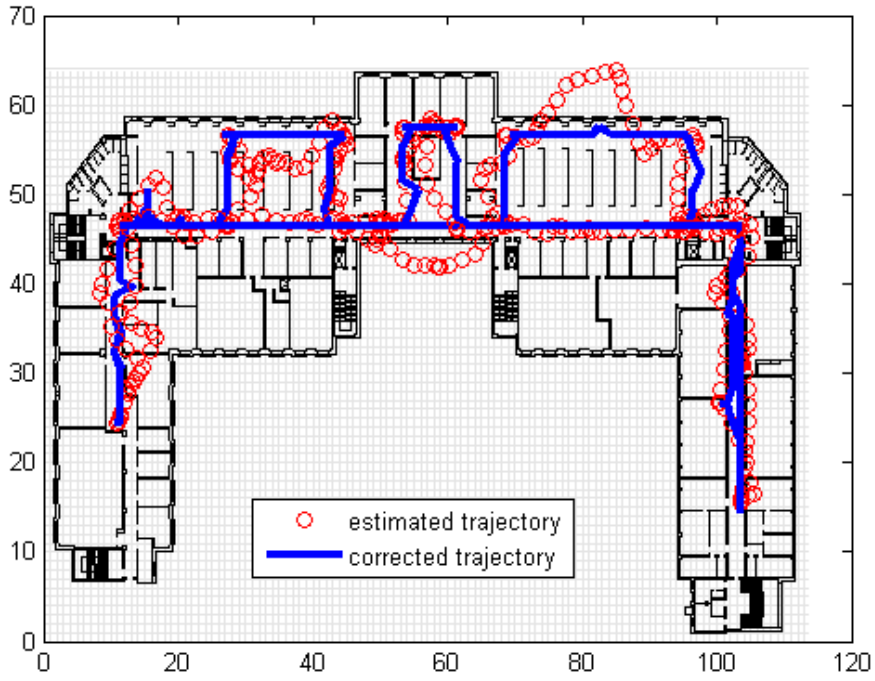



*Figure 54.* An example with solved divergence and zigzag problems.

# 8. Conclusions and Future Work

In this research a map matching system has been developed to improve the results of localization systems in terms of feasibility and accuracy. An offline CRF map matching algorithm has been proposed that can be easily loosely coupled to different localization systems; it uses only the output estimated trajectory of the localization system as an input and refines it based only on the map information, without the direct use of raw sensor data. The adoption of loose coupling means that the map matching algorithm does not need to know the implementation details of the primary localization system; this also then allows the map matching algorithm to be linked to variuos localization systems that produces (x, y) coordinate trajectory as a separate module and to use only its output.

In order to utilize the map information, a tool was developed to extract the map information from CAD files of the building plans; and to convert this information to be presented in a model suitable for the map matching algorithm; CAD files were chosen as they are widely used in architectural design. A grid-based model was developed that represents the map as square cells with attributes to represent characteristic of each cell (has obstacle, free for walk or on preferable lines of walk).

The research has demonstrated the successful use of CRF as a model in the offline map matching process. The CRF algorithm corrects trajectories produced by localization systems when given the appropriate map information. Grid cells of fixed length were used for the map representation, with a cell size of 1 m. The use of a larger cell size achieved less accurate results, mainly due to obstacles blocking entire cells; big cells are unsuitable for sites with narrow areas such as corridors. On the other hand, although smaller cells should provide more accurate results, they also potentially increase the computation time as well as the occurrence of the divergence problem as it allows the corrected trajectory to enter small semi-enclosed areas and stick there diverging from the correct path.

Problems found in the map matching results have been addressed and solved. The first problem is the divergence that occurs when trajectories get stuck inside small semi-enclosed areas. To solve this problem a post correction algorithm was applied, in which

the divergence point is detected and the CRF algorithm backward phase is restarted from that point. The second problem is the non-smooth sections in the output trajectories. This problem is a result of using the distance with the input observation as the only feature in addition to avoiding obstacles. To solve this problem the CRF algorithm is modified to include a new feature which uses the behaviour of the people while walking, particularly the habit walking in straight lines. The results show that the new feature enhanced the trajectories by smoothing them.

Even though the experiments were carried out with pedestrians in mind, the proposed map matching algorithm can be applied to other moving object. This is because the map matching system is independent of the localization system internal implementation; instead it uses only its output coordinates. However, the cell size or the buffer size of the map model might need to be adapted in some cases.

As future work, adding more semantic information such as rooms and doors to the map model to improve the results will be the next task. Another planned future work is to use the CRF-based map matching with different map models such as graph-based models and object-oriented models; the grid-based model has the advantage of enabling high accuracy but the big number of cells increases the computations as the computational complexity of the CRF algorithm which is linearly dependent on the number of cells. Using graphical map models with fewer nodes and links may help improve the speed of the algorithm. Through object-oriented models, more semantic information such as rooms, doors, and stairs can be used by the algorithm.

## 8.1 Publications

- ***Conditional Random Field-based Offline Map Matching for Indoor Environments.***

  Bataineh, S., Bahillo, A., Díez, L. E., Onieva, E., & Bataineh, I. (2016). Conditional Random Field-Based Offline Map Matching for Indoor Environments. *Sensors*, *16*(8), 1302.

  In this paper, an offline map matching technique designed for indoor localization systems based on conditional random fields (CRF) is proposed. The proposed algorithm can refine the results of existing indoor localization systems and match

them with the map, using loose coupling between the existing localization system and the proposed map matching technique. The algorithm was applied to several actual and simulated trajectories of different lengths, with the results then refined and matched with the map using the CRF algorithm.

- ***Enhancing improved heuristic drift elimination for step-and-heading based pedestrian dead-reckoning systems.***

  Díez, L. E., Bahillo, A., Bataineh, S., Masegosa, A. D., & Perallos, A. (2016, October). Enhancing improved heuristic drift elimination for step-and-heading based pedestrian dead-reckoning systems. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the* (pp. 4415-4418). IEEE

  In this paper, an enhanced the method known as improved heuristic drift elimination (iHDE) is proposed to be implemented in a Step-and-Heading (SHS) based PDR system that allows placing the inertial sensors in almost any location of the user's body. Particularly, wrist-worn sensors are used.

- ***Enhancing Conditional Random Field-based Map Matching with Behavioral Information.***

  Bataineh, S., Bahillo, A., & Díez, L. E., (2016). Enhancing Conditional Random Field-based Map Matching with Behavioral Information. Retrieved from http://www3.uah.es/ipin2016/usb/app/descargas/177_WIP.pdf

  This was a work-in-progress that was presented as a poster. In this work-in-progress an enhancement to the Conditional Random Field (CRF) based map matching algorithm was proposed to make the map matched trajectory smoother and more feasible. A new semantic layer model that depends on the behavioural areas of walking was added to the map to enhance the output trajectory in terms of accuracy and smoothness.

- ***Using of Behavioral Information for Enhancing Conditional Random Field-based Map Matching.***

  Bataineh, S., Bahillo, A., & Díez, L. E., (2017). Using of Behavioral Information for Enhancing Conditional Random Field-based Map Matching. In *European Navigation Conference*, Geneva, Switzerland. Accepted for publication.

  In this paper an enhancement to the Conditional Random Field (CRF) based map matching algorithm was proposed to make the map matched trajectory smoother and more feasible. A new semantic layer model that depends on the behavioural areas of walking was added to the map to enhance the output trajectory in terms of accuracy and smoothness.

# 9. References


Afyouni, I., Cyril, R., & Christophe, C. (2012). Spatial models for context-aware indoor navigation systems: A survey. *Journal of Spatial Information Science*, *1*(4), 85-123.

Aitenbichler, E., & Muhlhauser, M. (2003, May). An IR local positioning system for smart items and devices. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on* (pp. 334-339). IEEE.

Alhmiedat, T. A., & Yang, S. H. (2008). A ZigBee-based mobile tracking system through wireless sensor networks. *International Journal of Advanced Mechatronic Systems*, *1*(1), 63-70.

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, *50*(2), 174-188.

Ascher, C., Kessler, C., Wankerl, M., & Trommer, G. F. (2010, September). Dual IMU Indoor Navigation with particle filter based map-matching on a smartphone. In *Indoor positioning and indoor navigation (IPIN), 2010 international conference on* (pp. 1-5). IEEE.

Aurenhammer, F. (1991). Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, *23*(3), 345-405.

Autodesk, Inc. Drawing Interchange File Formats. Retrieved February 22, 2017 from http://www.autodesk.com/techpubs/autocad/acadr14/dxf/drawing_interchange_file_formats.htm

Azizyan, M., Constandache, I., & Roy Choudhury, R. (2009, September). SurroundSense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking* (pp. 261-272). ACM.

Bahillo, A., Marušić, B. G., & Perallos, A. (2015, December). A Mobile Application as an Unobtrusive Tool for Behavioural Mapping in Public Spaces. In *International Conference on Ubiquitous Computing and Ambient Intelligence* (pp. 13-25). Springer International Publishing.

Bataineh, S., Bahillo, A., Díez, L. E., Onieva, E., & Bataineh, I. (2016). Conditional Random Field-Based Offline Map Matching for Indoor Environments. *Sensors*, *16*(8), 1302.

Bataineh, S., Bahillo, A., & Díez, L. E. (2016) Enhancing Conditional Random Field-based Map Matching with Behavioral Information. Retrieved January 17, 2017 from http://www3.uah.es/ipin2016/usb/app/descargas/177_WIP.pdf

Bataineh, S., Bahillo, A., & Díez, L. E., (2017). Using of Behavioral Information for Enhancing Conditional Random Field-based Map Matching. In *European Navigation Conference*, Geneva, Switzerland. Accepted for publication.

Becker, C., & Dürr, F. (2005). On location models for ubiquitous computing. *Personal and Ubiquitous Computing*, *9*(1), 20-31.

Becker, T., Nagel, C., & Kolbe, T. (2010). *Discussion of Euclidean space and cellular space and proposal of an integrated indoor spatial data model*. Tech. rep., Institute of Geodesy and Geoinformation Science.

Becker, T., Nagel, C., & Kolbe, T. H. (2009, May). Supporting contexts for indoor navigation using a multilayered space model. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware* (pp. 680-685). IEEE.

Bentley, J. L., & Maurer, H. A. (1980). Efficient worst-case data structures for range searching. *Acta Informatica*, *13*(2), 155-168.

Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, *22*(1), 39-71.

Bernstein, D., & Kornhauser, A. (1998). An introduction to map matching for personal navigation assistants.

Bhatt, M., Dylla, F., & Hois, J. (2009, September). Spatio-terminological inference for the design of ambient environments. In *International Conference on Spatial Information Theory* (pp. 371-391). Springer Berlin Heidelberg.

Bishop, Christopher M. "Pattern recognition." *Machine Learning* 128 (2006).

Bohn, J., & Vogt, H. (2003). *Robust probabilistic positioning based on high-level sensor-fusion and map knowledge*. ETH, Swiss Federal Institute of Technology, Institute for Pervasive Computing.

Borenstein, J., Ojeda, L., & Kwanmuang, S. (2009, May). Heuristic reduction of gyro drift in IMU-based personnel tracking systems. In *SPIE Defense, Security, and Sensing* (pp. 73061H-73061H). International Society for Optics and Photonics.

Brajdic, A., & Harle, R. (2013, September). Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (pp. 225-234). ACM.

Chatila, R., & Laumond, J. P. (1985, March). Position referencing and consistent world modeling for mobile robots. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on* (Vol. 2, pp. 138-145). IEEE.

Chen, J. C., Wang, Y. C., Maa, C. S., & Chen, J. T. (2006). Network-side mobile position location using factor graphs. *IEEE Transactions on Wireless Communications*, *5*(10), 2696-2704.

Chen, Y., Lymberopoulos, D., Liu, J., & Priyantha, B. (2012, June). FM-based indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (pp. 169-182). ACM.

Choset, H., & Burdick, J. (2000). Sensor-based exploration: The hierarchical generalized Voronoi graph. *The International Journal of Robotics Research*, *19*(2), 96-125.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

Crowley, J. L. (1989, May). World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on* (pp. 674-680). IEEE.

Cui, Y., & Ge, S. S. (2003). Autonomous vehicle positioning with GPS in urban canyon environments. *IEEE transactions on robotics and automation*, *19*(1), 15-25.

Davidson, P., Collin, J., & Takala, J. (2010, October). Application of particle filters for indoor positioning using floor plans. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010* (pp. 1-4). IEEE.

Demyen, D. J., & Buro, M. (2006, July). Efficient triangulation-based pathfinding. In *Aaai* (Vol. 6, pp. 942-947).

Determine your Stride. (n.d.). Retrieved April 23, 2016 from https://cdn.shopify.com/s/files/1/0196/4616/files/Soleus-determing_your_stride.pdf

Díez, L. E., Bahillo, A., Bataineh, S., Masegosa, A. D., & Perallos, A. (2016, October). Enhancing improved heuristic drift elimination for step-and-heading based pedestrian dead-reckoning systems. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the* (pp. 4415-4418). IEEE.

Díez, L. E., Bahillo, A., Masegosa, A. D., Perallos, A., Azpilicueta, L., Falcone, F., ... & Villadangos, J. (2015, September). Signal processing requirements for step detection using wrist-worn IMU. In *Electromagnetics in Advanced Applications (ICEAA), 2015 International Conference on* (pp. 1032-1035). IEEE.

Dobson, Robin. "Adaptive antenna array." U.S. Patent No. 4,780,721. 25 Oct. 1988.

Doucet, A., Gordon, N. J., & Krishnamurthy, V. (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on signal processing*, *49*(3), 613-624.

Drawing Interchange and File Formats, Release 12. Retrieved March 22, 2016 from http://cnc25.free.fr/documentation/format_dxf/dxf.pdf

Dürr, F., & Rothermel, K. (2003, October). On a location model for fine-grained geocast. In *International Conference on Ubiquitous Computing* (pp. 18-35). Springer Berlin Heidelberg.

DXF reference, chapter1 – DXF format, Retrieved February 22, 2017 from http://www.autodesk.com/techpubs/autocad/acad2000/dxf/dxf_format.htm

Elhamshary, M., & Youssef, M. (2014, September). CheckInside: a fine-grained indoor location-based social network. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 607-618). ACM.

Elsevier Health Sciences. (2009, March 21). Moderate Intensity Walking Means 100 Steps Per Minute. *ScienceDaily*. Retieved April 22, 2016 from www.sciencedaily.com/releases/2009/03/090317094719.htm

Feldmann, S., Kyamakya, K., Zapater, A., & Lue, Z. (2003, June). An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation. In *International Conference on Wireless Networks* (pp. 109-113).

Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268-278.

Fox, D., Burgard, W., & Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, *11*, 391-427.

Foxlin, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer graphics and applications*, *25*(6), 38-46.

Franz, G., Mallot, H., Wiener, J., & Neurowissenschaft, K. (2005, August). Graph-based models of space in architecture and cognitive science-a comparative analysis. In *Proceedings of the 17th International Conference on Systems Research, Informatics and Cybernetics* (Vol. 3038).

Froese, T., Fischer, M., Grobler, F., Ritzenthaler, J., Yu, K., Sutherland, S., ... & Barron, A. (1999). Industry foundation classes for project management-a trial implementation. *Electronic Journal of Information Technology in Construction*, *4*, 17-36.

Frost, C., Jensen, C. S., Luckow, K. S., Thomsen, B., & Hansen, R. (2011, May). Bluetooth Indoor positioning system using fingerprinting. In *International Conference on Mobile Lightweight Wireless Systems* (pp. 136-150). Springer Berlin Heidelberg.

Fuchs, H., Kedem, Z. M., & Naylor, B. F. (1980, July). On visible surface generation by a priori tree structures. In *ACM Siggraph Computer Graphics* (Vol. 14, No. 3, pp. 124-133). ACM.

Fujii, K. (2013). Extended Kalman Filter. *Refernce Manual*.

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993, April). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing* (Vol. 140, No. 2, pp. 107-113). IET.

Grewal, M. S. (2011). *Kalman filtering* (pp. 705-708). Springer Berlin Heidelberg.

Gu, Y., Lo, A., & Niemegeers, I. (2009). A survey of indoor positioning systems for wireless personal networks. *IEEE Communications surveys & tutorials*, *11*(1), 13-32.

Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, *25*(7), 53-82.

Harle, R. (2013). A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys and Tutorials*, *15*(3), 1281-1293.

Hazas, M., & Hopper, A. (2006). Broadband ultrasonic location systems for improved indoor positioning. *IEEE Transactions on mobile Computing*, *5*(5), 536-547.

Hightower, J., & Borriello, G. (2001). Location systems for ubiquitous computing. *Computer*, *34*(8), 57-66.

Honey, S. K., Zavoli, W. B., Milnes, K. A., Phillips, A. C., White Jr, M. S., & Loughmiller Jr, G. E. (1989). Vehicle navigational system and method. *U.S. Patent No. 4,796,191*. Washington, DC: U.S. Patent and Trademark Office.

Hosmer, D. W., & Lemeshow, S. (2000). Introduction to the logistic regression model. *Applied Logistic Regression, Second Edition*, 1-30.

Hsiao, K., Miller, J., & de Plinval-Salgues, H. (2005). Particle Filters and their applications. *Cognitive Robotics*, *4*.

Hu, H., & Lee, D. L. (2004). Semantic location modeling for location navigation in mobile environment. In *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on* (pp. 52-61). IEEE.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical review*, *106*(4), 620.

Jensen, C. S., Lu, H., & Yang, B. (2009, May). Graph model based indoor tracking. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware* (pp. 122-131). IEEE.

Jiménez, A. R., Seco, F., Prieto, J. C., & Guevara, J. (2010, March). Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on* (pp. 135-143). IEEE.

Jiménez, A. R., Seco, F., Zampella, F., Prieto, J. C., & Guevara, J. (2012). Improved heuristic drift elimination with magnetically-aided dominant directions (MiHDE) for pedestrian navigation in complex buildings. *Journal of location based services*, *6*(3), 186-210.

Jiménez, A. R., Zampella, F., & Seco, F. (2013, October). Light-matching: A new signal of opportunity for pedestrian indoor navigation. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on* (pp. 1-10). IEEE.

Kaemarungsi, K., & Krishnamurthy, P. (2004, August). Properties of indoor received signal strength for WLAN location fingerprinting. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on* (pp. 14-23). IEEE.

Kainz, W., Egenhofer, M. J., & Greasley, I. (1993). Modelling spatial relations and operations with partially ordered sets. *International Journal of Geographical Information Systems*, *7*(3), 215-229.

Kawakami, S., & Ohtsuki, T. (2011). Localization using iterative angle of arrival method sharing snapshots of coherent subarrays. *EURASIP Journal on Advances in Signal Processing*, *2011*(1), 46.

Kim, S., & Kim, J. H. (2001). Adaptive fuzzy-network-based C-measure map-matching algorithm for car navigation system. *IEEE Transactions on industrial electronics*, *48*(2), 432-441.

Klepal, M., & Pesch, D. (2007, October). A bayesian approach for rf-based indoor localisation. In *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on* (pp. 133-137). IEEE.

Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., ... & Engelmann, W. H. (2001). The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science and Environmental Epidemiology*, *11*(3), 231.

Klinger, R., & Tomanek, K. (2007). *Classical probabilistic models and conditional random fields*. TU, Algorithm Engineering.

Kschischang, F. R., Frey, B. J., & Loeliger, H. A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, *47*(2), 498-519.

Lafferty, J., McCallum, A., & Pereira, F. (2001, June). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML* (Vol. 1, pp. 282-289).

Lamy-Perbal, S., Guénard, N., Boukallel, M., & Landragin-Frassati, A. (2015, October). A HMM map-matching approach enhancing indoor positioning performances of an inertial measurement system. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on* (pp. 1-4). IEEE.

Lan, K. C., & Shih, W. Y. (2013). On calibrating the sensor errors of a PDR-based indoor localization system. *Sensors*, *13*(4), 4781-4810.

Li, D., & Lee, D. L. (2008, April). A lattice-based semantic location model for indoor navigation. In *The Ninth International Conference on Mobile Data Management (mdm 2008)* (pp. 17-24). IEEE.

Li, F., Zhao, C., Ding, G., Gong, J., Liu, C., & Zhao, F. (2012, September). A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 421-430). ACM.

Li, X., Claramunt, C., & Ray, C. (2010). A grid graph-based model for the analysis of 2D indoor spaces. *Computers, Environment and Urban Systems*, *34*(6), 532-540.

Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Karstila, K., Reed, K., ... & Wix, J. (2012). Industry foundation classes IFC2x edition 3 technical corrigendum 1. *Online: http://www. buildingsmart-tech. org/ifc/IFC2x3/TC1/html [Accessed: 22 january 2017]*.

Liu, H., Darabi, H., Banerjee, P., & Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *37*(6), 1067-1080.

Liu, X., Makino, H., & Maeda, Y. (2008, August). Basic study on indoor location estimation using visible light communication platform. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* (pp. 2377-2380). IEEE.

Lopez de Ipina, D. (2002). *Visual sensing and middleware support for sentient computing* (Doctoral dissertation, University of Cambridge).

Lou, P., Zhang, H., Zhang, X., Yao, M., & Xu, Z. (2012, August). Fundamental analysis for indoor visible light positioning system. In *Communications in China Workshops (ICCC), 2012 1st IEEE International Conference on* (pp. 59-63). IEEE.

Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R. R., Handziski, V., & Sen, S. (2015, April). A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th international conference on information processing in sensor networks* (pp. 178-189). ACM.

Marušić, B. G., & Marušic, D. (2012). *Behavioural maps and GIS in place evaluation and design*. INTECH Open Access Publisher.

Medina, C., Segura, J. C., & De la Torre, A. (2013). Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. *Sensors*, *13*(3), 3501-3526.

Mekni, M. (2010). *Automated generation of geometrically-precise and semantically-informed virtual geographic environments populated with spatially-reasoning agents*. Universal-Publishers.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544-551.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *2*, 841-848.

Ni, L. M., Liu, Y., Lau, Y. C., & Patil, A. P. (2004). LANDMARC: indoor location sensing using active RFID. *Wireless networks*, *10*(6), 701-710.

Ochieng, W. Y., Quddus, M. A., & Noland, R. B. (2003). Map-matching in complex urban road networks.

Pereira, F. C., Costa, H., & Pereira, N. M. (2009). An off-line map-matching algorithm for incomplete map databases. *European Transport Research Review*, *1*(3), 107-124.

Priyantha, N. B., Chakraborty, A., & Balakrishnan, H. (2000, August). The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 32-43). ACM.

Quddus, M. A., Ochieng, W. Y., & Noland, R. B. (2006). Integrity of map-matching algorithms. *Transportation Research Part C: Emerging Technologies*, *14*(4), 283-302.;

Quddus, M. A., Ochieng, W. Y., & Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation research part c: Emerging technologies*, *15*(5), 312-328.

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *ieee assp magazine*, *3*(1), 4-16.

Rai, A., Chintalapudi, K. K., Padmanabhan, V. N., & Sen, R. (2012, August). Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking* (pp. 293-304). ACM.

Rajagopal, S. (2008). Personal dead reckoning system with shoe mounted inertial sensors. *Master's Degree Project, Stockholm, Sweden*.

Ramos, F. T., Fox, D., & Durrant-Whyte, H. F. (2007, June). CRF-Matching: Conditional Random Fields for Feature-Based Scan Matching. In *Robotics: Science and Systems*.

Remolina, E., & Kuipers, B. (2004). Towards a general theory of topological maps. *Artificial Intelligence*, *152*(1), 47-104.

Remolina, E., Fernandez, J. A., Kuipers, B., & Gonzalez, J. (1999, August). Formalizing regions in the spatial semantic hierarchy: An AH-graphs implementation approach. In *International Conference on Spatial Information Theory* (pp. 109-124). Springer Berlin Heidelberg.

Ren, M. (2012). *Advanced map matching technologies and techniques for pedestrian/wheelchair navigation* (Doctoral dissertation, University of Pittsburgh).

Rogalski, A. (2002). Infrared detectors: an overview. *Infrared Physics & Technology*, *43*(3), 187-210.

Saad, M. M., Bleakley, C. J., Ballal, T., & Dobson, S. (2012). High-accuracy reference-free ultrasonic location estimation. *IEEE Transactions on Instrumentation and Measurement*, *61*(6), 1561-1570.

Sayeef, S., Madawala, U. K., Handley, P. G., & Santoso, D. (2004, April). Indoor personnel tracking using infrared beam scanning. In *Position Location and Navigation Symposium, 2004. PLANS 2004* (pp. 698-705). IEEE.

Schweinzer, H., & Syafrudin, M. (2010, September). LOSNUS: An ultrasonic system enabling high accuracy and secure TDoA locating of numerous devices. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on* (pp. 1-8). IEEE.

Seitz, J., Vaupel, T., Meyer, S., Boronat, J. G., & Thielecke, J. (2010, March). A hidden markov model for pedestrian navigation. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on* (pp. 120-127). IEEE.

Shan, R. *Define: A naïve Bayes classifier*.Retrieved January 17, 2017 from http://quicktechie.com/cs/data-science-q-a/156-define-a-naive-bayes-classifier.

Shang, J., Hu, X., Gu, F., Wang, D., & Yu, S. (2015). Improvement schemes for indoor mobile location estimation: A survey. *Mathematical Problems in Engineering*, *2015*.

Shao, W., & Terzopoulos, D. (2005, July). Autonomous pedestrians. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (pp. 19-28). ACM.

Spassov, I., Bierlaire, M., & Merminod, B. (2006, May). Map matching for pedestrians via bayesian inference. In *Proceedings of the European Navigation Conference—Global Navigation Satellite Systems, Manchester, UK* (pp. 7-10).

Stoffel, E. P., Schoder, K., & Ohlbach, H. J. (2008, November). Applying hierarchical graphs to pedestrian indoor navigation. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems* (p. 54). ACM.

Subhan, F., Hasbullah, H., Rozyyev, A., & Bakhsh, S. T. (2011, April). Indoor positioning in bluetooth networks using fingerprinting and lateration approach. In *2011 International Conference on Information Science and Applications* (pp. 1-9). IEEE.

Sutton, C., & McCallum, A. (2010). An introduction to conditional random fields. *arXiv preprint arXiv:1011.4088*.

Syed, S., & Cannon, M. E. (2004, January). Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons. In *ION National Technical Meeting, San Diego, CA* (Vol. 1, pp. 26-28).

Symonds, J. (Ed.). (2009). *Auto-Identification and ubiquitous computing applications: RFID and Smart Technologies for Information Convergence*. IGI Global.

Taheri, A., Singh, A., & Emmanuel, A. (2004, November). Location fingerprinting on infrastructure 802.11 wireless local area networks (WLANs) using Locus. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on* (pp. 676-683). IEEE.

Tas, B., & Altiparmak, N. (2009, December). Low cost indoor location management system using infrared leds and Wii Remote Controller. In *2009 IEEE 28th International Performance Computing and Communications Conference* (pp. 280-288). IEEE.

Turgut, B., & Martin, R. P. (2009, November). Restarting p filters: an approach to improve the performance of dynamic indoor localization. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE* (pp. 1-7). IEEE.

Van Treeck, C., & Rank, E. (2004). Analysis of building structure and topology based on graph theory. In *Proceeding of the 10th International Conference on Computing in Civil and Building Engineering. Weimar*.

Vildjiounaite, E., Malm, E. J., Kaartinen, J., & Alahuhta, P. (2002, August). Location estimation indoors by means of small computing power devices, accelerometers, magnetic sensors, and map knowledge. In *International Conference on Pervasive Computing* (pp. 211-224). Springer Berlin Heidelberg.

Wallgrün, J. O. (2010). Qualitative spatial reasoning for topological map learning. *Spatial Cognition & Computation*, *10*(4), 207-246.

Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M., & Choudhury, R. R. (2012, June). No need to war-drive: unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (pp. 197-210). ACM.

Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems (TOIS)*, *10*(1), 91-102.

Ward, A., Jones, A., & Hopper, A. (1997). A new location technique for the active office. *IEEE Personal Communications*, *4*(5), 42-47.

Weinstein, R. (2005). RFID: a technical overview and its application to the enterprise. *IT professional*, *7*(3), 27-33.

Werner, S., Krieg-Brückner, B., & Herrmann, T. (2000). Modelling navigational knowledge by route graphs. In *Spatial cognition II* (pp. 295-316). Springer Berlin Heidelberg.

White, C. E., Bernstein, D., & Kornhauser, A. L. (2000). Some map matching algorithms for personal navigation assistants. *Transportation research part c: emerging technologies*, *8*(1), 91-108.

Widrow, B., Mantey, P. E., Griffiths, L., & Goode, B. (1967). Adaptive antenna systems. *The Journal of the Acoustical Society of America*, *42*(5), 1175-1176.

Widyawan, Klepal, M., & Beauregard, S. (2008, March). A backtracking particle filter for fusing building plans with PDR displacement estimates. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on* (pp. 207-212). IEEE.

Woodman, O. J. (2007). An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, *14*, 15.

Woodman, O., & Harle, R. (2008, September). Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 114-123). ACM.

Xiao, Z., Wen, H., Markham, A., & Trigoni, N. (2014, April). Lightweight map matching for indoor localisation using conditional random fields. In *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on* (pp. 131-142). IEEE.

Xu, M., Du, Y., Wu, J., & Zhou, Y. (2015). Map Matching Based on Conditional Random Fields and Route Preference Mining for Uncertain Trajectories. *Mathematical Problems in Engineering*, *2015*.

Yang, H., Cheng, S., Jiang, H., & An, S. (2013). An enhanced weight-based topological map matching algorithm for intricate urban road network. *Procedia-Social and Behavioral Sciences*, *96*, 1670-1678.

Yazici, A., Yayan, U., & Yücel, H. (2011, June). An ultrasonic based indoor positioning system. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on* (pp. 585-589). IEEE.

Yu, M. (2006). *Improved positioning of land vehicle in ITS using digital map and other accessory information* (Doctoral dissertation, The Hong Kong Polytechnic University).

Zampella, F., Bahillo, A., Prieto, J., Jiménez, A. R., & Seco, F. (2013). Pedestrian navigation fusing inertial and RSS/TOF measurements with adaptive movement/measurement models: Experimental evaluation and theoretical limits. *Sensors and Actuators A: Physical*, *203*, 249-260.

Zeytinci, M. B., Sari, V., Harmanci, F. K., Anarim, E., & Akar, M. (2013). Location estimation using RSS measurements with unknown path loss exponents. *EURASIP Journal on Wireless Communications and Networking*, *2013*(1), 1-14.