NTNU
Norwegian University of
Science and Technology

Multi-Target Localization and Tracking with Wireless Sensor Networks for Digital Storytelling

Sonia Mendieta Moreton

Master of Science in Electronics
Submission date: June 2017
Supervisor: Pierluigi Salvo Rossi, IES
Co-supervisor: Andrew Perkis, IES

Norwegian University of Science and Technology
Department of Electronic Systems

Abstract

In recent years, Wireless Sensor Networks (WSNs) have been considered for a wide variety of applications in areas such as medicine, industry, environmental issues or defense, due to their attractive characteristics which have motivated their rapid diffusion. Navigation, environmental monitoring, disaster prevention, health monitoring or area monitoring are some examples of their extended use.

Multiple target localization and tracking using WSNs has become an interesting and challenging topic of research in recent years. A wide variety of approaches have been proposed in the literature but further work has to be done in order to provide more accurate localization and tracking algorithms. There is no unique solution for finding a particular methodology that performs optimal object detection, localization and tracking due to the excessive amount of configurable parameters in a Wireless Sensor Network (*i.e.* topology, sensors features, communication protocols, network architecture) or the characteristics of the scenario under study (*i.e.* cooperative or non-cooperative targets, indoor or outdoor areas, environmental conditions).

Radio Tomographic Imaging (RTI) has been considered an interesting technique for tracking purposes. It allows an image reconstruction of the region of interest using the received signal strength (RSS) in every mote of the wireless network. Localization and target tracking is accomplished by analyzing the changes in the RSS in the set of estimated images due to the presence of targets. This technique has shown favorable results for area monitoring even in cluttered environments.

In the present work, a multi-target localization and tracking algorithm has been developed for digital storytelling applications in public spaces. RTI has been the selected methodology for these purposes due to the characteristics of the area under study. Due to the lack of RF sensors, environmental data (*i.e.* RSS changes in the area) has been generated for latter tests of the developed algorithm. From the generated data, image estimation and further processing has been performed, achieving successfully the localization and tracking tasks. Afterwards, Kalman Filter and Particle Filter are applied to the tracking results for better position estimation. A comparative study of different set-ups is also performed in order to understand the system behavior under different scenarios. Additionally, a user interface is created to make the algorithm configuration more user friendly for possible future users that may hardly understand the developed MATLAB program.

Key words: *Multiple Target Tracking*, *Object Detection*, *Wireless Sensor Networks*, *Radio Tomographic Imaging*, *Received Signal Strength*, *Image Estimation*, *RF sensors*, *Kalman Filter*, *Particle Filter.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Object Detection and Tracking

For many applications, such as surveillance or area monitoring, object detection and tracking has become an important and challenging topic of research. It may seem very intuitive for an average person to be able to distinguish an specific object in real life or even in a video playback but, what about accurately identifying the location of an object just with sounds? Or by knowing the surrounding temperature? This could seem a bit more complicated.

Object detection is the ability of detecting an specific object in a region of interest. Once an object is detected, object localization deals with accurately knowing the position of the object in the area. Object tracking is the process of identifying the same object over time and know its position inside the selected region. The same process can be also accomplished for tracking multiple objects in many different scenarios.

When monitoring an area, for security for example, some sensors can be deployed, such as cameras, and usually an operator is in charge of watching the video streams from all the monitoring cameras. If the number of cameras increases, it would be challenging for the operator to keep track of all the streams. In such cases, it is worthwhile processing the video streams and presenting some useful results to help the operator.

In computer vision, a non intelligent machine has to deal with tasks that the human visual system can easily accomplish such as distinguish elements from images or videos. Some of the typical tasks are object detection, object recognition, object identification, motion analysis, event detection or machine interaction. It has been broadly studied in medical areas (*e.g.* detection of tumors in X-ray images), industrial control (*e.g.* guidance of robots), multimedia and videogames industry (*e.g.* face recognition or virtual reality), navigation (*e.g.* autonomous vehicles), security issues (*e.g.* target identification and tracking for surveillance) and even for space applications (*e.g.* space exploration).

When analyzing a digital image for object detection and tracking, the first step is pre-processing the image for removing unwanted noise. Then, depending on the application, an important task is to identify and classify object shapes and its boundaries such as faces, bodies or vehicles, and extract other features like simple geometries and other points of interest. Hereafter, a filter is applied in order to remove unwanted areas and focus on the desired shapes. This process is called image segmentation. Afterwards, the remaining data is processed and algorithms are applied over it. Finally, a decision is made over the results (e.g. classifying an object as detected or updating tracks) and next steps are applied depending on the application.

On the other hand, when it comes to avoiding privacy issues, object detection and tracking becomes a more challenging task. In this case, one can no longer work with audio or video data. Other types of environmental data must be extracted in order to be able to detect and track an specific object inside a region of interest such as temperature, light, radio waves, or any other type of phenomena that the object can change just with its presence.

In this case, Wireless sensor networks are very interesting to perform detection and tracking tasks. Sensing devices as infrared sensors, RFID sensors, ultrasonic sensors, RF sensors, LIDAR sensors, RADAR sensors, lasers or photodetector sensors, among others, can be quiet useful technologies for providing input data to the corresponding detection and tracking algorithm.

When tracking a target, it is also worth noticing, regarding the types of sensors, that the target can be either classified as cooperative or non-cooperative, depending on whether the target is wearing or not a device that can be recognize by the sensing units (e.g. Tracking animals with RFID tags). In most cases, the target is non-cooperative, so the design of localization and tracking algorithms is generally more complex.

1.2 Overview of Wireless Sensor Networks

A Wireless Sensor Network (*WSN*) consists of a set of spatially distributed devices in an area of interest which are able to measure and monitor physical conditions in their surroundings in order to extract certain knowledge about the environment as a whole integrated system. Typically, a WSN is composed of sensor nodes or motes that collect the measurements, which can be choose from a wide variety of types, from temperature sensors to chemical sensors. A Gateway is responsible for sending the data of the sensors to another network where the data will be analyzed. Then, the Fusion Center (*FC*) receives all the data and processes it in order to extract the relevant information depending on the application. Sometimes, when the network is big enough, with some hundreds or even thousands of sensors, repeater nodes are also used as a bridge between the sensor nodes and the Gateway.

The architecture of a general WSN is represented in Figure 1.1.

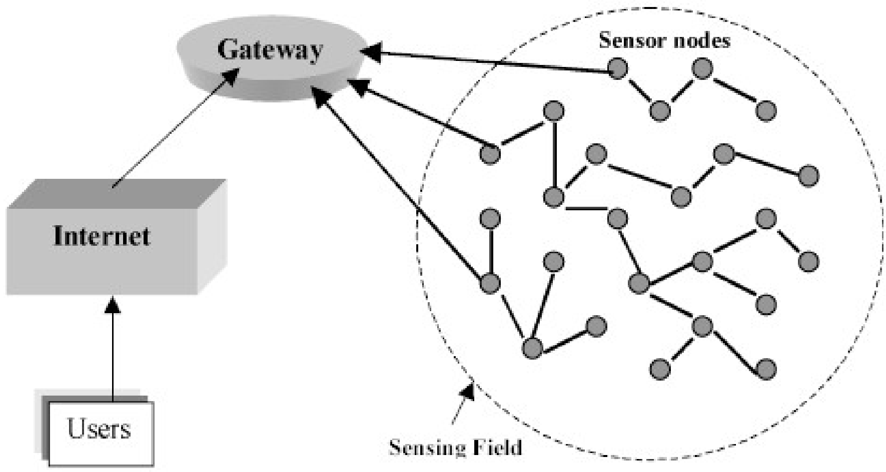


Figure 1.1: Wireless Sensor Network Architecture

The different network elements can be distributed in many different ways, forming different topologies as tree, star, ring, mesh, circular, grid, etc., which can be useful depending on the desired network features.

In general, WSNs has some remarkable characteristics [7] that have made them very popular in the present years for a wide variety of applications. Industrial monitoring and control (machine health, waste, structures), environmental monitoring (temperature, pressure, pollution, fire detection, disaster management), health monitoring (disease prevention, physiological data), automated agriculture, home applications (security, automation), tracking and monitoring objects (vehicles, people, birds) or military surveillance, among others, are some of the uses of these networks.

The main characteristics of WSNs that are worth highlighting are the following:

- **Reliability:** In case of a node failure, the network is able to change the routing path of all other sensors to the Gateway without affecting the network efficiency, even if the failure is produced in a master node.

- **Robustness:** The elements of a WSN can easily resist hard environmental conditions and also transmit their data via radio communications.

- **Scalability:** A WSN can be extended very easily by introducing new elements and configuring them. The topology of the network can change by adding or removing elements without affecting the network performance.

- **Heterogeneity of nodes:** Before transmitting data, the sensor nodes usually have integrated an A/D converter, so the data is previously converted into digital signals and then transmitted. Thus, the network can be made up of different types of sensors.

On the other hand, WSNs also have some constraints that must be taken into account when designing and deploying it, which are:

- **Power-consumption:** The sensor nodes have to use small batteries so that the amount of computational processing and memory has to be very low in order to reduce energy costs. Additionally, the transmitted bandwidth highly affects the power consumption. Low-power methods for radio communication and data acquisition are being investigated and techniques of energy harvesting are usually applied.

- **Low data rates:** The amount of transmitted data is proportional to the frequency of data acquisition so, in order to reduce energy consumption, there is a need of transmitting information with low data rates and narrow bandwidth.

- **Security and Communication Failures:** Due to the intrinsic characteristics of the radio channel, the transmitted signals can be affected, introducing non-desired errors. Additionally, techniques to protect the information, as encoding, are also restricted due to energy constraints, so the network security can be undermined.

The wireless sensor nodes or motes are mainly composed of a sensing unit, a processing unit, a power unit and a communication unit, as shown in Figure 1.2. The *sensing unit* consists of a sensing device which measures an external condition depending on its application. The measure is then sent to the processing unit after converting it into digital. The *processing unit* is typically a micro-controller with some predefined simple instructions. It performs easy tasks and also controls all the other units. The main goal of processing the sensor data is to send less information through the communication channel so that energy can be saved, prolonging the sensor lifetime, the transmitting bandwidth is reduced and the message is less prone to errors. However, by doing so, the network performance is inevitably deteriorated. The *communication unit* is basically an RF transceiver with an antenna that sends periodically small packets to other nodes of the network. The *battery unit* provides the power to all the units. Additionally, an internal or external memory or some other units can also be included.

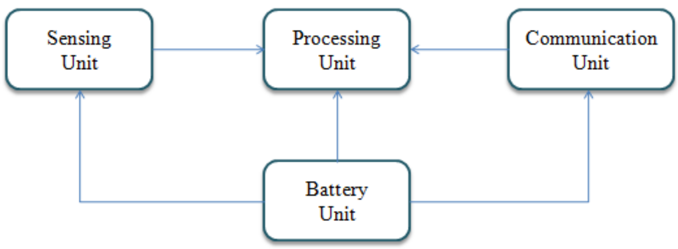


Figure 1.2: Block diagram of a Wireless Sensor Node

When deploying a network, there are different ways of positioning the wireless sensor nodes. In one case, the sensors can be located either in some predefined locations. However, it is also possible to deploy the sensor nodes randomly in the region of interest if the area is hard to reach (e.g. military terrain, forest, etc.) or if it implies a great expense. In this case, there could be some solutions to know the real position like inserting a GPS module, or establishing a communication protocol between nodes in order to know the relative positions between them. Furthermore, the fusion center can take this into account when processing the data and perform some position estimation algorithms.

Synchronization of the different network nodes is also important in order to prevent data collapses. This is usually made by adding a header in the transmitted packages, which implicitly increases the transmitted bandwidth.

The network can be configured in two different ways: centralized or distributed. Both configurations can be appreciated in Figure 1.3. The main difference between them is whether the data fusion is made in a fusion center or in a distributed manner, where each sensor node collaborates with its neighbors and make their own decisions.

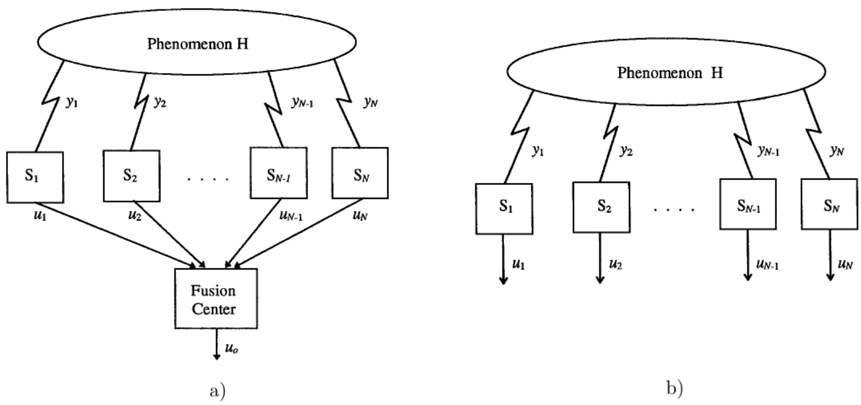


Figure 1.3: a) Centralized topology with fusion center. b) Distributed topology without fusion center [1]

In the first case, each sensor node sends the data to the fusion center. Then, the fusion center, after gathering all possible information, it implements an algorithm of data fusion and provides the operator the required presentation of the results. As the fusion center is generally located in an office or with enough power supply, it does not have energy constraints, so the applied algorithms can be computationally more complex than in the sensor nodes. On the other hand, a distributed configuration implies that each sensor node sends information to other nodes in its vicinity and the data fusion is achieved in some of the nodes with the gathered information. In this case, the complexity of the data fusion algorithm is highly reduced because of the lack of resources in the nodes.

In conclusion, Wireless Sensor Networks have become an interesting and highly competitive technology with many applications that is growing fast in the present years. It is clearly expected that the number of applications will increase in the following years. Thus, so much effort has to be made in the near future to enhance security, power consumption and other constraints as well as researching improved synchronization and fusion techniques.

1.3 Motivation

Target localization and tracking with WSN has become a demanded solution in the present years due to the continuous emergence of new applications in which knowing if an object is present in an area and its position is essential. Target tracking tasks can be useful for surveillance applications, area monitoring or industry control. In areas such as medicine or defense, the recognition and identification of objects is also an important field of study nowadays. Enemy recognition and continuous tracking in a battle field or identification of tumors in medical images can be crucial tasks to carry out. The use of connected sensors for these challenges reinforces the current systems features reaching levels inaccessible for traditional systems. It is an interesting topic of research how one sensor can tackle the problem of tracking targets as an individual and the value-added performance that achieve by working as a network.

Multiple target tracking with WSN is a challenging task to perform. Depending on the types of sensors and their topology, there exist many different ways of approaching the detection, localization and tracking problems. None of the existing solutions is yet considered the best and this makes one think that there is still a long way to go regarding tracking challenges using sensor technology.

The completion of this project will lead to the understanding of the advantages and practical implications of using WSN for localization and tracking applications, to gain knowledge about existing algorithms and estimation methods, to understand how WSN can handle the available information to perform detection, localization and tracking tasks in a specific region and, most importantly, to get experience in the design and development of a multi-target tracking algorithm using WSN for real applications.

1.4 Thesis Scope

The main goal of this thesis is the research and understanding of multi-target localization and tracking systems using wireless sensor networks (WSNs) for digital story-telling applications. The tracking systems under study focus on localization and tracking of human beings in public areas, so the interest is mainly on WSN with sensors different than audio and video, which implicitly avoid privacy issues, such as RF sensors, LIDAR sensors, temperature sensors, and so on.

The theoretical work includes literature research about distributed detection and estimation in wireless sensor networks, understanding the existing localization and tracking algorithms and the different techniques that can be applied in wireless sensor networks, study of data fusion techniques and literature research about the outcoming difficulties when tracking multiple targets with WSNs.

The practical work includes the design and development of a multi-target localization and tracking algorithm for tracking people in a particular area using a wireless sensor network. The algorithm will be implemented using MATLAB® software. Performance analysis of the developed algorithm and comparison of different cases are also encompassed. A user interface that enables the control and configuration of the multiple target tracking algorithm will be also implemented to help the user for the digital story-telling application.

1.5 Thesis Organization

The rest of this document is organized as follows:

- Chapter 2 encompass a general overview of the existing literature and state-of-the-art regarding target localization and tracking, from the challenges in object detection to target tracking, covering also other topics as recursive filters for position estimation.

- Chapter 3 describes the different steps followed to develop the multi-target localization and tracking algorithm scope of this thesis. Data generation, environmental simulation, image estimation and processing, tracking algorithm and estimation filters are some of the tasks explained here. It also covers radio propagation theory and the effect of moving and stationary targets in radio waves.

- Chapter 4 presents the performance analysis of the designed tracking system, including a comparison of the results with different set ups. Performance methods for tracking algorithms are addressed and implemented.

- Chapter 5 addresses the operability of a designed user interface to handle the developed tracking algorithm in a higher level. It presents a helpful tool with multiple configuration options to study the behavior of a tracking algorithm and how a minor change can affect its performance.

- Chapter 6 includes the conclusions of the project and future work.

Chapter 2

Background

2.1 Object Detection

One of the most important tasks that a tracker system must accomplish is to decide whether if a specific object is present in the scenario under study or not. In order to do so, the system must acquire environmental information and be able to process it in order to isolate and recognize specific objects. Depending on how the environmental information is acquired, many different recognition methods [8] can be applied. In wireless sensor networks, for example, the object detection and recognition is a task that can be accomplished by the individual sensors, in isolated groups or at the fusion center, depending on the characteristics of the network.

2.1.1 Object Recognition

An interesting field of study for object detection is the classification of objects. There are many applications that need to recognize specific shapes of the present objects to consider that an object is detected. Some examples of this type of applications could be facial recognition for video systems or fingerprint identification for security systems. Other systems may not need to recognize objects in detail but some characteristics, such as simple shapes. Furthermore, there are some systems in which an existing object is detected just because of the changes on the surrounding environment that provokes its own presence, in which the shape of the object is not even important for the application. An example of these systems could be a target tracking system with wireless sensors deployed in an area where the sensors detect the presence of a target and, working as a network, are capable of estimating the target position without targets spatial information.

When an image representation of the scenario can be obtained, the tracker system usually employs object recognition methods over the available image. Object representations [9] commonly used in tracking applications are points (e.g. one point in the middle or a set of distributed points around the object), simple geometric shapes (e.g. rectangle, ellipse), contours and articulated models (e.g. body parts of a human being), as can be seen in Figure 2.1.

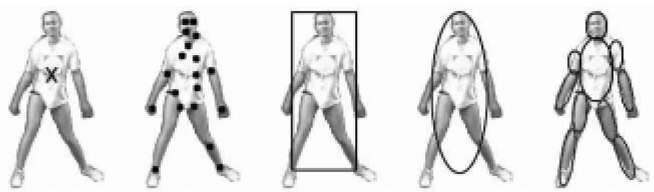


Figure 2.1: Different forms of object representation for detection

Additionally, the appearance features of an object can be analyzed in order to extract useful information. Probability density of the object appearance, use of templates, or motion transition analysis are some of the methods to describe appearance of present objects. Other useful characteristics that can be taken into account for the detection of objects in an image are colors, textures, boundaries, etc.

2.1.2 Distributed Detection

In traditional tracking systems where multiple sensors are deployed in a specific region, target detection and the corresponding estimations (*e.g.* position, velocity, etc.) are usually conducted in a centralized manner. In this case, all sensing devices send the available environmental information to a central processing center where optimal detection and estimation is performed.

On the other hand, in applications such as multiple target tracking in extended areas, the information of some sensors may not be relevant for the detection of a target. In such cases, decentralized detection can be accomplished. Depending on the network features (*e.g.* topology, type of sensors, etc.), different methods for distributed detection and data fusion [1][10][11] can be used. Preliminary decisions can be made locally at each sensor and then these decisions are sent to the fusion center. Decisions can be made at one sensor not only with its own data but with the available information from other sensors.

Distributed detection leads to a reduction in the transmission bandwidth, reducing costs and increasing sensors lifespan. However, the available information at the fusion center is less than in centralized systems, giving rise to a loss of performance. Thus, the corresponding detection and estimation algorithms carried out in the sensors (*i.e.* local decisions) and in the fusion center (*i.e.* global decisions and estimation) must be conducted in an optimal manner.

In a hypothesis testing context, two different optimal approaches can be addressed: Neyman-Pearson and Bayesian formulations.

- **Neyman-Pearson Formulation**: In order to decide whether if a target is present or not, a binary hypothesis testing problem is solved in every sensor with the available environmental information (*i.e.* observations at each sensor), where hypothesis H_1 corresponds to target presence and hypothesis H_0 corresponds to target absence. Each sensor transmits to the fusion center a $x_i = 1$ if H_1 is decided or $x_i = 0$ if H_0. With the available information at the fusion center, the corresponding optimal fusion rules are applied and a global decision is obtained, being $x_o = 1$ if H_1 or $x_0 = 0$ if H_0. The NP formulation states that, for a desired global probability of false alarm, a local and global optimal decision rules must be found that maximize the global probability of detection. Usually, decision rules are based on finding thresholds that fulfill some specific conditions, which is not a trivial process and is often very difficult.

- **Bayesian Formulation**: In this case, binary test hypotheses are also solved as in NP formulation. The main objective in the Bayesian framework is to find optimal fusion rules that minimize the Bayesian risk. In this case, local decisions are made in every sensor. Every decision is assigned a cost depending on some specific probabilities known a priori depending on the network characteristics.

In both cases, some system characteristics must be assumed or known a priori, such as independency of sensor measurements, known or unknown target characteristics (*e.g.* target power), known or unknown network characteristics (*e.g.* position of sensors), probability density functions, probabilities of detection in sensors, and so on, in order to guarantee the systems feasibility.

Finding optimal decisions rules in distributed detection systems is a complicated task and several approaches have been proposed in the literature [12][13][14][15][16]. Usually the algorithms are computationally complex and one of the main goals is to reduce this inherent complexity by modeling statistically certain system features and consequently worsening the system performance. However, in many applications, distributed detection is the best solution, and therefore it has become an important subject of study in recent years.

2.2 Object Localization

An important task of an object tracker is the determination of targets positions in the area of interest. There have been many proposed approaches [17] about the localization of objects using distributed sensor networks. Depending on the system, one method would be preferred over others. Tracked targets in a surveillance area can be either stationary or moving through the area. Some methods can handle both states but others can only work with targets in motion. In some applications, targets can be equipped with a specific devices that enables the communication with the network (cooperative target), in others targets may not collaborate (non-cooperative target) due to privacy issues (*e.g.* tracking targets in public spaces). In the last case, the system performs device-free location (DFL). Additionally, targets can be localized when an image of the area is reconstructed from environmental data acquire by the deployed sensors.

When surveillance cameras are deployed in the area, the location of the targets can be obtained from image processing over the recorded video frames. However, in some cases is not possible to extract information from cameras due to privacy requirements. Thus, other type of physical information is required. For localization purposes, it is very common the analysis of RF signals in the surveillance area due to their characteristics. Radio waves are able to penetrate harsh environments, such as smoke, fog, rain or even walls, while images from cameras are very noisy in these conditions and it is almost impossible to detect or locate objects. Location of targets with RF signals can be achieve by analyzing some particular signal features such as the direction or angle of arrival (AOA), the time of arrival (TOA) or the received signal strength (RSS). These characteristics are available at the physical layer (PHY) of all the sensors in a WSN so non additional hardware is required. In the following subsections, some of the most applied methods are addressed for both cooperative and non-cooperative cases, and imaging techniques for target localization are also outlined.

2.2.1 Cooperative Targets

In this case, the targets are equipped with wearable devices that allow the communication with the sensor network. Different technologies and methods have been investigated for determining the location of targets in a region of interest. Some of the approaches are as follows:

- **GPS**: Global positioning systems are a traditional technology widely used for localization purposes. However, having GPS receivers at each node and in the targets devices is costly and the tracking results may not be accurate enough due to interferences, especially in indoor environments. Thus, this technology may not be the optimal solution for tracking multiple targets with wireless sensor networks.

- **Infrared (IR)**: Infrared systems use the IR spectrum to transmit signals between the sensors and the targets device. The device of the target sends an identifier and the sensors in the surroundings will receive it. Then the sensors transmit this information to a centralized processor where the location of the target is estimated. This kind of systems are able to perform accurate estimations on the location of targets but the targets must be closed to the sensors in order to be detected due to the limited range of IR sensors. Additionally, the location of sensors has to be known at the central processor. Thus, it may not be the optimal technology for wide areas deployments and for harsh environments, where IR signal are highly degraded compared to RF signals.

- **Angle of Arrival (AOA)**: In this case, targets transmit RF signal which are receive by the sensors. The angle of arrival of the RF signal can be measures, for example, using directional antennas with narrow beams. Then, the overall information can be processed in a centralized system in order to estimate the location of the targets. However, the employed algorithms are usually time consuming and complex, and the hardware is costly.

- **Time of Arrival (TOA)**: In this kind of systems, the information regarding the distance between a target and a sensor can be acquired by measuring the time of arrival of RF signals. This can be accomplished in two different ways: measuring the time of arrival of a signal transmitted by the target (TOA) or measuring the time difference of arrival (TDOA) of a signal received in different sensors or the difference between two different signals transmitted by the target. These signals can have different nature, such as radio waves, acoustic waves, ultrasound waves, and so on. These systems perform accurate results when no obstacles are present and the signals are not reflected.

- **Received Signal Strength (RSS)**: In this case, the location of targets is achieved with the received signal strength available at every sensor in a WSN, independently of the type of sensors. Thus, no additional hardware is required. However, special algorithms have to be applied in order to compensate the radio channel changes or some previous measurements have to be acquired for channel estimations. In order to estimate the location of targets, an optimization problem has to be solved, in which the RSS at each sensor and the target position are related [18].

2.2.2 Non-cooperative Targets

In this case, the targets are completely device-free. The localization process is performed according to environmental measurements that can vary due to the presence of objects. There exist different approaches regarding device-free localization of targets [5][17][19], as follows:

- **Infrared (IR)**: This technology is also used in device-free localization systems. In this case, sensors are continuously transmitting IR rays. When a target is close to a sensor, the rays are obstructed and the sensor notices the change. Then, the available information at every sensor is sent to a central processor and an algorithm is applied to estimate the location of targets in the area.

- **Ultra-wideband (UWB) signals**: Ultra Wideband signals are transmitted from the sensors with specific features, such as amplitude or phases. UWB receivers notice the changes in the signal features due to the presence of moving targets in the surroundings. These systems are more expensive than narrowband systems, but they are able to obtain information regarding the pulse response in a multipath channel. Then, the information is processed and the estimation on the location is performed in a centralized base.

- **Narrowband signals**: The working principle in this case is similar to the one used in UWB systems, in which the changes in signal features can be measured. These systems are cheaper but when multipath effect is present in the channel, the system performance decreases considerably.

- **Received Signal Strength (RSS)**: In this case, the working principle is also similar to the abovementioned systems. The sensors receive RF signals from other transmitting sensors. As mentioned before, the RSS level is available at every sensor in the network. In the literature, it is state that the spatial variability of the RSS is due to multipath propagation and sensors distances. On the other hand, the temporal variability of the RSS is due to the presence of moving targets in the area. Thus, the localization is performed first comparing the RSS during an specified interval with a selected threshold and then estimating the target location with the corresponding algorithm.

2.2.3 Imaging Methods

An image reconstruction of the area under study can be performed from the acquired sensor measurements. Localization and tracking processes are then accomplished from the reconstructed image applying the corresponding image processing methods. The reconstructed image is a representation of some environmental characteristic that may change over time due to the presence of targets. Two different approaches are found in the literature:

- **Support Vector Machine (SVM)**: In this case, the image is reconstructed from the RSS available data at the sensor nodes. A SVM method is applied for estimating the probability density of targets in different regions of the surveillance area [20]. Afterwards, an estimation algorithm is applied in order to obtaine the positions of targets in the area.

- **Radio Tomographic Imaging**: In this case, an environmental image is reconstructed from the RSS measurements from every pair of sensors. Each sensor acts as a transmitter and as a receiver. When a target is present in the area under study, it will cause a shadowing effect in the signal transmitted from one sensor to another. With all the sensor measurements, an image representing this effect can be obtained and the position of targets can be estimated by the corresponding image processing [5][21][22][23]. This process is described in detail in later sections as it has been chosen for the multiple target localization and tracking algorithm developed in this work.

2.3 Object Tracking

When the information regarding the location of targets is available, the tracking system must be able to follow the different targets over time in a region of interest. This means that the trajectories of the targets must be estimated. Data Association Problems (DAP) are solved in this case for associating current observations from the sensors with previous acquired data. In target tracking algorithms, the main task is to associate one new location of a target with the previous location corresponding to the same target. In this section, the challenges related to target tracking algorithms and the classification of different tracking approaches are addressed [2][3][4][8].

2.3.1 Tracking Challenges

The following challenges may arise when a WSN is tracking targets:

- **Node failures**: As explained in the WSN overview, nodes can suffer from failures due to harsh environmental conditions or robbery. Thus, the network must be capable to automatically reconfigure itself. Target tracking algorithms must take into account this aspect.
- **Target recovery**: When the target tracking algorithm losses a target due to signal degradations or incorrect data association, the network must be capable to recover the position of targets.
- **Coverage**: If the area under surveillance is not totally covered by the WSN, the targets can be lost, with the resulting performance degradation. When an area is totally covered, the system performance is better.
- **Data redundancy**: In order to manage data redundancy, the tracking algorithm must be able to select the preferred data among all the sensor measurements.
- **Latency**: Tracking algorithms must take into account the time consuming of its execution. For real-time applications, target tracking at each step must not take too long in order to preserve the accuracy of the system.
- **Energy consumption**: Due to energy consumption constraints, the processes in the sensor nodes that are accomplished for target tracking must not be computationally complex and time consuming in order to guarantee a long life of the network.

2.3.2 Tracking Taxonomy

When classifying target tracking algorithms, the following categories can be distinguished:

- **Network structure**: Target tracking algorithms can be classified according to their architecture. Two main architectures can be distinguished: Centralized and distributed. In the centralized one, the data fusion is performed in a fusion center (FC). On the other hand, in a distributed architecture, all the nodes collaborate between them to perform the localization of targets through the network. Then a fusion center may exist or not, depending on the system. In the distributed architecture, three main hierarchical structures can be addressed: tree-based structure, cluster-based structure and face-based structure. In the tree-based structure, the nodes with desired information send it to the processing center through the shortest path in the tree topology. In a cluster-based structure, the network is formed by different groups (or clusters) and the available information is communicated to the cluster head in each group and then transmitted to the processing center. This topology allows collaborative data processing [24]. In the face-based structure, the network is distributed by sections (*i.e.* faces) in which all of them are distributed in a ring topology and the sensors belonging to one face know the features of the other sensors in same face, such as sensor positions or neighbors. In Figure 2.2, different methods are addressed regarding hierarchical classification (distributed fusion).

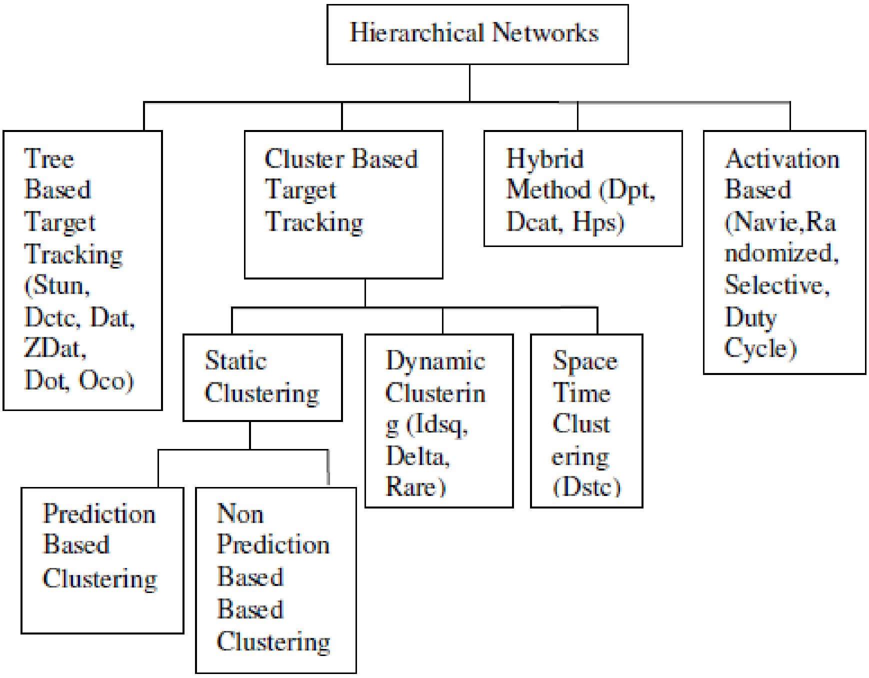


Figure 2.2: Hierarchical network taxonomy [2]

- **Number of Targets**: Target tracking algorithms can be developed for tracking and individual subject or for tracking multiple targets. In real applications, the second option is the most common. When tracking multiple targets, the number of sensors usually increases, thus increasing the power consumption of the network and complicating the tracking algorithms.

- **Type of Targets**: Two types of targets are addressed in the literature regarding their nature: discrete and continuous targets. Examples of discrete targets can be human beings, animals or vehicles, while continuous targets are environmental phenomena such as fires, gases, and so on. Usually, tracking systems are designed for the first type.

- **Prediction**: In order to reduce energy consumption, prediction methods can be used, as the Kalman Filter, for predicting the next position of targets. The sensors from the WSN will be activated just if the algorithm predicts that the target will be close to them. Otherwise, they will remain in sleeping mode.

- **Type of Sensors**: There can be distinguished two different types of sensors: binary or not. If the sensors are binary it means that they will send a decision on whether a target is present or not in the area just with one bit [25]. Otherwise, the sensor will sent other information (*e.g.* measurements, processed data), with more than one bit.

- **Recovery**: In this case, the ability of target recovery when it becomes lost after being tracked is addressed. Tracking systems must handle the situation of a missed target with recovery mechanisms.

The above described classification can be summarized as depicted in Figure 2.3.

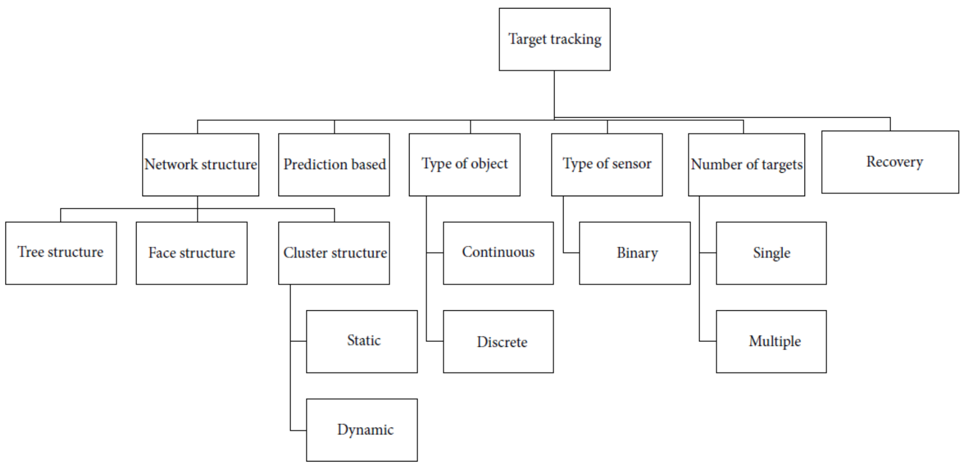


Figure 2.3: Target tracking classification methods [3]

A summary regarding tracking approaches that can be found in the literature are presented in the following figure, extracted from [4].

Structure		Method	Additional Components		
			Prediction	Energy Management	Target Recovery
Tree		DAB – Kung and Vlah [2003]	no	no	no
		DCTC – Zhang and Cao [2004a]	yes	yes	no
		DAT – Lin et al. [2006]	no	no	no
		OCO – Tran and Yang [2006]	no	yes	yes
		DAT+Shortcut – Liu et al. [2008]	no	no	no
		MOT – Sharma et al. [2014]	no	no	no
Cluster	Static	PES – Xu et al. [2004b]	yes	yes	yes
		RARE – Olule et al. [2007]	no	no	no
		Sharma et al. [2011a]	yes	yes	no
		Sharma et al. [2011b]	yes	yes	no
		Kumar and Sivalingam [2012]	yes	yes	no
	Dynamic	Jin et al. [2006]	yes	yes	yes
		Lee et al. [2007]	yes	yes	no
		ADCT – Yang et al. [2007]	yes	yes	yes
		DELTA – Walchli et al. [2007]	no	no	no
		PRECO – Hong et al. [2010]	yes	yes	no
		DMMT – Hamouda and Phillips [2011]	yes	yes	yes
	Hybrid	CODA – Chang et al. [2008]	no	no	no
		HCTT – Wang et al. [2010b]	yes	yes	yes
		TG-COD – Park et al. [2010]	yes	no	no
		HCMTT – Hajiaghajani et al. [2012]	no	yes	no
Face		DOT – Tsai et al. [2007]	no	yes	no
		FOTP – Ji et al. [2009]	yes	yes	yes
		PET – Bhuiyan et al. [2010]	yes	yes	yes
		POOT – Hsu et al. [2012]	yes	yes	yes

Figure 2.4: Comparative table of target tracking algorithms in the literature [4]

2.4 Recursive Estimation Methods

Tracking systems in which the estimated target location and other features are found from noisy measurements, lead to non accurate results. In order to accurately estimate the target features under study (*e.g.* position, velocity, acceleration) in the multiple target tracking process, estimation filters are usually required. The most commonly estimation methods used for target tracking purposes are Kalman Filtering and Particle Filtering. Both estimation methods act in a recursive way, taking into account previous decisions and current observations. In the following, the working principles of Kalman Filter and Particle Filter are briefly described.

2.4.1 Kalman Filtering

Kalman filtering is a recursive estimation filter commonly used in tracking systems. It is an algorithm able to accurately estimate target features, such as position or velocity, if it is properly configured. Using previously acquired information, performs a prediction of the real values of current variables. It assumes that the variable under study is Gaussian distributed. Theoretically, it can be modeled as a recursive Bayesian filter. For a simplified analysis, only the position of the target will be considered for estimation. The position of a target can be modeled as a sequence of states X. The Kalman Filtering model describes the true state at time k as a linear combination of the previous state as follows:

$$x_k = A_k x_{k-1} + w_k \tag{2.1}$$

where A_k is the state transition model that relates the previous state with the current state and w_k is the process noise, which is modeled according to a normal distribution with zero mean and covariance Q_k, as $w_k \sim \mathcal{N}(0, Q_k)$. On the other hand, the relationship between an observation at time k and the current state is described as follows:

$$z_k = H_k x_k + v_k \tag{2.2}$$

where H_k is the observation model which relates the current measurement and the actual state and v_k is the observation noise, which is modeled according to a normal distribution with zero mean and covariance R_k, as $v_k \sim \mathcal{N}(0, R_k)$. The process noise and the observation noise are assumed independent.

Given the abovementioned relations, the objective is the estimation of the current state x_k (*i.e.* the current position of a target). The Kalman Filter solves the problem in two steps: state prediction and state correction. The first step computes a prediction on the state estimate and the noise covariance as follows:

$$\hat{x}_k^p = A_k \cdot \hat{x}_{k-1} \tag{2.3}$$

$$\hat{P}_k^p = A_k \cdot \hat{P}_{k-1} \cdot A_k^T + Q_k \tag{2.4}$$

where $\hat{x}_k^p$ addresses a first approximation to the real state, A_k is the state transition in time k and $\hat{x}_{k-1}$ is the previous state estimation. On the other hand, $\hat{P}_k^p$ describes a prediction on the estimated covariance, $\hat{P}_{k-1}$ is the previous estimated covariance and Q_k is the process noise covariance.

The second step consists on estimating the current state with the predicted state and the current observation. The predicted step is updated with real measurements. The covariance P_k is also updated in order to use it for the following estimations. In this case, the updated states are expressed as follows:

$$\hat{x}_k = \hat{x}_k^p + K_k \cdot (z_k - H_k \cdot \hat{x}_k^p) \tag{2.5}$$

$$\hat{P}_k = (I - K_k \cdot H_k) \cdot \hat{P}_k^p \tag{2.6}$$

where K_k is the Kalman Gain, represented by:

$$K_k = \hat{P}_k^p \cdot H_k^T \cdot (H_k \cdot \hat{P}_k^p \cdot H_k^T + R_k)^{-1} \tag{2.7}$$

In the tracking process, this recursive filter is applied in every step, with every new measure, after the association process is accomplished. Kalman Filter can be deeply customized in order to take into account specific system characteristics. For example, an Extended Kalman Filter can be used if the system under study is a nonlinear system.

2.4.2 Particle Filtering

Particle filters, also known as Sequential Monte Carlo (SMC) methods, consist on non-linear and non-Gaussian genetic filters for recursive estimation. In this case, the state variables are not assumed to be distributed according to a Gaussian distribution. A particle filter uses discrete particles to estimate true states of variables. In tracking systems, the variable under study is the position of targets in a region of interest [26]. Each particle represents a possible state (*i.e.* possible position of the target). Particles are distributed along the state space (*i.e.* covered area) according to a given mean and covariance around one state. If a state is not available, there must be specified state space boundaries (*i.e.* limits of the area). When the number of particles increases, more processing time is required but the result is more accurate. State parameters can be specified, such as motion modeling, state transition, initial particle position and state space boundaries or sensing model.

As for the Kalman Filter, the estimation process is divided into two steps: state prediction and state correction. In the prediction stage, the Particle Filter algorithm proposes an estimate state using previous state estimations and a transition model. Afterwards, in the correction stage, the algorithm updates the state estimation using measure data in the current moment. All the particles at one time are weighted and used in the estimation process. Then, particles are redistributed in the state space in order to match the successive estimations.

The tracking of targets can be modeled as a set of states $\mathcal{X}$ (*i.e.* positions). In time k, the state X_k and the corresponding observation Z_k are assumed as follows:

$$X_k \mid X_{k-1} = x_k \sim p(x_k \mid x_{k-1}) \tag{2.8}$$

$$Z_k \mid X_k = z_k \sim p(z_k \mid x_k) \tag{2.9}$$

where $p(x_k \mid x_{k-1})$ is known as the transition probability densitiy and $p(z_k \mid x_k)$ is the conditional probability density function of an observation given X_k. A traditional system for estimation can be modeled as follows:

$$X_k = g(X_{k-1}) + W_k \tag{2.10}$$

$$Z_k = h(X_k) + V_k \tag{2.11}$$

where g is a transition function, h is an observation function, W_k is the process noise and V_k is the observation noise. If both noises are Gaussian-distributed, the solution of Particle filtering will be exactly the same as for the Kalman Filter. The corresponding formulas for both prediction and correction stages can be found in the literature [18]. As in the Kalman Filter case, modifications can be carried out in order to adapt the state estimation problem to a specific system.

Chapter 3

Algorithm Development

3.1 Introduction

In this section, the design and development of a multiple target localization and tracking algorithm implemented for a wireless sensor network is presented. First, both the area where the sensors will be deployed and the propagation and target motion model are analyzed. Then, in order to test if the algorithm is working properly in the last stage (i.e. when all the data has been collected by the sensors and sent to the fusion center), the trajectories of multiple targets have been generated randomly in the corresponding area of study, including their influence in the environment. Afterwards, the characteristics of the deployed sensors and possible topologies are presented. In addition, in order to model a realistic scenario, before sending the acquired data from the sensors to the fusion center, the corresponding values are quantized and the channel errors are taken into account.

In the fusion center, the developed algorithm is applied to the gathered data. The algorithm is outlined step by step. It mainly consists of an estimation of a tomography image of the covered area through time and the corresponding digital processing, including thresholding and clustering to locate multiple targets in the area. Then, the tracking stage is accomplished as a data-association problem, where hypothesis and observations are matched under particular conditions, explained in the corresponding subsection. Finally, some estimation methods are applied to the tracking results in order to compensate possible position deviations to achieve more accurate outputs regarding the actual positions of the targets.

3.2 Scenario

The system is designed to work most effectively in open environments. Particularly, it has been designed for the purpose of locating and tracking people in public spaces where they can pass through freely and stay in the area or leave it as desired. Evidently, the system must take into account the human rights and guarantee the privacy and other rights of persons with respect to their identities, such as data protection. In order to avoid privacy issues, the considered sensors units that would be deployed in the area would be different than audio or video sensors. Indeed, the developed tracking algorithm has been designed for RF sensors, which measure the signal strength of radio waves in their surroundings and are unable of reconstructing the shape of human beings and, thus, clearly incapable of recognizing their identity.

The main purpose of the tracking system under study is to be implemented for digital story-telling applications. Apart from the WSN, other devices may be deployed in the area, such as spotlights, screens or other audiovisual devices required for the specific application. Depending on the positions of the targets, the technological infrastructure deployed in the area with the purpose of digital story-telling will change the appearance of the area giving the users a new virtual experience. The design and development of the digital story-telling features fall outside the scope of this thesis.

The selected geographic area of study is located in the city of Trondheim, in Norway. It is known as Adressaparken (Figure 3.1) and it is meant to be an outdoor laboratory for researchers and students. It is a joint cooperative project between Trondheim municipality, NTNU (Norwegian University of Science and Technology) and Adresseavisen, which is the main newspaper company in Trondheim. Some technological infrastructure has already been deployed, such as screen projectors, LED light strips, temperature sensors and LIDAR sensors.

Figure 3.1: Technological infrastructure and geographical location of Adressaparken in Trondheim (Norway)

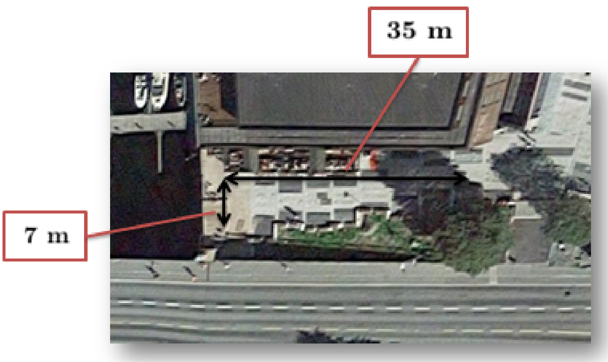


Figure 3.2: Dimensions of the covered area

Regarding technical aspects about the selected area, the dimensions of the area that have to be covered are approximately 7 meters in width by 35 meters in length, as it can be appreciated in Figure 3.2. The covered area is not uniform in the edges but in order to simplify the problem, the area will be treated as a rectangle. The problem can be easily extrapolated to a non-uniform area by locating the sensors in the edges of interest and identifying if a target is out of bounds or not.

On the other hand, it seems that the terrain has no obstacles in between, such as trees or columns, but apparently there are some corners in the benches and in the front part of the restaurant that should be taken into account when locating the sensors so that they can have direct line of sight. If not possible, the system performance could be affected by the non-line of sight. In addition, there are some buildings and other obstacles around the area that can also affect the radio signals by the reflection of the radio waves in their surfaces, which is known as multipath propagation.

3.3 Propagation Model

A wireless sensor network can be composed of several types of sensors. In this case, the focus of the study for the development of an algorithm for locating and tracking targets concerns only radio frequency (RF) sensors. In future works, the integration of other types of sensors such as LIDAR or temperature sensors can be convenient in order to improve the system performance, but the algorithm will require the corresponding modifications.

In this case, each RF sensor sends a signal through the radio channel and the rest of sensors measure the receive signal strength (RSS) with its intrinsic degradation due to the imperfections of the radio channel. Therefore, in this section we focus on the propagation of radio waves through open environments and how people can affect their behavior.

3.3.1 Radio Waves Propagation

When radio waves travel in open environments they can be affected by several factors as reflection in surfaces, attenuation, absorption in the atmosphere, refraction, diffraction, scattering, changes in polarization, and so on. The abovementioned aspects must be taken into account when analyzing a received radio signal.

The equation that represents the received signal strength of a radio signal travelling through open space with line-of-sight is defined as follows (in *dBm*):

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{FS} - L_M + G_{RX} - L_{RX} \tag{3.1}$$

where $\mathrm{P_{TX}}$ and $\mathrm{P_{RX}}$ are the transmitted and received signal powers, $\mathrm{G_{TX}}$ and $\mathrm{G_{RX}}$ are the corresponding antenna gains both in transmitter and receiver, $\mathrm{L_{TX}}$ and $\mathrm{L_{RX}}$ are the transmitter and receiver losses (*i.e.* impedance mismatches due to imperfections in connectors), $\mathrm{L_{FS}}$ is the free space loss (Equation 3.2) where λ is the wavelength and R is the distance between the antennas, and $\mathrm{L_M}$ encompasses the rest of losses such as polarization mismatch, absorption, fading margin, and so on.

$$L_{FS} = \left(\frac{4\pi R}{\lambda}\right)^2 \tag{3.2}$$

In real environments, when obstacles are present between transmitter and receiver (*e.g.* present targets), the radio signal experiences also multipath propagation and the line of sight becomes blocked. This effect causes random fluctuations in the signal amplitude, which can be modeled according to a Rayleigh distribution. Doppler Effect can also appear by the presence of moving objects. In addition, the signal is affected by noise that is typically modeled as AWGN.

On the other hand, obstacles can provoke an additional path loss and shadow fading. A variety of deterministic and experimental models [27] [28] [29] [30] have arisen to address this effect depending on the environment (*e.g.* indoor, outdoor, cluttered, etc.). In a simple form, the attenuation of a signal can be model with an exponential distribution (Equation 3.3), a power law function (Equation 3.4) or a Gaussian distribution (Equation 3.5), where x is the distance between the transmitter and the receiver. The relation between the distance of the power source and the receiving antenna and the emitted power regarding the abovementioned models can be appreciated in Figure 3.3.

$$\alpha_{Exponential} = e^{\lambda x} \tag{3.3}$$

$$\alpha_{Power-Law} = \frac{1}{\sqrt{1 + (\lambda x)^{\alpha}}} \tag{3.4}$$

$$\alpha_{Gaussian} = e^{-\dfrac{(x-\mu)^2}{2\sigma^2}} \tag{3.5}$$

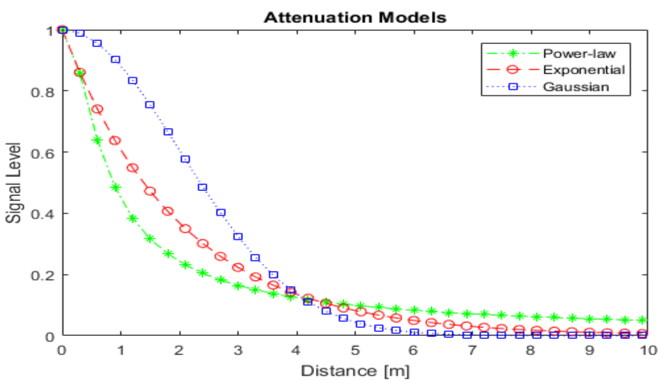


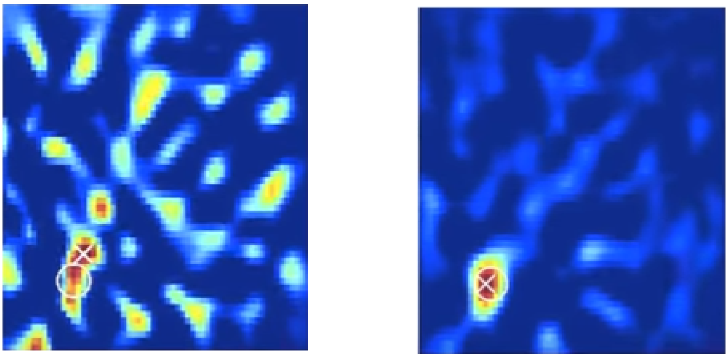
Figure 3.3: Example of Attenuation Models in Radio Propagation

3.3.2 Fade Level

The received signal strength (RSS) by one sensor when a target is present in the surveillance area will be highly affected by an effect known as multipath fading. In [31] the fading effect is characterized by a fade level and it is shown that the RSS during human motion is strongly dependent on the RSS fade level measured when the human being is standing. The fade level depends both in the frequency and in the positions or distance between the communicating devices. It is a measurable parameter by the receiving devices and, as it affects strongly to the RSS, it is considered a key parameter for the system performance.

Two different scenarios can be considered as the extremes of fade level: anti-fade and deep-fade. When a link between two sensors has a deep-fade level is because it has destructive multipath interference. If suddenly a target is present obstructing the link, the average of the RSS increases. It has been also demonstrated that if a target is far away from the link, the RSS also changes. On the other hand, in the case of an anti-fade link, where constructive multipath interference is dominant, the average RSS decreases and, even if the target is far away, the RSS does not present changes. Thus, it is an important task to identify the fade level of each link and weight it according to its contributions.

In [32] it is stated that channel diversity can make a real difference in the localization accuracy of targets even in cluttered environments, as can be seen in Figure 3.4. Therefore, if the receiving devices are able to distinguish between frequency channels, it is interesting to measure the fade level in all the channels of every link.



a) b)

Figure 3.4: Comparison of Radio Tomography Images (RTI) using Fade Level method for a moving target in the same area with and without channel diversity, a) without channel diversity and b) with channel diversity.

The fade level in a channel of one link between two devices can be described by the following equation (in dBm):

$$F_{l_i,c_j} = P_{R,l_i,c_j} - P_{T,c_j} + L_{l_i,c_j} + w_{l_i,c_j} \tag{3.6}$$

where P_{R,l_i,c_j} is the received power by the sensor in the link i on channel j, P_{T,c_j} is the transmitted power on channel j, L_{l_i,c_j} addresses the losses in the channel and w_{l_i,c_j} is the measurement noise due to the imperfections in the sensing device.

The losses and the noise are not known so the fade level cannot be calculated directly. Hence, an estimation of the fade level in each case can be accomplished when the surveillance area is empty. In this case, the average RSS is measure in each channel (or link if channel diversity is not used) during an intended calibration period. Then, for all the channels of one link, the minimum average is taken as reference to define the level of fading.

The fade level is then estimated as follows:

$$\hat{F}_{l_i,c_j} = \bar{P}_{R,l_i,c_j} - \underset{c}{argmin} \bar{P}_{R,l_i,c_j} \tag{3.7}$$

The average change in the received signal strength provoked in each link is considered as the input data to the developed algorithm. This parameter can be calculated in one instant as the weighted average of the difference between the RSS at that instant and the average RSS measured in the calibration period. The following equation expresses the RSS change for link l at instant k:

$$y_{l_i}(k) = \frac{1}{\sum_c F_{l_i,c_j}} \sum_c F_{l_i,c_j} \cdot |P_{R,l_i,c_j}(k) - \bar{P}_{R,l_i,c_j}| \tag{3.8}$$

3.3.3 Target Radiation Model

Due to the lack of a physical system that collects real data, the change of the RSS in the surveillance area has been simulated. First, before simulating the radiation caused by the targets, random paths are generated within the area.

The following parameters can be configured:

Parameter	Description
`users`	Number of users
`dimensions`	Dimensions of the surface in meters [m]
`total_positions`	Maximum number of positions
`step_distance`	Distance of an step of a person in meters [m]
`mean_distance`	Average straight distance of a target

Table 3.1: Configurable parameters for target paths

It is set that targets can enter randomly along the four edges of the area in any time. The selected entrance gate and instant of one target is chosen following a uniform distribution. When moving across the area, targets can change their direction and can either be moving or standing in the same point for a while and start walking again. This is modeled with a uniform distribution, by changing a direction vector.

Finally, if a target cross any of the area boundaries, it is considered gone. In Figure 3.5 it can be seen an example of the targets paths inside an area. For simplicity, all the simulations have been done with a squared area.

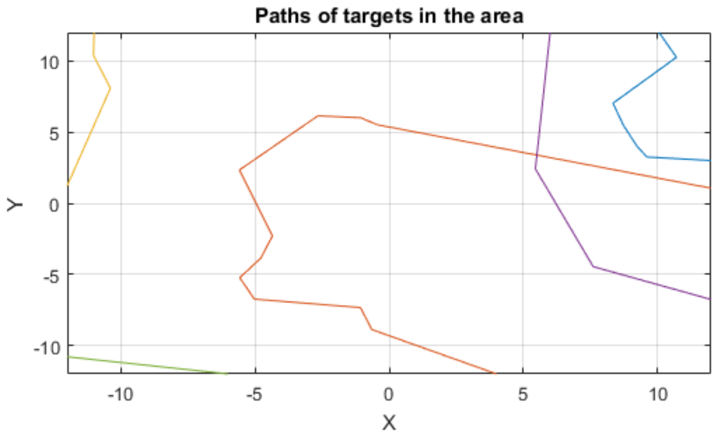


Figure 3.5: Random generated paths in the area

After generating the paths in the area, the area is discretized into voxels (or pixels) and the radiation caused by each target is modeled having a different value in every voxel within the surface. The attenuation is modeled as a Gaussian function with respect to the distance between one target and the sensors as it has been described in the previous section.

In order to model a more realistic environment, two different noises are considered. These are the measurement noise, modeled with an exponential distribution, and the background AWGN (Additive White Gaussian Noise) following a Gaussian distribution with a configurable standard deviation. Furthermore, multipath fading is also modeled according to a Rayleigh distribution as the radial component of the complex sum of two uncorrelated Gaussian random variables. Figure 3.6 shows the generated radiation by several targets when passing through the area.

In this case, the following parameters can be configured:

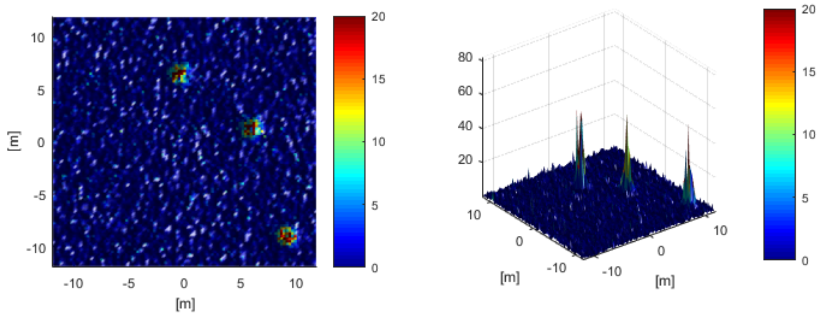


Figure 3.6: Generated radiation of targets in the area

Parameter	Description
`users_path`	Array with the positions of all the targets
`precision`	Meters per voxel
`calibration_steps`	Number of instants for the calibration period
`radiation_amplitude`	Radiation in the position of the target
`target_width`	2-D Array with the width of the target [x y]
`rotation`	Angle of rotation of the target radiation
`noiselevel`	AWGN standard deviation
`radius`	2-D Array with the standard deviation of the Gaussian function $[r_x \; r_y]$

Table 3.2: Configurable parameters for target radiation

3.4 Sensors Characteristics

After simulating the radiation present in the environment when targets are moving through the area, the next step is the deployment of the sensors and the acquisition of data to work with. As previously mentioned, the sensors will be radio-frequency sensors deployed in the surveillance area. Each sensor will received signals from all the rest, identifying them with the corresponding protocol (*e.g.* IDs, time-spacing). Every two sensors will form a link, as shown in Figure 3.7.

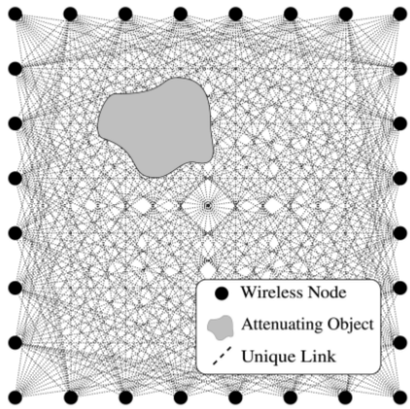


Figure 3.7: Links between RF sensors in a surveillance area [5]

The total number of links can be calculated as follows:

$$N_{links} = \frac{K_{sensors}^2 - K_{sensors}}{2} \tag{3.9}$$

In this case, as the fade level acquisition is a tedious task to perform due to the lack of real sensor measurements and a real environment, the change in RSS can be modeled as a linear combination of the radiation previously simulated in each voxel as follows:

$$y_{l_i}(k) = \sum_{j}^{V} w_{l_i j} x_j + n_{l_i} \tag{3.10}$$

where x_j is the generated amplitude in voxel j, n_{l_i} is the noise of the link i and $w_{l_i j}$ is the corresponding weighting of voxel j for the link i under consideration. Each voxel will affect each link in a different way, depending on the positions of the sensors. In order to calculate the contribution of each voxel, we consider the ellipse model [21][22][5][33] where the sensors of one link are located in the focal points of the ellipse, as can be seen in Figure 3.8.

In the majority of related articles, the weighting is set to $w_{l_i j} = 1$ if the voxel j is

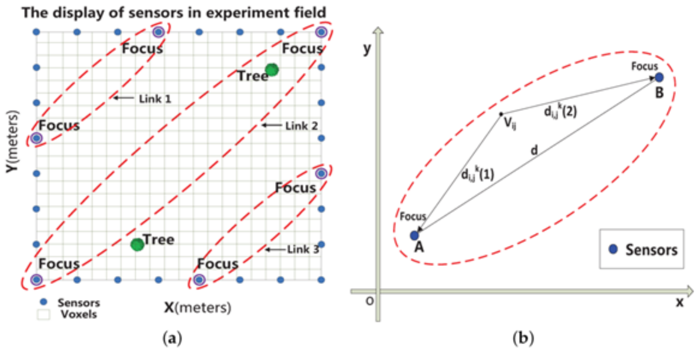


Figure 3.8: Ellipse model for sensor links [6] a) Example of sensor deployment and b) Ellipse model in one link.

positioned inside the ellipse and $w_{l_ij} = 0$ if not. In [6] they propose a new elliptical model (2016) where the weight is set according to the position of the voxel inside the ellipse, taking into account also the line of sight, and also the distance between the voxel and the sensors. Mathematically,

$$w_{l_ij} = \begin{cases} \dfrac{1}{d_{A,B}}(k_1 + max(d_{A,j}, d_{B,j})) & \text{if } d_{A,j} + d_{B,j} < d_{A,B} + \lambda, d_{A,j} + d_{B,j} \neq d_{A,B} \\ \dfrac{1}{d_{A,B}}(k_2 + max(d_{A,j}, d_{B,j})) & \text{if } d_{A,j} + d_{B,j} < d_{A,B} + \lambda, d_{A,j} + d_{B,j} = d_{A,B} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where k_1 is a coefficient that takes into account if a target or obstacle is in a NLOS path, which is normally set to $k_1 = 2$ by empirical experiments [6], k_2 is a coefficient that represents the LOS path, which is normally set to $k_1 = 2.5$, $d_{A,B}$ is the distances between sensors A and B, $d_{A,j}$ and $d_{B,j}$ are the distances between the voxel j and sensors A and B, respectively, and λ is a parameter that measures the ellipse eccentricity.

In Figures 3.9 and 3.10 are represented two examples with 2 sensors and 40 sensors. It can be appreciated the area covered by each link in both cases. With a few simulations, it is interesting to see the coverage of each link and of the whole system in order to select an appropiate network topology and to realize if there are any ghost points in the region of interest.

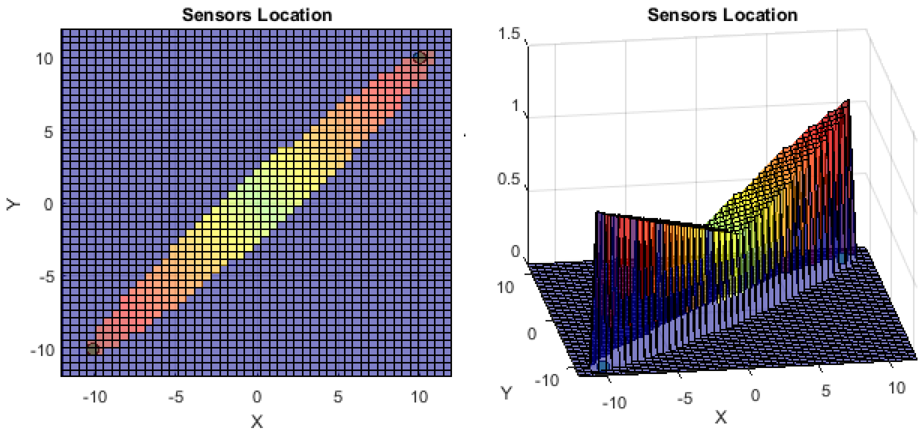


Figure 3.9: Coverage of 2 sensors with ellipse model

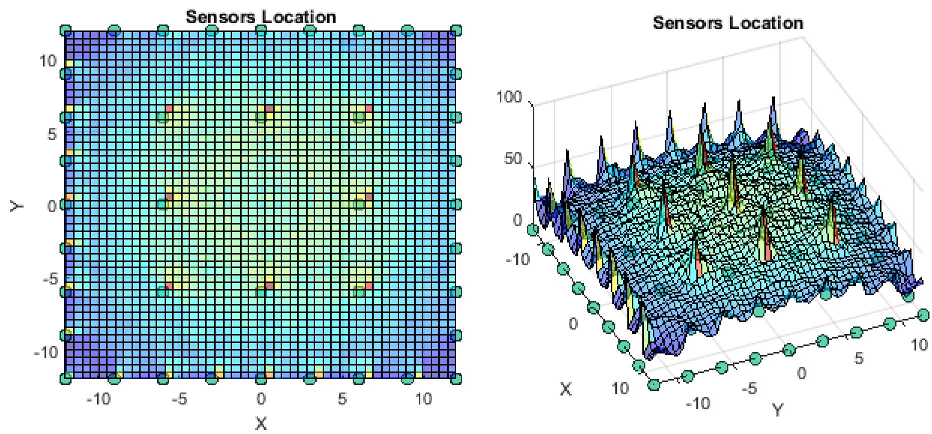


Figure 3.10: Coverage of 40 sensors with ellipse model

In this case, the following parameters can be configured:

Parameter	Description
`sensor_x`	X-coordinate of the sensor location
`sensor_y`	Y-coordinate of the sensor location
`lambda_ellipse`	Ellipse excess path

Table 3.3: Configurable parameters for sensor network

The RF sensor technology available in the current market is extensive and many solutions can be considered for proposed system. Usually, RF modules for wireless sensor networks operate in the 2.4 GHz ISM band, which suits with the IEEE 802.15.4 standard for low-rate wireless personal area networks (WPAN). Zigbee is an standard based on the IEEE 802.15.4 for low-cost low-powered RF modules for WPAN better than Bluetooth or Wi-Fi. It defines high-level communication protocols suitable for wide wireless sensor networks (WSNs). After some research, the following RF modules are proposed:

- **CC2500**: Low Cost, Low-Power 2.4 GHz RF Transceiver Designed for Low-Power Wireless Apps in the 2.4 GHz ISM B.
- **CC2520**: Second generation 2.4 GHz ZigBee/IEEE 802.15.4 RF transceiver.
- **CC2531**: System-on-Chip Solution for IEEE 802.15.4 and ZigBee Applications.
- **NRF24L01+**: Ultra low power 2.4 GHz RF Transceiver IC.
- **MC13202**: 2.4 GHz RF transceiver forIEEE 802.15.4.

3.5 Data Quantization

Before sending to the fusion center the acquired data of each link, a quantization stage is introduced in order to reduce the amount of information from each sensor due to power-consumption restrictions. In this analog-to-digital stage, the number of quantization levels is configurable, where 2 to 32 bits can be used for describing each signal value. After some simulations, it has been seen that with less than 5 bits (*i.e.* 32 quantization levels) the performance is highly degraded and with less than 3 bits (*i.e.* 8 quantization levels), the system is not able to distinguish any targets in the surveillance area.

3.6 Channel Errors

Here we consider the imperfections of the radio channel, where the radio signals can suffer from degradation and the information sent from the sensors to the fusion center can be received with some errors. In [13], error-prone channels are also considered for target detection. These errors introduced by the channel can be modeled with a configurable BER (Bit-Error Rate). After the quantization process, where the information is in form of bits, some of the bits change their value depending on the selected channel error probability. The bits that change are selected randomly according to a uniform distribution. The number of erroneous bits is calculated as the product of the channel BER and the number of bits that are transmitted in the signal, as stated in Equation 3.12.

$$N_{errorbits} = BER \cdot N_{totalbits} \tag{3.12}$$

3.7 Data Fusion

In the fusion center, the information received by the sensors is processed in real time to be able to perform the corresponding digital story telling event depending on the position of the targets in the area. Here the algorithms for locating and tracking are applied. With the information of the received signal strengths in each link, a radio tomographic image can be reconstructed and analyze as it will help to localize targets considerably accurately, even if there is more than one target present in the area. However, it has been seen that when the number of targets increases, the system performance is degraded and sometimes it is not even possible to distinguish between targets and tracks can be lost or mismatched.

The developed algorithm consists on the reconstruction of a radio tomographic image. In order to do so, first image estimation is necessary. Then some mechanisms of image processing are required such as image filtering, in order to remove non desired parts, and cluster identification, in order to distinguish between different targets. Finally, according to the obtained information, an association algorithm for the observed data must be implemented so that the tracking can be performed. These tasks are explained in the following sections.

3.8 Image Estimation

Image reconstruction from the available information of a set of sensors has become an important field of study in areas such as medicine, astronomy or military defense. When performing Radio Tomography Imaging (RTI), image estimation becomes a challenging task [34]. Different propagation fields can lead to the same measurement data so the solution is not unique and, thus, one reconstructed image can correspond to different real environments.

In our system, the RSS change caused in each link has been modeled with the following expression:

$$y_{l_i}(k) = \sum_{j}^{V} w_{l_i j} x_j + n_{l_i} \tag{3.13}$$

which can be understand as a spatial integral of the propagation fields in the area, and is described as a linear combination of the RSS change caused in each discretized part of the area (*i.e.* each voxel).

In a general form, the available data at the fusion center can be represented as:

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{n} \tag{3.14}$$

where $\boldsymbol{W}$ is a MxN matrix representing the weighting of every voxel (*i.e.* N voxels) for every link (*i.e.* M links) and it can be seen as a transfer matrix of the $\boldsymbol{x}$ parameters. Thus, the reconstructed image will be created by estimating the RSS in every voxel of the map (*i.e.* $\boldsymbol{x}$-vector). In order to do so, the first approach is to find the optimal solution for a least-squared problem, which can be expressed as follows:

$$\boldsymbol{x_{LS}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{y} \tag{3.15}$$

However, this only works if the $\boldsymbol{W}$ is full-rank, which is not the case for radio tomography imaging. The noise of measurement data leads to a non-unique solution of the least-squared approximation. Thus, regularization methods [35] have to be applied, introducing further constraints that can determine a unique solution.

In this case, a regularized least-squares approach is used [21]:

$$\hat{\boldsymbol{x}}_{\boldsymbol{RLS}} = (\boldsymbol{W}^T\boldsymbol{W} + \boldsymbol{C}_{\boldsymbol{x}}^{-1}\sigma_N^2)^{-1}\boldsymbol{W}^T\boldsymbol{y} \tag{3.16}$$

where σ_n is the standard deviation of the measurement noise and $\boldsymbol{C}_{\boldsymbol{x}}$ is the covariance matrix of the voxels. $\boldsymbol{C}_{\boldsymbol{x}}$ is defined by σ_x, which is the standard deviation of the voxel measurements and an exponential decay given by the distance $d_{i,j}$ and the correlation distance δ of between every voxel (Equation 3.17). The covariance matrix is computationally intensive, but it only has to be calculated once, so it fits well with real-time tracking requirements.

$$\boldsymbol{C}_{\boldsymbol{x}} = \sigma_x^2 e^{-d_{i,j}/\delta} \tag{3.17}$$

In this stage, the following parameters can be configured:

Parameter	Description
`sensors_data`	RSS change data transmitted from sensors to FC
`link_weights`	Weighting of the voxels for each link
`sigma_n`	Noise deviation
`sigma_x`	Voxels measurement deviation
`delta_c`	Voxels correlation distance

Table 3.4: Configurable parameters for image estimation

An example of image estimation compared with the original environment (simulated) using the regularized least-squares method can be appreciated in Figure 3.11.

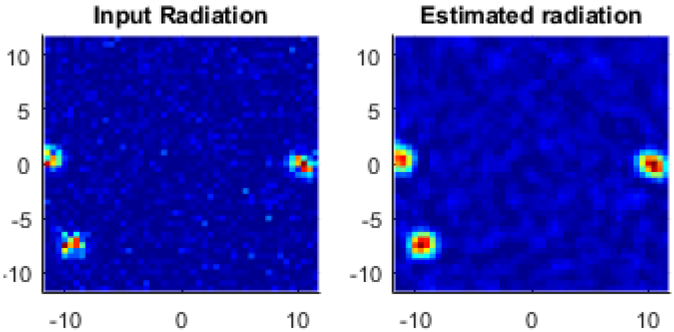


Figure 3.11: Comparison between RSS change in the area and estimated image

3.9 Thresholding

Once an image representing the RSS change in the area is obtained, it is convenient to apply an image filter in order to remove non-desired parts which appear often by the presence of noise or spurious blobs. With the simulated radiation the estimated image results very clear and a blob that emerges by the presence of a target is clearly recognizable. However, measures acquired in real environments can be really noisy and spurious blobs can appear even when the area is empty as a result of constructive interference of the emitted radio waves. Even so, applying an image filter in our system will help to reduce the image noise and to improve the performance of the tracking algorithm.

In this process, we assume that the higher blobs correspond to the presence of targets so the image filter consists on applying a threshold to the RSS change values in every voxel of the discretized image and removing all the values below the selected threshold by setting its value to $x_{i,j} = 0$.

The thresholding process consists in two different steps. First, an initial threshold is chosen taking into account the measurements taken in the calibration period (*i.e.* when the area is empty). Here, the RSS change caused by the absence of targets is measured and then the intensity average is calculated. Afterwards, when targets can be present in the area, the minimum RSS change is measured and a recursive average is calculated at each step (*i.e.* every time a new measure is acquired), following the Equation 3.18.

$$I_{LPF}[n] = \alpha I_{LPF}[n-1] + (1-\alpha) I_{min}[n] \tag{3.18}$$

A flow diagram representing the minimum intensity low-pass filtering is shown in Figure 3.12.

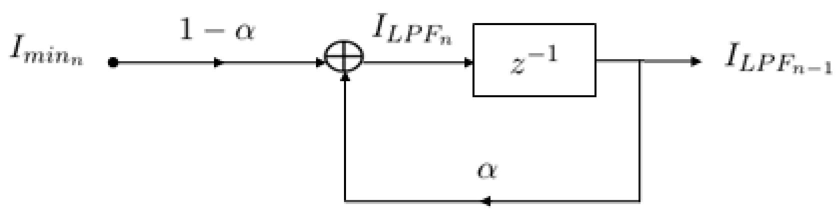


Figure 3.12: Low-pass filtering of minimum intensity flow diagram for image filtering

The intensity threshold is then set such that, in every step, when no target has been detected yet, the threshold is two times the average intensity $\bar{I}_c$ measured in the calibration period and, when targets are being track, the threshold is switched to a weighted low-pass filtered intensity $I_{LPF}(k)$ (Equation 3.19) In this expression, $\mathbb{T}$ represents the set of targets in the area and β is the weighting parameter for the low-pass filtered intensity.

$$T(k) = \begin{cases} \beta I_{LPF}(k) & \text{if } \exists t \in \mathcal{T} \\ 2\bar{I}_c & \text{otherwise} \end{cases} \tag{3.19}$$

It can be appreciated in Figure 3.13 the difference between an estimated image (described in the previous section) and the same image after the filtering process.

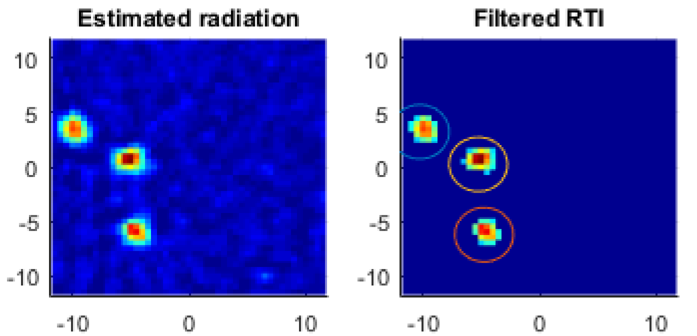


Figure 3.13: Comparison between estimated RTI image and filtered image

In this case, the following parameters can be configured:

Parameter	Description
`rss_change_estimate`	Image estimation of the RSS change in the area
`alfa`	Weighting parameter for Intensity Low-Pass Filtering
`beta`	Weighting parameter for dynamic thresholding

Table 3.5: Configurable parameters for image thresholding

3.10 Clustering

After the filtering process, the resulting image has enough information to distinguish the shape of targets in the surveillance area. We assume that the shape of a RSS change blob caused by the presence of a human being is almost cylindrical [5][32]. Thus, the remaining voxels with values different than zero are analyzed and a clustering algorithm is applied in order to group the voxels that can correspond to the same target. In this way, the algorithm is able to distinguish between different targets, achieving the purpose of tracking multiple targets at the same time in the region of interest.

Although there are a wide variety of clustering algorithms [36], the one chosen for this case is a hierarchical clustering algorithm. This algorithm is very suitable for the multiple target tracking system because there is no a priori assumption on the number of targets that will be present in the area while the tracking is performed. The hierarchical clustering is based on the disparities of the observations, separating those that are not compatible with the settled measure of dissimilarity. For the tracking purpose, the chosen dissimilarity has been the average linkage distance between voxels. Firstly, all the remaining voxels will be considered as separated clusters. Then, the distances between all the clusters will be computed (Equation 3.20). If the distance between two voxels is below a selected threshold, the voxels will be merged forming a bigger cluster. On the other hand, if the distance is beyond the threshold, the voxels will remain in separated clusters. It is proposed that the distance that can separated two different human beings can be around 2 meters.

$$d_{\mathcal{A},\mathcal{B}} = \frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y) \tag{3.20}$$

where x and y represent two different voxels and $\mathcal{A}$ and $\mathcal{B}$ represent two different subsets (or clusters). In this case, the following parameters can be configured:

Parameter	Description
`filtered_image`	Filtered radio tomography image
`cluster_width`	Diameter of the clusters in meters [m]

Table 3.6: Configurable parameters for data clustering

In Figure 3.14 it is shown the result of applying the clustering algorithm to an estimated image and the corresponding clusters. Then, in Figure 3.15 it is represented the dendrogram of the clustering process of Figure 3.14, where there can be appreciated the formation of different clusters.

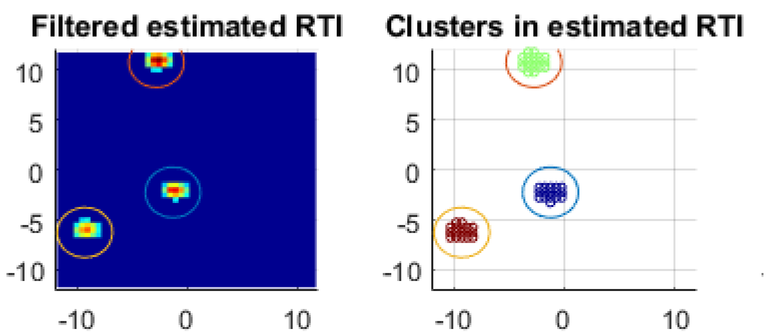


Figure 3.14: Example of a filtered image and the corresponding clusters

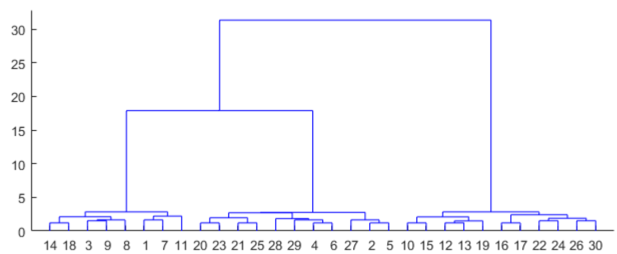


Figure 3.15: Dendrogram after applying the Hierarchical clustering algorithm

3.11 Multi-Target Tracking Algorithm

Once the data is grouped in clusters and spatial information over time is available, the target tracking algorithm can be applied. The tracking algorithm has to lead with issues such as intersecting trajectories of different targets, disappearing targets (*e.g.* due to the merging of two different clusters representing two different targets or because the blob representing a target disappears suddenly), spurious blobs, and so on. In order to do so, the first step consists on selecting the cluster head h_c of each formed cluster, which corresponds to the voxel in the cluster with the highest amplitude value.

$$h_c = arg \max_{c \in \mathcal{C}} \hat{x}_c \tag{3.21}$$

Then, for the tracking purpose, a settled number of previous cluster heads are stored in memory to compare them with the current observations. One of the target tracking tasks consists on identifying the same target during time. Thus, in the algorithm, each cluster head is assigned an ID number. One observation (*i.e.* one cluster) is compared with the hypotheses made for previous instants. If the voxels distance between one observation and a previous hypothesis is below an established length, this new observation is assigned to the same hypothesis (*i.e.* the same ID). Two observations at the current instant cannot be assigned to the same ID. This can be modeled as a circumference with a specified radius r_c around a new observation and if a previous assignation drops inside the circled area, the same ID is assigned.

$$\left\| \hat{x}^{(k)} - h_c^{(k-n)} \right\| < r_c \tag{3.22}$$

The tracking process can be appreciated in Figure 3.16 where the smallest circles represent current observations and crosses represent cluster heads in the previous instant.

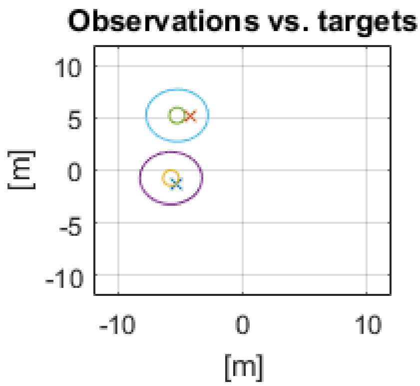


Figure 3.16: Comparison between current observations and previous target hypotheses

In order to identify a real target within a set of clusters that may correspond to real targets or non-desired spurious blobs, we define three different categories for classifying the observations:

- **Candidate**, when one ID has been assigned less than N times.
- **Confirmed**, when N of Nmax observations have the same ID.
- **Deleted**, when one ID has not been assigned for M times after being assigned before.

Some restrictions are specified before considering a target *candidate* to avoid assigning IDs to cluster heads that appear suddenly in a zone of the area impossible to reach by a human being between one measure and the next one. Thus, an entrance and exit regions are considered with this purpose. These regions are defined in the boundaries of the surveillance area, where targets can firstly appear. The amplitude of these regions can be adjustable for the desired application.

In this case, the range of the entrance/exit regions is set to 2 meters as this is approximately the width of the radiation caused by the presence of a target in the area.

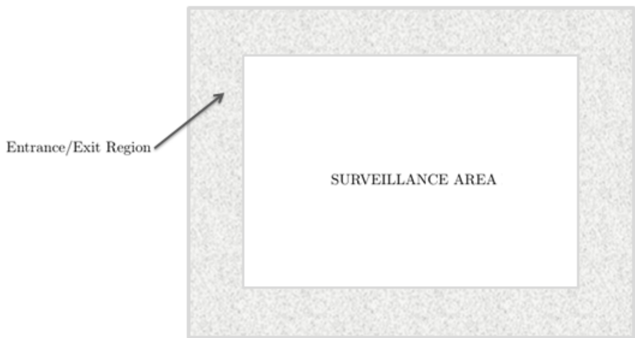


Figure 3.17: Entrance/Exit region in the region of interest

Thus, if a blob appears in this region, it will be considered as a *candidate*. When a *candidate* has been assigned the same ID certain number of times, it will then be considered as a *confirmed* target. In case that a *candidate* or a *confirmed* track is not assigned for a while, it will be deleted from the memory. Indeed, if the same real person leaves the area and enters again, it will be identified with a different ID. On the other hand, if a cluster is found outside the entrance/exit region (*i.e.* in the center part of the area), and it is far away from the position of previous assignations so that it cannot be identified, it is considered as a spurious blob and is discarded.

When two targets are closed to each other, the clusters representing both targets can merge forming an individual cluster. In order to handle intersecting trajectories of several targets, the radius around each observation is increased so that if one blob disappears suddenly and reappears again it can be reassigned to an existing hypothesis, as it can be appreciated in Figure 3.18.

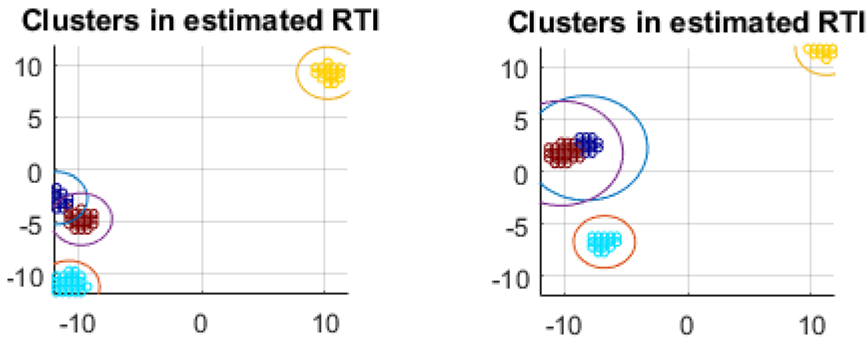


Figure 3.18: Change in the radius around a target when targets are close

In this case, the following parameters can be configured:

Parameter	Description
`clusters`	Clusters observations in current instant
`targets`	Hypothesis of the tracks in previous steps
`Re_width`	Width of the Entrance/Exit Region
`target_radius`	Radius around a target
`n_memory`	Number of previous instants for data association of new observations
`n_confirm`	Number of previous targets with same ID for being a confirmed target
`n_max_confirm`	Maximum previous instants for confirming a target
`n_delete`	Number of frames before deleting a target
`target_id`	Current target ID for next new target
`target_candidates`	Array with the IDs of the targets considered as candidates
`confirmed_targets`	Array with the IDs of the confirmed targets
`deleted_targets`	Array with the IDs of the deleted targets

Table 3.7: Configurable parameters for target tracking

In Figure 3.19 the flow diagram of the multiple target tracking algorithm is shown.

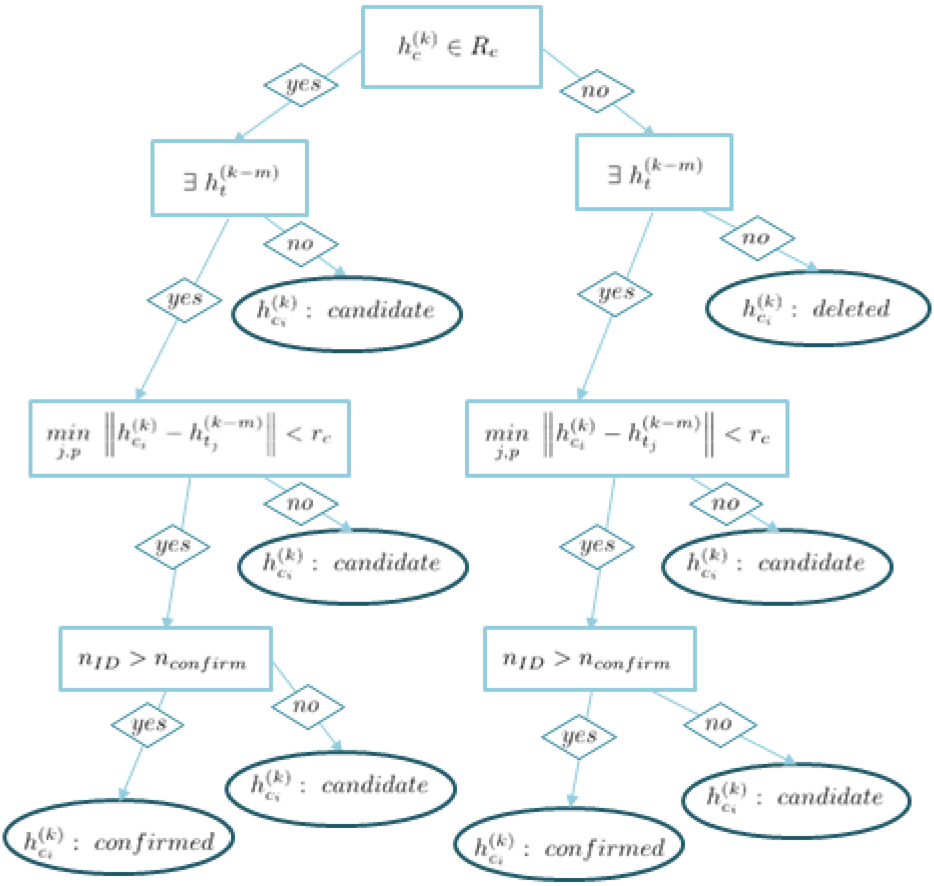


Figure 3.19: Flow diagram of the multiple target tracking algorithm

As an example of multiple target tracking, in Figure 3.20 can be observed the different IDs assigned to observations in the surveillance area compared with the simulated paths of different targets. It can be appreciated that when an observation is considered as a candidate, its ID is pictured in dark red and when it becomes a confirmed target its ID changes to green.

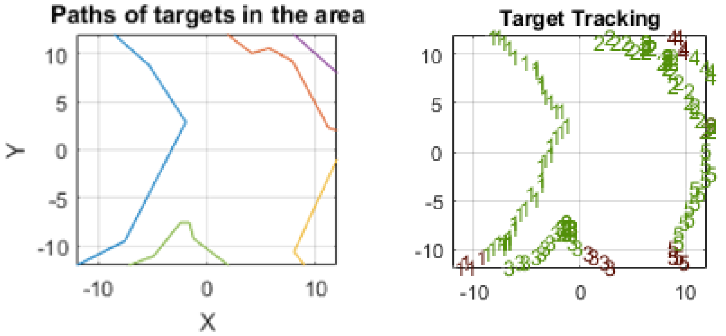


Figure 3.20: Example of multiple target tracking in the area

3.12 Estimation Filters

In practice, the tracked positions of the targets (*i.e.* the corresponding cluster heads) at a time may not correspond to the real position of human beings in the area. Thus, two different estimation filters are applied in order to smooth the trajectories of the targets and the abrupt changes of directions. These are the Kalman Filter and the Particle Filter. Their working principles have been explained in previous sections. An example of the correction in the position after applying both filters (separately) can be appreciated in Figure 3.21.

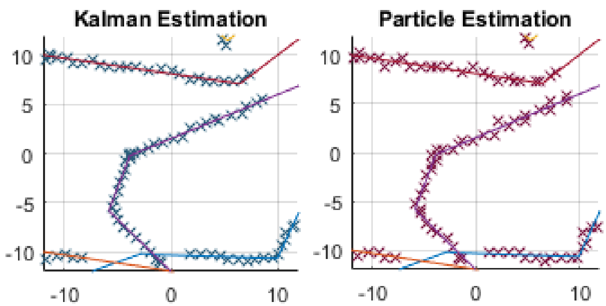


Figure 3.21: Estimated positions with Kalman Filter and Particle Filter

In this case, the parameters that can be configured for Kalman Filtering and for

Particle Filtering are presented in the following tables, respectively:

Parameter	Description
`R`	Observation Noise Covariance
`Q`	Process Noise Covariance
`A`	State transition model
`H`	Observation model

Table 3.8: Configurable parameters for Kalman Filtering

Parameter	Description	Value
`StateEstimationMethod`	State estimation model	'mean'
`ResamplingMethod`	Resampling particle modeling	'Systematic'
`n_particles`	Number of particles	1000

Table 3.9: Configurable parameters for Particle Filtering

Chapter 4

Performance Analysis

4.1 Performance Metrics

An ideal multiple target tracking system should be able to determine the correct number of people present in the area, to accurately estimate their positions, and to be able to identify each target with the same ID in the surveillance period. In this stage, the developed tracking algorithm is tested and the tracking results are compared with the input generated data. In order to do so, the set of observations (*i.e.* identified tracks) are analyzed and compare with the real paths followed by each target in the area. One observations and one real position will be associated if the distance between them is below a selected threshold, which is settled equal to 2 meters (as the radius selected in previous sections for a human being), and if it is the minimum distance compared to other observations. There exist different metrics to analyze the tracking performance [37][38]. Here we distinguish between precision metrics, which measure position deviation of the target estimates, and accuracy metrics, which count the number of occurred errors in the surveillance period.

4.1.1 Precision Metrics

The estimated position of real targets is normally suffers from some deviation inherent to the estimation mechanisms. After the tracked observations and the real objects are associated, the computation of the position deviation is performed. In our case, three different deviations are calculated due to the following estimation processes: image estimation (*i.e.* cluster heads), Kalman filtering and Particle filtering. The maximum deviation and the average deviation are also calculated for the three methods.

$$d = \|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\| \tag{4.1}$$

where $\hat{\boldsymbol{x}}_t$ is the estimated position of target t and $\boldsymbol{x}_t$ is the real position of target t at the evaluated instant. The maximum and average deviation can be expressed as follows, respectively:

$$d_{max} = arg \max_{t \in \mathcal{T}} \|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\| \tag{4.2}$$

$$\bar{d} = \frac{1}{M} \sum_{m=1}^{M} d_m \tag{4.3}$$

where M is the total number of matches during the surveillance period.

4.1.2 Accuracy Metrics

In particular, the following errors can take place when the system is tracking:

- **Miss**: one real object is not assigned to any current observation. This could happen, for example, if the RSS change caused by the target is below the threshold in the image filtering process and is not taken into account as an observation. It also could happen if the system losses the track of a target because its blob disappears as the result of the intersection of two different targets (*i.e.* two clusters are merged) and when their blobs are separated again, it falls out of the covered area of the previous hypothesis. The ratio of misses can be computed as the total number of misses m over the total number of real objects N in all the steps during the measurement period, as follows:

$$\bar{m}\ (\%) = \frac{1}{N}\sum_{n=1}^{N} m_n \tag{4.4}$$

- **False Positive**: one observation is considered as a target when no real object exists. This could happen because of the presence of spurious blobs in the image in the entrance/exit region. The ratio of false positives can be computed as the total number of false positives fp over the total number of real objects N in all the steps during the measurement period, as follows:

$$\bar{fp}\ (\%) = \frac{1}{N}\sum_{n=1}^{N} fp_n \tag{4.5}$$

- **Mismatch**: one real object is assigned a different ID (*i.e.* a different observation) than the one assigned in previous instants. This could happen if two targets have intersecting trajectories and the tracking algorithm is not able to distinguish correctly the previous paths of the targets and the new observations are assigned with each others path. Thus, each target will be track with a different ID than in previous steps. In this case, when a target changes its ID it is considered as one mismatch, and in the following steps, as the target is being tracked, it is not considered an error. The ratio of mismatches can be computed as the total number of mismatches *mme* over the total number of real objects N in all the steps during the measurement period, as follows:

$$m\bar{m}e\ (\%) = \frac{1}{N}\sum_{n=1}^{N} mme_n \tag{4.6}$$

In [37] it is addressed a tracking system performance metric which takes into account all the occurred errors during the tracking period. This metric is known as MOTA (Multiple Object Tracking Accuracy) and it is able to measure the system accuracy. Thus, it is the preferred metric for evaluating our system performance. It can be expressed as follows:

$$MOTA\ (\%) = 1 - \frac{1}{N}\sum_{n=1}^{N}(m_n + fp_n + mme_n) \tag{4.7}$$

4.2 Comparison between different Set-Ups

In this section, the tracking algorithm will be tested and different set-ups will be compared in order to analyze the system performance in different situations. The first simulation will be done with the parameter values shown in the following tables. Then, a variation in the parameters will be analyzed.

Parameter	Value
Number of targets	3
Number of steps	150
Step distances [m]	0.85
Average straight distance [steps]	10
Area - X coordinates [m]	[-12,12]
Area - Y coordinates [m]	[-12,12]
RSS change in target position [dB]	20
Calibration steps	10
Precision [m/voxel]	0.5
Target width - X [m]	1
Target width - Y [m]	1
Target angle [rad]	0
AWGN deviation	0.25

Table 4.1: Input data - First simulation

Parameter	Value
Number of sensors	41
Ellipse Excess path [m]	0.2
Network topology	Mesh
Quantization bits	16
Channel BER	1e-6

Table 4.2: Sensors - First simulation

Parameter	Value
Noise standard deviation [dB]	1
Voxel standard deviation [dB]	0.25
Correlation distance [m]	3
Intensity LPF weighting	0.9
Thresholding weighting	0.8
Cluster width [m]	2
Entrance/Exit region [m]	2
Radius around target [m]	2.5
Memory [steps]	5
Confirmation memory [steps]	5
Maximum confirmation memory [steps]	10
Steps before deleting	5

Table 4.3: Algorithm configuration - First simulation

4.2.1 Number of targets

First, the number of targets is varied in order to test the tracking capabilities of the algorithm. For one target that follows the path shown in Figure 4.1, there are no misses, false positives, or mismatches and the maximum precision is acquired with the Particle Filter, with a distance deviation of 0.695 meters.

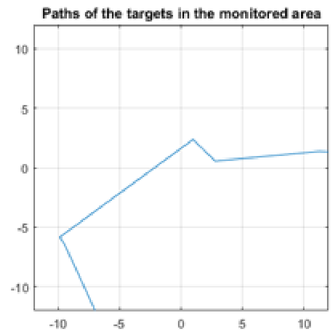


Figure 4.1: Example. Path followed by 1 target

For two targets following the paths represented in Figure 4.2, the system performs the tracking without misses, false positives or mismatches. The best performance is acquired again with the Particle Filter, obtaining 0.975 meters of maximum deviation.

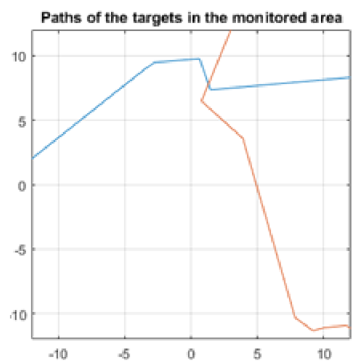


Figure 4.2: Example. Path followed by 2 targets

In case of 4 targets with no intersecting trajectories, as shown in Figure **??** (left), the minimum deviation is achieved with the simple tracking for this simulation. In this case the deviation is 1 meter. There are no misses, false positives or mismatches. In case of 4 targets with intersecting trajectories, as shown in Figure 4.3 (right), the minimum deviation is achieved with the simple tracking and is of 1.118 meters. The minimum average deviation is achieved with the Particle Filer, and is of 0.196 meters. In this case, there have been counted 3 misses, 9 mismatches and 0 false positives, due to the complexity of the paths.

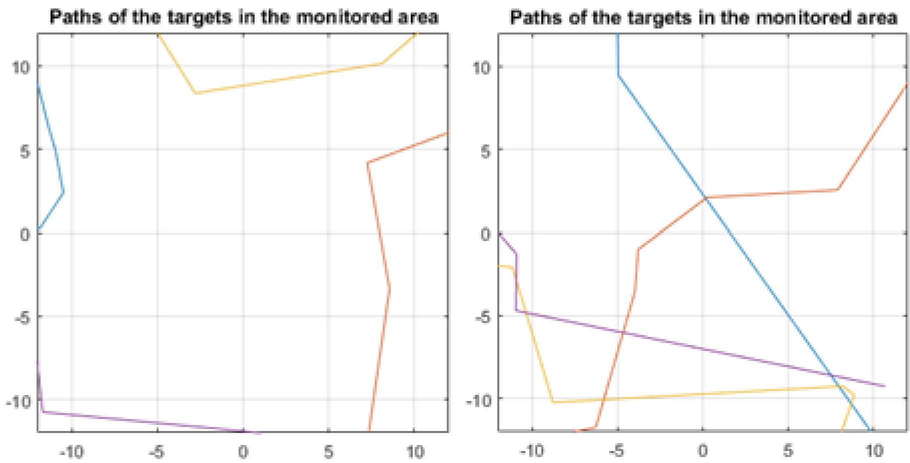


Figure 4.3: Example. Path followed by 4 targets with intersecting (right) and non intersecting trajectories (left)

When the number of targets is increased, for example, up to 10 targets in the area with intersecting trajectories, as shown in Figure 4.4, the system performance is highly degraded, obtaining in this case 40 misses, 2 false positives and 15 mismatches of 479 measures. For the tracked targets, the maximum deviation is 2 meters, obtained with the simple tracking algorithm. Thus, if the region of interest is crowded, it would be difficult to perform accurate tracking over all the present targets.

Figure 4.4: Example. Path followed by 10 targets with intersecting trajectories

4.2.2 Fade Level Amplitude

The following table present the results for three different simulations changing the fade level amplitude at the position of the target. This amplitude cannot be selected by the user but it can be understand as a measure of the fade level that has to be present in the environment due to multipath propagation.

Fade level:	5 dB	10 dB	20 dB
Max deviation	0.283	1.573	1
Estimator	Kalman Filter	Particle Filter	Simple Tracking
Misses	62	1	0
False positives	0	1	0
Mismatches	9	1	0

Table 4.4: Comparative study. Fade level variation

In this case, it can be appreciated from the above results that when the fade level decreases, the tracking task becomes unstable. This is because the fading level is comparable to the background noise.

4.2.3 Tracking Parameters

First, the tracking memory is varied and the following results are obtained. It can be appreciated that the tracking is better when more memory is available.

Tracking memory:	1 step	3 steps	5 steps
Max deviation	0	1	0.997
Estimator	Kalman Filter	Simple Tracking	Particle Filter
Misses	45	0	0
False positives	0	1	0
Mismatches	24	0	0

Table 4.5: Comparative study. Tracking memory

Then, the radius around the target is analyzed. This parameter represent when a current observation is assigned the same ID than a previous observation because it falls inside of its circunference. If the radius decreases, targets can be missed easily.

Tracking radius:	1 m	2 m	3 m
Max deviation	0.707	1	0.986
Estimator	Simple Tracking	Simple Tracking	Particle Filter
Misses	44	0	0
False positives	1	0	0
Mismatches	12	0	0

Table 4.6: Comparative study. Tracking radius

4.2.4 Number of sensors

The obtained tracking results are presented for the sensor topologies that are shown in the following figures. It can be understand from the results, that in order to perform a relevant radio tomographic image, the network need to be dense.

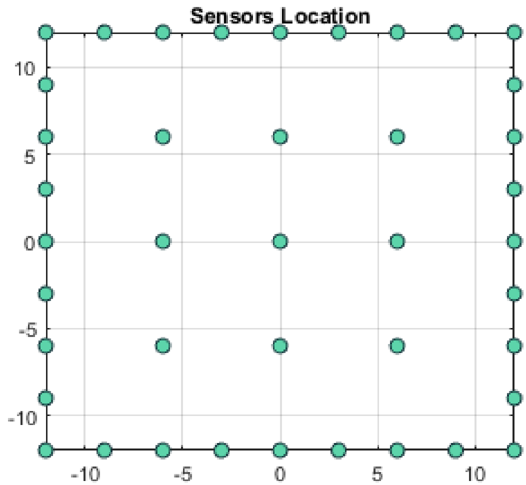


Figure 4.5: Comparison study. Network with 41 sensors in mesh topology

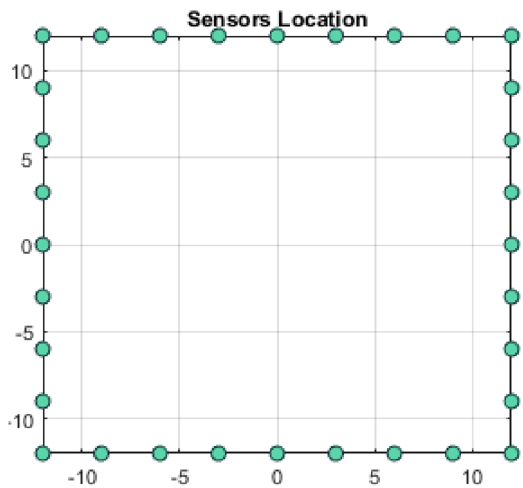


Figure 4.6: Comparison study. Network with 32 sensors

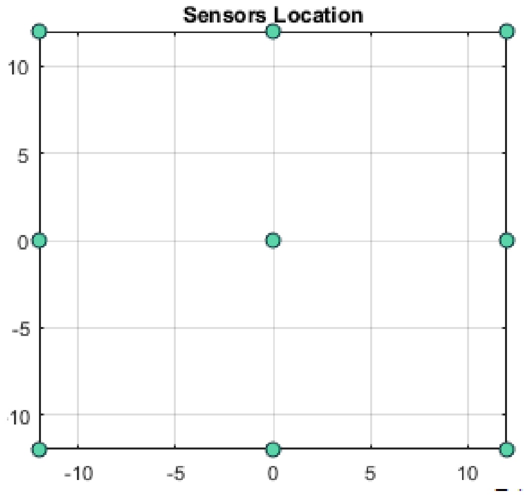


Figure 4.7: Comparison study. Network with 9 sensors

Sensors:	41 sensors	32 sensors	9 sensors
Max deviation	0.99	1	19.433
Estimator	Particle Filter	Simple Tracking	Particle Filter
Misses	0	0	1
False positives	0	0	98
Mismatches	0	0	36

Table 4.7: Comparative study. Sensors

Chapter 5

User Interface

5.1 Introduction

A user interface (UI) has been created with MATLAB® in order to facilitate the customization and control of the multi-target localization and tracking algorithm explained in previous sections. Several windows have been created distinguishing between input simulation and radiation model, wireless sensor network settings and algorithm configuration, and finally tracking visualization in the surveillance area and performance analysis of the algorithm for the selected set-up.

5.2 Input Simulation

The *Input Simulation Window* allows the variation of the following characteristics:

- **Number of targets**
- **Surveillance area**, where the maximum and minimum limits can be set both in X- and Y-axes.
- **Number of observation steps**
- **Target motion model**, where the average straight distance of a person and the distance covered by one step of a person can be configured.

When the button *Create Paths* is pressed, the paths are created for each target over the area and are presented in a graph in the right side of the window, as shown in Figure 5.1.

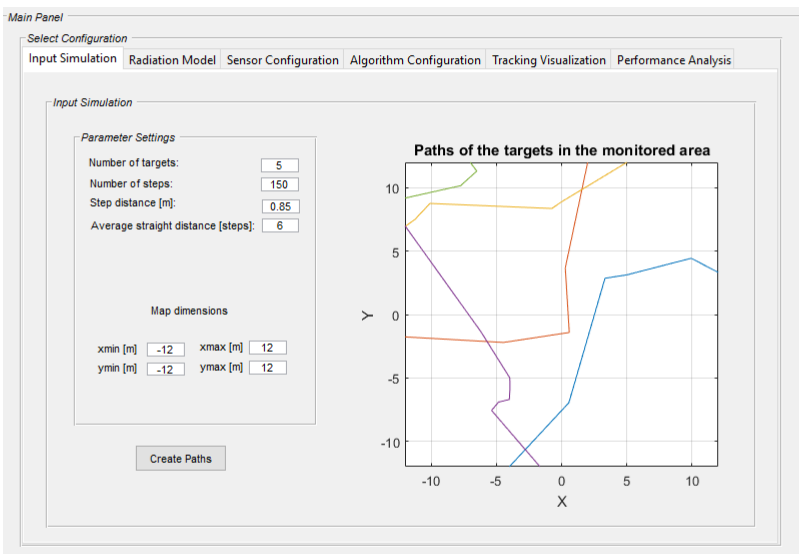


Figure 5.1: User Interface. Input Simulation

5.3 Radiation Model

The *Radiation Model Window* performs the simulation of the receive signal strength changes in the scenario due to the presence of targets in the area. It allows the variation of the following features:

- **Fade amplitude** in the position of a target
- **Number of calibration steps**, which is the time while the system is taking reference measurements of the environment when the area is empty.
- **Image precision**, in meters per voxel.
- **Target width**, in meters. It can be adjusted both in X- and Y-axes.
- **Target rotation angle**, in radians.
- **Noise deviation** of the environmental noise, modeled as AWGN.

When the *Create Radiation* button is pressed, the radiation with the selected features is generated in the area and the *Play* buttons become enable. The simulated radiation can be played by the user in 2-D or 3-D views. It also can be paused at one point and then continue the video in the same point, or stopped to replay it again. Additionally, two tabs are added in the left side for future implementation of other models of environmental information such as temperature o LIDAR radiation. Figure 5.2 shows the appearance of this window.

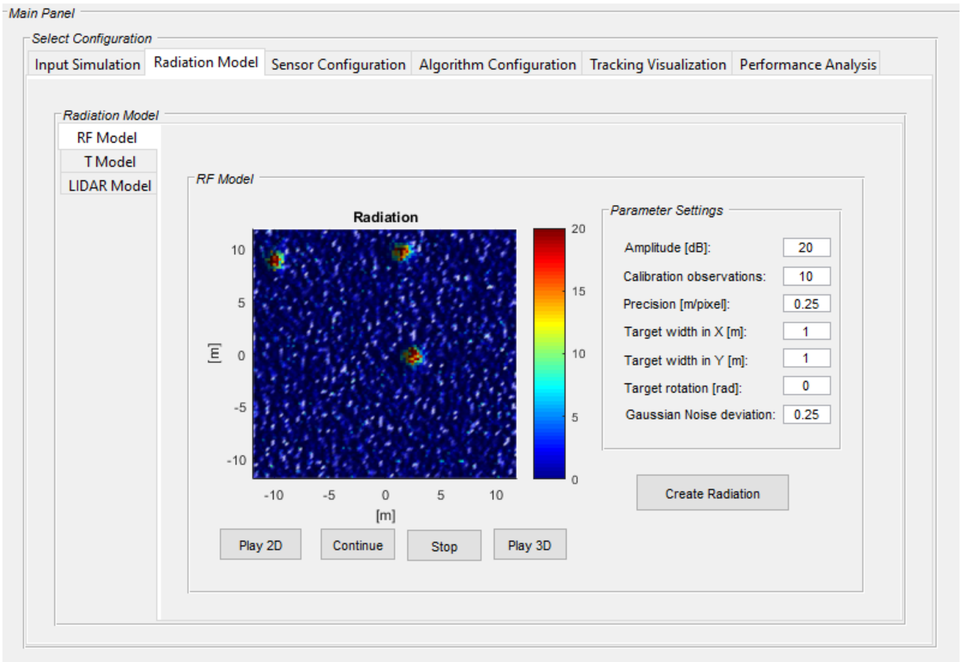


Figure 5.2: User Interface. Radiation Model

5.4 Sensor Configuration

The *Sensor Configuration Window* allows the user to select the topology of the wireless sensor network. For doing so, it is possible to select the coordinates of the sensors positions either introducing them manually or selecting the point in the graph that represents the covered area, as it can be seen in Figure 5.3. For future implementations, it is also possible to select the type of sensors in both cases, while in this first version it is only possible to select the RF type.

When introducing a new sensor, this will appear in the *Sensors Overview Panel*, as well as the rest of sensors that have been already added. It also will appear as a mote in the figure that represents the surveillance area. The panel shows the position and type properties of each sensor. If the user wants to delete some sensors, it can be done by selecting them in the panel and then pressing the *Delete Sensors* button. The chosen topology can be saved by pressing the *Save As* button to save them in a new file or the *Save Sensors* button to update the current file. The file is saved with *.csv* format. It is also possible to load other network topologies by pressing the *Load Sensors* button if they have been saved in a *csv* file with the correct format.

One important characteristic that the topology should have is the appropriate coverage in order to be able to cover all the desired area. As it has been mentioned in previous sections, the ellipse model is used to model the coverage in each link.Therefore, the ellipse excess path can be configured and the corresponding ellipses can be pictured in the graph by pressing the *Show Ellipses* button and be hidden again by pressing the *Hide Ellipses* button.

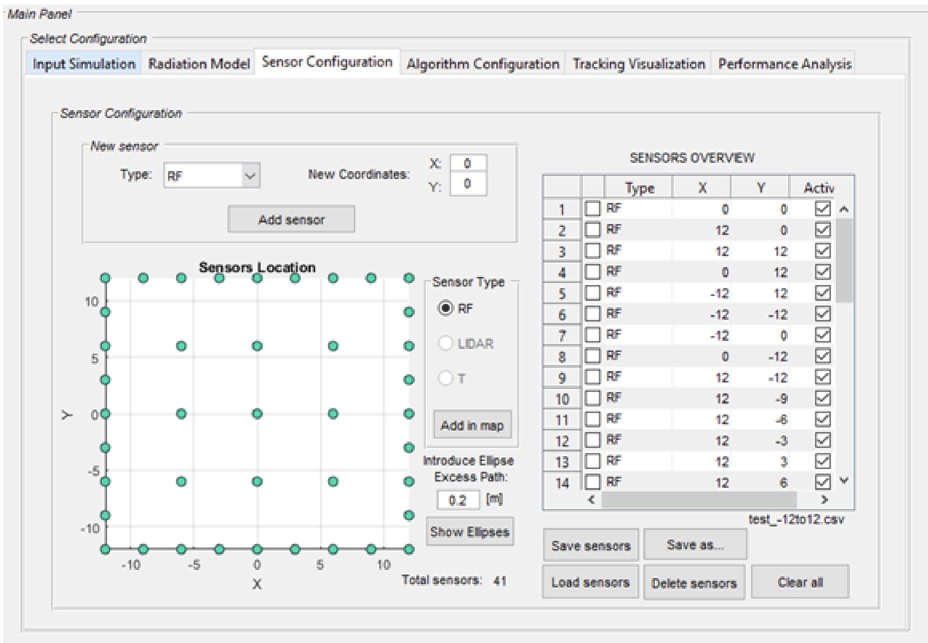


Figure 5.3: User Interface. Sensor Configuration

5.5 Algorithm Configuration

The *Algorithm Configuration Window* allows the user to set some internal parameters of the tracking algorithm. Most of these parameters depends on the characteristics of the area and have to be acquired or estimated by experimental measurements and tests. In this case, the following parameters can be configured:

- **Noise standard deviation** for image reconstruction
- **Voxels standard deviation**, which addresses the possible variation of voxel measurements.
- **Correlation distance** between voxels
- **Filtering weighting** α, for intensity low pass filtering.
- **Thresholding weighting** β, for dynamic thresholding.
- **Cluster width**
- **Quantization bits**
- **Probability of error**
- **Entrance/Exit region dimensions**
- **Radius around target**, for data association.
- **Tracking memory**, for comparing new observations to previous hypotheses.
- **Confirmation memory**, for confirming a track
- **Maximum number of observations** for confirming a target
- **Number of non-assignations** for deleting a target

When the *Apply* button is pressed, the tracking algorithm is computed over the previous data and the results in each step are stored. The tracking has to be performed for all the selected target positions, which takes some time to be completed so a pop-up window appears showing the progress. In the command window of MATLAB it is logged every time a target becomes candidate, confirmed or deleted, as can be seen in the example shown in Figure 5.4.

```
--------------------------------------
New target CANDIDATE! Target ID: 4

--------------------------------------
TRACK CANDIDATES:
4
CONFIRMED TRACKS:
1
3
DELETED TRACKS:
2
```

Figure 5.4: Target Tracking Log

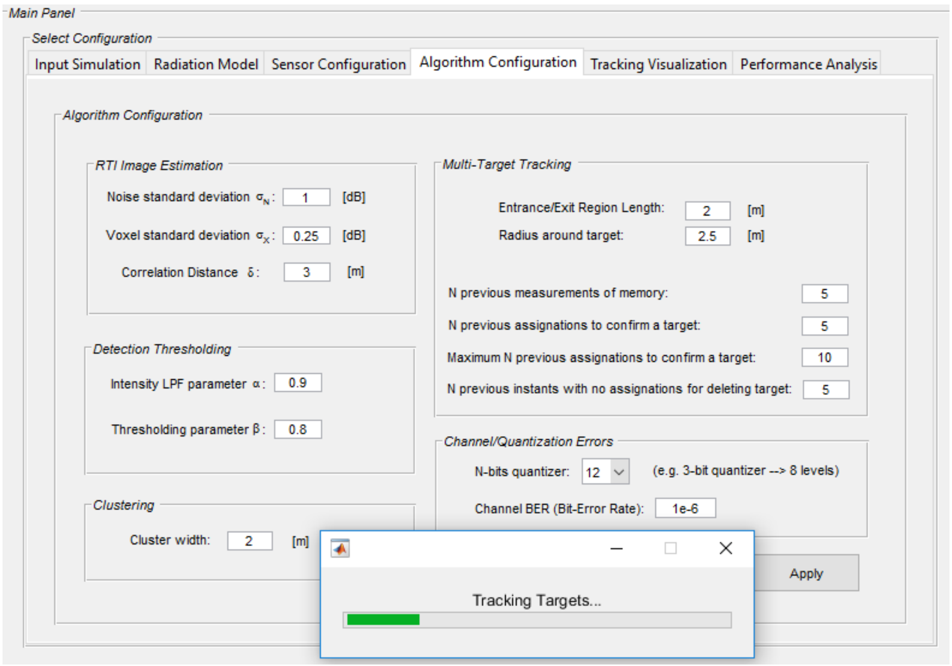


Figure 5.5: User Interface. Algorithm Configuration

5.6 Tracking Visualization

The *Tracking Visualization Window* presents different figures where the different stages of the target tracking algorithm can be seen while the tracking is performed. By pressing the Play button, the tracking visualization starts.

The different tracking phases are divided into the following categories, from left to right in Figure 5.6:

1. Input Radiation of the targets in the area
2. Estimated Radiation after image reconstruction
3. Filtered RTI Image
4. Clustering process
5. Comparison between current observations and previous hypotheses
6. Target IDs assignations
7. Estimation of targets positions with Kalman Filter
8. Estimation of targets positions with Particle Filter

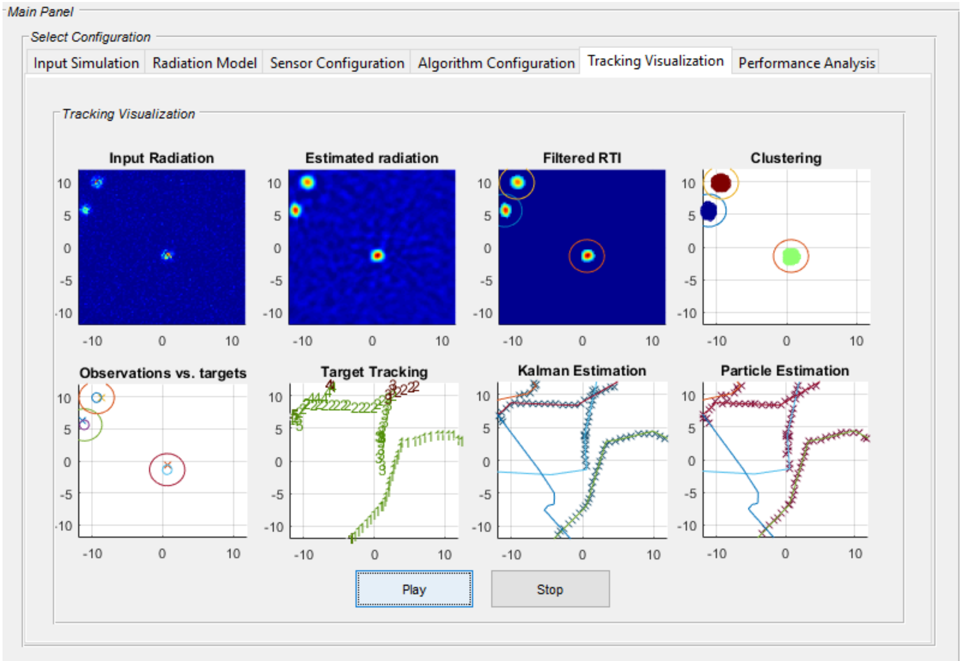


Figure 5.6: User Interface. Tracking Visualization

5.7 Performance Analysis

The *Performance Analysis Window* presents the performance results of the tracking algorithm after applying the corresponding metrics described in previous sections. In this window two different panels can be distinguished: the *Tracking Precision Panel* and the *Tracking Accuracy Panel*. In the first one, the maximum and average deviation in the estimated position of the target, compared with its real position, are presented for the simple tracking algorithm, when using Kalman Filtering and also when using Particle Filtering. For each case, it is also included a histogram of the deviation errors. On the other hand, in the second panel the tracking accuracy is displayed.

The following accuracy metrics are analyzed:

- **Ratio of misses**
- **Ratio of false positives**
- **Ratio of mismatches**
- **MOTA**, Multiple Object Tracking Accuracy (%)

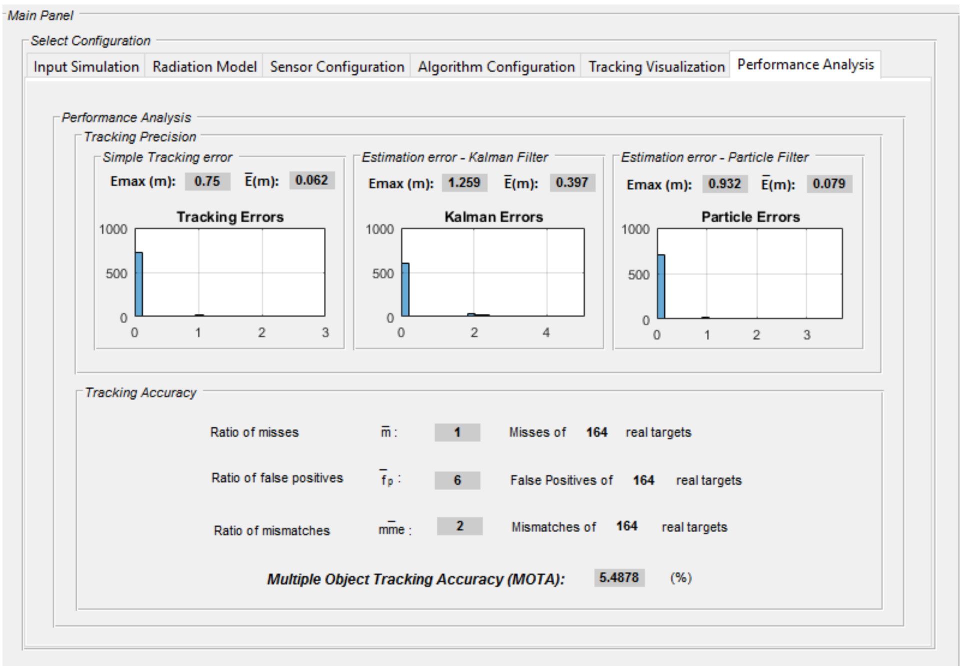


Figure 5.7: User Interface. Performance Analysis

Chapter 6

Conclusions and Future Work

An algorithm for localization and tracking of multiple non-cooperative targets for digital storytelling applications in public spaces has been developed applying Radio Tomographic Imaging techniques. Using a dense sensor network with RF sensors, which fulfills privacy requirements in public areas, has been possible the accurate tracking of multiple targets in open environments, which suffer from multipath propagation and shadowing, considering error-prone radio channels. Additionally, Kalman filtering and Particle filtering have been applied in order to achieve a better system performance.

In this report the use of wireless sensor networks for multiple target tracking have been analyzed. A review of the state-of-the-art of multiple target tracking techniques and localization and detection methods is outlined. Multiple target tracking challenges by using WSNs are also analyzed and taken into account for the tracking algorithm under study.

A comparative study of different configurations of the developed tracking algorithm has been performed. From multiple simulations of the developed tracking algorithm, it has been shown that the best tracking accuracy is achieved when the number of nodes increases. When the number of targets increases in the surveillance area, the tracking system suffers from performance degradation.

As future steps, real environmental data can be processed, instead of generated data, in order to evaluate the real system performance of the developed algorithm. Besides, more specific environmental models can be taken into account in the tracking algorithm in order to perform better location estimations. The algorithm can be modified to include other types of sensors and jointly generate more accurate results. Other tracking techniques can also be applied, such as distributed detection in the sensor nodes instead of having a centralized detector, in order to reduce the amount of data that has to be sent from every sensor in the network due to energy constraints and other network requirements. Finally, tests with real sensors can be made either in a laboratory in Adressaparken in Trondheim so that multiple target tracking can be achieved for digital storytelling applications.

Bibliography

[1] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors i. fundamentals. *Proceedings of the IEEE*, 85(1):54–63, Jan 1997.

[2] K Ramya, K Praveen Kumar, and V Srinivas Rao. A survey on target tracking techniques in wireless sensor networks. *International Journal of Computer Science and Engineering Survey*, 3(4):93, 2012.

[3] Asmaa Ez-Zaidi and Said Rakrak. A comparative study of target tracking approaches in wireless sensor networks. *J. Sensors*, 2016:3270659:1–3270659:11, 2016.

[4] Éfren L. Souza, Eduardo F. Nakamura, and Richard W. Pazzi. Target tracking for sensor networks: A survey. *ACM Comput. Surv.*, 49(2):30:1–30:31, June 2016.

[5] Joey Wilson and Neal Patwari. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing*, 9(5):621–632, May 2010.

[6] Qian Lei, Haijian Zhang, Hong Sun, and Linling Tang. A new elliptical model for device-free localization. *Sensors*, 16(4):577, 2016.

[7] Ian F. Akyildiz, Tommaso Melodia, and Kaushik R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51:921–960, 2007.

[8] Himani S Parekh, Darshak G Thakore, and Udesang K Jaliya. A survey on object detection and tracking methods. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(2):2970–2979, 2014.

[9] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), December 2006.

[10] Pramod K. Varshney. *Distributed Detection and Data Fusion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1996.

[11] R. S. Blum, S. A. Kassam, and H. V. Poor. Distributed detection with multiple sensors ii. advanced topics. *Proceedings of the IEEE*, 85(1):64–79, Jan 1997.

[12] Amirali K. Gostar, Reza Hoseinnezhad, and Alireza Bab-Hadiashar. Multi-bernoulli sensor-selection for multi-target tracking with unknown clutter and detection profiles. *Signal Process.*, 119(C):28–42, February 2016.

[13] Domenico Ciuonzo and Pierluigi Salvo Rossi. Distributed detection of a non-cooperative target via generalized locally-optimum approaches. *Information Fusion*, 36:261–274, 2017.

[14] Tiancheng Li, Juan M. Corchado, Shudong Sun, and Javier Bajo. Clustering for filtering: Multi-object detection and estimation using multiple/massive sensors. *Information Sciences*, 388389:172 – 190, 2017.

[15] H. Yigitler, R. Jantti, O. Kaltiokallio, and N. Patwari. Detector based radio tomographic imaging. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2017.

[16] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion*, (1):28–44, January 2013.

[17] F. Viani, P. Rocca, G. Oliveri, D. Trinchero, and A. Massa. Localization, tracking, and imaging of targets in wireless sensor networks: An invited review. *Radio Science*, 46(05):1–12, Oct 2011.

[18] Zhe Chen. Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. Technical report, McMaster University, 2003.

[19] D. Zhang, Y. Liu, and L. M. Ni. Rass: A real-time, accurate and scalable system for tracking transceiver-free objects. In *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 197–204, March 2011.

[20] F Viani, P Rocca, M Benedetti, G Oliveri, and A Massa. Electromagnetic passive localization and tracking of moving targets in a wsn-infrastructured environment. *Inverse Problems*, 26(7):074003, 2010.

[21] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian. Multiple target tracking with rf sensor networks. *IEEE Transactions on Mobile Computing*, 13(8):1787–1800, Aug 2014.

[22] S. Nannuru, Y. Li, Y. Zeng, M. Coates, and B. Yang. Radio-frequency tomography for passive indoor multitarget tracking. *IEEE Transactions on Mobile Computing*, 12(12):2322–2333, Dec 2013.

[23] Y. Zhao, N. Patwari, J. M. Phillips, and S. Venkatasubramanian. Radio tomographic imaging and tracking of stationary and moving people via kernel distance. In *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 229–240, April 2013.

[24] Lionel M. Ni and Dian Zhang. Dynamic clustering for tracking multiple transceiver-free objects. *2009 IEEE International Conference on Pervasive Computing and Communications*, 00:1–8, 2009.

[25] Natallia Katenka, Elizaveta Levina, and George Michailidis. Tracking multiple targets using binary decisions from wireless sensor networks. *Journal of the American Statistical Association*, 108(502):398–410, 2013.

[26] M. M. Noel, P. P. Joshi, and T. C. Jannett. Improved maximum likelihood estimation of target position in wireless sensor networks using particle swarm optimization. In *Third International Conference on Information Technology: New Generations (ITNG'06)*, pages 274–279, April 2006.

[27] C. Perez-Vega and J. L. G. Garcia. A simple approach to a statistical path loss model for indoor communications. In *1997 27th European Microwave Conference*, volume 1, pages 617–623, Sept 1997.

[28] S. Srinivasa and M. Haenggi. Path loss exponent estimation in large wireless networks. In *2009 Information Theory and Applications Workshop*, pages 124–129, Feb 2009.

[29] T. K. Sarkar, Zhong Ji, Kyungjung Kim, A. Medouri, and M. Salazar-Palma. A survey of various propagation models for mobile communication. *IEEE Antennas and Propagation Magazine*, 45(3):51–82, June 2003.

[30] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi. An empirically based path loss model for wireless channels in suburban environments. *IEEE J.Sel. A. Commun.*, 17(7):1205–1211, September 2006.

[31] J. Wilson and N. Patwari. A fade-level skew-laplace signal strength model for device-free localization with wireless networks. *IEEE Transactions on Mobile Computing*, 11(6):947–958, June 2012.

[32] O. Kaltiokallio, M. Bocca, and N. Patwari. Enhancing the accuracy of radio tomographic imaging using channel diversity. In *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, pages 254–262, Oct 2012.

[33] N. Patwari and P. Agrawal. Effects of correlated shadowing: Connectivity, localization, and rf tomography. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 82–93, April 2008.

[34] Liu Heng, Wang Zheng-huan, Bu Xiang-yuan, and An Jian-ping. Image reconstruction algorithms for radio tomographic imaging. In *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 48–53, May 2012.

[35] Joey Wilson, Neal Patwari, and O Guevara Vasquez. Regularization methods for radio tomographic imaging. In *in 2009 Virginia Tech Symposium on Wireless Personal Communications*, 2009.

[36] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2 edition, 2009.

[37] Keni Bernardin, Er Elbs, and Rainer Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment.

[38] B. Ristic, B. N. Vo, D. Clark, and B. T. Vo. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Transactions on Signal Processing*, 59(7):3452–3457, July 2011.