DOTTORATO DI RICERCA IN
COMPUTER ENGINEERING AND SCIENCE

SCUOLA DI DOTTORATO IN
INFORMATION AND COMMUNICATION TECHNOLOGIES

XXIV Ciclo

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

TESI PER IL CONSEGUIMENTO DEL TITOLO DI DOTTORE DI RICERCA

Tecniche di Data Management per Applicazioni Active RFID

Tesi di:
Razia HAIDER
Relatore:
Prof. Federica MANDREOLI
Il Direttore:
Prof. Giorgio Matteo VITETTA

DEGREE OF DOCTOR OF PHILOSOPHY IN
COMPUTER ENGINEERING AND SCIENCE

DOCTORATE SCHOOL IN
INFORMATION AND COMMUNICATION TECHNOLOGIES

XXIV Cycle

UNIVERSITY OF MODENA AND REGGIO EMILIA

INFORMATION ENGINEERING DEPARTMENT

Ph.D. DISSERTATION

Data Management Techniques for Active RFID Applications

Candidate:
Razia HAIDER
Advisor:
Prof. Federica MANDREOLI
The Director of the School:
Prof. Giorgio Matteo VITETTA

Keywords:

RFID Data Management Systems
Probabilistic models
Uncertainty Management
RFID Data Aggregation
Intruder Localization

Abstract

In the last several years, RFID technology has gained significant popularity due to its ability of detecting objects and people carrying small RFID tags in an environment equipped with RFID readers. RFID applications usually rely on RFID deployments to manage high-level events such as tracking the location that products visit for supply-chain management, monitoring the location and status of patients in hospital environment, localizing intruders for alerting services, and so on. A fundamental relation for these purposes is the location of people and objects over time. However, the nature of RFID data stream is noisy, redundant and unreliable and thus streams of low-level tag-readings must be transformed into meaningful relation instances. Nevertheless, the management of RFID data in transforming low-level streams in to high-level events poses a number of challenges. Moreover, RFID deployments usually produce huge volume of data that can reach in practical cases to the size of gigabytes in a day.

This thesis presents the design, implementation and experimental evaluation of realtime system that addresses the above mentioned data management issues in the context of RFID location tracking systems.

In this thesis, an *RFID Data Online Filtering and Uncertainty Management Module* based on probabilistic models and techniques is developed that operates on unreliable and imprecise real-time data stream in order to transform them into probabilistic data streams that can be meaningful to the applications or by which we can extract information of interest.

Moreover, to handle the huge data volume generated by RFID deployments, this thesis proposes a simple on-line summarization mechanism by developing an *RFID Probabilistic Data Aggregation Module*, which is able to provide small space representation for massive RFID probabilistic real-time data streams while preserving the meaningfulness of the information.

This thesis also presents an innovative system for intruder detection relying on the joint use of cameras and RFIDs, in real noisy and complex wide open areas. We propose a new architecture and developed specific algorithms that have been finally tested on real-cases.

Sommario

Negli ultimi anni, la tecnologia RFID sta guadagnando una certa popolarità grazie alla sua capacità di rilevare oggetti e persone dotati di piccoli tag in un ambiente attrezzato di antenne e lettori RFID. Le applicazioni RFID spesso si basano sull'omonima tecnologia per gestire eventi di alto livello, come il tracciamento della posizione di prodotti in una suply-chain, il monitoraggio della posizione e lo stato dei pazienti in ambiente ospedaliero, servizi di localizzazione di intrusi, e cos via. Una relazione fondamentale per questi scopi é la posizione di persone e oggetti nel tempo. Tuttavia, il flusso di dati RFID é per natura rumoroso, ridondante e inaffidabile e quindi questo flusso di basso livello deve essere trasformato in una sequenza temporale di istanze di posizione. Inoltre, le applicazioni RFID di solito producono enormi quantità di dati che possono raggiungere in casi pratici la dimensione di un gigabyte in un giorno.

Questa tesi presenta la progettazione, realizzazione e valutazione sperimentale di un sistema in tempo reale che risolve i problemi di gestione dei dati RFID suddetti.

Il sistema é dotato di un modulo per RFID Data Online Filtering & Uncertainty Management basato su modelli probabilistici che é in grado di trasformare il flusso di dati grezzi provenienti da tag RFID in informazioni probabilistiche, che risultano significative per le applicazioni o dalle quali é possibile estrarre informazioni di interesse.

Inoltre, per gestire grandi volumi di dati generati da applicazioni RFID, questa tesi propone un semplice meccanismo on-line che é in grado di riassumere quantità massicce di dati in streaming pur preservando la significatività delle informazioni trasmesse.

Infine, in questa tesi abbiamo anche progettato e realizzato un programma innovativo per la localizzazione di intrusi. Si tratta della prima proposta di utilizzo congiunto di telecamere e RFID in tempo reale su vaste aree aperte, rumorose e complesse. A questo scopo nella tesi viene proposta una nuova architettura e vengono sviluppati algoritmi specifici che sono stati testati su casi reali.

Contents

List of Figures

Chapter 1

Introduction

Data streams are possibly infinite sources of data that stream continuously while observing a physical phenomenon, e.g. temperature or humidity levels, telephone call records or audio video streaming, and so on. Data streams could be generated in different scenario by different devices, such as audio and video devices, Global Positioning System(GPS), Radio Frequency Identification (RFID) and other types of sensors. Among these RFID is one of the emerging technology for a wide-range of applications, including supply chain and asset management [Gonzalez et al., 2006b], monitoring the location and status of patients in hospital environment [Kim et al., 2008], localizing intruders for alerting services [Cucchiara et al., 2011] and so on. RFID offers a promising alternative to barcode identification system. In an RFID system, an environment is deployed with the RFID readers and antennas while users and objects carry RFID tags. There are various types of RFID technology. In particular, some systems use active RFID tags (i.e., battery powered tags), while others use passive tags. RFID readers detect the presence of tags in its vicinity and generate stream of low-level observation in the form of TRE (Tag Read Event): $(tag\_id, antenna\_id, time)$ that shows when and where tags are being sighted. These low-level observations must be transformed into high-level events meaningful to applications. For example, "Tag 101 was seen at antenna 12 at 10:00" must be transformed into meaningful relation instance such as " Alice entered office O1 at 10:00". Nevertheless, the management of RFID data in transforming low-level streams in to high-level events poses a number of challenges [Chawathe et al., 2004, Hu et al., 2005]. In particular, the nature of an RFID data stream is noisy, redundant and unreliable, making it unsuitable for direct use in applications. RFID deployments, generally, produce imprecise data mainly because of three main reasons:

1. *Conflicting Readings*: Readings in the presence of contradiction i.e., when an RFID tag is simultaneously detected by two antennas that cover ad-

jacent areas, it becomes difficult to establish the actual location of tag [Jeffery et al., 2006b]

2. *Missing Readings*: Loss of reading instances in which RFID tags are not detected by the antenna while actually being present within its coverage area. This is a phenomenon whose causes are entirely separate from the specific application scenario and amputations technologies used in the construction of the devices, the incidence of this phenomenon is, however, high and not negligible because recent studies report that an RFID reader is usually able to detect only 60% -70% of tags that are in its vicinity [Floerkemeier and Lampe, 2004, Jeffery et al., 2006b]

3. *Data-Information Mismatch*: Mismatch between the information to which the application is concerned and the data produced by the sensors. Typically an application is particularly interested in high-level information such as "who is in a certain place at a given time", "the place where he can be", for example, a room, a specific area, or near by an object. The sensors are limited to providing data in form of low-level signaling i.e., "when a tag is detected by a certain antenna".

For all of these reasons the generated stream of raw data become unreliable for RFID applications and hence, making them not suitable to be directly used for further analysis of sophisticated queries. To this end, this thesis proposes a *RFID Data Acquisition, Online Filtering & Uncertainty Management* mechanism that operates on unreliable and imprecise data stream in order to transformed them into reliable probabilistic data streams that can be meaningful to the applications or by which we can extract information of the interest.A common way of dealing with such kind of imprecise data is to built a model of the data and uses stream of raw reading as input to the model. To this end, *RFID Data Online Filtering & Uncertainty Management Module* makes use of a temporal graphical model [Koller and Friedman, 2009], the simplest of which is the Hidden Markov Model (HMM) [Rabiner, 1990] that continuously infers hidden variables (locations, in case of above example) based on sensor readings. Such a relation, therefore, is a probabilistic relation `At(tagID,location,time,prob)` that is usually stored in a (probabilistic) database table and queried to detect complex events meaningful to applications [Ré et al., 2008]. An example tuple is `(101,O1,10:00,0.7)`, which indicates that tag 101 at time 10:00 was in office O1 with probability 0.7.

Since active RFID tags continually send out their IDs at pre-programmed intervals (one second) as the case of this thesis and for each tag read, the number of probabilistic tuples equals the number of reference locations. Therefore, an HMM for RFID deployments produces huge volume of probabilistic data that can reach in practical cases to the size of gigabytes in a day. Storing all these probabilistic tuples in the probabilistic database is extremely expensive and, even more important, it is not always useful. For instance, if person stays at same location for two hours and there are five mapped locations in this system, 36,000 probabilistic tuples are produced for two hours which report more or less the same location information for him.

In this context, the thesis proposes a simple on-line summarization mechanism, which is able to provide small space representation for massive RFID probabilistic data streams while preserving the meaningfulness of the information. The mechanism draws inspiration from the field of clustering [Jain et al., 1999]. The main idea behind the proposed approach is to keep on aggregating tuples until a state transition is detected. The *RFID Probabilistic Data Aggregation Module* processes probabilistic tuples as they arrive, i.e. it takes the output of the *RFID Data Online Filtering & Uncertainty Management Module* as its input, hence avoiding the use of expensive and offline disk based operations such as sorting and summarization, and promptly stores the output in the probabilistic database MayBMS [Huang et al., 2009] in such a way that a wide range of temporal probabilistic queries can be applicable and answered effectively.

All the modules and methods presented in this thesis are implemented in a framework named *RFID Probabilistic Data Management (RPDM) System* in the context of location tracking. However, they can be applicable in other contexts of RFID data management applications. *RPDM System* works in two phases. In first phase, it receives raw RFID data from RFID devices and performs online filtering & uncertainty management and finally, stores filtered probabilistic tuples in probabilistic database for query processing. In second phase, aggregation module receives the output of the first phase, which is filtered probabilistic tuples and aggregates them by applying our proposed summarization method. Finally, it stores aggregated probabilistic tuples in probabilistic database for query processing. We evaluate the performance of *RPDM System* under real-cases, it also includes an easy-to-use GUI that can be used for both online and offline operations.

Wide open areas represent challenging scenarios for surveillance systems, especially when sensors are affected by noise, uncertainty, distractors and complex scenarios. Moreover, the identification of intruders become difficult task when moving in group of people. The coordination between the cameras can be certainly used but the tasks of localizing and identifying targets (e.g., people) in such environments require to go beyond the use of camera-only deployments. In this context, in a joint effort with imageLab of University of Modena and Reggio Emilia, we present an innovative system for wide open area intruder detection based on the joint use of cameras and RFIDs, allowing us to map RFID tags to people detected by cameras and, thus, highlighting potential intruders.

This is the first proposal of joint use of cameras and RFIDs in real noisy and complex wide open areas for intruder localization. We propose a new architecture and specific algorithms for intruder detection relying on:

- sophisticated filtering techniques for singular sensor modality that preserve the uncertainty of data in the form of probabilities and overcome the heterogeneity of sensors through the introduction of common locations which the data coming both from cameras and RFIDs are mapped to. This is basically the *RFID Data Online Filtering & Uncertainty Management Module* of *RPDM System*;

- an evidential fusion architecture, based on Transferable Belief Model (TBM) [Smets, 1994], that processes uncertain data, combines the two sources of information and manages conflicts between them in order to map RFID tags to people detected from

Finally, proposed system has been tested on real-cases.

Organization of the Thesis
The rest of the thesis is organized as follow:

- *chapter 2* illustrates a brief introduction to RFID technology with its main feature, constructions and deployments. Furthermore, it presents some of the important RFID applications.

- *chapter 3* introduces RFID Data Management Systems by giving an overview of the fundamental issues faced by RFID systems and solutions provided in literature regarding these issues. A description of the main interesting RFID Data Management Systems is also included from the literature.

- *chapter 4* gives a general description and schematic architecture of proposed *RPDM System*. It also includes the details of RFID devices used for data acquisition in *RPDM*.

- In *chapter 5*, we introduce our *RFID Data Acquisition, Online Filtering & Uncertainty Management* mechanism in the context of location tracking. This mechanism mainly concerns with the transformation of unreliable and imprecise RFID data streams into reliable probabilistic data streams that can be meaningful to the application or by which we can extract information of the interest.

- *chapter 6* presents our proposed on-line summarization mechanism, for providing small space representation for massive RFID probabilistic data streams.
- In *chapter 7*, we present an innovative system for wide open area intruder detection relying on the joint use of cameras and RFIDs.
- Finally, in *chapter 8*, we made some concluding remarks and discuss future work.

Chapter 2

RFID Systems and Applications

In last several years, RFID technology has gained significant popularity due to its ability of detecting objects and people carrying small RFID tags in an environment equipped with RFID readers. RFID applications usually rely on RFID deployments to manage high-level events such as tracking the location that products visit for supply-chain management, monitoring the location and status of patients in an hospital environment, localizing intruders for alerting services, and so on.

This chapter provides a brief introduction of the RFID technology to the readers by describing its main features, construction and deployments (Section 2.1 and 2.2). Moreover, it discusses some of the important applications built on the RFID technology (Section 2.3).

2.1 RFID

RFID technology enables a system to transmit the identity of an object or person wirelessly from an electronic tag, called RFID tag in the form of a unique serial number, known as label, using radio waves. The purpose of an RFID system is to transmit data by a portable device, called a tag, which is read by an RFID reader and processed according to the requirements of a specific application. The tag transmits the data that may provide identification or location information of persons or objects, or other specifics about the objects tagged, such as price, color, date of purchase, etc.

Unlike, the bar-code technology, it is not essential for the RFID technology to "show" the tag to the reader device. In other words it does not require contact or line of sight for transmission or communication. RFID data is readable through the human body, clothing and non-metallic materials.

Operating frequency for RFID devices [Ward et al., 2006] ranges from 100 Hz to beyond 2.5 GHz. There is no global governing unit for frequencies used by RFID devices. However, every country put some regulatory restrictions on use of the radio-frequency spectrum, therefore, only a few frequencies are usually used [rfi, 2005]. The two most commonly used frequencies in HF band and UHF band are: in high frequency tags 13.56 MHz and in Ultra-high frequency around 900 MHz. The HF frequency is usable globally without any license, while the UHF frequencies are usable only in the U.S., E.U., and Japan (and vary among them). The characteristics of the sensing environment are influenced by frequency range being used. For instance, UHF signals can travel longer when there are no obstacle or barriers. Therefore, UHF tags are a good option for tagging items for the supply chain industry. UHF signals cannot propagate easily through certain objects but HF signals transmit more easily through plastic, paper, and moisture. As a result, HF tags are better choice for applications such as tagging bottles for the pharmaceutical industries.

2.2 System Overview and Components

A basic RFID system consists of following components:

- *Tags* (also called transponders) electronically programmed with unique information knows as "label", which are used to identify goods or assets
- *Readers*, which exchange information with the tags and host computer systems
- various *Antenna* types/characteristics for different applications
- *Middleware* that provide interfacing between readers and application softwares in an RFID environment

An overview of RFID system based on above mentioned component is shown in Figure 2.1 and it works in the following way: When an RFID tag passes through the electromagnetic zone of the reader, it detects the reader's activation signal. The tag respond to the reader by sending return signal and the reader decodes the data encoded in the tag's integrated chip. Finally, the data is passed to the host computer for processing. The antenna is used in order to activate the tag for data read and write purposes by emitting radio signals. The detailed description of each component of an RFID system is given below.

2.2.1 Tags

Figure 2.2 shows an RFID tag which is the most important and basic component of an RFID system, because they store the information that describes the object

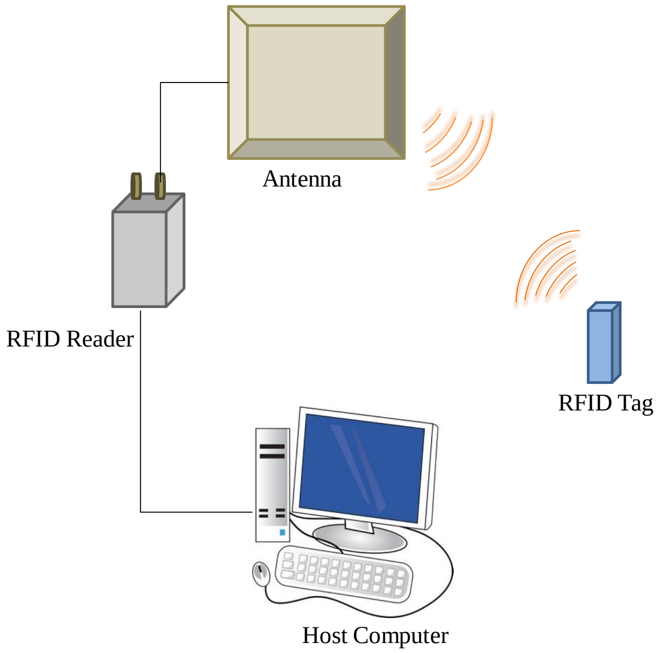


Figure 2.1: An Overview of RFID System

being tracked. RFID tag is a single-chip small device which is composed of a chip , an antenna and optional power source (depend on the type of tag) mounted on a substrate. These tags store specific object information in their memory and RFID readers access it through the radio signal. The chip can store as much as 2 kilobytes of data. Radio waves tuned on same frequency are used to transfer data between a tag and a reader. A transceiver sends a signal to the RFID tag in order to obtain the information from a tag, in return the tag transmits its information to the transceiver. The signal is then read by the transceiver, converts it to a digital format, and transmits it to a specified application such as a supply chain management system.

In past few years many different kind of tags have been introduced. Among them one kind of tags have very limited abilities and offers little more functionality than providing a unique identifier only. These tags can be manufactured at very low cost [Sarma, 2001] as they have very simple construction. Hence, it is economically viable to attach such tags to a huge number of objects, even very inexpensive ones (e.g., cups in a store). Furthermore, these tags are not application specific and can be used across application domains. Therefore, standards developed for managing RFID data are applicable to a wide cross-industry applications.

There are various kinds of RFID technology. In particular, some systems use

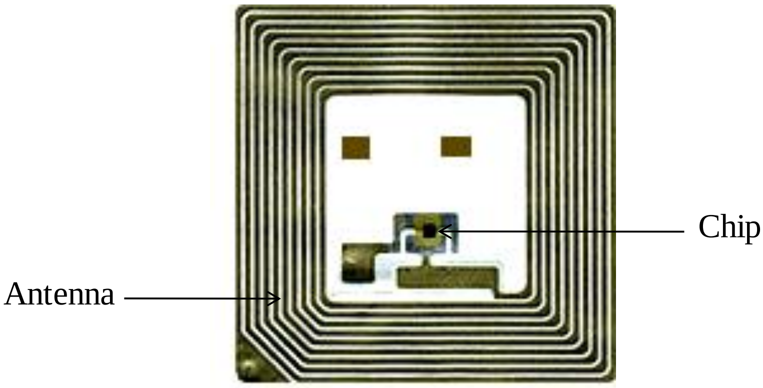


Figure 2.2: An RFID Tag

RFID tags that may obtain the energy to work from an on-tag battery while others use power from the electromagnetic radiation emitted by tag readers. Moreover, these tags may respond in different ways i.e., passively or actively [whi, 2002].

RFID tags can be categorized as active, passive or semi-passive and read-only, write-once, or read-write. Below is the description of each:

- *Active tags* refer to RFID tags that use an internal power source in form of battery, within the tag to continuously runs the microchip's circuitry. Active tags are capable of receiving extremely low-level RF signals from the reader/interrogators (since the reader/interrogator does not power the tag), and the active tag can boost the return signal back to the reader/interrogator. The read range of active tags ranges from tens of meters to hundreds of meters approximately 300 feet (100 meters). Because of continuous power to active RFID tags, they are normally used when a longer tag read distance is desired, regardless whether they are in the reader/interrogator field or not. Usually active tags are costly as compared to passive tags because of their size and construction. Active tags have internal battery, therefore, they have limited life of few years.

- *Passive tags* refer to RFID tags that do not have internal battery. Instead, they are powered by the RFID reader/interrogator. The reader/interrogator sends Radio frequency (RF) waves, which induce a current in the silicon chip on the tag when it is within range of the reader's RF field. This energy is temporarily stored by the tag in order to generate the tag responses back to the reader/interrogator. Usually, passive tags need strong RF signals from the reader/interrogator, because the strength of RF signal returned from the tag is restricted to very low levels by the small amount of energy. Passive tags are good choice to use when the tag and reader are close to each other.

The read range of passive tags is approximately 30 feet.

- *Semi Passive tags* refer to RFID tags that contain an internal power source usually a low cost battery which can be used to monitor environmental conditions (e.g. temperature), but not to boost range. Like passive tags, semi-passive tags still need RF energy transferred from the reader to power a tag response. The communication in semi-passive tags is still passive; no power is transmitted by the tags, just tags reflect back some of the power sent by the reader. The difference between semi-passive and passive tags is that semi-passive tags contain an internal battery, that can be used to complete some other functions e.g., to monitor environmental conditions (temperature, humidity) and that may extend the tag signal range.

- *Read-only tags* have specific data, such as a serialized tracking numbers that are pre-written onto them by the tag manufacturer or distributor and cannot be changed. Read-only tags are usually the cheapest, because they commonly need the least amount of memory and cannot have any additional information. They depend on an infrastructure and readily available database to retrieve useful information and any updates to that information have to be maintained in the application software.

- *Write-once Read-many tags* allow a user to write data to the tag once during production or distribution. This information can include a serial number or some other data, such as a lot or batch number.

- *Full read-write tags* enable a user to update a new data to the tag as needed.

2.2.2 Readers

A typical RFID reader is a device that converts radio waves from RFID tags into a digital form that can be passed to middleware software. Every RFID system requires a reader in order to retrieve the data stored on an RFID tag. Usually, a reader has one or more antennas that send radio waves and receive signal back form the tag and pass it on in digital form to the computer system. Depending upon the reader power output and the RF used, it sends radio waves in ranges of anywhere from one inch to 100 feet or more and it detects the reader's activation signal, whenever an RFID tag passes through the RF field. Some readers can also write remotely to the tags depending on the application and technology used.

An RFID tag reader communicates with the RFID chip by using antennas. Every application has its own reader requirements depending on the type of task. In order to make a successful system almost all applications require multiple forms

Figure 2.3: RFID Readers

of readers. There are a wide range of various reading systems and technologies. These include:

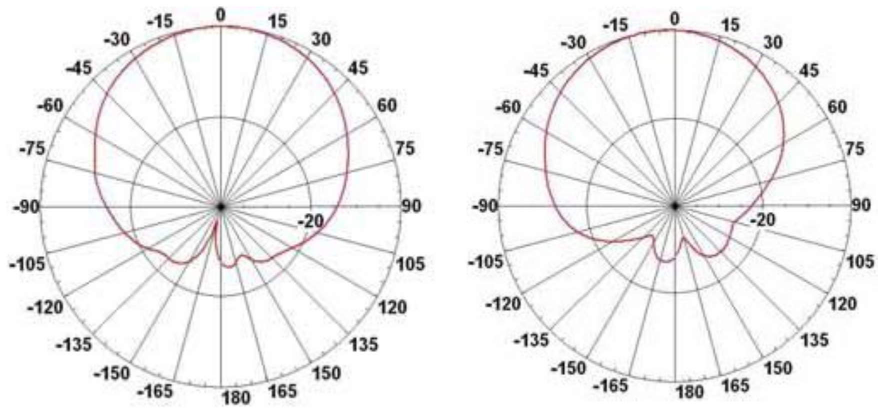
- *Handheld readers* that work like a handheld bar-code scanner.
- *RFID readers* embedded into mobile data collection devices.
- *Fixed readers*, which are mounted to read tags automatically as items pass by or near them.

A few examples of different readers are shown in Figure 2.3[1].

2.2.3 Antennas

In an RFID system, the antenna is a conductive element that allows the tag to exchange data with the reader. The RFID reader antenna transmits electromagnetic waves for communication and data transfer. Depending on kind of task and application, a variety of antennas can be used. The antennas are categorized by their characteristics such as polarization, apex angle, and gain. The right choice

[1] `http://www.skyrfid.com/RFID_Antenna_Tutorial.php`



(a) (b)

Figure 2.4: Elliptical Polarized Antenna (a) Elevation; (b) Azimuth

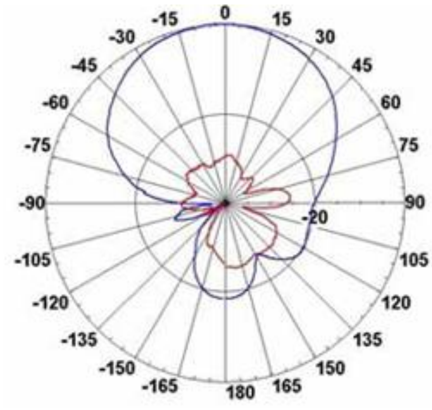
of antenna (characteristics) and receive sensitivity can achieve optimal fit to the read zone. Since the antennas are passive system elements, no tuning is required, which facilitates installation and maintenance.

Generally, RFID reader antennas are polarized in two ways: Elliptical Polarized Antennas and Linear Polarized Antennas shown in Figures 2.4 [2] and 2.5 [3]

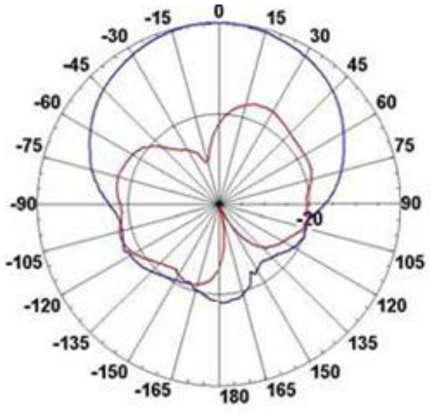
- *Elliptical Polarized Antennas*- emit the electromagnetic waves that propagate in two planes creating a circular effect, making one complete revolution in a single wavelength time frame. As the RFID antenna continuously emits a wavelength the rotational field will finally detect any tag that is in its path. Because of the wide apex angle of these antennas a large read zone can be achieved, which is required when a large quantity of tags need to be read at one time, or when tags moving at great speeds need to be interrogated. As the antenna is elliptically polarized, orientation of the tag relative to the antenna is not important: therefore, it is the best choice to use when tags orientation is unknown. The tag may be horizontally or vertically polarized along the line of sight of the antenna when the tag is in front of the antenna.

- *Linear Polarized Antennas*- emit the electromagnetic waves that travel entirely in one plane i.e., Vertical or Horizontal plane in the direction of the signal propagation. Because of the smaller apex angle, this antenna is more useful for selective data collection and restriction on read zones. The antennas field is either vertically or horizontally polarized,depending

[2] `http://www.skyrfid.com/RFID_Antenna_Tutorial.php`
[3] `http://www.skyrfid.com/RFID_Antenna_Tutorial.php`



(a)



(b)

Figure 2.5: Linear Polarized Antenna (a) Elevation; (b) Azimuth

on the direction of mounting, requiring the tag to have the same orientation. This is the best choice to use when the tag orientation is known and fixed. To achieve the best read rates the RFID antenna and RFID tag should be matched in polarization. Longer read ranges can be achieved with this antenna compared to the elliptical polarized antenna because of its greater gain.

The electromagnetic waves of RFID propagate in both vertical and horizontal dimensions. The field coverage of the wave and its signal strength both are partially controlled by the number of degrees that the wave expands as it leaves the antenna. The higher number of degrees means a bigger wave coverage pattern and lower strength of the signal.

- *Azimuth(AZ)* is the horizontal radiation plane of the antenna wave. It depicts the amount of expansion horizontally from the centerline of the antenna at a maximum variation of 3 dB by using degrees. (This 3 dB deviation is known as Beam width, it means when tag is near or at the 3 dB Beam width angle there will be 50% loss in signal strength)

- *Elevation (EL)* is the vertical radiation plane of the antenna. It depicts the amount of expansion vertically from centerline of the antenna at a maximum variation of 3 dB by using degrees. Elevation Beam width follow the same signal strength information as the Azimuth Beam width.

AZ and EL degrees can be different for both Linear and Elliptical polarized antenna thus providing different read patterns depending on specific requirements. The greater the number of degrees mean the wider the wave read zone and thus the lower the wave signal strength.

2.2.4 Middleware

The RFID Middleware provides an interface/platform between the interrogators and enterprise management systems to work, control and send the data captured by the RFID hardware.

RFID middleware differs from traditional middleware in a way that it works on the edge of the network and moves data at the same point of the transactions. The basic functionalities of RFID middleware includes monitoring, and data management devices. A major issue comes while implementing an RFID solution is the lack of sufficient middleware to link RFID systems and enterprise applications. In fact, middleware software or applications are required to manage the flow of data from readers and then send the data to back-end management systems. RFID middleware usually assist with the following tasks:

- Data retrieval from readers
- Data filtration for application software
- Generating inventory movement notifications
- Monitoring tag and reader network performance
- Capturing history
- Analyzing tag-read events for application tuning and optimization

2.3 Current and Potential Applications

This section includes a comprehensive overview of RFID's different current and potential applications. The RFID technology can be used in many applications [Nath et al., 2006, Domdouzis et al., 2007]. An RFID tag can be attached to any object and used to track and manage inventory, assets, people or animals etc. For example, these objects can be cars, computer equipment, books, mobile phones etc. It is possible that companies and RFID vendors will develop many new applications to solve common and unique business problems as soon as RFID technology advances and becomes less expensive and more robust. Some of the applications of RFID are discussed below.

2.3.1 Asset Tracking

Among the uses of RFID, it's one of the most common use is in asset tracking [Hayashi et al., 2003]. RFID tags can be attached to assets and virtually from any location one can monitor the location of assets, their status and availability, in order to manage the assets on demand by tracking their location and increase production by locating them at time when they are needed without delay [Lampe and Strassner, 2003]. Furthermore, lost or stolen assets or those which are not utilized can be tracked easily by using RFID technology.

2.3.2 Manufacturing

RFID technology has been used in manufacturing plants to track parts and work in process in order to reduce defects, increase throughput and manage the production of various versions of the same product [Günther et al., 2008, Strassner and Fleisch,].

As an item progresses with the time, manufacturers can track and record its in-process assembly information into the RFID tag. In this way, RFID tag maintains the current item information. For instance, an RFID reader can be used by assembly line personnel to verify which processes have been completed, in order to find which inspections or tests are further required and to automatically update the item state in central production database. Similarly, RFID tags can be used by production planners and inventory control personnel to automatically update the customer database and finished goods in the inventory. Thus, the use of RFID technology can avoid the errors in the system introduce by manually creating data entry sheets.

2.3.3 Supply Chain Management

A number of companies use RFID technology for supply chains management or to automate parts of the supply chain [Kärkkäinen, 2003]. The use of RFID may help in improving the visibility at multiple stages of the supply chain [SANTOSH and Smith, 2008]. For example, the RFID tag attached to the product at the production stage, can be tracked by RFID readers which gather information about the location of tagged products as they make their way from the manufacturer, to a warehouse or series of distribution centers, and to the final destination, store [Zhou et al., 2007].

2.3.4 Toll Road Payments

Now a days RFID technology has also been used for convenient and fast payments. In this regard, one of the most popular use of RFID is to pay for road tolls without stopping [Blythe, 1999]. The toll collection mechanism gets better by using RFID applications while improving traffic flow, as vehicles can pass through toll stations without stopping for payment. The affixed RFID automatically identifies the account holder and makes faster transactions without stopping the vehicles. On the other hand, this application can help to keep good traffic flow and to identify traffic patterns using data mining techniques that can inform the administration or decision support systems in order to report the traffic conditions or to extend and develop future policies [Shepard, 2005].

2.3.5 Security and Access Control

Another application of RFID is it's use as an electronic key which provides hands-free access control to office buildings or areas within office buildings [Sarma et al., 2003]. It has many advantages over traditional access control badges and systems. In traditional systems, such as bar-code, magnetic stripe, and proximity readers, user must hold the badge and place it close to, or make physical contact with the reader. While in case of RFID, user can have complete hand-free access as he/she does not need to hold up a badge , a key or swiping a magnetic stripe card to unlock a door. As there is no contact between the card and reader, there is less wear and tear, hence less maintenance. The use of RFID applications in secure areas, not only give permission to and revoke for the users/persons in particular area but also record individual access and the length of their stay in that particular area. At first low-frequency RFID tags has been used for access control systems. Now a days, long read range tags have been introduced by vendors.

2.3.6 Healthcare Applications

In healthcare, adoption of RFID applications has been widespread and very effective in order to have better patient care by saving important resources. The first combine use of active and passive RFID has been seen in hospitals, where active tags are used for tracking high-value, or frequency moved items, and passive tags are used to track only items to be read on small range. In healthcare, the number of errors can be reduced by attaching RFID tags to the medical objects e.g. patients files and medical equipment.

RFID further improves healthcare by providing RFID based timely information about the location of objects that would increase the efficiency and effectiveness of paramedical staff leading to improved patients care [Ahsan et al., 2009a, Hakim et al., 2006, Ahsan et al., 2009b].

An RFID system can also be used to track patients, doctors and expensive equipment in hospitals in real time. The ID bracelets of all patients can be affixed with RFID tags, or just patients requiring special attention, so that they can be tracked continuously [Kim et al., 2008]. Also the patient's RFID tag can be used to access his/her information for review and update through a hand-held computer.

Furthermore, these RFID applications can be combined with RFID access control to allow only authorized personnel to access to critical areas of the hospital or to restrict access to drugs and pediatrics.

2.3.7 Library Management

The RFID technology can be used for effective and efficient library management [Rizvi et al., 2005]. The RFID tag can maintain information about the library items, such as a book's title or material type, without having to be pointed to a separate database instead this information is read by an RFID reader.

In literature, RFID has many library applications that can be highly beneficial [Choi et al., 2006]. The RFID readers, tags and sensors can be installed in the library for its efficient management. Since RFID tags are readable through an item, therefore there is no need to open a book cover or DVD case to scan an item. RFID allows borrowers to scan an entire pile of books in one go, instead of one at a time. In this way, complete shelf of book can be cataloged within few seconds, without ever having taken off book from the shelf [Wadham, 2003, Ehrenberg et al., 2007]. Moreover, this technology provides a number of ways for checking out books and also helps in avoiding theft of books.

Chapter 3

RFID Data Management Systems

In last few decades, RFID technology has emerged significantly with many real time applications, such as product tracking and asset management, object and people authentication, health care etc. Nevertheless, the data management in these RFID applications poses a number of challenges [Chawathe et al., 2004]. In particular, the nature of an RFID data stream is noisy, redundant and unreliable, making it unsuitable for direct use in applications. For all these reasons, the unreliable data streams must be transformed into reliable streams that can be meaningful to applications. Since the nature of RFID data is significantly different from traditional data, thus traditional data management systems are not an appropriate choice for RFID applications. This originates need of an RFID data management system that particularly deals with RFID data management issues.

In this chapter, we give an overview of the fundamental issues faced by RFID systems and solutions provided in literature (Section 3.1). Finally, at the end, we present some of the famous RFID Data Management Systems from literature (Section 3.2).

3.1 Design Issues of an RFID Data Management System

The management of RFID data in transforming low-level streams into high-level events poses a number of challenges, since the nature of RFID data stream is noisy, redundant and unreliable. Regardless of a variety of RFID applications, an RFID-based information framework shares following common fundamental issues:

- *Incomplete, Unreliable and Duplicate readings of Data*: The data readings in RFID applications are inherently noisy and unreliable. The incomplete and unreliable readings meant to false positive and false negative readings.

False positive readings are those which are unexpectedly generate by RFID devices also know as noise reading while false negative readings are actually missing data readings i.e. when a tag is present in vicinity of RFID reader but reader failed to read that tag. Other than unreliable readings RFID applications usually face the problem of duplicated readings. This problem arises because multiple readers overlap the read ranges. Hence, these problems make necessary the use of filtering and smoothing techniques for RFID applications.

- *Data-Information Mismatch*: Usually, RFID devices generate low-level data streams in the form of TRE $(tag\_id, antenna\_id, time)$, which are not directly useable in applications. However, RFID-based tracking and monitoring applications require more meaningful and actionable information. Hence, these raw RFID data streams need pre-processing, where complex data-information transformation are carried out in order to resolve the mismatch between data and information.

- *Huge Volume of Data Streams*: RFID tags continuously send out their IDs at pre-programmed intervals. Usually, RFID applications receive high data volume which can easily overwhelm an application. Therefore, there is need of small space representation for massive RFID data streams while preserving the meaningfulness of the information.

In literature different deterministic and probabilistic approaches have been proposed for filtering and smoothing of RFID data streams. For instance, in [Jeffery et al., 2006a] authors proposed a framework named ESP (Extensible Sensor Stream Processing) for sensor data filtering and smoothing for use in pervasive applications. ESP introduces the concept of temporal and spatial granules and designs as a pipeline using declarative filtering procedures that involve deduping, removing outliers and smoothing data collected from various sources. ESP transforms the raw sensor data into cleaned data, hence applications remain unaffected by the unreliability of sensor data. ESP frameworks are easy to deploy and evolve because of following characteristics: (i) by using declarative queries ESP is very easy to program, (ii) it works as a pipeline i.e., cleaning stage can be programmed independently hence, reused across deployments, and (iii) it is particularly a cleaning framework designed to address directly the errors in the sensor data. In [Jeffery et al., 2006b, Jeffery et al., 2008] authors discussed sliding window based approach called SMURF. It is a declarative, adaptive smoothing filter for RFID data. Based on the characteristics of the underlying data stream SMURF automatically adjusts the window size; therefore, applications do not need to set the size of smoothing window. The main idea of SMURF is to model the unreliability of RFID readings to see RFID data streams as statistical sample of the tags in the physical world, and incorporates sampling theory techniques, such as binomial sampling and π-estimators, to perform its cleaning process. Another, deferred RFID data cleaning framework introduced in [Rao et al., 2006], where the RFID cleaning framework uses declarative sequence based rules at the time of query execution to correct RFID data anomalies. In [Deshpande et al., 2005a, Deshpande et al., 2005b], Deshpande et al. has discussed techniques based on probabilistic model in order to handle input errors and inaccuracies. These techniques are based on temporal and spatial correlations to predict missing values and identify outliers. Furthermore, these correlations also provide a way to give approximate answers to queries. Similarly, many other approaches that deal with the incomplete and duplicated RFID data presented in [Khoussainova et al., 2006, Mahdin and Abawajy, 2009, Darcy et al., 2010, Bai et al., 2006, Chen et al., 2010, Peng et al., 2008, Derakhshan et al., 2007].

Likewise, a number of probabilistic techniques have been proposed for the analysis and transformation of RFID low-level data streams into meaningful information in order to deal with data-information mismatch problem. These techniques, exploit the probabilistic nature of RFID data and manage their inherent uncertainty in the form of probabilities and correlations, so to achieve even higher effectiveness in the application scenarios they are applied to [Ré et al., 2008, Tran et al., 2009, Khoussainova et al., 2008a, Welbourne et al., 2008]. For instance, [Ré et al., 2008, Kanagal and Deshpande, 2008] generate probabilistic streams by inference on an HMM. Then, probabilistic inference is required in order to extract high-level complex events from the low-level atomic events acquired by the readings. For example, in tracking applications, the location of the objects is unknown to the system and observed low level sensor data is translated into precise and more reliable estimates about the location of these objects [Ré et al., 2008, Tran et al., 2009].

For RFID data compression, recently several proposals have been discussed in literature. A graph-based model is discussed in [Cocci et al., 2008] for providing the compression in RFID systems. This model captures the possible object locations and their containment relationships.However, high detection rates at the RFID readers required by graph model in order to have accurate results. In [Gonzalez et al., 2006c], a new model for warehousing RFID data has been proposed. The propose model provides significant data compression and path-dependent aggregates while preserving the object transitions. The proposed work basically takes advantage of object movements in bulk, of data generalization and the merge or collapse of the path segment that RFID objects follow. Though, this work did not handle missing and erroneous data tuples. In [Bleco and Kotidis, 2009] authors present an aggregation mechanism for RFID data streams based on temporal and spatial aggregations. The proposed algorithm exploits the time and space dimension to reduce the volume of input

RFID data streams. A special data cube termed as Flowcube is introduced in [Gonzalez et al., 2006a] for RFID systems. The Flowcube is a data cube computed for a large collection of paths. The Flowcube computes the movement trends of each specific item instead of computing aggregated measurements like in traditional data cube. Basically, the Flowcube examines item flows in an RFID system. Since, RFID data has different flow of information from the traditional data, therefore data storage and query processing tasks are difficult. Lee et al. has discussed this aspect of RFID data in [Lee and Chung, 2008]. They proposed an efficient storage scheme and query processing for supply chain management. They used an effective path encoding method to represent the flow information representing movements of products. A storage scheme is developed to process tracking queries and path oriented queries efficiently based on path encoding scheme and numbering scheme. Another approach for RFID data compression in supply chain scenario is presented in [De Virgilio et al., 2009]. This approach takes advantage of the property that objects move together in supply chain scenario and also this approach is capable of representing aggregations of objects which are not dependent on movements along the supply chain. In particular, this work represents an incremental aggregation approach based on various combinations of attributes describing RFID data other than paths and locations. Using this compression approach, the authors develop a lossless, relational-based storage model which preserves information about both path dependent and path independent items. In [Fazzinga et al., 2009], a lossy compression technique is proposed for RFID data streams. In particular, authors define a data structure to represent compressed RFID warehouses. Moreover, they proposed an architecture that gathers reading from RFID readers and store them in a compact way.

In the following, we have presented some of the famous RFID data management systems that address above mentioned design issues of an RFID data management system.

3.2 RFID Data Management Systems: State of the Art

3.2.1 Data Furnace

The *Data Furnace* project [Garofalakis et al., 2006] was developed at Intel Research and UC-Berkeley. The main objective of this project was to provide a probabilistic data management infrastructure for pervasive computing environments, appropriate for applications like the Digital Home. The proposed framework manages the uncertainty present in such data as a first class citizen through a standard framework based on probabilistic models and inference techniques.

Moreover, the *Data Furnace* offers a uniform and declarative means for higher-level applications in order to store, query and learn from given probabilistic data.

Data Furnace vision is to provide a central repository for both application data and metadata as well as a variety of various services at different level of abstraction that ranges from device and web connectivity, to data archiving, pattern learning and probabilistic reasoning.

The architecture of the *Data Furnace* consists of three layers: (1) The *Hardware Layer* for managing physical system resources, e.g. storage, processing and communication, (2) The *Metadata Layer* provides the repository for environment metadata. The basic *Data Furnace* interface between the physical world and higher-level applications is defined by this layer, and (3) The *Service Layer* is one of the most important layer of the *Data Furnace* framework. This layer basically offers the functionality of information management for the target application scenarios. Other services included by this layer are query processing and optimization, data archiving, complex event processing, pattern and model learning, probabilistic reasoning, and so on.

Some of the major probabilistic data management challenges addressed by the *Data Furnace* project are: (1) Voluminous Streams of Uncertain, Low-Level, Correlated Sensory Data, (2) Definition and Real-Time Tracking of Complex, Hierarchical Probabilistic Events, (3) Efficient Querying and Learning over both Probabilistic and Deterministic Information, and (4) System Support for Managing, Maintaining, and Reasoning over a Multitude of Probabilistic Models.

3.2.2 HiFi

The High Fan-in systems are basically distributed systems containing various receptors such as sensors, RFIDs at their edges and conventional host computers at their internal nodes, managed by using the notion of successive aggregation. The *HiFi* system [Franklin et al., 2005, Cooper et al., 2004] build at UC Berkeley, provides data management infrastructure for High Fan-in architectures. The architecture of the *HiFi* system based on the principle that the use of stream query processing and streaming views can be seen as a unified declarative infrastructure for data access across all of the different scales (i.e., an entire High Fan-in environment). As a result of successive stream query processing at each level of the system, data flow from edges to internal nodes. The *HiFi* system called this data flow as cascading streams. For programming simplification and optimizations *HiFi* makes use of stream-oriented query languages at each level. Many data manipulation tasks such as data cleaning, event monitoring, data stream correlation, outlier detection and aggregation, needed by High Fan-in systems can be done by using these stream-oriented queries.

There are three major components of the *HiFi* system that defines its functionality and services provided by it. (1) the *Metadata Repository (MDR)* is a globally accessible catalog for system-wide information. This metadata consist of three kinds: schema, views and system information. The schema hold by the *MDR* is the mediated schema of the system. All application queries and views are written on this schema. In case of *HiFi*, the mediated schema is the sensor and RFID data. The views stored in the *MDR* are those exported by each node in the system. The system information hold by the *MDR* comprises of node capabilities, authorization and privacy controls, and information relating to organizational boundaries and administrative domains. Moreover, runtime information such as the currently running queries on each node, present network usage, and unavailable/unreachable nodes to help guide and optimize system behavior are also maintained by the *MDR*, (2) the *Data Stream Processor (DSP)* is responsible for all single site data stream processing within a *HiFi* node. The *DSP* should simply have the ability to process continuous queries, add continuous queries and sources on-the-fly and cancel queries. In principle, the *DSP* could be any stream processor such as TelegraphCQ [Chandrasekaran et al., 2003] or Aurora [Abadi et al., 2003], and (3) the *HiFi Glue* runs on each *HiFi* node and provides binding between the systems. It manages its local DSP, made communication with other *HiFi* nodes, and manages incoming and outgoing streams. The *HiFi Glue* includes local and global sets of services.The *local HiFi Glue* services perform actions that involve local decisions only. The *Global HiFi services* require non-local knowledge and interaction with other nodes in the system. An initial prototype of the *HiFi* system demonstrated by the authors, which highlight the core technology of the HiFi approach and its different aspects such as data acquisition, data filtering, and data aggregation from multiple devices by using data stream query processing including sensor motes, RFID readers, and low power gateways organized as a High Fan-in system.

3.2.3 SASE

SASE (Stream-based And Shared Event processing) [Gyllstrom et al., 2006], is a complex event processing system that filters real-time RFID data and transforms the low level data to events that provide meaningful, actionable information to target applications. The *SASE* system allows applications to encode their complex logic for such data-information transformation by providing an expressive and user friendly event language [Diao et al., 2007]. This language is an extension of already existing languages such as complex event languages [Zimmer and Unland, 1999] developed for active databases and stream languages [Chandrasekaran et al., 2003, Rizvi et al., 2005] with additional support for sequence patterns involving temporal order of events, negation, value-based predicates, sliding windows, etc.

The proposed event language is efficiently implemented by providing query plan-based method. The approach is base on a new abstraction of complex event processing, that is a dataflow paradigm with pipelined operators as in relational query processing. Based on a Non-deterministic Finite Automata based model, the *SASE* system formulates native sequence operators that are capable of reading query-specific event sequences efficiently from continuously arriving events. The basis of each plan are then made by using these operators and also use for pipelining the event sequences to subsequent operators such as selection, window, negation etc.

The new abstraction of event query processing provides the optimization for two significant issues in complex event processing: large sliding windows and intermediate result sets. Generally, in monitoring applications large sliding windows are used and sequence generation from events in such windows are widely dispersed that can be an expensive operation. The *SASE* system develops optimizations to deal with this issue where these optimizations make use of novel sequence indexes to advance the sequence operators. Another aspect that affects the query processing is large intermediate results. To deal with this issue, *SASE* reduces the intermediate results by employing techniques [Zimmer and Unland, 1999] that drive some of the predicates and windows down to the sequence operators. Indexing relevant events both in temporal order and across value-based partitions are used for these optimizations.

The architecture of the *SASE* system consists of three layers: (1) the *Physical Device Layer* consists of RFID devices i.e. RFID readers, tags and antennas. In this layer RFID devices receive the data and passed it to the next layer, (2) the *Cleaning and Association Layer* performs two important functions. First, it performs filtering and smoothing functions [Wu et al., 2006] on received RFID data from physical device layer. Second, it creates events by using the attributes such as product name, expiration date etc. in order to facilitate processing and decision making in later components. Basically, this layer works in five sub-layers: (i) anomaly filtering layer, (ii) temporal smoothing layer , (iii) time conversion layer, (iv) deduplication layer, and (v) event generation layer, and (3) the *Complex Event Processor Layer* is responsible for continuous long running queries that are written in the *SASE* over event streams. For storage purposes the *SASE* includes a storage component for querying historical data and to join the stored data with the query results coming from stream processor. Moreover, *SASE* contains a user interface through which user can issue continuous queries for RFID stream and ad-hoc queries on the event databases.

3.2.4 Siemens RFID Middleware

The *Siemens RFID Middleware* [Wang and Liu, 2005] is an RFID data management system that brings together all technologies to make one efficient system. It is based on an event-based expressive temporal oriented data model i.e., Dynamic Relationship ER Data Model(DRER) for RFID data. The main purpose of the *Siemens RFID Middleware* is to offer a combined framework for RFID applications while providing following services: automatic data acquisition, filtering and transformation based on declarative rules; expressive data modeling and effective query support of RFID object tracking and monitoring.

The architecture of the *Siemens RFID Middleware* comprises of three basic components (1) the *RFID readers*, (2) the *Event Managers* that work at the front end of the system. This component basically receives data from the RFID readers, filters received data and passes it to the RFID Data Server which is the next component of the the *Siemens RFID Middleware*. The *Event Managers* can work in parallel at various clients and each *Event Manager* can connect to several readers while simultaneously processes the data generated from the readers. An *Event Manager* consists of three sub-components; (i) Reader Adapters: software component to support communication with RFID readers with integrated interface for RFID middleware to access readings, (ii) Filter: data filtering component to remove duplicates, errors from the raw RFID data, and (iii) Writer: for formatting RFID data with its standard language and send it to different target applications in the form of messages or streams, (3) the *RFID Data Server* consists of five sub-components; (i) RFID Data Manager: offers expressive data modeling, semantic data filtering, data aggregation, RFID object tracking and monitoring, and decision making support. The RFID Data Manager is composed of three layers: Data Processing Layer, Query Layer, and Decision Making Layer; (ii) RFID Data Store: gives schemas devised from DRER, and stores RFID data for RFID object tracking and monitoring, and decision making; (iii) RFID Data Archive: for storing non-active data into history partitions; (iv) Product Data Store: preserves static information associated with EPC objects; (v) Data Integration: is an application integration layer which combines the system with other applications. The *Siemens RFID Middleware*'s architecture is flexible and adaptable to various RFID applications while needing minimum configurations.

3.2.5 SPIRE

SPIRE (Scalable Processing of RFID Event Streams) [Cocci et al., 2007, Cocci et al., 2008] is a distributed system which provides correct interpretation of incomplete and noisy raw RFID data. Specifically, it uses probabilistic algorithms in order to infer locations of unobserved objects and inter-object relationships. Moreover, it carries out online interpretation, allowing online compression by identifying and discarding redundant data to deal with the enormous volumes of RFID data.

The *SPIRE* system aims to manage fundamental challenges faced by RFID data such as data-information mismatch for monitoring, incomplete and insufficient data for track and trace, scalability and low-latency by employing following methods: (1) the *Data Cleaning* techniques employed on data directly coming from readers to enhance reader reliability and raw data quality. Basically, this layer removes duplicate readings, and applies filtering and smoothing function to deal with error prone and missed readings, (2) the *Data Compression* layer reduces the volume of cleansed data coming from *Data Cleaning* layer. It also minimizes the inaccuracies in the data before passing it to the Event Generation phase, (3) the *Event Processing* phase of the *SPIRE* consists of two sub-phases. In first phase, event generator receives the compressed tag readings from *Data Compression* layer, generates event from them. In second phase, complex event processor allows a user to specify continuous queries over both incoming and historical data and monitors the generated event stream to search for the events that satisfy user defined continuous queries, and (4) in order to achieve the scalability the *SPIRE* system has introduce the concept of *Distributed Event Processing*. Up to this point, the compression and initial event processing are carried out at the local level. By employing Distributed Event Processing they can be integrated globally where further event processing can be done to monitor data at enterprise level.

The architecture of the *SPIRE* is an extension of above discussed *SASE* system. The architecture of local *SPIRE* has three layers. (1) *Physical Device Layer* consists of RFID devices, (2) *Cleaning, Compression and Association Layer* performs three important functions. First, it performs filtering and smoothing functions [Wu et al., 2006, Jeffery et al., 2006b] on received RFID data from physical device layer. Second, it employs compression in two ways on received data that is *location compression* and *containment compression* to efficiently reduce volume of RFID data between physical layer and event processor. Third, it creates events by using the attributes such as product name, expiration date etc. in order to facilitate processing and decision making in later components, and (3) *Complex Event Processor* is responsible for continuous long running queries that are written in the *SASE* over event streams. The architecture of the *SPIRE* was designed in such a way that it can address issues associated with large scale RFID based information systems. Finally, authors discussed about extension of *SPIRE* single warehouse to multiple local warehouses and one or more centralized global locations as in case of distributed networks. A prototype of *SPIRE* that implements inference & compression substrate is presented in [Cocci et al., 2012] and interpretation & compression substrate in [Nie et al., 2009]

3.2.6 Cascadia

The *Cascadia* system [Khoussainova et al., 2008b, Welbourne et al., 2008] was developed at the Department of Computer Science and Engineering, University of Washington with the aim to provide an infrastructure for user centric RFID based applications to specify, extract and manage meaningful high-level RFID events from raw RFID data. In particular, *Cascadia* has the following main contributions: (i) it enables application developers and user to specify events by using graphical interface providing an intuitive iconic language or by a declarative query language, (ii) it offers an API to facilitate the management of high-level events, and (iii) it efficiently extracts the specified events, passes them to target applications and stores them for later use such as for historical queries.

Cascadia consists of two main components: *PEEX* and *Scenic*. (1) *PEEX (Probabilistic Event Extraction System)* [Khoussainova et al., 2007b, Khoussainova et al., 2008a, Khoussainova et al., 2007a] is an RFID data management system that takes event specification as input in form of *PeexL*, a SQL-like event language for defining high-level events, continuously extracts the specified events from RFID data streams and stores them in the *At* relation. *PEEX* makes use of probabilistic techniques for event extraction from RFID data streams. *PEEX* contains two main components: the *Event Detector* for event extraction and the *Confidence Learner* to learn event confidences using RFID data collected within a specified time range. (2) *Scenic* [rfi, 2007] is a framework through which developers and user can specify the definition of spatio-temporal events graphically by using an intuitive visual language. *Scenic*, then translates these graphical event definitions into *PeexL*. *Scenic* iconic language composed of three basic components. (i) *Scenes* represent events in a sequence, (ii) *Actors* represent five kinds of entities in an event: person, group of persons, object, group of objects and place, each of these represented by a sperate icon, and (iii) *Primitives* represent primitive events such as with, without, inside, outside, near, far and lasts.

The system architecture of *Cascadia* is as follow: at the lowest level, it receives and stores the raw RFID data in the form of TRE coming from the RFID readers. These TRE are then processed by the particle filter (sampling based probabilistic inference method) to populate the *At* relations. These *At* relations passed to *PEEX* which continuously extracts and stores the higher-level events. Finally, for simplification *Cascadia* offers *Scenic* a user level tool for specifying events that helps non-experts users to specify common high-level events.

The authors demonstrated *Cascadia* by building a digital diary application in the form of calendar. *Cascadia* automatically populates the calendar with meaningful RFID events for the user. In order to deal with the ambiguity in RFID data, *Cascadia* transforms the raw RFID readings into probabilistic events and it is shown by authors that *Cascadia* approach outperforms deterministic event detection techniques.

Chapter 4

The RPDM System

In this chapter, we present the schematic architecture of our proposed *RFID Probabilistic Data Management (RPDM) System*. *RPDM System* is a probabilistic data management system particularly for RIFD data that applies probabilistic methods to filter raw RFID data streams and efficiently stores them in probabilistic database for query processing purposes. Furthermore, a general description of RPDM System and hardware details are also given in this chapter.

The chapter organization is as follow: Section 4.1 describes the *RPDM System* with its architecture; in Section 4.2 the details of RFID devices used for data acquisition in *RPDM* are discussed.

4.1 The RPDM Architecture

The *RPDM System* has been designed in a modular way. Figure 4.1 depicts the complete architecture of *RPDM System*.

At the lowest level of *RPDM* architecture there is *Data Acquisition Layer*, which consists of RFID devices including RFID tags, RFID readers etc. RFID tags are attached to the objects and people while RFID readers receive data from these tags in the form of radio signals and convert them in digital form to pass it to application softwares. Basically, these RFID devices are the source of raw RFID data streams. As already mentioned in previous chapters, raw RFID data streams are in the form of TRE: $(tag\_id, antenna\_id, time)$. The main focus of *RPDM System* is the part of the system architecture, which lies above the *Data Acquisition Layer*.

The second layer of the architecture is the *Data Management Layer*. This layer plays the primary role in the *RPDM System*, and thus, main focus of our research activity. This layer is composed of two modules: (1) *RFID Data Online Filtering & Uncertainty Management Module*, and (2) *RFID Probabilistic Data*

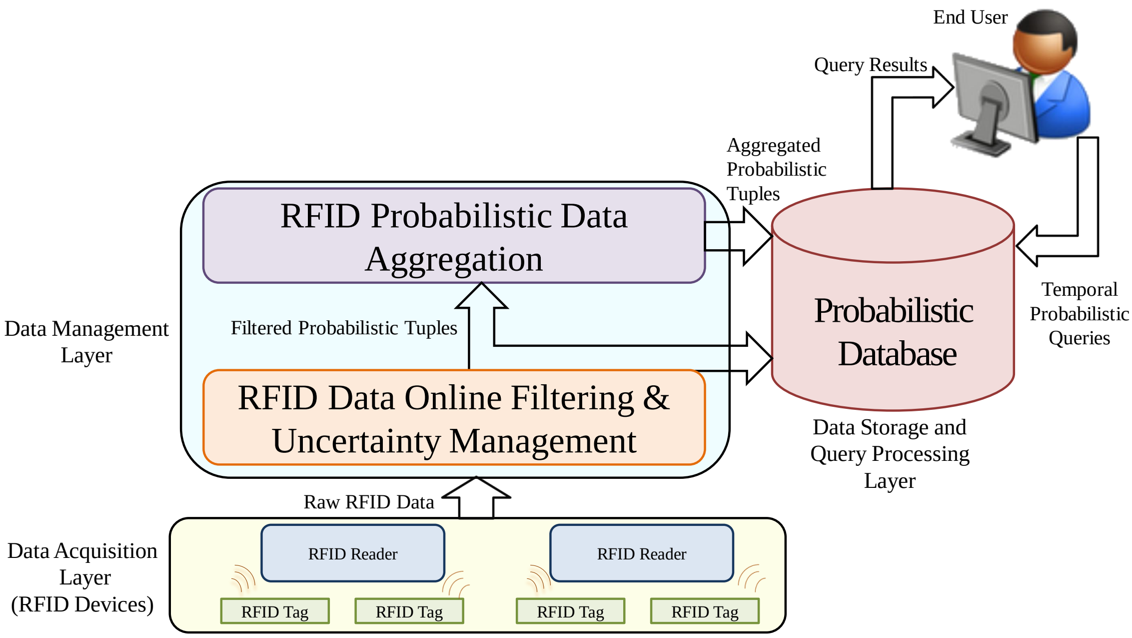


Figure 4.1: Architecture of RPDM System

Aggregation Module. A brief description of these modules is given in sub-sections 4.1.1 and 4.1.2. This layer typically lies between the RFID devices and application softwares to clean the received RFID data and gives filtered and meaningful data to applications.

The third layer is for *Data Storage and Query Processing* purposes. This layer includes a probabilistic database management system called MayBMS [may,]. MayBMS is a probabilistic state of the art DBMS developed at Cornell University and is configured as an extension of the famous open-source relational DBMS called Postgres [pos,]. It is a robust and scalable system for managing data that implements probabilistic and uncertain mechanisms for efficient representation and storage. In particular, probabilistic MayBMS stores data by means of these special U-relational database providing a complete and concise representation of the large number of possible worlds that are generated in the presence of probabilistic tuples [Antova et al., 2008]. It also provides an expressive query language that supports the entire set of capabilities offered by SQL and extends it with features designed to support the probability and to work with uncertainty, such as $conf(), aconf()$ for calculating the confidence of the tuple, $repairkey()$ to introduce the uncertainty and $argmax(), esum(), ecount()$ are the approximate aggregation functions. This language is sufficiently general because it adopts semantics independent of the details related to the mode of data representation and composition. Since it provides a comprehensive set of constructs for data transformation; therefore its compatibility with the relational algebra and standard SQL is important. It also includes several optimization techniques for query processing.

Finally, users can submit temporal probabilistic queries on the stored RFID probabilistic tuples in MayBMS by using its front-end interface.

4.1.1 RFID Data Online Filtering & Uncertainty Management Module

This module is responsible for transforming the low-level unreliable, noisy and redundant data incoming from RFID devices into more manageable and meaningful relational instances suitable for the interaction on the application level.

RFID Data Online Filtering & Uncertainty Management Module includes a specially designed data model which is based on well-known approach temporal graphical model, known as HMM. This module uses the online raw RFID data stream as input and produces the stream of probabilistically cleaned and filtered RFID data.

4.1.2 RFID Probabilistic Data Aggregation Module

The main goal of this module is to propose and implement a robust solution that can efficiently store data in probabilistic DBMS in an aggregated form without loss of information. In this module, we have proposed an aggregation mechanism which is able to provide small space representation for RFID data. The mechanism draws inspiration from the field of clustering.

The input to the algorithm is a re-elaborated version of the stream of tuples generated by the *RFID Data Online Filtering & Uncertainty Management Module* of *RPDM System*. The algorithm maintains a compact description of the processed tuples as clusters of tuples. The proposed algorithm tests boundary condition to discriminate when a cluster has to be closed in order to avoid distortion. To this end, this module makes use of three different models, which provide different indices for the boundary condition test.

The aggregation module processes probabilistic tuples as they arrive, hence avoiding the use of expensive and offline disk based operations such as sorting and summarization, and promptly stores the output in the probabilistic database MayBMS in such a way that a wide range of temporal probabilistic queries can be applicable and answered effectively.

Figure 4.2: i-PORT MB Reader

4.2 Data Acquisition

In this section, we discuss about the RFID devices used during the implementation of *RPDM System* for data acquisition purposes. Figures 4.2, 4.3, 4.4 and 4.5 present the RFID devices, which were used in this thesis. We use IDENTEC i-B2 tags, i-PORT MB readers and both Elliptical and Linear polarized antennas during our experiments in different scenarios. A brief description each of these is given below:

4.2.1 i-PORT MB

Figure 4.2 shows the i-PORT MB fixed reader used for our experiments in this thesis. The reader for the i-B series of Intelligent Long Range®(ILR®) broadcast tags is i-PORT MB. It can interrogate the i-B tags at distances of up to 300 feet (100 meters). The i-PORT MB reader establishes the connection to host system by using RS422 interface.

For data exchange a simple master/slave protocol is used by i-PORT MB reader. Other than data exchange between reader and tag, protocol can also give some additional information such as time of data reception, signal strength and the number of times the tag has been read by the reader.

4.2.2 Antennas

The IDENTEC SOLUTIONS antennas are shown in Figures 4.3 and 4.4, which were used in this thesis. The choice of antenna depends on the type and require-

Figure 4.3: Elliptical Polarized Antenna

Figure 4.4: Linear Polarized Antenna

ment of application. Primarily, *Elliptical Polarized* antennas are well suited to our purposes and have been used for final experimentation but some initial tests were also conducted by *Linear Polarized* antenna.

- *Elliptical Polarized Antenna*: It has wide apex angle of $(120°)$, which enables it to cover large read zone. Therefore, it is capable of reading a large number of tags at one time even with the fast speeds. The orientation of the tag relative to antenna is not important in elliptical polarized antennas.

- *Linear Polarized Antenna*: It is more suitable for applications in which read zones are restricted and data collection must be selective. This antenna has smaller apex angle of $(60°)$. The field of antenna is either horizontally or vertically polarized depending on the mounting direction, thus requiring the

Figure 4.5: i-B2 Active Tag

tag to have the same orientation. Since it has greater gain, therefore longer read ranges are obtainable with this antenna as compared to the *Elliptical Polarized* antennas.

4.2.3 i-B2 Tag

The beacon generation of IDENTEC SOLUTIONS ILR®active RFID tags is known as i-B2 series. Based on UHF radio frequency, i-B2 active tags are capable of providing long range for wireless applications, it can transmit data at distance up to the 300 feet (100 meters) to either fixed reader or handheld mobile reader. The i-B2 tag continuously sends static data written in its memory on pre-programmed interval known as ping rate that can be one second to four minutes. In our setup of RFID tags the ping rate is one second.

Due to the ultra-low power consumption of i-B2 active tags an operational lifetime of up to 6 years can be expected. They can be used in combination with i-PORT R2 and i-CARD R2 and several hundred tags can be detected simultaneously. In particular, i-B2 active tag is suitable for identification and tracking applications. Figure 4.5 shows i-B2 tag used in our *Intruder Localization and Identification* and *RPDM* systems.

Chapter 5

RFID Data Online Filtering & Uncertainty Management

The nature of an RFID data stream is inherently noisy, redundant and unreliable. Therefore, the management of RFID data generated by applications poses a number of challenges considering that RFID deployments produced huge amount of data. A fundamental relation for RFID applications is the location of people and objects over time. Ideally, it would be beneficial to organize RFID stream of tuples according to the scheme i.e. (Person, Location, Time) that can show the position of each person with an RFID tag on each instant. However, this becomes difficult due to the fact that RFID data in these scenarios are inherently inaccurate as mentioned above and there is data-information mismatch between the information to which the application is concerned and the information produced by the sensors. In order to filter out the inaccuracies and to manage the unreliability and data-information mismatch of RFID data, in this chapter, we propose an *RFID Data Acquisition, Online Filtering & Uncertainty Management* mechanism in the context of location tracking that operate on unreliable and imprecise RFID data streams in order to transformed them into reliable probabilistic data streams that can be meaningful to the application or by which we can extract information of the interest. For instance, in case of location tracking this information would be location of people or objects over the time. A common way of dealing with such kind of imprecise data is to built a model of the data and use stream of raw reading as input to the model. To this end, *RFID Data Online Filtering & Uncertainty Management Module* makes use of a temporal graphical model, the simplest of which is HMM that continuously infers hidden variables based on sensor readings. In order to perform inference on hidden variables HMM uses sample based algorithm called sequential Monte Carlo Particle Filtering. Finally, we conduct a series of real scenarios experiments showing the effectiveness and accuracy level of the proposed approach.

The rest of the chapter is organized as follow: In Section 5.1, we give an overview of the probabilistic graphical models with their representation (subsection 5.1.1) by giving details of used model HMM in the presented approach, inference methods (subsection 5.1.2) and learning procedures (subsection 5.1.3) in these models. Section 5.2 presents our presented module for filtering and uncertainty management with its representation (subsection 5.2.1), learning (subsection 5.2.2) and inferencing method (subsection 5.2.3). Section 5.3 discusses the experiments, we conducted in order to evaluate the performance of presented approach. Finally, in Section 5.4, we discuss the literature briefly.

5.1 Background : Probabilistic Graphical Models

5.1.1 Representation

Graphical models [Lauritzen, 1996] are the combination of probability theory and graph theory. They provide a natural tool for dealing with uncertainty and complexity problems that exist in many real world applications.The graphical models basically work on the concept of modularity; a complex system is built by combining simpler parts. Where the probability theory provides connection between these combined parts by ensuring that the system all together is consistent and gives ways to interface models to data. While graph theory of graphical models provides an easy and intuitively appealing interface to humans by which they can easily model high-interacting set of variables. Moreover, the resulting data structure not only lends itself naturally to the design of efficient general-purpose algorithms, but also can be used with all the well-developed and highly efficient graph algorithms suitable for the specific network topology.

Probabilistic graphical models [Koller and Friedman, 2009] are graphs in which nodes represent random variables. Arcs, or the lack of arcs, represent conditional independence assumptions. Therefore, they provide a compact representation of joint probability distributions. A number of multivariate probabilistic systems developed in various fields are the special cases of the general graphical model formalism. The examples of these models include Dynamic Bayesian networks(DBNs) [Mihajlovic and Petkovic, 2001, Murphy, 2002], HMMs [Rabiner, 1990], Kalman filters (KFM) [G:Welch and G.Bishop, 2002] and Ising models.

There are two types of probabilistic models: undirected and directed graphical models. DBNs are the example of directed graphical models of stochastic processes. They used to represent compactly the stochastic evolution of a set of variable over time where graph structure captures the complex interdependencies between the variables of the process. DBNs generalize HMM and linear dynamic system (LDS) [G:Welch and G.Bishop, 2002] by representing the hidden (and observed) state in terms of state variables, which can have complex interdependencies. The graphical structure provides an easy way to specify these conditional independencies, and hence, to provide a compact parameterizations of the model.

The difference between a DBN and an HMM is that in an HMM, the state space consists of a single random variable S_t. On the other hand, a DBN represents the hidden state in terms of a set of random variables, $S_{t^1}, ..., S_t^{Nh}$. The difference between a DBN and a KFM is that a KFM requires all the conditional probability distribution (CPDs) to be linear-Gaussian, whereas a DBN allows arbitrary CPDs. In addition, HMMs and KFMs have a restricted topology, whereas a DBN allows much more general graph structures.

The proposed solution, we presented in this chapter is based on HMM, so we will focus only on HMM and for readers, we will describe its further details in order to understand it better.

Hidden Markov Model

HMM is the simplest kind of DBN, which has one discrete hidden node variable and one discrete or continuous observed node variable per slice. HMMs are a often used model for time series data. They are used in various applications such as image recognition, pattern recognition, data compression and speech recognition. It represents probability distributions over sequences of observations.

Definition: The HMM formally defined as a finite set of discrete states (can be multidimensional), each of which is associated with a probability distribution. Since the states are discrete, transition among the states are controlled by set of probabilities called transition probabilities $A_{ij} = P(S_{t+1} = S^{(i)}|S_t = S^{(i)})$. In particular state observation can be generated, according to the associated probability distribution. It is only the observed value, not the state visible to an external observer hence, states are hidden.

Parameters of HMMs: In order to define an HMM, following parameters are required.

- N, the number of states in model $S = \{S_1, S_2, ...S_N, \}$
- M, the number of possible observations
- The initial state distribution $\Pi = \{\Pi_i\}$ where $\Pi = P\{q_1 = S_i\}, 1 \leq i \leq N$
- The state transition probabilities $\{A_{ij}\}$ where $A_{ij} = P\{q_{t+1} = S_j|q_t = S_i\}\ 1 \leq i, j \leq N$ where q_t denotes the current state. Normal stochastic constraints should be satisfied by

transition probabilities, i.e.

$$A_{ij} \geq 0, 1 \leq i, j \leq N \tag{5.1}$$

and

$$\sum_{j=1}^{N} A_{ij} = 1, 1 \leq i \leq N \tag{5.2}$$

- The observation/emission probabilities $B = b_i(j)$.

  $$b_i\left(j\right) = P\left\{o_t = v_k | q_t = j\right\}, 1 \leq j \leq N, 1 \leq k \leq M \tag{5.3}$$

  where V_k denotes the k^{th} observation and O_t the current parameter vector.

  If the observations are continuous then continuous probability density function is used instead of a set of discrete probabilities.

Assumptions for HMMs: Usually, following assumptions are made while defining an HMM for mathematical and computational tractability.

- **The Markov assumption**: The Markov property states that the next state depends only upon the current state in time. HMM based on this assumption are known as first order HMM. The first order HMMs are the frequently used though some efforts have been made to use the higher order HMMs too.
- **The stationary assumption**: This assumptions states that state transition probabilities are independent of the actual time at which the transitions take place.

  $$P\left\{S_{t_1+1} = j | S_{t_1} = i\right\} = P\left\{S_{t_2+1} = j | S_{t_2} = i\right\} \tag{5.4}$$

  for any t_1 and t_2
- **The output independence assumption**: According to this assumption the current observation is statically independent of the previous observations.

  If sequence of observation is given by $O = o_1, o_2, ...o_T$

  Then, according to the assumption for an HMM λ,

  $$P\left\{O | S_1, S_2, ..., S_T, \lambda\right\} = \prod P(o_t | S_t, \lambda) \tag{5.5}$$

Nevertheless unlike other two assumptions, this assumption is a kind of limitation to HMM. In some cases this assumption may not be good enough for system design.

Problems for HMMs: Given an HMM, there are three basic problems of interest that must be solved for the model to be useful in real-world applications.

- **Evaluation**: Given the observation sequence $O = o_1, o_2, ..., o_t$ and a model $\lambda = (N, M, \Pi_i, A_{ij}, b_i(j))$, how do efficiently compute $P(O|\lambda)$, i.e. the probability of the observation sequence, given the model?

  The most straightforward way to calculate the probability of the observation sequence is by using simple probabilistic arguments. Since there are N possible transitions from each state to other, there are N^T such state sequences.

  To reduce the complexity of the problem we can use the time invariance of the probabilities, which makes use of *forward variable* $\alpha_t(i)$. The complexity of this method, known as the forward algorithm [Rabiner, 1990] is proportional to N^2T, which is linear w.r.t., T. Similarly, we can define the backward variable $\beta_t(i)$ as the probability of the partial observation sequence, given that the current state is i.
- **Decoding/Inference**: Given the observation sequence $O = o_1, o_2, ..., o_t$ and a model $\lambda = (N, M, \Pi_i, A_{ij}, b_i(j))$, how do choose a corresponding state sequence $S = s_1, s_2, ...s_t$ which is optimal in some meaningful sense (i.e. best explain the observation). For each new measurement o_t, the Viterbi algorithm [Rabiner, 1990] can be applied in order to efficiently compute the maximum likelihood path through the states $S_{0:t} = s_0, s_1, ...s_{t-1}, s_t$ that best accounts for the sequence of measurements $O_{0:t} = o_0, o_1, ...o_{t-1}, o_t$.
- **Learning**: Given the observation sequence $O = o_1, o_2, ..., o_t$ and a model $\lambda = (N, M, \Pi_i, A_{ij}, b_i(j))$, how do adjust model parameter in order to maximize $P(O|\lambda)$. There may be various optimization criteria for learning depending on the application. Generally, two main optimization criteria found in literature; Maximum Likelihood (ML) [Bishop and en ligne), 2006] and Maximum Mutual Information (MMI) [Chow, 1990].

5.1.2 Inference

In various real-world data streams, the elements of interest may not be directly observable (e.g. location information in raw data stream coming from RFID tracking application [Ré et al., 2008, Kanagal and Deshpande, 2008, Tran et al., 2009]), or it may be very expensive to measure them. A common way to process such kind of data streams is to continuously infer the value of the hidden variables by using observed data. Different types of methods allow us to combine prior domain knowledge about the system behavior with the actually observed variables to compute the best possible estimate of the hidden variables. This task is known as "inference". A number of inference techniques have been developed for efficient inference in special cases (e.g. Kalman filters, HMM), and many general purpose inference algorithms (e.g. Monte Carlo techniques [Doucet et al., 2001], junction tree algorithm [Madsen and Jensen, 1999, Jensen and Jensen, 1994]) are also adopted.

In the perspective of graphical models, inference is generally the computation of a particular distribution in a graphical model given evidence. In case of dynamic probabilistic models, this distribution could be the marginal probability of a node, the joint probability of a set of nodes, or the conditional probability of one set of nodes given another in addition to many other possibilities to consider in terms of applying evidence and querying distributions. We can compute the joint probability of the whole network given the evidence in order to estimate these distributions and then marginalize out the unwanted variables. There are many orders in which we can perform marginalization of the variables in which we are not interested.

Exact Inference

In simple cases, such as linear dynamic systems (LDS), "exact inference" algorithms describe efficient ways of performing this marginalization while handling the intermediate terms that arise as efficiently as possible. Moreover, in these cases exact inference is tractable. Many exact inference algorithms fall into two classes. First is the query driven algorithms such as variable elimination [Zhang and Poole, 1994], bucket elimination [Dechter, 1998], while second is the message passing algorithms such as junction tree algorithm. However, in general, for many graphical models in real-world applications exact inference is infeasible. Usually, real-world scenarios have high data rates while exact inference on a specific model is not capable enough to keep up with the these high data rates. Additionally, graphical models need a certain minimum number of nodes and edges in order to accurately model physical dynamics. On the other hand, most of the exact inference algorithms face severe challenges for large, densely connected models with high update rates.

Approximate Inference

In order to handle the intractability in real-world scenarios "approximate inference" algorithms have been developed. In the case of continuous state-spaces, when everything is linear-Gaussian, exact inference can be applicable but if we have all non-linear or non-Gaussian distributions, exact inference is not possible. In such cases, approximate inference should be used. Approximate inference is further applicable in two ways: Deterministic and Probabilistic.

Deterministic algorithms are those that usually use the structure of exact inference algorithms in order to make ceratin approximations while performing inference tractably. For instance, belief propagation algorithm is an exact inference algorithm for acyclic undirected graphical models but the same algorithm is applicable to cyclic graphical models for obtaining approximate results. This algorithm is known as loopy belief propagation [Murphy et al., 1999]. The Boyen-koller [Kwon and Murphy, 2000] algorithm, is another example, which is the simple modification of junction tree algorithm.

A *probabilistic* representation is a good way to communicate uncertainty to higher level modules and reasoning probabilistically is appealing because it naturally accounts for the ambiguity and uncertainty of sensor data. There are situations in which models consists of continuous, non-Gaussian random variables or non-linear statistical relationships among variables may originate the integrals that evolve in belief propagation making them impossible to compute. In most of the problems of interest the relationship between the hidden variables and observed measurements is not deterministic, so probabilistic method must be used. There are a number of algorithms for handling arbitrary continuous distributions in graphical models. The important classes of stochastic algorithm includes sample based sequential Monte Carlo methods known as "particle filters" and Markov Chain-Monte Carlo (MCMC).

Next, we describe and discuss several general methods that have been used for processing raw sensor readings into useful information, emphasizing probabilistic approaches.

Probabilistic Inference techniques

Maximum Likelihood Estimate: Maximum likelihood is a commonly known method for performing probabilistic inferences in the absence of prior assumptions. The relationship between observed measurements and state variables is often described by a state-conditional probability $P(O_t|S_t)$. Given an observation O_t, the maximum likelihood estimate of the state is given by the state that maximizes the state-conditional probability:

$$\hat{S}_t = argmax_{st} P(O_t|S_t) \tag{5.6}$$

Maximum A Posterior (MAP) Estimate: A maximum *a posteriori* probability (MAP) [Gauvain and Lee, 1994] estimate is a mode of the posterior distribution. MAP estimates depend on prior probabilities about the state variables. For the continuous state case, the *a posteriori* probability distribution of state given a measurement is given by Bayes' rule:

$$P(S_t|O_t) = P(Ot_t|S_t)P(S_t)/P(O_t) \tag{5.7}$$

MAP can be considered as a regularization of MLE, the only difference between MAP and ML is that MAP includes *a prior* assumptions while ML does not consider this assumption. Usually, it is easy to make reasonable prior assumptions about object's location for tracking applications, thus MAP can be an easy way to improve accuracy.

MLE and MAP are memoryless techniques which mean they lack the ability to exploit the dynamic models of the tracked user, such as expectations of possible speeds and feasible paths. They also do not consider past measurements in computing posterior probability distribution and just concentrate on instantaneous position.

Recursive Estimates: In order to address above mentioned problems with memoryless inferencing techniques recursive filtering techniques have been developed. These techniques maintain a probabilistic distribution of state that implicitly includes the effect of all past measurements and dynamic assumptions. They also give a way for updating this distribution with new arrived measurements. By looking at history, a recursive filter looks at the path of a tracked user instead of just instantaneous position like the memoryless techniques discussed above. Recursive filters are able to examine a sequence of measurements in time, along with a dynamic model, which is an effective way to deal with ambiguous measurements, thus these abilities make recursive filtering the best way to process sensor data in many different applications. Below, we discuss three recursive filtering techniques: Kalman filter, HMM, and Particle filter.

- *Kalman Filter*: The basic idea behind the Kalman filter is to simplify assumptions about both the measurement process and system dynamics. Actually, Kalman filter is a special case of general inference algorithms for DBNs and refers to specific analytical inference algorithm for the LDS. It uses observed variables over time, containing Gaussian noise and other inaccuracies, and the conditional distributions in order to estimate a distribution over the hidden variables. The Kalman filter assumes that the relationship between the measurement vector O_t and state Vector S_t is linear with zero-mean, additive, Gaussian noise. It also assumes that the relationship between the previous state S_{t-1} and current state S_t is linear with zero-mean, additive, Gaussian noise. Mathematically, these assumptions are

$$S_t = F_k S_{t-1} + W_{k-1} O_t = H S_{t-1} + V_k \tag{5.8}$$

Where F_k is an $m * m$ matrix, which is the state transition model, H is an $l * m$ matrix, the observation model, W_{k-1} is the process noise and V_k is the observation.

The Kalman filter can be represented in a single equation, though, it is usually described as two distinct phases: Predict and Update. In the predict phase, also known as *a prior state estimate*, it estimates the state at current timestep by using the state estimates from the previous timestep while does not include observation information from the current timestep. The Kalman update equations [Gelb, 1999], also termed as *a posterior state estimate* are used to improve the predicted estimates. It basically give a simple means of updating the previous state vector with new measurements using closed-form matrix math, in order to refine the state estimate. However, Kalman filter has some limitations: (i) It only works for linearly defined system dynamics. This means that sharp turns can be hard to model, and it does not provide any means of constraining a path from passing through a wall or other barriers; (ii) Measurements must also be linear in state. Most sensor models must be greatly simplified to adapt to this assumption; and (iii) Representation of state is Gaussian. This means the belief state must be unimodal, which is inappropriate and often very simple for many problems, especially those involving discrete variables.

More complex systems, usually, can be nonlinear. These nonlinearities have been addressed with the Extended Kalman Filter (EKF), where the basic Kalman filter is modified to deal with multimodal distributions. Several techniques for reasoning about data association has developed by radar tracking community in the context of Kalman filtering [Blackman, 1986]. In its basic state, though, the Kalman filter has been surpassed by HMMs and particle filters.

- *Hidden Markov Model*: See Section 5.1.1 for details

- *Particle Filter*: The Particle filter is a Monte Carlo sampling based technique for implementing recursive Bayesian filter [Doucet et al., 2000]. The basis of the method is to represent the posterior density by a set of random particles with associated weights and then compute estimates based on these samples and weights. The higher weights specify more probable states. In contrast to HMM, where the states are discrete and predefined, here each of these particles is continuous and evolves as new measurements are processed. When a new measurement O_t arrived, a new set of particles is computed in two phases: Predict and Update.

- Predict: Each particle is modified according to transition probability $P(S_t|S_{t-1})$ in order to generate randomly a new sample $S_t^{(i)}$ from each $S_{t-1}^{(i)}$.
- Update: Then each particle's $S_{t^{(i)}}$ weight is re-evaluated according to state- conditional probability $P(O_t|S_t)$. In order to avoid the degeneracy and loss of diversity, usually re-sampling step is performed after updation phase.

These weighted samples give a versatile way of approximating a posterior state probability distribution. Only knowledge of the transition probabilities $P(S_t|S_{t-1})$ and the state-conditional probabilities $P(O_t|S_t)$ required by each iteration during two phases of algorithm. These two probability functions can be arbitrarily complex, allowing for the modeling of realistic dynamics and sensors.

The Particle filters have high computational complexity. In addition, it is difficult to determine the optimal number of particle in advance. To deal with this Particle filter exploits the independence of the state variables such as Partitioned Sampling [MacCormick and Blake, 2000].

5.1.3 Learning

A probabilistic graphical model usually represented by the Conditional Probability Distributions (CPD), which are referenced as parameters of the model. These CPDs are used to define the transition model $P(S_t|S_{t-1})$ and the observation model $P(O_t|S_t)$. "Learning" is the process of estimating these parameters from training data.

Maximum Likelihood Estimation (MLE) [Myung, 2003] is one of the most widely used statistical techniques to learn the parameters of a CPD. From the training data, this provides an estimates of the values of the parameter θ of the CPD, which maximizing the likelihood of observing that data. Specifically, given a data sample $X_1, ..., X_n$, assumed to be independent and identically distributed (iid) from a parametric distribution with unknown parameters, the purpose of MLE is to estimate the value of the unknown parameters. If the probability distribution has the form f parametric parameters θ, the approach is to maximize the likelihood function:

$$L(\theta) = f(X_1, X_2, ..., X_n|\theta) \tag{5.9}$$

And, since the samples are iid, the expression can be simplified in the following:

$$L(\theta) = \Pi^i f(X_i|\theta) \tag{5.10}$$

Or, if we take the logarithm of the above function

$$log(L(\theta)) = \Sigma^i log(f(X_i|\theta)) \tag{5.11}$$

In this way, the MLE method allows to derive the joint probability distribution $P(O_t, S_t)$. Finally, by applying Bayes' Theorem, we can obtain the conditional probability distribution of the observations $P(O_t|S_t)$.

5.2 RFID Data Acquisition, Online Filtering & Uncertainty Management

The detailed block diagram of phase I of *RPDM System* is shown in Figure 5.1. It receives the raw data from RFID devices and performs online filtering and uncertainty management and finally, stores the probabilistic filtered tuple in probabilistic database. The details of the *Online Filtering & Uncertainty Management Module* are discussed in subsequent sub-sections.

5.2.1 Representation

This module of *RPDM System* includes a specifically designed data model based on HMM. As we discussed in Section(5.1.1) the HMM is graphical model typically used in temporal contexts. HMM infers the state of "hidden" variables, i.e. when the variables are not directly observable, based on a sequence of events that are observable somehow, in connection with the hidden variables. For example, consider the reference scenario of location tracking, where the interest of the application is to infer the position of people or objects over time (which are not being directly observable itself and considered as the hidden variables) on the basis of RFID readings collected by the reader (observable events, or simple observations). Thus, this module uses an HMM to produce, at each timestamp, a distribution over each tag location (i.e. the hidden variables or states) based on observations that, being sensor readings or observations. These observations include four type of information: 1) the identifier of the tag the reading is concern to; 2) the identifier of the antenna the tag is seen by; 3) the timestamp of the reading; 4) the Received Signal Strength Indicator (RSSI) of the reading. Nevertheless, the main important feature of this kind of models is that they allow to combine prior domain knowledge about the system behavior with the actual observations

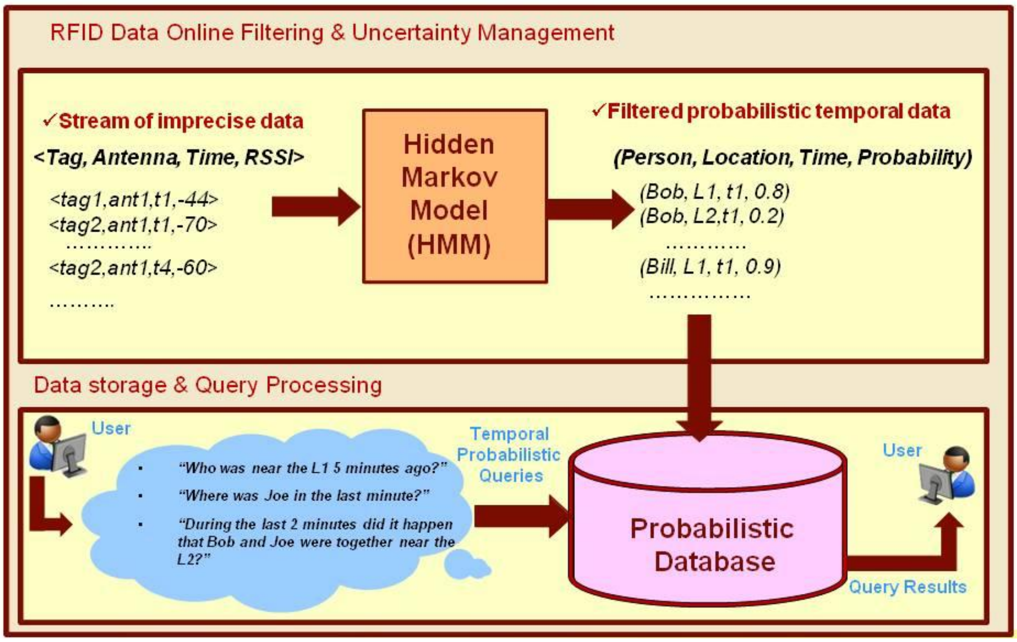


Figure 5.1: Block Diagram of Phase I of RPDM System

to compute the most likely values of the hidden variables. While observations are directly evaluable, the prior knowledge about the system is represented by Conditional Probability Distributions (CPD) which are referenced as the parameters of the HMM. A graphical representation of designed HMM for *RFID Online Filtering & Uncertainty Management Module* is shown in Figure 5.2. The nodes of the graph represent the variables (hidden states and observations) of the modeled system, while the directional arcs represent the concept of "causality" whose degree is indicated by the corresponding CPD. Specifically, darker nodes in the graph represent the observations and thus correspond to measurements collected by RFID antenna, while clear nodes represent states and thus coincide with the positions of the people. In the graphical representation in Figure 5.2 time is shown through the use of vertical "lanes"; each of single lane is therefore represents the situation of the system in a single instant in time. In this regard, it is noted that,according to the well-known Markov principle, these models typically assume that the variables at time t directly depend on the variables at time t and $t-1$ only and, hence, two consecutive time instances are sufficient for completely representing the whole system. The other parameters of the HMM, or the CPD that describe the relationship of causality between the variables that are represented as directional arcs in Figure 5.2, are listed below:

1. the *initial states distribution* $P(L_0)$, encodes knowledge about the initial state of the system (i.e. the time instant 0);

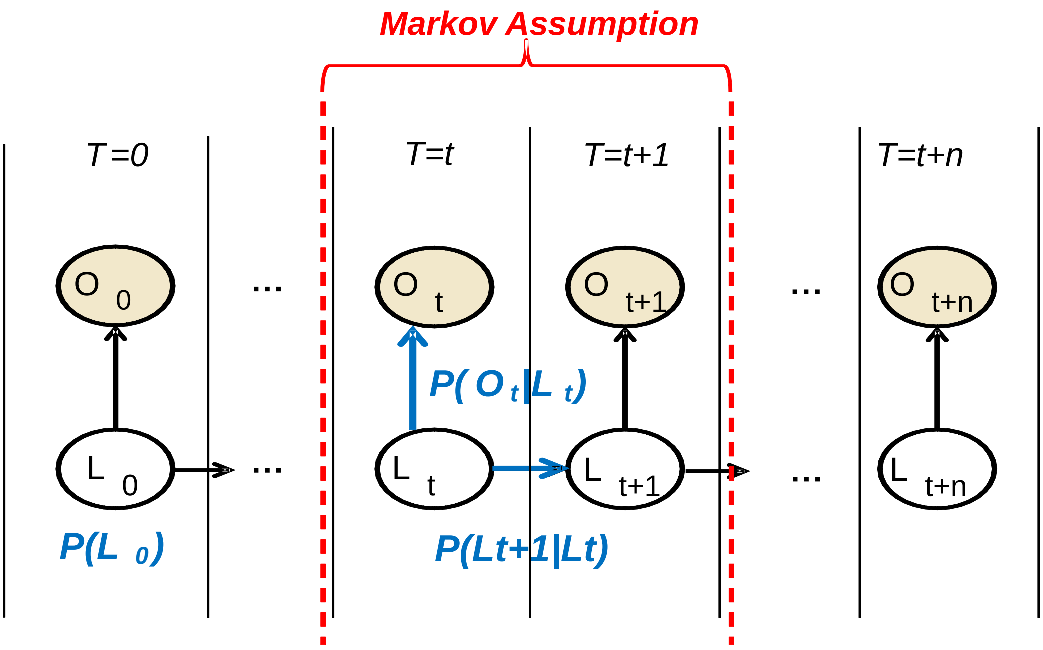


Figure 5.2: Graphical Representation of Hidden Markov Model Used

2. the *transition probability distribution* $P(L_{t+1}|L_t)$, encodes the knowledge of how the state of the hidden variables at time instant $t+1$ depends on the state at time instant t;

3. the *observation probability distribution* $P(O_t|L_t)$, encodes the knowledge of how the observations at time instant t depend on the state of the hidden variables at time instant t.

5.2.2 Learning

In the effective implementation of *RFID Online Filtering & Uncertainty Management Module* in the context of location tracking, these CPDs are modeled respectively, as follow:

1. the *initial states probability* $P(L_0)$: we assumed this parameter to be a uniform distribution among all the possible locations;

2. the *transition probability*$P(L_t|L_{t-1})$: It is modeled as a matrix whose rows ad columns are associated to the available locations so that each cell $[i,j]$ contains the probability value of having a movement from location i to location j (as an example, if two locations are separated by a wall the corresponding cell contains the value 0)

3. Finally, the *observation probability* $P(O_t|L_t)$: This information is typically not available and, thus, has to be learned from training data. To this end, we adopt the popular statistical method called *Maximum Likelihood Estimation (MLE)* (discussed in subsection 5.1.3) which, given learning data, estimates the value of the probability function parameter that maximizes the likelihood of the observed data (i.e. that makes the learning data "most likely"). Actually, MLE allows us to compute the conjunctive probability $P(O_t, L_t)$, from which observation probability $P(O_t|L_t)$ can be easily computed by applying the Bayes theorem. In the current implementation of *RFID Online Filtering & Uncertainty Management Module*, we have integrated an open-source statistical package called GRETL [gre,] that allows to exploit many different statistical techniques as well as MLE.

5.2.3 Inference

Nevertheless, the final aim of modeling a stochastic process with an HMM is to obtain the *posterior probability distribution* $P(L_t^{\tau_i})$ over the hidden variable $L_t^{\tau_i}$ (location of tag τ_i at time instant t) given the observed measurements. This task is called "inference" and different algorithms can be used to this purpose. Among the others, we decide to exploit a popular Monte Carlo algorithm called *Particle Filtering* [Arnaud et al., 2005], usually adopted in sample-based inference processes. The algorithm works by computing and constantly maintaining sets of particles to describe the historical and present states of the model. Figure 5.3 represents a schema of the steps executed by the algorithm at each time instant t. Specifically, given the observed values $o_t^{\tau_i}$ for each identified tag τ_i, the algorithm works by iteratively executing the following steps:

Initialization: during this phase, an initial set of particles is created by randomly sampling from the initial states probability $P(L_0)$.

Prediction: during this phase, the state of hidden variables at time t is estimated by using their state at time $t-1$ and exploiting the parameters of the HMM. More precisely, for each existing particle p_{t-1}^i at time $t-1$ a new particle p_t^i is created for time t by sampling from $P(L_t|L_{t-1})$.

Filtering: in this phase, the observation o_t arrived at time t are used to update the states previously estimated for time t. More precisely, each particle p_t^i is assigned a weight based on the values of the observed variables at time t and on the observation probability $P(O_t|L_t)$. This weight is proportional to $P(O_t = o_t^{\tau_i}|L_t = \lambda)$ where λ is the location of p_t^i.

Re-sampling: in this phase, the particles created in the *Filtering* step are re-sampled in order to generate a new set of particles, all with the same weight. This task is necessary in order to avoid degeneracy, i.e. the case where a single particle

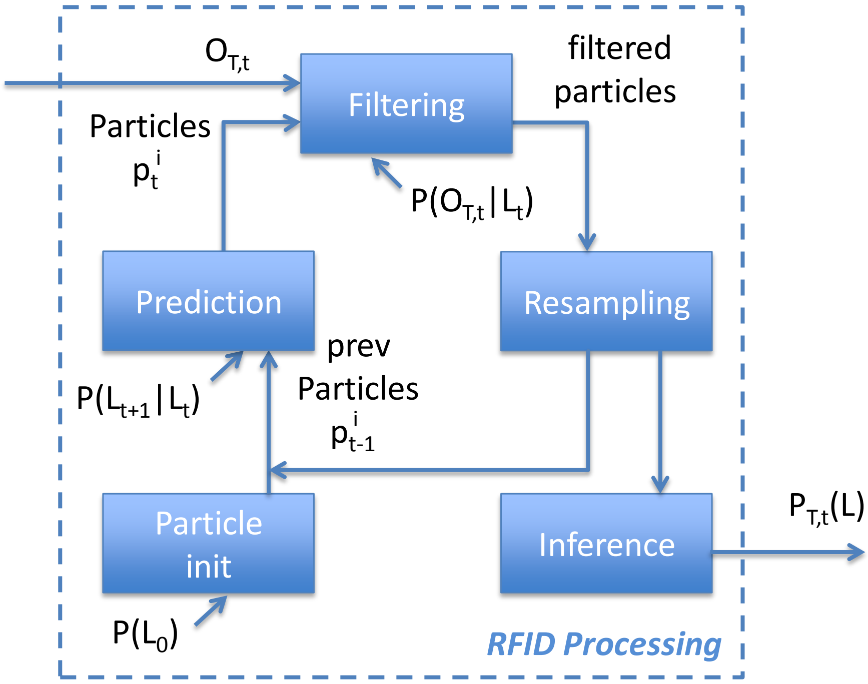


Figure 5.3: Detail of RFID Online Filtering & Uncertainty Management Module Schema

has all the weight.

Broadly speaking, each particle p_t^i represents a guess about the location of tag τ_i. Then, after a number of iterations, the inference task is performed: to compute the posterior probability $P\left(L_t^{\tau_i}\right)$ we can indeed simply count the number of particles in each location and divide it by the total number.

5.2.4 Data Storage & Query Processing

As shown in Figure 5.1, the result of applying Particle Filtering on HMM is the transformation of raw data collected by the RFID tags into a stream of "filtered" tuples organized according to the schema $(Person, Location, Time, Probability)$. In particular, the last element of the schema represents the probability i.e. the degree of correctness of reported data in a tuple. Therefore, generated tuples inherently probabilistic in nature, hence, need to stored by means of an appropriate tool that can handle the associated uncertainty.

For this purpose, *RPDM System* includes a separate probabilistic database management system called MayBMS (see Section 4.1 for details). Figure 5.4 shows few examples of supported temporal probabilistic queries for our stored probabilistic data.

Q1
```
--Who was at ‘L1’ in first minute?
SELECT tagId, conf()
 FROM Pw
WHERE LocationId='L1'
AND instant<=(select starttime()
 + interval '00:01:00')
GROUP BY TagId;
```

Q2
```
--Where was ‘P1’ in the last 20 seconds?
SELECT LocationID, conf()
FROM Pw
WHERE TagId=‘P1'
AND instant >= (select endtime()) - '00:00:20'
GROUP BY LocationId;
```

Q3
```
--In the last 2 minutes was it that the
 ‘P1’ and ‘P2’ were simultaneously
 present at ‘L2’? If so, when?
SELECT p1.instant,conf()
FROM Pw p1, Pw p2
WHERE p1.TagId=‘P1‘
AND p2.TagId=‘P2'
AND p1.LocationId= 'L2‘
AND p1.LocationId=p2.LocationId
AND p1.instant = p2.instant
AND p1.instant>= (select endtime()) - '00:02'
group by p1.instant
```

Q4
```
--Was ‘P1'at ‘L1’ 1 minute ago?
SELECT conf()
FROM Pw
WHERE LocationId ='L1'
AND tagId= ‘P1'
AND instant = '17:18:49' - interval '00:01:00' ;
```

Figure 5.4: MayBMS (SQL) Example Queries

5.3 Experimental Evaluation

In this section, we discuss the different experiments that we have conducted in order to evaluate the effectiveness of our proposed approach.

5.3.1 Experimental Setup

For evaluating the effectiveness of our presented approach, we performed experiments in different scenarios, collecting data from persons wearing RFID tags. The experimental scenarios are all set in three indoor locations (denoted by $L1$, $L2$, and $L3$) and capture different possible movement behaviors. Figure 5.5 shows the overview of the testbed where it gives an idea of the setup with locations represented by bounded areas and the antenna indicated by black color. Upon this setup, we have collected data from RFID tags in two different scenarios: 1)“ No Stay”, where people rapidly move between locations without staying on any specific one; and 2)“Stay”, where people move between locations and spend some time on each of them. Both types of scenarios have been tested with one/multiple tags.

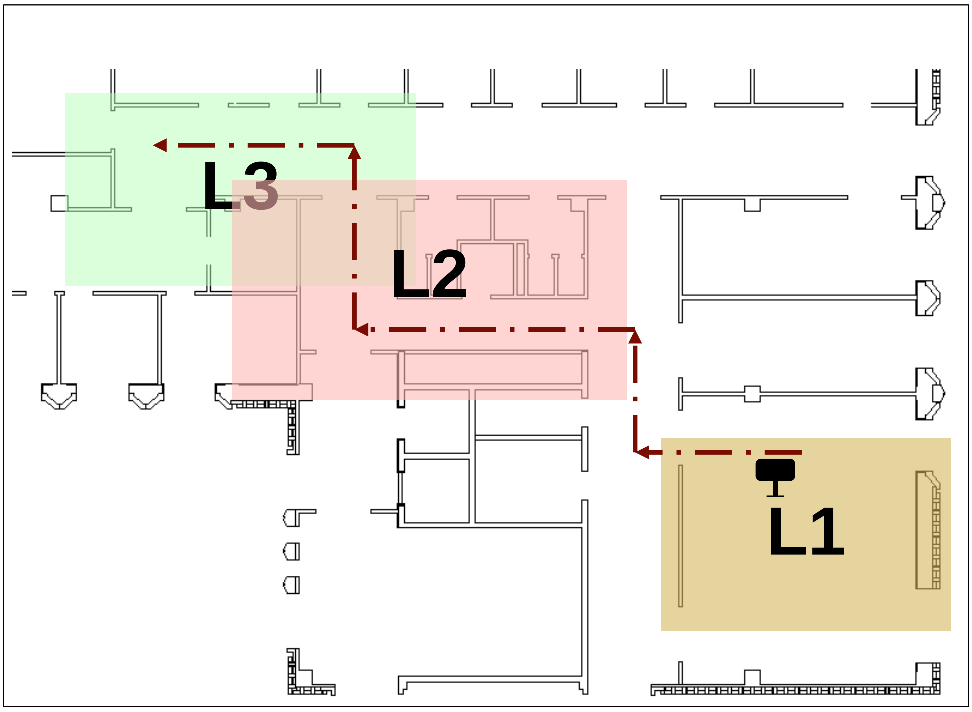


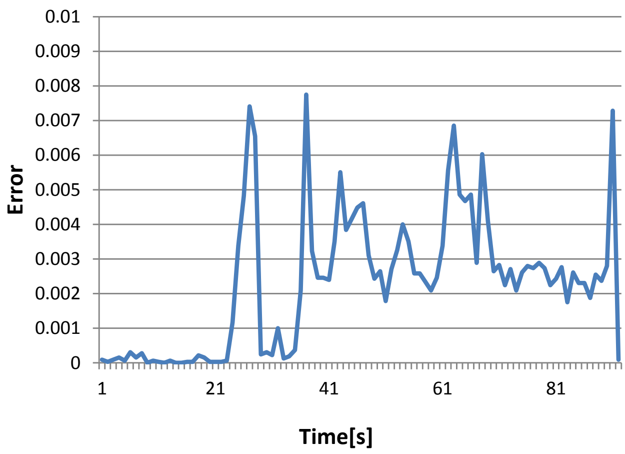
Figure 5.5: An Overview of Testbed used with Mapped Locations

During the training phase, we have used a single person as a probe to collect RSSI samples from the tag for each of three locations ($L1$, $L2$, and $L3$), and then perform MLE on them in order to map the locations and to learn observation probability. During the testing phase, instead, particle filtering is applied to infer/track the location of the RFID tags attached to persons or objects. Particle filtering has been initialized with 500 particles where initial probability distribution for each location is uniform. Regarding the prediction, a uniform transition matrix has been defined according to a map of locations, e.g. the probability of moving from one location to others is uniform for all but the case of two locations which are not directly connected with each other or separated by some barrier (e.g. wall), where the probability is set to zero.

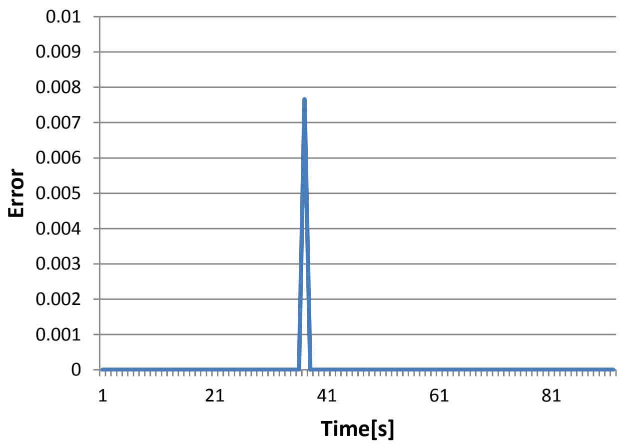
5.3.2 Experimental Results

In this subsection, we will report the descriptions and the results of the different experiments performed.

For each experiment, we evaluate the results on the basis of two parameters i.e. average location error and estimated vs ground truth error. Where average location error is calculated by means of an average Euclidean distance between the ground truth and the estimated values for each second and estimated vs ground truth error also calculate the average Euclidean distance but only for the time where estimated values differ from ground truth.



(a) Average Location Error

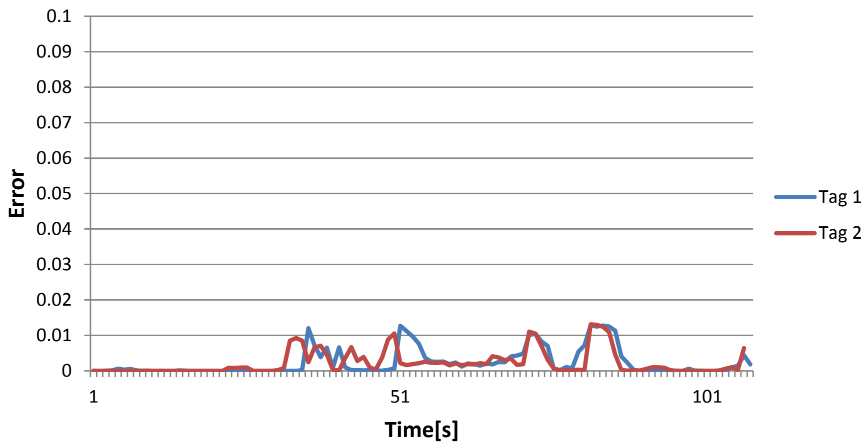


(b) Estimated vs Ground truth Error

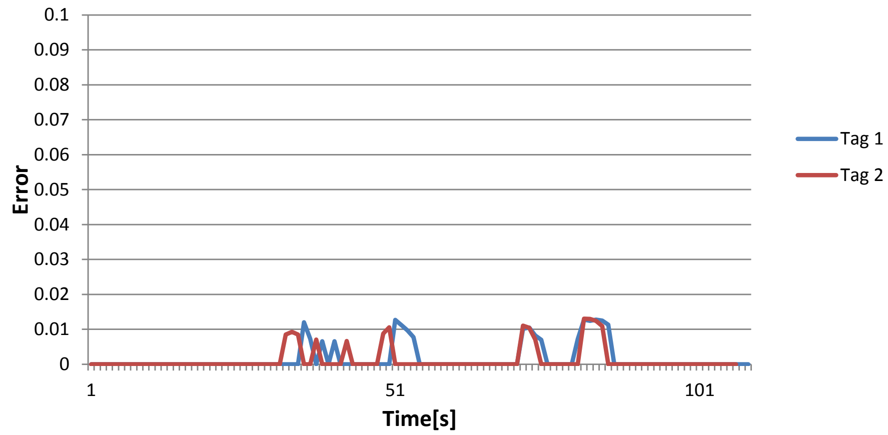
Figure 5.6: Case 1: Experiment 1: No Stay with 1 Tag

In the following, there is description of each case and the obtained results from these cases.

Case 1: No Stay with 1 Tag: in this case, a person wearing RFID tag that transmits every second rapidly move between $L1$, $L2$ and $L3$ and does not stay at any of them. Figure 5.6 (a,b) shows the average location error and estimated vs ground truth error respectively for this case. In Figure 5.6 (b) at only one second estimated value reported wrong location. Similarly, if we consider per second error i.e. average location error, it is clear from Figure 5.6 (a) that highest peak of error is nearly 0.008 which is wrong location according to ground truth and all other values are less than it and reports correct locations.



(a) Average Location Error

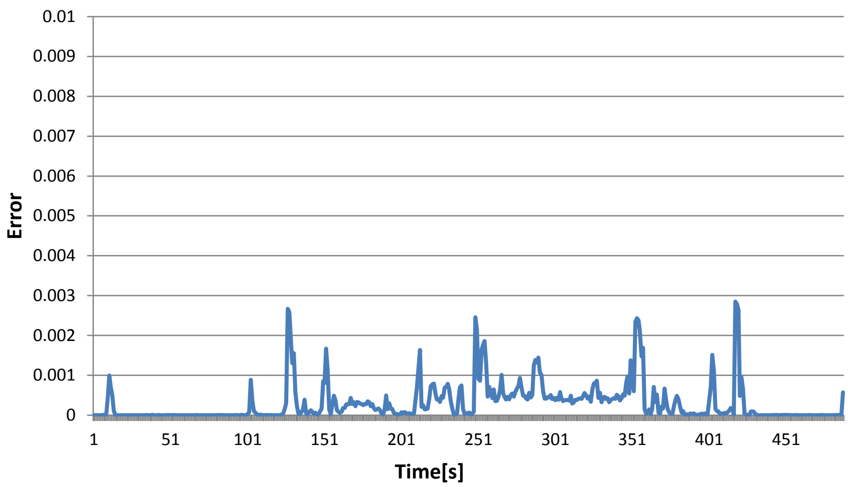


(b) Estimated vs Ground truth Error

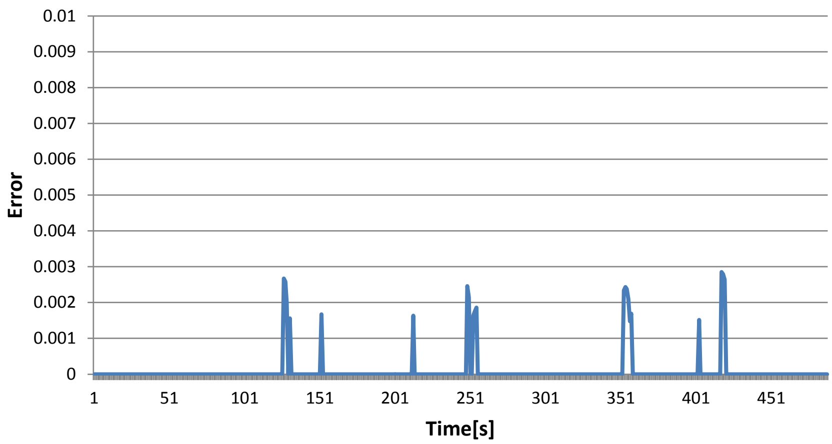
Figure 5.7: Case 2: Experiment 2: No Stay with 2 Tags

Case 2: No Stay with 2 Tags: this case considers the same movement scenario of case 1 but number of people is 2, holding RFID tags and walking side by side. Figure 5.7 (a,b) shows the obtained results for this case. Since, both persons were walking side by side and changing there locations together, and thus, this behavior of movement is very clear from both the graphs by overlapping results of two tags. In this case, the average precision for tag 1 is 88.39% and for tag 2 is 85.71%.

The movement scenario of case 1 and 2 are typically, difficult situation for the capturing the exact locations due to the fact that people move rapidly and do not stay on particular point. Therefore, system may not get stable RSSI values from RFID antennas and thus, may contain lot of noise i.e. may reports the wrong locations.



(a) Average Location Error

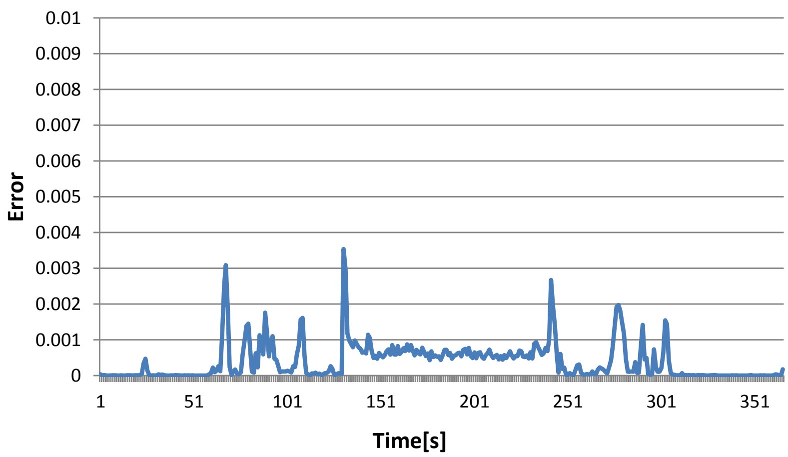


(b) Estimated vs Ground truth Error

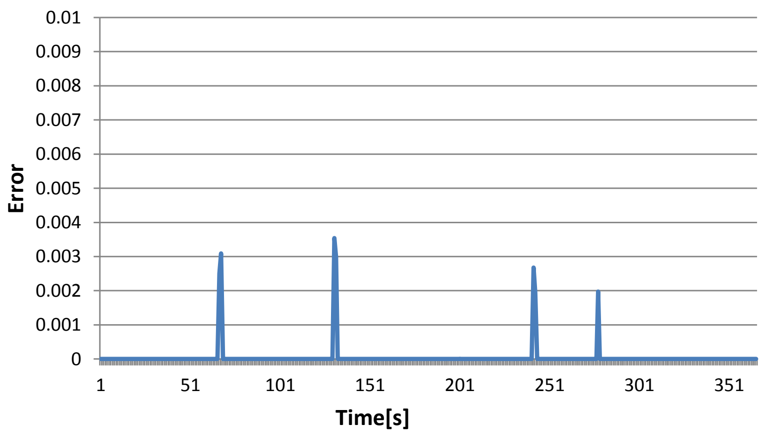
Figure 5.8: Case 3: Experiment 3: Stay with 1 Tag

Case 3: Stay with 1 Tag: this case considers one person with RFID tag moving between locations but stay for some time on each of the location. As compared to previous movement scenario this gives more stable RSSI values but even then it may contain lot of noise. Figures 5.8 (a,b), 5.9 (a,b) show the achieved results for this case. The average precision is 96.95% for Figure 5.8 (b) while it is 97.52%

for Figure 5.9 (b), which is even more better. Moreover, the accuracy of estimated results is 89.55% for Figure 5.8 (a) and 87.40% for Figure 5.9 (a).



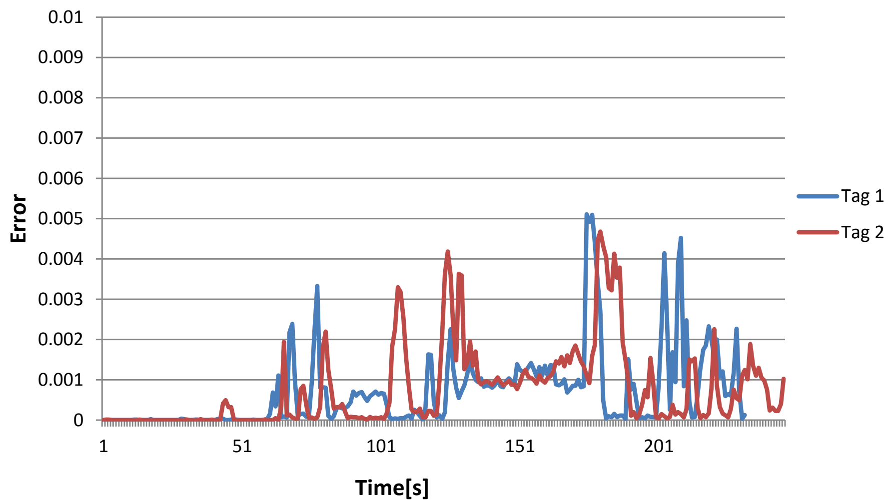
(a) Average Location Error



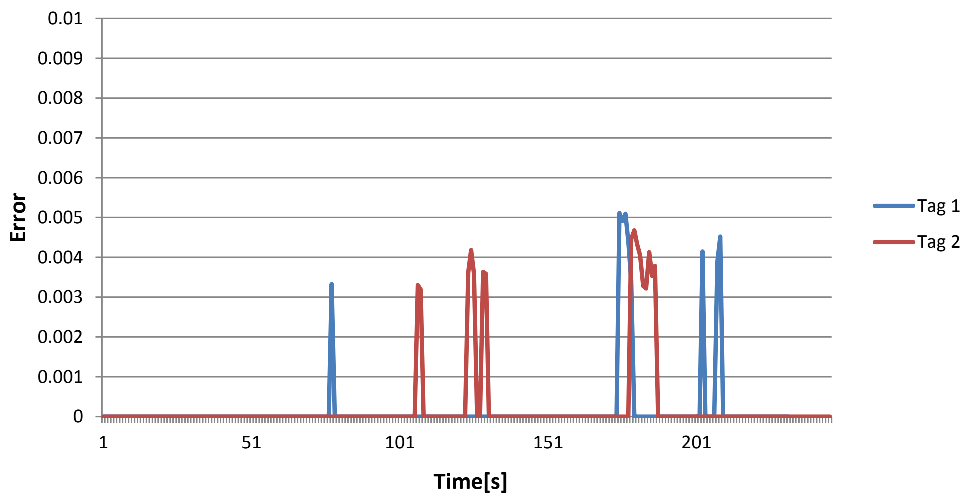
(b) Estimated vs Ground truth Error

Figure 5.9: Case 3: Experiment 4: Stay with 1 Tag

Case 4: Stay with 2 Tags: in this case, two people wearing RFID tags walk side by side and stay on each location. Figures 5.10 (a,b) and 5.12 (a,b) show the results of experiments done in this case. Since, again in this case both persons were walking together and change their locations on the same time instants thus, this typical behavior is shown by all graphs in overlapping results for both tags. The percentage of accuracy in Figure 5.10 (a) for tag 1 is 87.48% and for tag 2 is 87.21%. Whereas, the average precision in Figure 5.10 (b) for tag 1 is 94.14% and tag 2 is 96.65%.



(a) Average Location Error



(b) Estimated vs Ground truth Error

Figure 5.10: Case 4: Experiment 5: Stay with 2 Tags

For the result shown in Figure 5.12 (a,b), we consider the different path shown in Figure 5.11 than the path shown in Figure 5.5. The purpose of this experiment is to test, whether different path will effect the results or not. It is evident from both graphs that the change of path does not effect the results significantly. The average precision for Figure 5.12 (a) of tag 1 is 88.63% and tag 2 is 89.97%. It is less than the results shown in Figure 5.10 (a) due to the fact that the followed path in this case may have long gap to cover between locations. Whereas, the accuracy percentage for Figure 5.12 (b) of tag 1 is 87.11% and tag 2 is 86.61%.

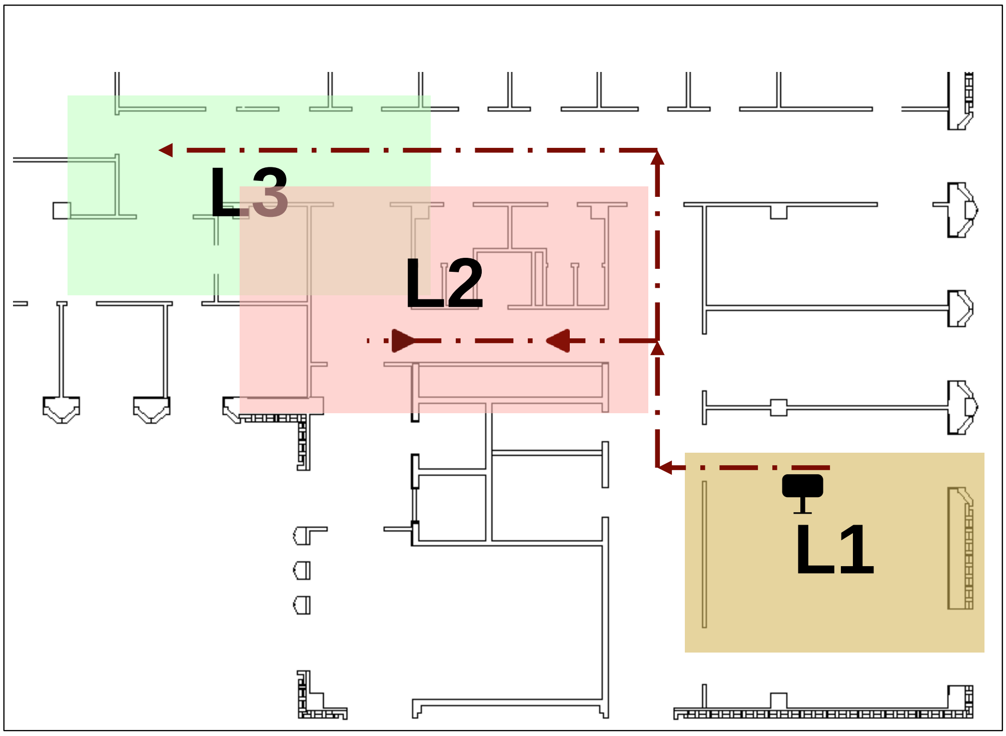


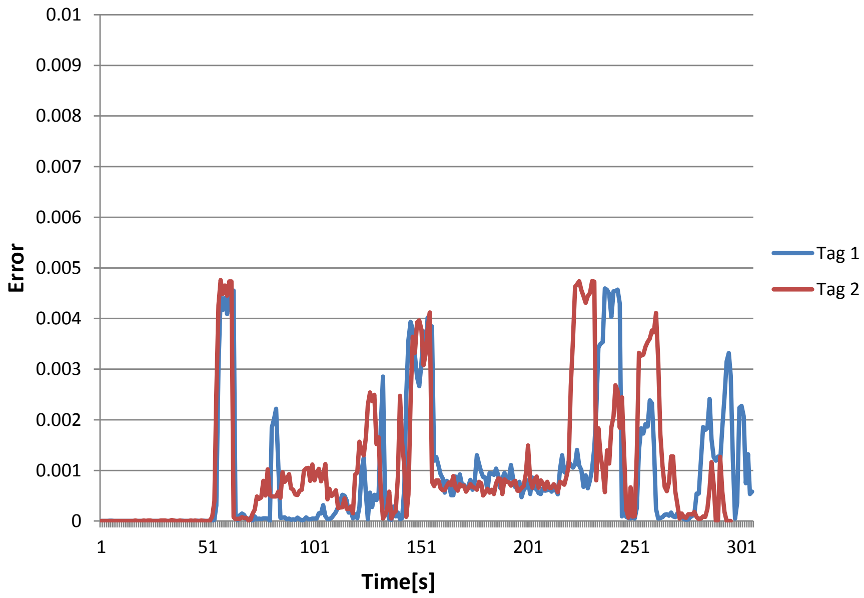
Figure 5.11: Path followed by Users in Case 4: Experiment 6

5.4 Related Works

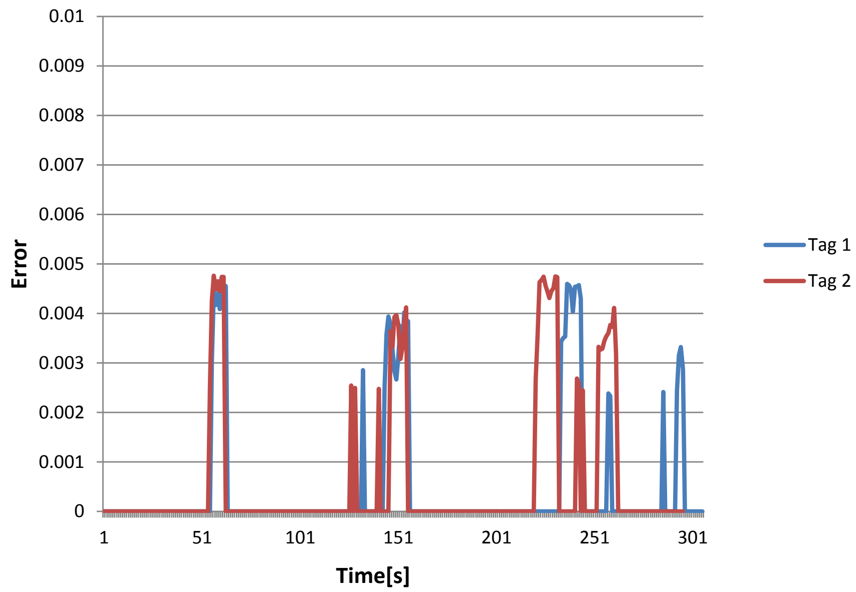
In last few decades, RFID technology has emerged significantly with many real time applications, such as product tracking and asset management, object and people authentication, health care etc. Nevertheless, the data management in these RFID applications poses a number of challenges [Chawathe et al., 2004].

In particular, the nature of an RFID data stream is noisy, redundant and unreliable, making it unsuitable for direct use in applications: among the issues that need to be effectively faced in most RFID deployments, the most common are conflicting readings (a tag is read by multiple antennas in conflicting ways) and missed readings (readers commonly detect only about 60%-70% of the tags in their range) [Ré et al., 2008].

For all these reasons, the unreliable data streams must be transformed into precise, reliable streams that can be meaningful to applications. Several techniques have been proposed for the analysis and processing of raw noisy RFID data [Rao et al., 2006]. A number of techniques propose to clean the data streams deterministically. For instance, [Franklin et al., 2005] proposes a declarative framework for RFID data cleaning and processing which makes use of a window-based adaptive smoothing filter, producing more reliable RFID data streams by interpolating missed readings.



(a) Average Location Error



(b) Estimated vs Ground truth Error

Figure 5.12: Case 4: Experiment 6: Stay with 2 Tags

Other techniques, instead, exploit the probabilistic nature of RFID data and manage their inherent uncertainty in the form of probabilities and correlations, so to achieve even higher effectiveness in the application scenarios they are applied to [Ré et al., 2008, Tran et al., 2009, Khoussainova et al., 2008a]. For instance, [Ré et al., 2008, Kanagal and Deshpande, 2008] generate probabilistic streams by inference on an HMM. Then, probabilistic inference is required in order to extract high-level complex events from the low-level atomic events acquired by the readings. For example, in tracking applications, the location of the objects is unknown to the system and observed low level sensor data is translated into precise and more reliable estimates about the location of these objects [Ré et al., 2008, Khoussainova et al., 2008a]. Note that all such RFID systems define locations on the basis of actual places/areas which are of interest to the final users (e.g. a restricted-access room), as reflected also by the supported queries and the produced results (e.g. "Find out which rooms entered Paul today"). Our presented approach in this chapter takes inspiration from these probabilistic systems in order to perform online filtering and uncertainty management.

Chapter 6

RFID Probabilistic Data Aggregation

RFID applications usually rely on RFID deployments to manage high-level events. A fundamental relation for these purposes is the location of people and objects over time. However, the nature of RFID data streams is noisy, redundant and unreliable and thus streams of low-level tag-reads can be transformed into probabilistic data streams that can reach in practical cases the size of gigabytes in a day. In this chapter, we propose a simple on-line summarization mechanism, which is able to provide small space representation for massive RFID probabilistic data streams while preserving the meaningful information. The main idea behind the proposed approach is to keep on aggregating tuples in an incremental way until a state transition is detected. Probabilistic tuples are processed as they arrive, hence avoiding the use of expensive offline disk based operations, and the output is stored in a probabilistic database in such a way that, as we also experimentally prove, a wide range of probabilistic queries can be applicable and answered effectively.

The rest of the chapter is organized as follow: In Section 6.1, we give an overview of the problem statement for RFID data stream aggregation. Section 6.2 introduces the Phase II of *RPDM System* and presents our aggregation method with its implementation details and different boundary condition tests. In Section 6.3, we report the experimental results. Finally, in Section 6.4, we briefly discuss about related works.

6.1 Overview

In the last several years, RFID technology has gained significant popularity due to its ability of detecting objects and people carrying small RFID tags in an environment equipped with RFID readers. RFID applications usually rely on RFID

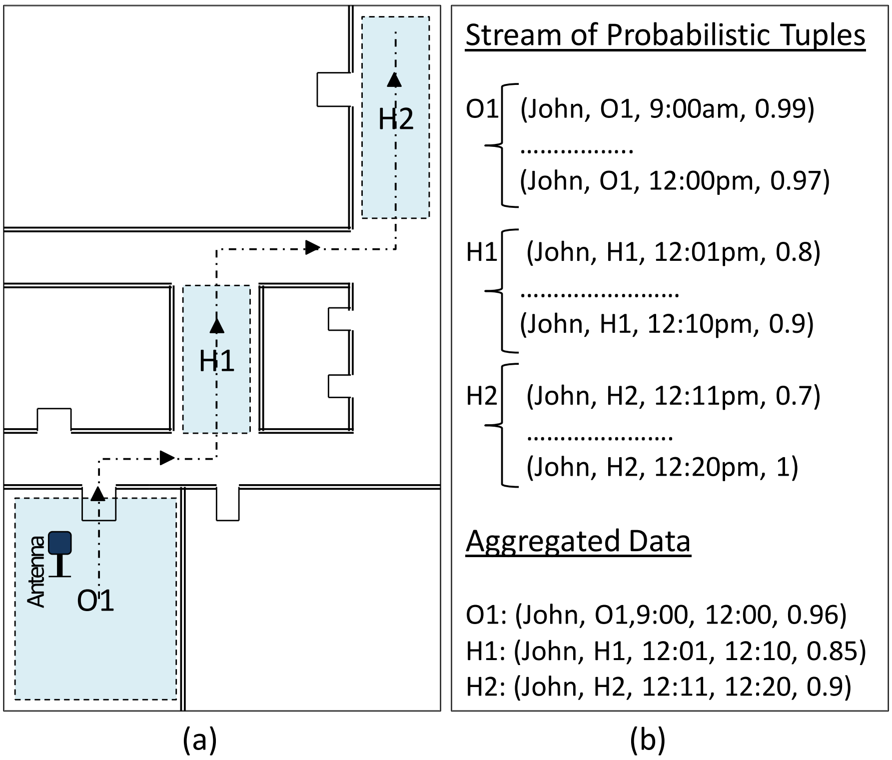


Figure 6.1: (a) A visual representation for John movements; (b) The stream of probabilistic tuples before and after applying the summarization mechanism

deployments to manage high-level events such as tracking the location that products visit for supply-chain management [Gonzalez et al., 2006b], monitoring the location and status of patients in hospital environment [Kim et al., 2008], localizing intruders for alerting services [Cucchiara et al., 2011], and so on. A fundamental relation for these purposes is the location of people and objects over time. However, the nature of RFID data stream is noisy, redundant and unreliable and thus streams of low-level tag-reads such as “Tag 101 was seen at antenna 12 at 10:00” must be transformed into meaningful relation instances such as “Tag 101 entered office 1-10 at 10:00”. To this end, a common approach for real-time applications is to use an HMM that continuously infers locations based on sensor readings [Cucchiara et al., 2011]. Such a relation, therefore, is a probabilistic relation `At(tagID,location,time,prob)` that is usually stored in a (probabilistic) database table and queried to detect complex events meaningful to applications [Ré et al., 2008]. An example tuple is `(101,1-10,10:00,0.7)`, which indicates that tag 101 at time 10:00 was in office 1-10 with probability 0.7.

RFID tags continually send out their IDs at pre-programmed intervals (few seconds) and for each tag read, the number of probabilistic tuples equals the number of reference locations. Therefore, an HMM for RFID deployments produces huge volumes of uncertain data that can reach in practical cases the size of gigabytes in a day. Storing all these probabilistic tuples in the probabilistic database is extremely expensive and, even more important, it is not always useful. For instance, Figure 6.1 (a,b) depicts one sample scenario, having a total duration of 3 hours and 20 minutes. In Figure 6.1(a), John, a user wearing an RFID tag that transmits every second, works in his office for three hours. Then, John goes to the coffee room (H2) by passing through the hall (H1), where he stays for some minutes talking with one of his colleagues. Since the number of locations in this scenario is three, $32,400$=(60 seconds * 180 minutes * 3 locations) probabilistic tuples are produced for the first three hours which report more or less the same location information for him (stay in office). This represents a rather realistic scenario, as usually person or good movements are noticeably slower than RFID transmission rates.

The main contribution of this chapter is that we propose a simple on-line summarization mechanism implemented in *RFID Probabilistic Data Aggregation Module* of *RPDM System*. The proposed method is able to provide small space representation for massive RFID probabilistic data streams while preserving the meaningful information. The mechanism draws inspiration from the field of clustering [Jain et al., 1999]. The main idea behind the proposed approach is to keep on aggregating tuples until a state transition is detected. This can be seen in Figure 6.1(b): only one tuple shows John location from 9:00am to 12:00pm i.e. in his office O1. An object or person is said to have state transition if its location changes from one to other, as in Figure 6.1(b) where John moves from O1 to H1 and consequently to H2. In this case, the proposed summarization method stores only 3 probabilistic tuples instead of $36,000$=(60 seconds * 200 minutes * 3 locations) probabilistic tuples, while these 3 stored probabilistic tuples give enough information about John's movements. The approach has been implemented in an RFID probabilistic data management system whose architecture is shown in Figure 4.1. The aggregation module processes probabilistic tuples as they arrive, hence avoiding the use of expensive and offline disk based operations such as sorting and summarization, and promptly stores the output in the probabilistic database MayBMS [Huang et al., 2009] in such a way that a wide range of probabilistic queries can be applicable and answered effectively.

6.2 RFID Probabilistic Data Aggregation, Storage and Query Processing

The detailed block diagram of phase II of *RPDM system* is shown in Figure 6.2. It receives the probabilistic RFID data streams from phase I and aggregates them by applying proposed summarization method and finally, store the probabilistic aggregated tuple in probabilistic database. The details of this module are discussed in subsequent sub-sections.

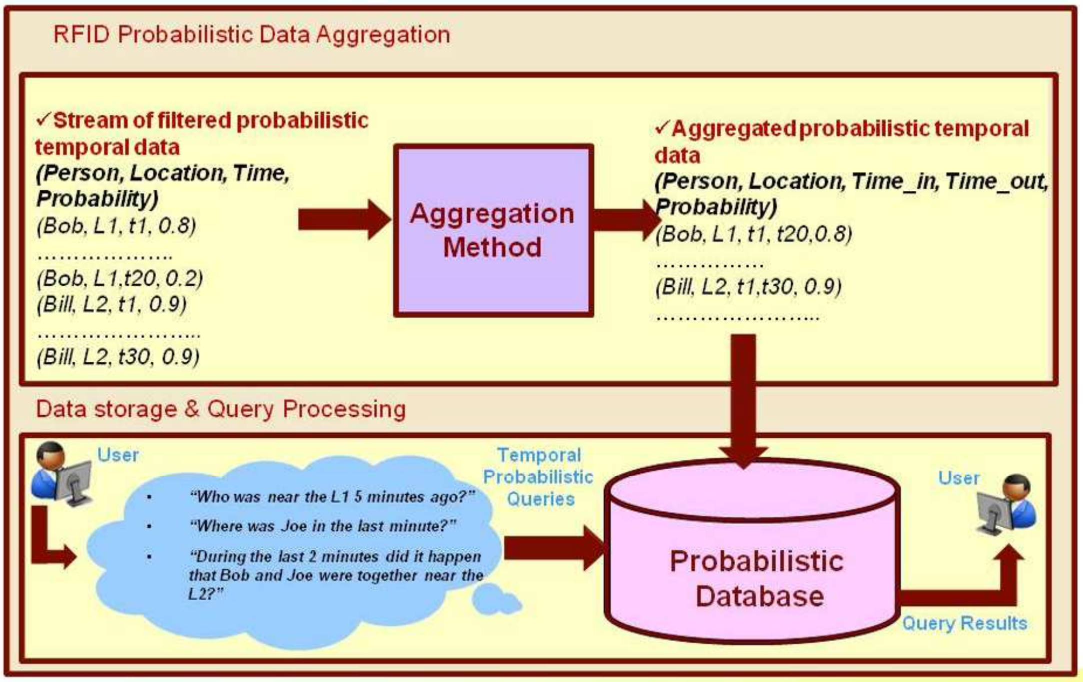


Figure 6.2: Block Diagram of Phase II of RPDM System

6.2.1 Aggregating tuples

In this section, we describe the details of our on-line aggregation algorithm (see Algorithm 1) that is implemented in the *aggregation module* shown in Figure 6.2.

Given m tags and n locations, the *RFID Online Filtering & Uncertainty Management Module* 5.2 performs inference on an HMM to produce a stream of timestamp ordered probabilistic tuples:[1]

$$X_1^{T_1}, X_1^{T_2}, \ldots, X_1^{T_m}, X_2^{T_1}, \ldots, X_2^{T_m}, \ldots$$

where each tuple X_t^T has the form:

$$(T, t, P_{T,t}(L^1), P_{T,t}(L^2), \ldots, P_{T,t}(L^n))$$

[1]For ease of presentation and without loss of generality, we assume that tuples arrive in tag order. For the same reason, the discrete probability distribution of the location random variable is represented as one tuple instead of n different tuples.

and each $P_{T,t}(L^i)$ is a score representing the probability that the considered tag T is in location L^i at time t. This is received in input by the aggregation algorithm that in turn outputs a stream of probabilistic tuples of the form:

$$X^T_{[t_s,t_e]} = (T, t_s, t_e, P_{T,[t_s,t_e]}(L^1), P_{T,[t_s,t_e]}(L^2), \ldots, P_{T,[t_s,t_e]}(L^n))$$

such that:

- for each pair of tuples on the same tag T, $X^T_{[t_{s_1},t_{e_1}]}$ and $X^T_{[t_{s_2},t_{e_2}]}$, $[t_{s_1}, t_{e_1}] \cap [t_{s_2}, t_{e_2}] = \emptyset$;
- for each source tuple X^T_t, a result tuple $X^T_{[t_s,t_e]}$ exists such that $t \in [t_s, t_e]$.

The aggregation algorithm works on the intuition that if a person wearing a tag T is stationary or resides at the same location for a period of time $[t_s, t_e]$, the corresponding probabilistic tuples $X^T_{t_s}, \ldots, X^T_{t_e}$ should show "similar" probability distributions. Therefore, in order to derive $X^T_{[t_s,t_e]}$ it draws inspiration from the large dataset clustering field [Guha et al., 1998] in that it incrementally groups together consecutive "similar" tuples. To this end, at each timestamp t the algorithm maintains at most m clusters, one for each tag T, and for each cluster c^T_t, it treats the tuple region collectively through some statistics $stat^{c^T_t}$ providing a summarized description for the cluster. When a new tuple X^T_{t+1} arrives, the algorithm tries to add it to the cluster associated to the corresponding tag c^T_t by updating the corresponding $stat^{c^T_{t+1}}$ values (see lines 3–5 of algorithm 1). Then, a boundary condition is checked (line 6) and, if it is the case, the tuple is inserted into the cluster by replacing its statistics with the newly computed ones $stat^{c^T_{t+1}}$ (line 7). On the other hand, if a violation is detected:

- c^T_t is closed and discarded from the set of current clusters S (line 10);
- a tuple $X^T_{[t_s,t]}$ describing the behavior of the tag T in the period in which the cluster c^T_t was active is stored in the database (line 11);
- a new cluster for T is created including tuple X^T_{t+1} only, its statistics is computed and it is added to S (lines 12 and 13).

Until now, we intentionally left our aggregation model generic. In the following, we show how output tuples and cluster statistics are computed.

6.2.2 Output tuples

In many clustering applications, the resulting clusters have to be represented or described in a compact form to achieve data abstraction. Basically, the most typical compact description of a cluster is given in terms of cluster prototypes or

Algorithm 1 Tuple aggregation algorithm

Require: n number of locations, p number of tags , B critical boundary
$S =$ current set of clusters; //S contains at most p elements
repeat
  receive the next stream point X_{t+1}^T
  c_t^T = identifyCluster(X_{t+1}^T, S) //$stat^{c_t^T}$ is extracted from c_t^T
  $stat^{c_{t+1}^T}$= updateStatistics($stat^{c_t^T}$,X_{t+1}^T)
  if testBoundaryCondition($stat^{c_{t+1}^T}$) **then**
    c_{t+1}^T = add(X_{t+1}^T,c_t^T); //$stat^{c_t^T}$ is replaced with $stat^{c_{t+1}^T}$
    update S with c_{t+1}^T;
  else
    close and discard c_t^T from S;
    insert $stat^{c_t^T}$ in the database;
    c_{t+1}^T = createNewCluster(X_{t+1}^T);
    add $X_{[t_s,t]}^T$ to S;
  end if
until data stream ends

representative patterns such as the *centroid* [Jain et al., 1999]. The centroid is the logical center of the cluster, usually computed as the average of all cluster points. The use of the centroid to represent a cluster is a very popular schema and works well when the clusters are compact, as in our case.

Therefore, we represent tuples in the n-dimensional Cartesian space as points whose coordinates are the probability values for the n locations. This tuple representation actually exhibits tight clustering as long as the state does not change and a good separation in case of state transition. Figure 6.3 shows the cartesian plane representation of the sample scenario discussed in Section 6.1. Since the number of locations is three in this scenario, each tuple generated by the *RFID Online Filtering & Uncertainty Management Module* is a point in a 3-dimensional space whose coordinates are the probability values for the locations O1, H1 and H2 (the graph shows only the first two dimensions since the third is linearly dependent from the others). We can see that, since John is residing at a same place (his office) for a long period, a large number of points are concentrated in the O1 region; all these points can be aggregated in one point which will be representative of the behavior of all of them. Instead, as John moves from O1 to H1 and consequently to H2, there is a transition that can be seen in the form of some scattered points on the graph plane. Hereinafter, whenever the context is clear, we will use X_t^T to denote either a probabilistic tuple $(T, t, P_{T,t}(L^1), P_{T,t}(L^2), \ldots, P_{T,t}(L^n))$ or its representation in the Cartesian space $(P_{T,t}(L^1), P_{T,t}(L^2), \ldots, P_{T,t}(L^n))$.

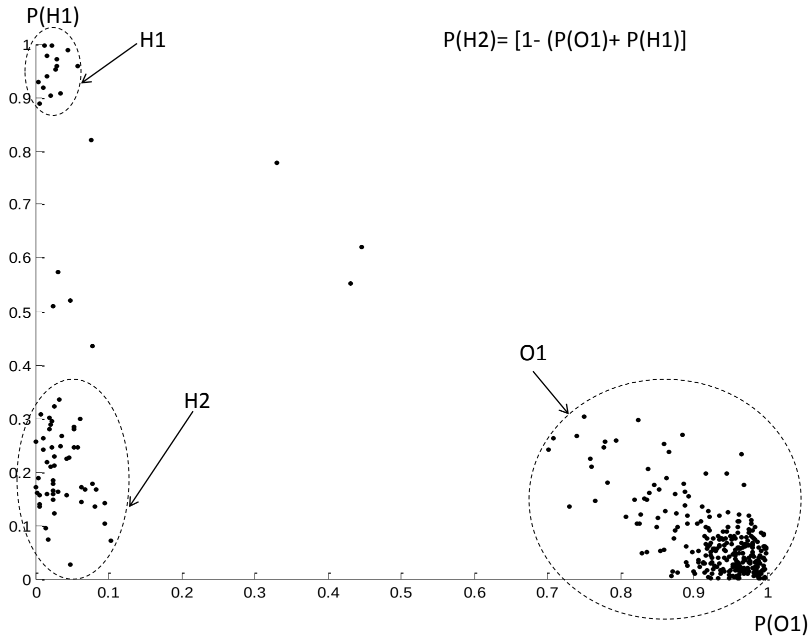


Figure 6.3: Cartesian Space representation of the probabilistic tuples of our sample scenario

Then, we incrementally compute the centroid $V_{c_t^T}$ of each cluster c_t^T while it evolves and, when it is closed, we store $X_{[t_s,t]}^T$ as $(T, t_s, t, V_{c_t^T})$.

6.2.3 Boundary conditions

The main objective of the boundary condition test, is to be able to discriminate when a cluster has to be closed in order to avoid distortion. To this end, we draw inspiration from techniques at the state of the art for cluster validity measurement [Legány et al., 2006]. Two measurement criteria are typically used for evaluating a clustering schema [Legány et al., 2006]: compactness and separation. While the former expresses the requirement that the members of each cluster should be as close to each other as possible, the latter refers to the fact that the clusters themselves should be widely separated and it is not particularly interesting for our scenario; we thus focus on compactness and consider three different methods for quantifying it. The three models, which provide different indices that can be used in the boundary condition test, are:

- *Maximum Probability Change (MPC)*: it monitors the probability distribution trends. To this end, let $\overline{L}_{X_t^T}(\overline{L}_{c_t^T})$ be the location with the maximum probability value in $X_t^T(c_t^T)$. For each cluster c_t^T, MPC maintains $\overline{L}_{c_t^T}$ as statistics, and the boundary condition is satisfied when $\overline{L}_{c_t^T} = \overline{L}_{X_{t+1}^T}$. The

Table 6.1: Performance Evaluation of (a) MPC, (b) DM and (c) CLRC

(a) MPC								
EXP	Scenario	#Tags		#Locs	#Clusters	%SP	%TAL	AvgLocError
1	No Stay	1		3	3 (=)	0.033	98.91	0.0136
2	Stay	1		5	13 (+160%)	0.026	95.93	0.0452
3	No Stay	2	Tag 1	5	9 (+80%)	0.080	86.61	0.1549
			Tag 2	5	11 (+120%)	0.097	83.04	0.1957
4	Stay	2	Tag 1	4	8 (+100%)	0.033	92.89	0.0707
			Tag 2	4	10 (+150%)	0.041	95.82	0.0453
5	Stay	2	Tag 1	4	16 (+300%)	0.053	82.16	0.1929
			Tag 2	4	13 (+225%)	0.043	86.96	0.1495
Mean					+141%	0.050	90.29	0.108

(b) DM								
EXP	Scenario	#Tags		#Locs	#Clusters	%SP	%TAL	AvgLocError
1	No Stay	1		3	3 (=)	0.033	98.91	0.0136
2	Stay	1		5	5 (=)	0.010	96.95	0.0383
3	No Stay	2	Tag 1	5	5 (=)	0.044	88.39	0.1419
			Tag 2	5	5 (=)	0.044	85.71	0.1739
4	Stay	2	Tag 1	4	5 (+25%)	0.020	94.14	0.0608
			Tag 2	4	8 (+100%)	0.033	95.82	0.0445
5	Stay	2	Tag 1	4	6 (+50%)	0.020	84.95	0.1696
			Tag 2	4	8 (+100%)	0.026	87.96	0.1374
Mean					+34%	0.040	91.60	0.0975

(c) CLRC								
EXP	Scenario	#Tags		#Locs	#Clusters	%SP	%TAL	AvgLocError
1	No Stay	1		3	3 (=)	0.033	98.91	0.0136
2	Stay	1		5	5 (=)	0.010	96.95	0.0383
3	No Stay	2	Tag 1	5	5 (=)	0.044	88.39	0.1410
			Tag 2	5	5 (=)	0.044	85.71	0.1739
4	Stay	2	Tag 1	4	4 (=)	0.016	94.14	0.0596
			Tag 2	4	5 (+25%)	0.020	96.65	0.0393
5	Stay	2	Tag 1	4	4 (=)	0.013	88.63	0.1190
			Tag 2	4	5 (+25%)	0.016	89.97	0.1185
Mean					+6%	0.024	92.41	0.0879

main disadvantage of this method is that it is very sensitive to noise and thus makes more clusters with fewer points in it;

- *Diameter-oriented (DM)*: it measures how large the cluster shape is. To this end it uses the cluster diameter as statistics and checks whether the latter is within a threshold B: $\max_{X,Y \in c^T_{t+1}} \{d(X,Y)\} \leq B$. The main disadvantage of this approach is the time and space complexity, due to the fact that the distance between all pairs of points have to be computed and constantly kept updated on the arrival of new data elements. This function is also very sensitive to noise, since the maximum cluster diameter can quickly become large in a noisy environment;

- *Centroid Vs Latest Reading Comparison (CLRC)*: it gives a measure of the mutual distance between the centroid $V_{c^T_t}$ and the latest point X^T_{t+1}. To this end, it checks whether $d(V_{c_t^T}, X_{t+1}^T) \leq B$. The main advantage of this method w.r.t. the DM model is that computations are less time and space consuming, as $V_{c_t^T}$ can be computed incrementally.

Regarding distance $d(\cdot, \cdot)$ between tuples, our approach is independent from the actually adopted function. Several alternatives are possible for its implementation since we only require it is applicable in a n-dimensional space. In our experiments we adopted the Euclidean distance. Finally, note that for both DM and CLRC, we can control the quality of the clustering process by properly selecting the threshold B: low values of B produce a high number of small and tight clusters, while we have an opposite behavior for high values of B.

6.3 Experimental Evaluation

In order to evaluate the performance of the presented approach, we have conducted several experiments in different scenarios, collecting data from persons wearing RFID tags. The experimental scenarios are all set in three indoor locations (denoted L1, L2 and L3) and capture different possible movement behaviors: (i) "No Stay", where people rapidly move between locations without staying on any specific one; and (ii) "Stay", where people move between locations and spend some time on each of them. Both types of scenarios have been tested with one/multiple tags. In all the experiments, we apply the aggregation methods, we propose to the stream of tuples generated by the *RFID Online Filtering & Uncertainty Management Module*.

The goal of our evaluation studies was two-fold: (i) to validate and compare the effectiveness of each method in precisely summarizing the movement behaviors which actually took place in the scenarios (Section 6.3.1); and (ii), to evaluate the best performing method on a possible target application, i.e. to compare the results which can be obtained by querying the RFID data via a temporal probabilistic database with and without applying the aggregation method to the involved data (Section 6.3.2).

6.3.1 Effectiveness of Aggregation Methods

In this subsection, we analyze the performance of the presented aggregation methods by means of five experiments conducted on different movement scenarios types (stay/no stay) and with a varying number of actually visited locations and tags. The experimental setup and the obtained results are summarized in the left and right parts of Table 6.1, respectively. For each experiment, we measure the effectiveness of the methods based on four parameters: (a) number of output clusters (#Cluster); (b) percentage of occupied space w.r.t. non-aggregated data (%SP); (c) percentage of time at actual location (%TAL); and (d) average location error (AvgLocError) between clustered and actual locations. The basic intuition for (a) is that the nearer it is to the number of actually visited locations, the more effective is the method; (b) provides a clear quantification of the space required by the aggregated tuples (the smaller the percentage the higher the saved space); beyond these "overview" approaches, (c) and (d) provide us with more detailed information on the actual contents of the generated clusters. More specifically, the %TAL is the percentage of time for which aggregated data reports the same location as of ground truth; besides correctness, this gives us an idea about the promptness of each method to adjust the output to the ground truth over the experiment duration (the higher the value the better). Moreover, average location error takes into account how much the summarized description of each generated cluster is near to the actual ground truth values. We devised the measure so to highlight what we really think is crucial in this evaluation, i.e. how long and how much each method differs from the ground truth: It is calculated by means of an average Euclidean distance between the ground truth and the aggregated summarized descriptions over the total time span, only considering those time instants when a "wrong" location is reported values of AvgLocError are between 0 and 1, therefore the lower the value the better the estimate.

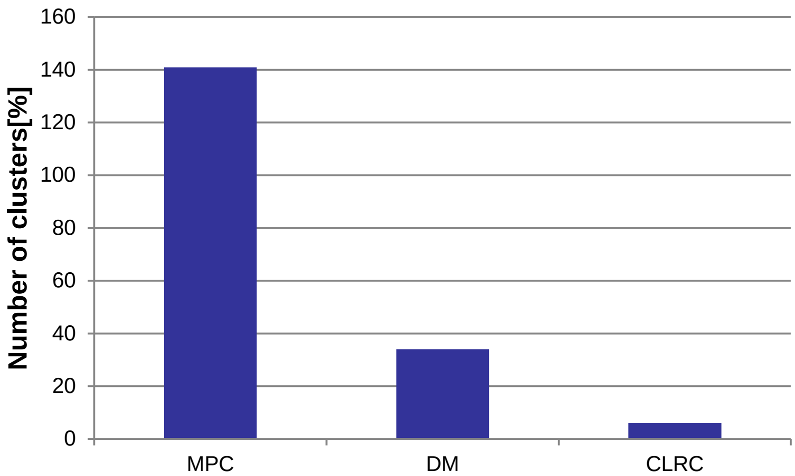


Figure 6.4: Number of Clusters

Figures 6.4,6.5,6.6 and 6.7 show the graphical comparison between the three indices on the basis of the experimental results obtained from (right part of Table 6.1 (a, b, c)).

From the obtained experimental results, we found that MPC is very sensitive to noise and thus performs poorly in the presence of noisy data. On average it makes 141% more clusters than expected (up to 300% more in EXP5), while average location error is quite high, for instance with values of 0.19 for EXP3 and EXP5

(0.108 on mean for all the experiments). TAL is about 90% on mean, with the lowest values being 83% (EXP3) and 82% (EXP5).

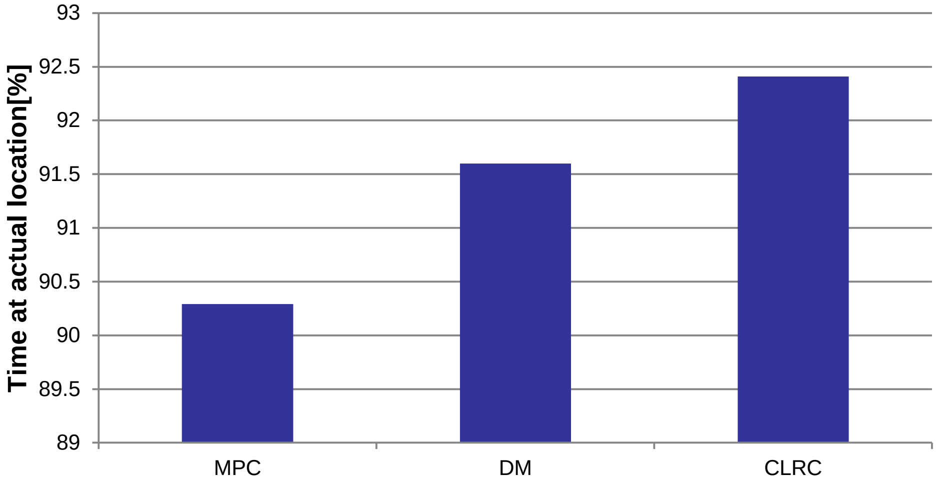


Figure 6.5: Time at Actual Location (%TAL)

DM performs better than MPC but its diameter can quickly become very large in presence of noisy data. DM has an average location error of 0.0975 and average TAL of approximately 92%, while it makes 34% more clusters than expected.

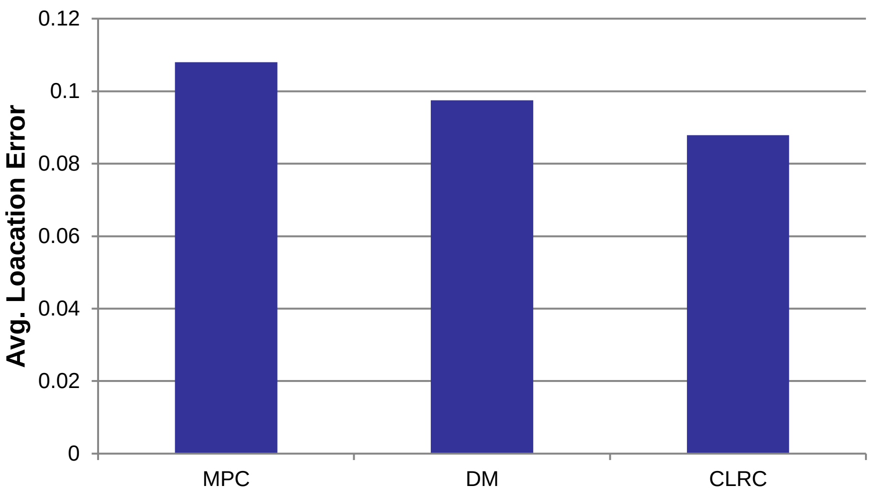


Figure 6.6: Average Location Error

CLRC shows superior performance to MPC and DM, giving good results even in noisy environments. The average TAL is about 92%, whereas the average location error is approximately 0.0879; on average, it only makes 6% more clusters than expected, which, together with the other figures, represents a very encouraging result. The same holds for the very consistent space savings produced by all methods (ranging from 0.05% of the space required by non-aggregated data to the most compact 0.024%, given by MPC and CLRC, respectively).

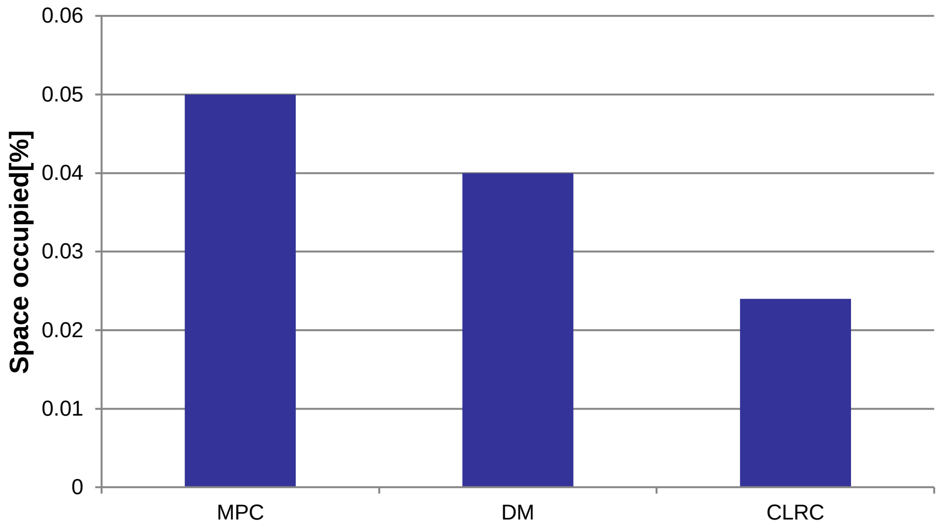


Figure 6.7: Percentage of Space Occupied

6.3.2 Temporal Probabilistic Query Processing

After having evaluated the goodness of the output data *per se*, we now want to assess the performance of a probabilistic DBMS in answering some typical queries over the summarized versus non-summarized data of our five experiments. For the tests in this section we will exploit the CLRC method, since it has been proven the best performing one (see Section 6.3.1).

As mentioned earlier, the output of the *RFID Online Filtering & Uncertainty Management Module* is a probabilistic stream of tuples where we keep the probabilities of each object being on a specific location at a specific instant. In order to handle the uncertainty associated to these probabilistic streams, we use the MayBMS database management system [Huang et al., 2009] and validate the results obtained on the aggregated and complete data over a number of queries. The queries contain constraints (interval or snapshot) over the temporal history of the RFID data and are used to identify and track RFID objects in the test environment.

In the following, we discuss six of the most significant queries, named Q1 to Q6, that we used in the tests. For each query, we will show its plain text form, its MayBMS (SQL) form, and discuss the obtained results as summarized in Table 6.2. In particular, the table shows, for each of the five experiments (columns) and of the six queries (rows), from left to right, the actual (expected) and computed output results over aggregated and non-aggregated data.

Table 6.2: Probabilistic Query Results for Aggregated and Non-Aggregated data

	EXP1					EXP2				
	Actual	*Aggregated Data*		*Non-Aggregated Data*		*Actual*	*Aggregated Data*		*Non-Aggregated Data*	
			conf		*conf*			*conf*		*conf*
Q1	P1	P1	0.983	P1	0.996	_	P1	0.027	P1	0.004
Q2	L2	L1 L2 L3	0.105 0.831 0.062	L1 L2	0.482 0.518	L3	L2 L3	0.213 0.786	L2 L3	0.606 0.394
Q3	2:09:34	2:09:34	0.983	2:09:34	0.78	5:20:55	5:20:55	0.984	5:20:55	1
Q4	L1,P1	L1,P1	0.983	L1,P1	0.994	L1,P1	L1,P1	0.975	L1,P1	1
Q5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Q6	2:09:35	2:09:35 2:09:46	0.817 0.022	*	*	5:14:44	5:14:48 5:16:47 5:19:50	0.879 0.005 0.0001	*	*

	EXP3					EXP4				
	Actual	*Aggregated Data*		*Non-Aggregated Data*		*Actual*	*Aggregated Data*		*Non-Aggregated Data*	
			conf		*conf*			*conf*		*conf*
Q1	P1 P2	P1 P2	0.981 0.988	P1 P2	1 1	_	P1 P2	0.033 0.004	P1 P2	0.016 0.04
Q2	L2	L1 L2 L3	0.026 0.944 0.03	L2 L3	0.976 0.024	L2	L1 L2 L3	0.053 0.898 0.048	L1 L2	0.216 0.784
Q3	4:41:24	4:41:24	0.94	4:41:24	0.516	6:05:10	6:05:10	0.989	6:05:10	0.801
Q4	L1,P1 L1,P2	L1,P1 L1,P2	0.988 0.981	L1,P1 L1,P2	1 1	L1,P1 L1,P2	L1,P1 L1,P2	0.996 0.989	L1,P1 L1,P2	1 1
Q5	L1 L2 L3	L1 L2 L3	0.998 0.974 0.686	L1 L2 L3	1 1 0.999	L1 L2 L3	L1 L2 L3	0.987 0.991 0.654	L1 L2 L3	1 1 1
Q6	4:40:03	4:40:04 4:40:20 4:41:03	0.701 0.028 0.001	*	*	6:05:11	6:05:12 6:06:08	0.882 0.001	*	*

	EXP5				
	Actual	*Aggregated Data*		*Non-Aggregated Data*	
			conf		*conf*
Q1	_	P1 P2	0.037 0.021	P1	0.002
Q2	L3	L2 L3	0.166 0.833	L2 L3	0.146 0.854
Q3	6:26:45	6:26:39	0.988	6:26:56	0.518
Q4	L1,P1 L1,P2	L1,P1 L1,P2	0.998 0.988	L1,P1 L1,P2	1 1
Q5	L1 L2 L3	L1 L2 L3	0.988 0.969 0.73	L1 L2 L3	1 1 1
Q6	6:26:46	6:26:40 6:28:02	0.842 0.022	*	*

Q1. *Find who was at location 'L1' 10 seconds ago?*

```
SELECT TagId, conf()
FROM cluster_data
WHERE LocationId='L1'
AND time_in<'T'- interval '00:00:10'
AND time_out>'T'- interval '00:00:10'
GROUP BY TagId;
```

---

where `conf()` is the MayBMS function for calculating the confidence of the answer. Note that, in some of the experiments (EXP1, EXP4 and EXP5) the actual answer to this query should be "no one" ("-" in Table 6.2). In all cases, we can see that the results on summarized data are correct and with a confidence which is very near (almost identical) to the non-aggregated data results; this shows that, even if data in aggregated form contain less detailed information, they provide accurate answers to the queries.

Q2. *Find where was person 'P1' at time 'T'?*

```
SELECT LocationId, conf()
FROM cluster_data
WHERE TagId='P1'
AND time_in<'T'
AND time_out> 'T'
GROUP BY LocationId;
```

Again, all the answers on the aggregated data are correct. Moreover, from this and some of the following queries we can see that in some cases the confidence of the correct answer is higher on the summarized data, due to the noise that is present in the non-summarized data.

Q3. *Find when 'P1' was seen last time at location 'L1'?*

```
SELECT time_out, conf()
FROM cluster_data
WHERE LocationId= 'L1' AND TagId='P1'
AND time_out=(SELECT max(time_out)
FROM prob_stream
WHERE LocationId= 'L1' AND TagId='P1'
AND probability>0.5)
GROUP BY time_out;
```

---

where `start_time()` is a user-defined function for retrieving the startup time of the used data set.

Q4. *Find where and which persons are detected at the first moment by the system?*

```
SELECT LocationId, TagId, conf()
FROM cluster_data
WHERE time_in=(SELECT start_time())
```

```
GROUP BY LocationId, TagId;
```

Q5. *Whether it happened that two persons are together at the same location at the same time? Where?*

```
SELECT c1.Locationid, conf()
FROM cluster_data c1, cluster_data c2
WHERE c1.TagId= 'P1'
AND c2.TagId= 'P2'
AND c1.LocationId = c2.LocationId
AND c1.time_in >= (SELECT start_time())
AND c1.time_out <= (SELECT end_time())
GROUP BY c1.Locationid;
```

Note that this query is not applicable to EXP1 and EXP2, since only one tag is used.

Q6. *Find when 'P1' moved from location 'L1' to 'L2'?*

```
SELECT c2.time_in,conf()
FROM cluster_data c1, cluster_data c2
WHERE c1.TagId='P1' and c1.TagId=c2.TagId
AND c1.LocationId='L1' and c2.LocationId='L2'
AND (c2.time_in-c1.time_out)<='00:00:02'
AND (c2.time_in-c1.time_out)>='00:00:00'
GROUP BY c2.time_in;
```

This is an interesting case involving transition detection between two locations. As expected, the results we got from the DBMS experimentally prove that transitions are much easier to identify on the aggregated data, since the complete data contain a lot of "noise" producing a very large quantity of irrelevant and/or incorrect results ("*" in Table 6.2).

6.4 Related Works

The efficient management of RFID data involves a large number of issues in a wide range of applications. One of the main concerns for data management is that the rate of RFID data streams is quite fast and, therefore, the resulting volume of the stream is quite huge.[2] For these reasons, clustering becomes one of the more challenging tasks to perform. In the database community various algorithms have been proposed for a number of clustering problems and several methods working on very large amounts of data gained popularity, such as DBSCAN [Ester et al., 1996], CURE [Guha et al., 1998] and BIRCH [Zhang et al., 1996].

Besides purely deterministic approaches, the vague and uncertain nature of the data stream has recently captured a lot of research attention and many clustering algorithms have been proposed which also take into account the probabilities associated to the involved data. In this context, a fuzzy version of DBSCAN has been presented as FDBSCAN [Kriegel and Pfeifle, 2005]. This algorithm, instead of finding regions with high density, identifies regions with high expected density, based on the probability distributions of the objects.

Another probabilistic extension is P-DBSCAN [Xu and Li, 2008], which takes advantage of the probability distribution information of the object locations in the definition and computation of probabilistic core object and probabilistic density-reachability.

In [Ngai et al., 2006], an extension of the K-means algorithm is proposed, named as UK-means algorithm, which considers expected distance between the object and the representative of the cluster.

As UK-means is based on classical K-means algorithm, it can be sensitive to noise. UMicro [Aggarwal and Yu, 2008] uses a general model of the uncertainty and keeps track of the standard errors of each dimension within each cluster, showing that the use of even general uncertainty model during the clustering process is enough to improve the quality of results over purely deterministic approaches. Other similar related approaches are the two-phase clustering algorithm discussed by Zhang et al. in [Zhang et al., 2009], named as LuMicro, and PW-Stream [Hu and Cheng, 2010], which has been proposed for the specific problem of sliding windows.

The objective of most of the methods discussed above is to analyze the incoming data and judge on their "certainty", thus producing the highest quality possible clusters both in terms of compactness and high probability, discarding low quality ones. Further, they work on the assumption of knowing specific information characterizing the uncertainty, such as having the entire probability density function or standard error data available. The number of clusters to be produced is also usually known in advance. On the other hand, our methods are targeted for a different objective, i.e. a summarization task in a location tracking context, and are thus designed to work on a different perspective. More specifically, our ultimate goal is to correctly identify and highlight state transitions, while avoiding redun-

[2] A part of the related work on RFID data stream aggregation is already discussed in chapter 3 Section 3.1.

dant information produced in stable states. In this context, not only one active cluster per tag suffices but, even more importantly, we never have to judge on the quality (probability) of the created clusters; instead, we purely and "objectively" summarize the received data in order to make it available to subsequent modules in a more compact but equally meaningful way. In this way, as experimentally proven, a probabilistic database such as MayBMS can effectively answer a wide range of probabilistic queries on the summarized version of the data, which only take up a fraction of the original space.

Chapter 7

A Reasoning Engine for Intruders' identification & Localization in Wide Open Areas using Cameras and RFIDs

Wide open areas represent challenging scenarios for surveillance systems, especially when sensors are affected by noise, uncertainty, distractors and complex scenarios. Moreover, the identification of intruders become difficult task when moving in group of people. The coordination between the cameras can be certainly used but the tasks of localizing and identifying targets (e.g., people) in such environments require to go beyond the use of camera-only deployments. In this chapter, in a joint effort with imageLab of University of Modena and Reggio Emilia, we present an innovative system for wide open area intruder detection relying on the joint use of cameras and RFIDs, allowing us to "map" RFID tags to people detected by cameras and, thus, highlighting potential intruders. To this end, sophisticated filtering techniques preserve the uncertainty of data and overcome the heterogeneity of sensors, while an evidential fusion architecture, based on Transferable Belief Model (TBM), combines the two sources of information and manages conflict between them. The conducted experimental evaluation shows promising results, especially in treating groups of people.

The rest of the chapter is organized as follow: First of all, in Section 7.1, we give an overview of problem statement for surveillance in wide open areas with brief description of our proposal. Then, in Section 7.2, we present high level architecture for proposed system with its internal details in subsection 7.2.1, 7.2.2 and 7.2.3. Section 7.3 discusses the experimental details and results. Finally, Section 7.4 presents literature review.

7.1 Overview

Wide open areas represent challenging scenarios for surveillance systems for two main reasons. Firstly, they are ”wide” and thus require multiple sensors in order to cover the interested area. This calls for methodologies for data fusion which need to handle noise and data redundancy. Secondly, the term “open” refers to unrestricted areas where no obliged entrances or constrained paths are present.

These conditions make the surveillance job much more difficult. In fact, one key objective of a surveillance system is to use cameras for localizing targets (e.g., people) in the area and identifying them, in order to specifically detect and localize *intruders* [Valera and Velastin, 2005]. For instance, it may be useful for acquiring zoomed images of the intruder and proceeding with recognition of gathering of evidences. Actually, these two tasks in wide open areas are competing when only video analysis is adopted: identification might require zooming on the person's face and localization needs an unzoomed view to find the correct position with respect to the scene, even if many people are potentially present. Coordination between cameras can be used, in order to have PTZ cameras zooming on the person's face while keeping the other camera fixed, but this solution becomes unfeasible in the case of multiple targets to be identified.

This problem is even more challenging when groups of people are present since it is often difficult to distinguish, within a group, authorized people from intruders. Pure vision-based approaches to handle groups of people have been proposed in past [Calderara et al., 2008], but, generally speaking, the data fusion of information coming from a network of cameras should be enriched with other information acquired by different sensors. Among the many alternative sensors, the RFID technology gained much attention thanks to its ease of use, low cost and touch-less way-of-reading. RFID technology enables applications to identify people carrying small RFID tags in an environment equipped with RFID readers. Therefore, the joint use of cameras and RFIDs allows us to take the best from both of them: camera-based systems can localize all the people in the scene (regardless if they are intruders or not), while RFIDs can identify allowed people only. In this scenario, an intruder is any person which is localized by cameras but not identified by RFID readers (thus, potentially not holding any tag).

The two tasks of localization and identification are certainly made more challenging when sensors (both cameras and RFIDs) are affected by noise, uncertainty, distractors and complex scenarios: illumination changes, occlusions and reflexes can make the task for computer vision algorithms applied to cameras hard, while multiple signal sources and the presence of metallic objects can introduce much noise in RFID signals.

In particular, the contribution of the chapter is, we propose a system for intruder detection relying on:

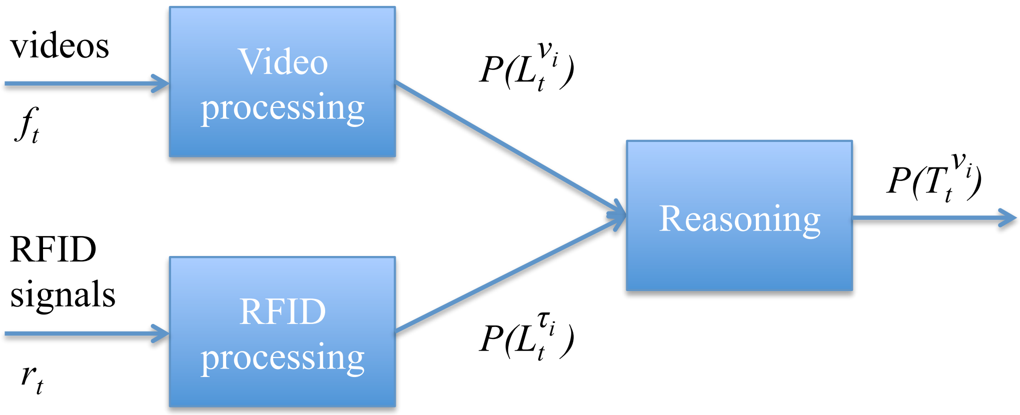


Figure 7.1: High level system description

- sophisticated filtering techniques for singular sensor modality that preserve the uncertainty of data in the form of probabilities and overcome the heterogeneity of sensors through the introduction of common locations which the data coming both from cameras and RFIDs are mapped to;
- an evidential fusion architecture, based on TBM [Smets, 1994], that processes uncertain data, combines the two sources of information and manages conflicts between them in order to "map" RFID tags to people detected from the cameras, thus highlighting potential intruders.

As to our knowledge, this is the first system that exploits two well-consolidated and low-cost technologies to localize intruders in wide open areas.

7.2 System Description

The high-level scheme of the proposed system is shown in Fig. 7.1. The two sources of information are the current frame f_t provided by the distributed cameras and analyzed by the *Video processing* module, and the RFID signals r_t provided by the tags and elaborated by the *RFID processing* module.

In the remainder of the paper we will refer to random variables using the uppercase letter and to single value with the lowercase letter. Specifically, we will refer to the following entities:

- $\mathcal{V}$ as the visual objects (typically people) detected by the video processing module, $\mathcal{V} = (\nu_{i=1,\ldots,n})$;
- $\mathcal{T}$ as the tags deployed to the authorized personnel, which can be then identified by the RFID processing module, $\mathcal{T} = (\tau_{i=1,\ldots,s})$;

- $\mathcal{L}$ as the locations in the scene, $\mathcal{L} = (\lambda_{i=1,\ldots,k})$. Locations are used to correlate data coming from cameras with those coming from RFID.

The video processing module computes the set of visual objects at time t and estimates the probability distribution $P(L_t^{\nu_i})$ of the random variable $L_t^{\nu_i}$ over the set of locations $\mathcal{L}$, one for each visual object ν_i. In other words, for each location λ_j, $P(L_t^{\nu_i} = \lambda_j)$ represents the probability that object ν_i is in λ_j.

Since a wide area can be monitored using multiple cameras only, our system makes use of a sophisticated algorithm for *consistent labeling* in partially-overlapped fields of view. The videos acquired by each camera are processed using the Sakbot (Statistical And Knowledge Based Object deTector) system [Cucchiara et al., 2003]. This motion detection algorithm is specifically designed to ensure a robust and reliable background estimation even in complex outdoor scenarios and is based on constructing a background model by means of a pixel-by-pixel temporal median model with a selective knowledge-based update stage. Once people are detected, they are tracked along time using a probabilistic appearance-based tracking algorithm [Vezzani and Cucchiara, 2008] that takes into account not only the status vector containing position and speed, but also the memory appearance model and the probabilistic mask of the person's shape. Finally, consistent labeling among different cameras is obtained using the homography-based approach presented in [Calderara et al., 2008].

Similarly, the RFID processing module will elaborate the raw RFID signal r_t to estimate the probability distribution $P(L_t^{\tau_i})$ of the random variable $L_t^{\tau_i}$ over the set of locations $\mathcal{L}$, one for each tag τ_i.

Finally, the reasoning module takes the two probability distributions, $P(L_t^{\nu_i})$ and $P(L_t^{\tau_i})$, as input and, for each visual object ν_i, it outputs the probabilities of carrying any of the tags in $\mathcal{T}$ together with the probability of carrying none of them. Formally, it outputs the random variable $T_t^{\nu_i}$ whose probability distribution is defined over the set of tags $\Omega = \mathcal{T} \cup \{\Bbbk\}$, where $\Bbbk$ is the dummy tag virtually held by an intruder. The higher is $P(L_t^{\nu_i} = \Bbbk)$ the higher is the probability that the visual object ν_i is an intruder.

In the following, we will first focus on the RFID processing module and then on the reasoning module, where we present two different reasoning engines used in our experiments. Finally, we will make some considerations on the role of locations in the system.

7.2.1 RFID Processing

The *RFID Processing* module is basically, the *RFID Online Filtering & Uncertainty Management Module* of phase I of *RPDM System*. See chapter 5 Section 5.2 (subsections 5.2.1, 5.2.2 and 5.2.3) for details.

7.2.2 Reasoning Engine

The reasoning engine has the main objective (see Fig. 7.1) to fuse the inference coming from video and RFID processing modules by means of the TBM [Smets, 1994].

TBM is a model that represents quantified belief (or weighted opinions) held by a “belief holder”, called *System* hereinafter, based on the belief function theory. Given a general *frame of discernment* $\Delta = \{H_1, \dots, H_b\}$ containing b mutually and exhaustive hypothesis related to a given problem (*closed world assumption*), belief can be represented by a *basic belief assignment* (*bba*), which is a function $m : 2^\Delta \to [0, 1]$ that satisfies $\sum_{A:A\subseteq\Delta} m(A) = 1$ and assigns a value in $[0, 1]$ to each subset $A \subseteq \Delta$ representing the part of System’s belief that is allocated to the hypothesis A. Every subset $A \subseteq \Delta$ where $m(A) > 0$ is called *focal element*. Work with focal elements only avoids the exponential complexity of the TBM. The symbol $|\cdot|$ indicates the cardinality of the set.

The advantage of the TBM over the classical Bayesian approach resides in its ability to represent every state of partial beliefs: total ignorance ($m(\Delta) = 1$), partial ignorance and total knowledge ($m(A) = 1$). It is a powerful model to deal with *uncertainty*, which may results from sensor noise, misreading or semantic noise.

In order to map tags with people, the reasoning engine goes through the following main steps. First, we present the details of reason engine I and then reasoning engine II that we used in our experiments.

Reasoning Engine I

Step 1 – Belief on locations. The probabilities over the locations of tags, $P\left(L_t^{\tau_i}\right)$, and visual objects (or people, in our case), $P\left(L_t^{\nu_i}\right)$, are translated in a Bayesian belief function [Smets, 2007] on the frame of discernment $\Delta \equiv \mathcal{L}$, which represents the set of locations of the scene. Therefore, for the tag τ_i (the same for the person ν_i) at time t, the resulting *bba* is: $m_t(\lambda_j) = P\left(L_t^{\tau_i} = \lambda_j\right), \forall \lambda_j \in \mathcal{L}$, where the subscript t has been added to indicate time and for congruency with previous notation. Each of these new data provided by sensors is then used to update System’s knowledge about localization until time $t - 1$, encoded as a set of belief functions (one for each tag and each person). The new information about τ_i (resp. ν_i) updates only the belief function for that particular tag (resp. person). The updating task is performed using an appropriate combination rule. Among others [Smets, 2007], we use Dubois-Prade’s conjunctive combination rule because it merges coherent information in a conjunctive way and conflicting ones in a disjunctive way:

$$
\begin{aligned}
m_{1\sqcap 2}(A) &= \sum_{X\cap Y=A} m_1(X)m_2(Y) + \sum_{\substack{W\cap Z=\emptyset \\ W\cup Z=A}} m_1(W)m_2(Z) \\
m_{1\sqcap 2}(\emptyset) &= 0
\end{aligned}
\tag{7.1}
$$

where $m_1 = m_{t-1}$, $m_2 = m_t$ and $A, X, Y, W, Z \subseteq \mathcal{L}$.

Step 2 – Similarity between locations. It is worth noting that the more the localization of tags and people is accurate, the higher is the mass for the same (set of) location(s) of a tag and its holder. Therefore, the comparison between the beliefs on localization of tag τ_j (m_{τ_j}) with the one of person ν_i (m_{ν_i}) returns a similarity value that indicates the support to the decision of mapping τ_j to ν_i derived from all information available to System at this moment. Defining $fe(x)$ as the set of focal elements (i.e. a subset $A \subseteq \mathcal{L}$ where $m_x(A) > 0$) of the belief function relative to a generic x, we use a measure which accounts for the similarity between focal elements through the Jaccard index [Jousselme et al., 2001]:

$$
\psi(\nu_i, \tau_j) = \sum_{A\in fe(\nu_i)} \sum_{B\in fe(\tau_j)} m_t^{\nu_i}(A) \cdot m_t^{\tau_j}(B) \cdot \frac{|A\cap B|}{|A\cup B|}
\tag{7.2}
$$

Step 3 – Evidence generation. The above mentioned similarity values are exploited to generate a new piece of information (*evidence*) encoded as a *bba* on the frame of discernment $\Delta \equiv \Omega = \mathcal{T} \cup \{\Bbbk\}$ relative to the person ν_i. Let $(\psi(\nu_i, \tau_1), \ldots, \psi(\nu_i, \tau_q))$ be the set of similarity values between the person ν_i and the q tags sensed at this moment (with $q \leq s$, where s is the total number of available tags, as defined in Section 7.2), ordered by decreasing value. We create focal elements of the evidence using the following criterion:

$$
m\left(\Gamma_j\right) = \begin{cases} \psi\left(\nu_i, \tau_{j-1}\right) - \psi\left(\nu_i, \tau_j\right) & \text{, if } j < q \\ \psi\left(\nu_i, \tau_j\right) & \text{, if } j = q \end{cases}
\tag{7.3}
$$

where $\Gamma_j = \bigcup\limits_{1\leq i\leq j,\ j\leq q} \tau_i$.

Moreover, to equal the sum of masses to 1 and considering that a person could be an intruder, we define respectively:

$$
m(\Omega) = (1 - \psi(\nu_i, \tau_1)) * \beta \tag{7.4}
$$

$$
m(\Bbbk) = (1 - \psi(\nu_i, \tau_1)) * (1 - \beta) \tag{7.5}
$$

with $\beta < 1$. A rule of thumb suggests $\beta = 0.7$.

Step 4 – Belief update and decision. For each person we combine the relative evidence with System's belief on the mapping between people and tag using again

Eq. (7.8), but this time m_1 represents the System's belief function on the mapping for person ν_i, m_2 is the new evidence relative to ν_i, and $A, X, Y, W, Z \subseteq \Omega$. Because System has to choose which tag is held by each person ν_i, it constructs a probability function on Ω in order to make the optimal decision, using the following "pignistic transformation":

$$P\left(L_t^{\nu_i} = \omega\right) = \sum_{A:\omega \in A \subseteq \Omega} \frac{m(A)}{|A|(1 - m(\emptyset))} \tag{7.6}$$

where $\omega \in \Omega$ can be either a tag or $\Bbbk$. $P\left(L_t^{\nu_i} = \omega\right)$ denotes the probability that the tag ω is held by the person ν_i.

Step 5 – Decision and belief reinforcement. The latter information can be useful to strengthen the mapping. For each tag τ_j, we consider only the *maximum probability value* over all people: $mpv(\tau_j) = \max_i P(L_t^{\nu_i} = \tau_j)$. In other words, we assume that a tag τ_j is held only by the person which is more likely to hold it at time t. We thus generate, for the person having the $mpv(\tau_j)$, the bba: $m(\tau_j) = mpv(\tau_j)$ and $m(\Omega) = 1 - mpv(\tau_j)$. If a person does not hold any tag with the highest probability, he/she is considered as an intruder, because we assume that a tag can be held by only one person. For each of these persons ν_i we thus generate the bba: $m(\Bbbk) = P(L_t^{\nu_i} = \Bbbk)$ and $m(\Omega) = 1 - P(L_t^{\nu_i} = \Bbbk)$. Finally, using Eq. (7.8) we combine these evidences with the System's belief on the mapping for that person (therefore reinforcing its belief on that particular tag) and the negated evidence (Eq. (7.7) below) on the others. Again, we need to re-define the meaning of m_1 and m_2 of Eq. (7.8): in this case, m_1 is the System's belief function on the mapping for the considered person and m_2 is the evidence just generated for him/her. We define also a belief function negate operator consistent with the closed world assumption:

$$m(A) = \sum_{B \subseteq \Omega : \overline{B} = A} m(B) \tag{7.7}$$

where $\overline{B} = \begin{cases} \Omega - B & \text{, if } (|B| = 1) \text{ and } (B \neq \{\Bbbk\}) \\ \Omega & \text{, otherwise} \end{cases}$. This operation is coherent with the fact that a tag can be held by one person only at a given time, except for the dummy tag $\Bbbk$.

Reasoning Engine II

Step 1 – FoD updating. At time t, we define the FoD $\mathcal{V}_t$ as the set of people seen and the FoD $\mathcal{T}_t$ as the set of tags sensed.

Given that a new person $\nu_i \notin \mathcal{V}_t$ appears at time $t+1$, the hypothesis ν_i is added to FoD: $\mathcal{V}_{t+1} = \mathcal{V}_t \cup \{\nu_i\}$. To reflect this change in all the belief functions

$bMap^{\nu_j}$ (see *e)* in the following) we need to apply a *deconditioning* process [**?**], which simply appends (in the set union sense) to each focal element the missing hypothesis. If at time $t+1$ a person $\nu_i \in \mathcal{V}_{t-1}$ disappear, the hypothesis ν_i is removed from FoD: $\mathcal{V}_{t+1} = \mathcal{V}_t \setminus \{\nu_i\}$, and ν_i is removed from the focal elements of all $bMap^{\nu_j}$ in the *conditioning* process. Moreover, a normalization step is obtained by transferring the mass of the empty set to the total ignorance set $(m(\mathcal{V}_{t+1}) = m(\mathcal{V}_t) + m(\emptyset))$.

The same considerations apply when a tag τ_j appears or disappears, except that the FoD is $\mathcal{T}_t$ and the belief functions to modify are $bMap^{\tau_i}$.

Step 2 – Belief on locations. The probabilities over the locations of tags $\tau_i \in \mathcal{T}_t$, $P\left(L_t^{\tau_i}\right)$, and visual objects $\nu_i \in \mathcal{V}_t$ (or people, in our case), $P\left(L_t^{\nu_i}\right)$, are translated in a Bayesian belief function [Smets, 2007] on the FoD $\Delta \equiv \mathcal{L}$, which represents the set of locations of the scene. Therefore, for the tag τ_i (the same for the person ν_i) at time t, the resulting *bba* is: $m_t(\lambda_j) = P\left(L_t^{\tau_i} = \lambda_j\right), \forall \lambda_j \in \mathcal{L}$, where the subscript t has been added to indicate time and for congruency with previous notation. Each of these new data provided by sensors is then used to update System's knowledge about localization until time $t-1$, encoded as a set of belief functions (one for each tag and each person). The new information about τ_i (resp. ν_i) updates only the belief function for that particular tag (resp. person). The updating task is performed using an appropriate combination rule. Among others [Smets, 2007], we use *Dubois-Prade's conjunctive combination rule* because it merges coherent information in a conjunctive way and conflicting ones in a disjunctive way:

$$\begin{aligned} m_{1\sqcap 2}(A) &= \sum_{X\cap Y=A} m_1(X)m_2(Y) + \sum_{\substack{W\cap Z=\emptyset \\ W\cup Z=A}} m_1(W)m_2(Z) \\ m_{1\sqcap 2}(\emptyset) &= 0 \end{aligned} \tag{7.8}$$

where $m_1 = m_{t-1}$, $m_2 = m_t$ and $A, W, X, Y, Z \subseteq \mathcal{L}$.

Step 3 – Similarity between locations. It is worth noting that the more the localization of tags and people is accurate, the higher is the mass for the same (set of) location(s) of a tag and its holder. Therefore, the comparison between the beliefs on localization of tag τ_j (m^{τ_j}) with the one of person ν_i (m^{ν_i}) returns a similarity value that indicates the support to the mapping $\langle \nu_i, \tau_j \rangle$ between ν_i and τ_j derived from all information available at this moment. Defining $fe(x)$ as the set of focal elements of the belief function relative to a generic x, we use a measure which accounts for the similarity between focal elements through the Jaccard index [Jousselme et al., 2001]:

$$\psi(\nu_i, \tau_j) = \sum_{A\in fe(\nu_i)} \sum_{B\in fe(\tau_j)} m_t^{\nu_i}(A) \cdot m_t^{\tau_j}(B) \cdot \frac{|A\cap B|}{|A\cup B|} \tag{7.9}$$

Step 4 – Evidence generation. The above mentioned similarity values are exploited to generate new pieces of information (*evidences*), encoded as a *bba*, representing all the knowledge that System is able to extract from data available at time t.

First, for each person $\nu_i \in \mathcal{V}_t$, the *bba* on the FoD $\mathcal{T}_t$ encodes the belief on which tag(s) can be held by ν_i. Let $\Psi(\nu_i) = \{\psi(\nu_i, \tau_1), \ldots, \psi(\nu_i, \tau_r)\}$ be the set of similarity values between ν_i and the $r = |\mathcal{T}_t|$ tags sensed at this moment ordered by decreasing value. We create the focal elements of the evidence using the following criterion, where $\Gamma_j = \bigcup_{1 \le i \le j,\ j \le r} \tau_i$:

$$m_t^{\nu_i}(\Gamma_j) = \begin{cases} \psi(\nu_i, \tau_{j-1}) - \psi(\nu_i, \tau_j) & \text{, if } j < r \\ \psi(\nu_i, \tau_j) & \text{, if } j = r \end{cases} \tag{7.10}$$

$$m_t^{\nu_i}(\mathcal{T}_t) = (1 - \psi(\nu_i, \tau_1)) \tag{7.11}$$

Second, for each tag $\tau_j \in \mathcal{T}_t$, the *bba* on the FoD $\mathcal{V}_t$ encodes the belief on which person(s) can hold the tag τ_j. Let $\Psi(\tau_j) = \{\psi(\nu_1, \tau_j), \ldots, \psi(\nu_q, \tau_j)\}$ be the set of similarity values between τ_j and the $q = |\mathcal{V}_t|$ traces seen at this moment ordered by decreasing value. We create the *bba* as before, replacing in (7.10) and (7.11) $r, \nu, \tau, \mathcal{T}_t$ with $q, \tau, \nu, \mathcal{V}_t$, respectively. Note that $\psi(\nu_i, \tau_j) \equiv \psi(\tau_j, \nu_i)$ by definition.

Step 5 – Belief updating. System entertains a belief on two kinds of mapping. The first is the mapping between a person ν_i and the tags, encoded as a belief function $bMap^{\nu_i}$ on the FoD $\mathcal{T}_t$. For each person $\nu_i \in \mathcal{V}_t$ we combine the relative evidence $m_t^{\nu_i}$ with $bMap^{\nu_i}$. Similarly, the second mapping is between a tag τ_j and the people, encoded as a belief function $bMap^{\tau_i}$ on the FoD $\mathcal{V}_t$. For each tag $\tau_j \in \mathcal{T}_t$ we combine the relative evidence $m_t^{\tau_j}$ with $bMap^{\tau_j}$.

To combine new evidences with System's belief we use eq. (7.8) with $m_1 = bMap^{\nu_i}$ (resp. $bMap^{\tau_j}$), $m_2 = m_t^{\nu_i}$ (resp. $m_t^{\tau_i}$) and $A, W, X, Y, Z \subseteq \mathcal{V}_t$ (resp. $\mathcal{T}_t$).

Step 6 – Betting. The belief function $bMap^{\nu_i}$ represents the System knowledge updated with all information available till now on which tag(s) can be held by the person ν_i.

For each $bMap^{\nu_i}$ System constructs a probability function Bet^{ν_i}, over the set of hypotheses $\mathcal{T}_t$, in order to make the optimal decision. If System must decide *now* what is the right tag to map to ν_i, the decision is taken accordingly to Bet^{ν_i}: this approach translates the saying that "beliefs guide our action". We then apply the following *pignistic transformation*, where $Bet^{\nu_i}(\tau_j)$ denotes the probability that the tag τ_j is held by the person ν_i:

$$Bet^{\nu_i}(\tau_j) = \sum_{A : \tau_j \in A \subseteq \mathcal{T}_t} \frac{m(A)}{|A|(1 - m(\emptyset))} \tag{7.12}$$

The same considerations are valid also when System constructs a probability function Bet^{τ_j} over the set of hypotheses $\mathcal{V}_t$ starting from $bMap^{\tau_j}$ to find the right person to map with τ_j.

Step 7 – Decision. System estimates, at best of its knowledge, which mappings $\langle\nu_i, \tau_j\rangle$, among all possible mappings between a single person and a single tag, are the most likely. In the set $fe(bMap^{\nu_i})$, the focal element Q^{ν_i} with highest mass is, thus, the (set of) hypothesis which has the highest support. Similarly, in the set $fe(bMap^{\tau_j})$, the focal element R^{τ_j} with highest mass is the (set of) person(s) which is more likely to hold τ_j.

The confidence C on $\langle\nu_i,\tau_j\rangle$ is determined as follows:

$$C\langle\nu_i,\tau_j\rangle = \frac{1}{2}\left(\frac{m(Q^{\nu_i})}{|Q^{\nu_i}|}Bet^{\tau_j}(\nu_i) + \frac{m(R^{\tau_j})}{|R^{\tau_j}|}Bet^{\nu_i}(\tau_j)\right) \tag{7.13}$$

which merges System's knowledge on which tag can be held by a given person with the one on which person can hold a given tag.

By thresholding the confidence on the mappings (see Sec. 7.3), System is able to determine which people in the scene are authorized, also assigning them their own tag. The more high confidence mappings are discovered, the more confident is System to detect as intruder who is not mapped to any tag.

TBM is then very suited to our application because (i) it supports the combination of different pieces of evidence coming from different sources, as the cameras and RFIDs in our case, (ii) it allows the information to be updated in time, (iii) it allows to propagate the uncertainty associated to the available information and (iv) knowledge is refined using only available information with no further assumption. The reasoning engine is thus capable to map tag and people, and to identify intruders when mapped with the dummy tag $\mathbb{k}$.

7.2.3 On the choice of locations

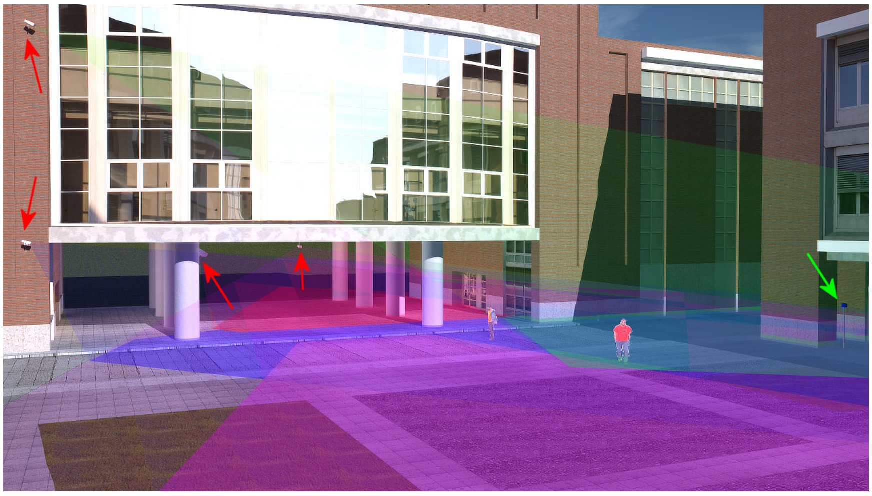
In the previous sections, we have described the inner architecture of our combined RFID/camera system in detail. In this subsection, we shortly discuss the criteria for choosing locations. The output of the system, consisting in the probability of a given tag being held by a given person, fully reflects our ultimate goal, i.e. associating the tags to the people so to be able to identify and locate possible intruders: locations are not part of the final output, even if a precise localization of each visual object is still of course possible thanks to the video analysis. Differently from most proposed RFID-only deployments, in our system the locations are not necessarily known to the final users and they do not necessarily have to coincide with actual places of interest. Instead, locations are the mean which data coming from both the camera and RFID processing modules are mapped to, thus location choice becomes an internal parameter which can be fine-tuned so to maximize the cooperation between the two modules and the final effectiveness of the system. This can be done by subdividing the open area so that: a) the resulting locations can still be correctly distinguishable by the RFID and video processing modules in most situations (e.g. sufficient size, disposition compatible with the deployed cameras/antennas configuration, and so on); b) the number of locations is sufficiently large to allow a substantial amount of location changes to be identified by the RFID and video modules, so that enough observations can be fed to the reasoning engine, which will in this way eventually provide more accurate results. The latter requirement could be easily satisfied by a preliminary offline analysis of the paths typically covered by people in the area, for instance by reviewing previously captured videos. As an alternative, an automatic method to determine the locations which maximize their identification by RFIDs can be employed (see for instance the work in [Cucchiara et al., 2010]).

7.3 Experimental Results

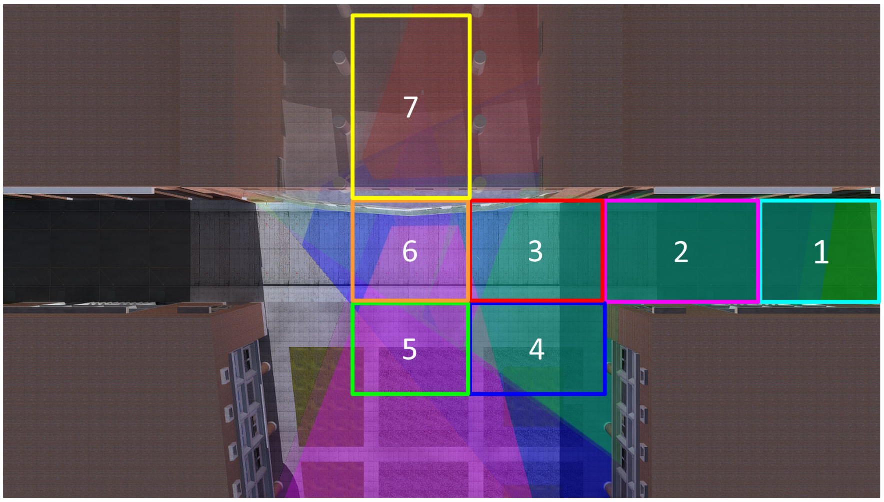
For evaluating the effectiveness of the envisaged application we have conducted experiments in different challenging situations, consisting of real wide open areas in our Campus, where several cameras are installed. In these scenarios it would be almost impossible to achieve good results without a reasoning engine capable of dealing with imprecise, uncertain and missing data.

Fig. 7.2 shows the overview of the testbed: Fig. 7.2(a) shows a 3D reconstruction of the area with the cameras (indicated by a red arrow) and the antenna used in our tests properly highlighted; Fig. 7.2(b) reports a bird-eye view of the setup with the locations represented by bounded areas and the antenna indicated by a green arrow. Upon this challenging setup, we have collected data from cameras and RFID tags in two cases. In the first case (*Case 1*) only one camera and one RFID antenna (covering the whole area to be monitored thanks to the active technology of our RFIDs) have been used, dividing the area in four locations (from 1 to 4 in Fig. 7.2(b)): different scenes of increasing complexity (with more people and tags, and with an intruder) have been analyzed. In *Case 2*, four cameras and seven locations have been used, allowing a larger scene coverage (thanks to the consistent labeling techniques described in Section 7.2) and considering a more realistic scenario.

During the training phase, we have used a single person as a probe to collect RSSI samples from the tag in the different chosen locations, and then perform MLE on them in order to map the locations $\lambda_i \in \mathcal{L}$. During the testing phase, instead, particle filtering is applied to infer/track the location of the RFID tags. Particle filtering has been initialized with 500 particles where initial probability



(a) 3D view of the scene



(b) bird-eye view with locations superimposed

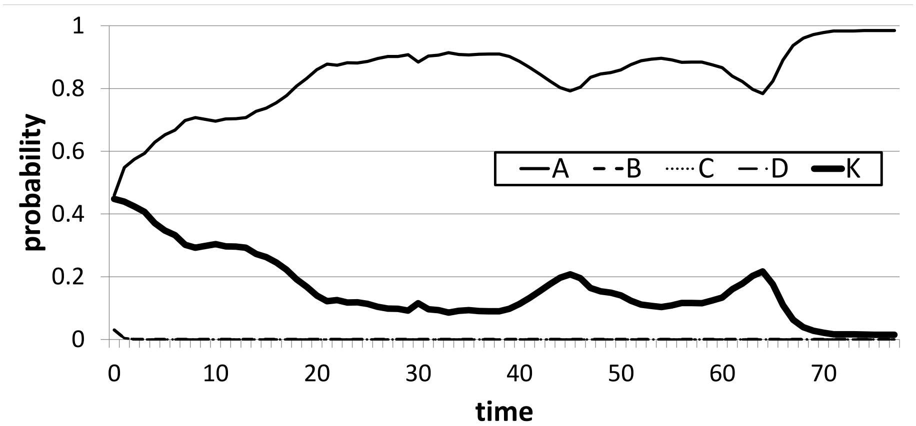
Figure 7.2: Computer graphics rendered images of our scenario. (Courtesy of Davide Baltieri)

distribution for each location is uniform. Regarding the prediction, a uniform transition matrix has been defined according to a map of locations, e.g. the probability of moving from one location to others is uniform for all but the case of two locations which are not directly connected with each other or separated by some barrier (e.g. wall), where the probability is set to zero.

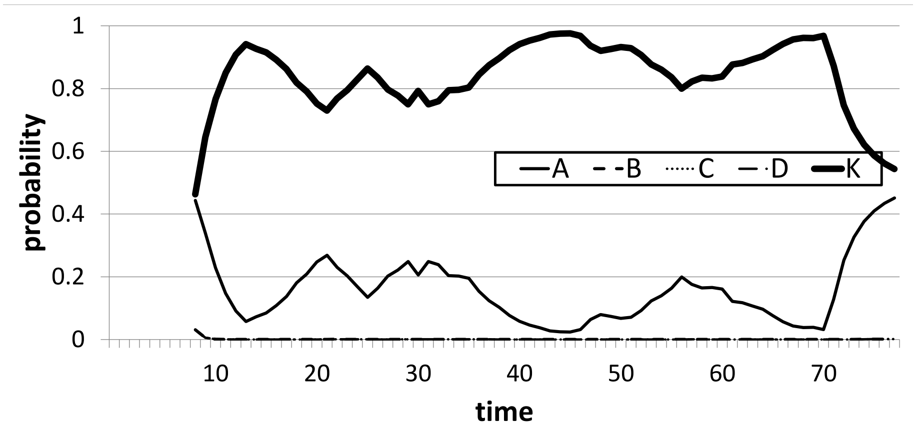
In the following we report the descriptions and the results of the different tests performed with *Reasoning Engine I* (Section 7.2.2), where $\Omega = \{A, B, C, D\} \cup \{\mathbb{k}\}$.

***Case 1*: two people, one authorized (tag A), the other intruder - Fig. 7.3**: in this case, the authorized people (ν_1) and the intruder (ν_2) walk side by side, making identification more challenging. Fig. 7.3(a)(a) and Fig. 7.3(b)(b) show the probability of different tags to be mapped to the respective person. Average precision and recall values (see caption of Fig. 7.3) are 100.0%.

***Case 1*: situation with four people, two authorized (tags A and B), and two intruders - Fig. 7.4**: this scenario presents several difficulties. Three people, one authorized (ν_1) with tag A and two intruders (ν_3, ν_4), walk in group and after a while (around time 60) the intruders move away and then leave the scene. So, because they are very close to each other, it is not surprising that tag A is mapped to the wrong person for some time.

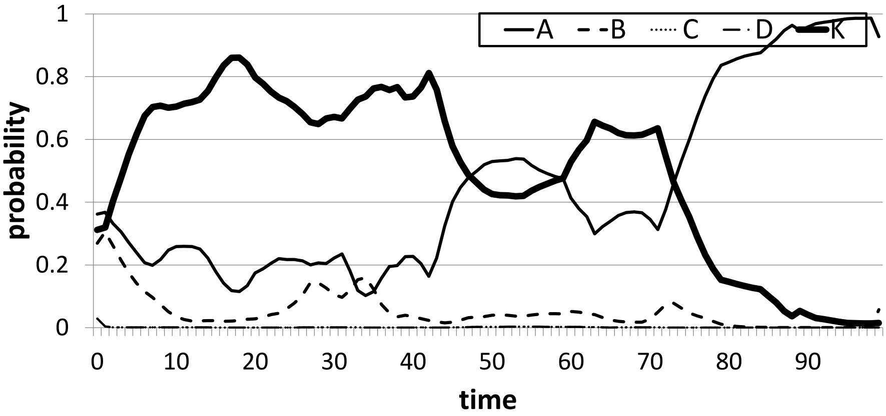


(a) Probability of ν_1

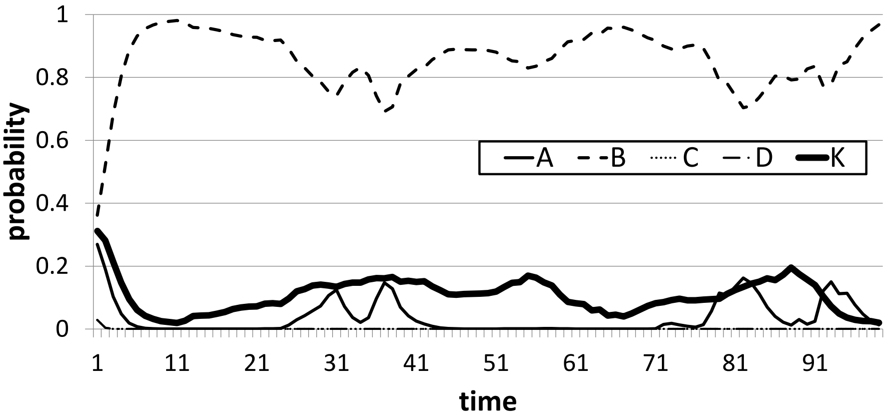


(b) Probability of ν_2

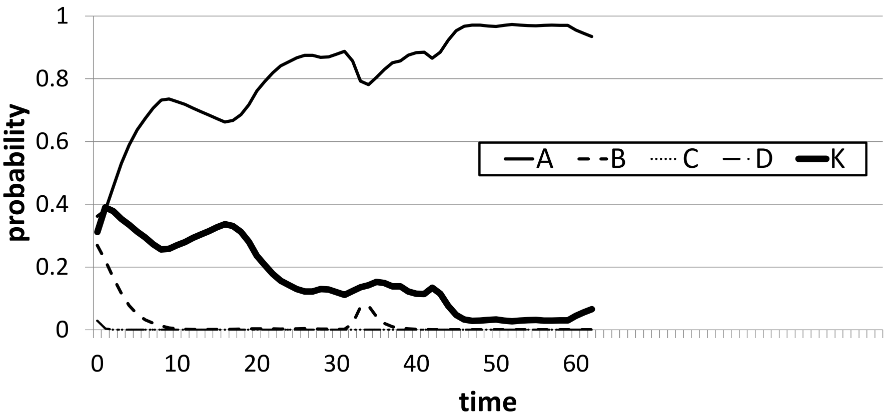
Figure 7.3: *Case 1*, test with one authorized person (ν_1) and one intruder (ν_2). Avg. precision=100.0%, avg. recall=100.0%.



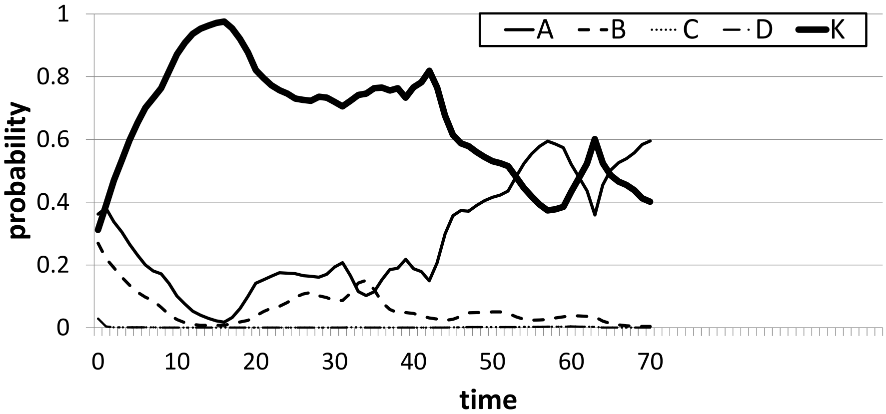
(a) Probability of ν_1



(b) Probability of ν_2



(c) Probability of ν_3



(d) Probability of ν_4

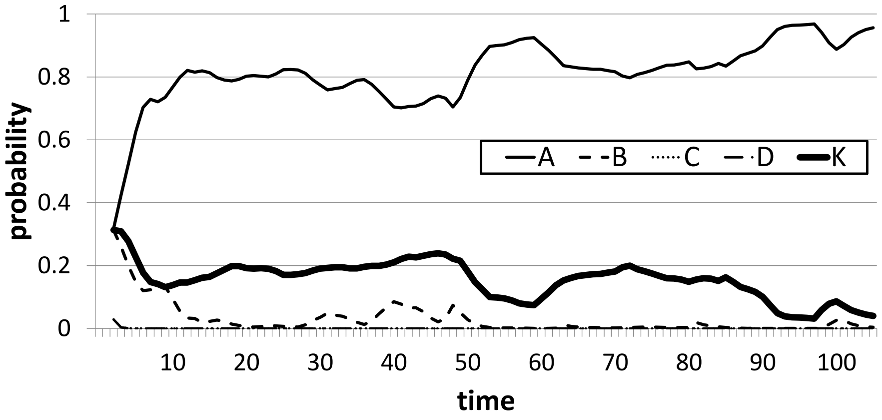
	A	B	C	D	K
A	39	0	0	0	61
B	0	99	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
K	77	0	0	0	57

(e) Confusion matrix

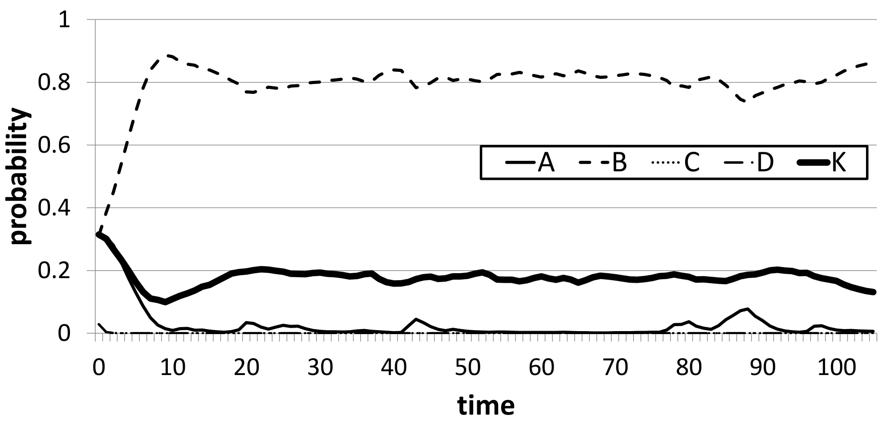
Figure 7.4: *Case 1*, test with two authorized people (ν_1 and ν_2) and two intruders (ν_3 and ν_4). Avg. precision=57.6%, Avg. recall=56.0%.

It is worth noting, however, that System is very confident that the tag A and the two intruders are part of the group, because the group localization over time is similar to the one of tag A and there are no other tag with similar localization. The correct mapping is obtained after the group splits. The other authorized person (ν_2) holding tag B, which is located far enough to not be confused with the other tag, is correctly mapped. In this case, Fig. 7.4(e) also reports the confusion matrix. The poor average precision and recall values are caused by the mapping errors between tags A and $\mathbb{k}$, which are the System's most probable decision at every time. Nevertheless, the errors are bounded to the people in the same group, which cannot be distinguished without other cues.

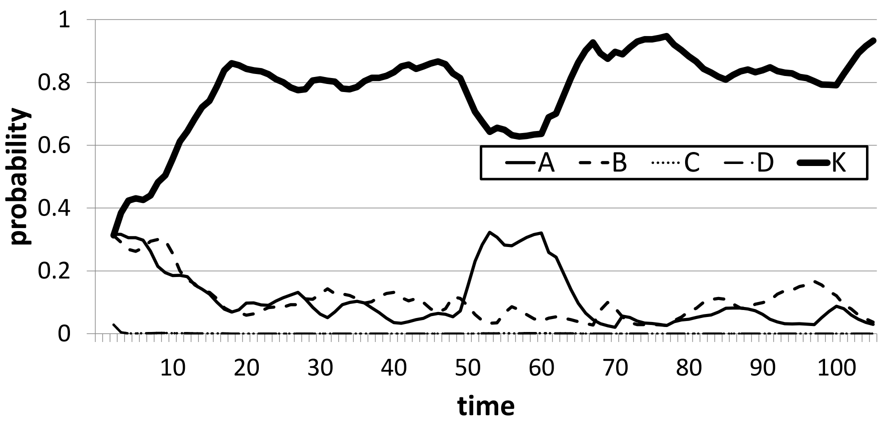
***Case 2*: three people, two authorized (tags A and B) and one intruder - Fig. 7.5**: in this case the two authorized people walk side by side, and the intruder follows them at some distance. Aside from the multi-camera issues, this scenario is more complex than the previous one because the two people walk together for the entire test. As a consequence, the mapping process relies only on the similarity



(a) Probability of ν_1



(b) Probability of ν_2

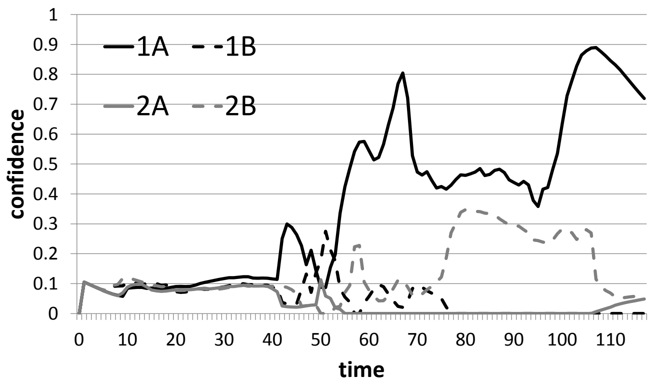


(c) Probability of ν_3

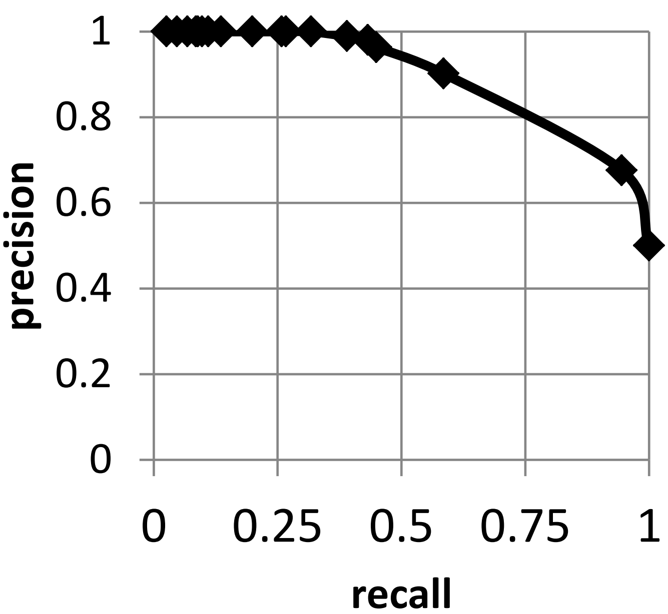
Figure 7.5: *Case 2*, test with two authorized people (ν_1 and ν_2) and one intruder (ν_3). Avg. precision=99.4%, Avg. recall=99.4%.

between tags and people localization. Once the data allow the system to discover a correct mapping, the reinforcement step excludes this mapping for the other person. The intruder is well mapped because, even if he/she is very often in the same location as the others, he/she moves from one location to another with some delay with respect to them. These differences in the localization are enough to get and keep the correct mapping. By increasing the number of locations, it is possible to cover a larger area, and thus track people for longer time. The more data are available, the more reliable is the mapping.

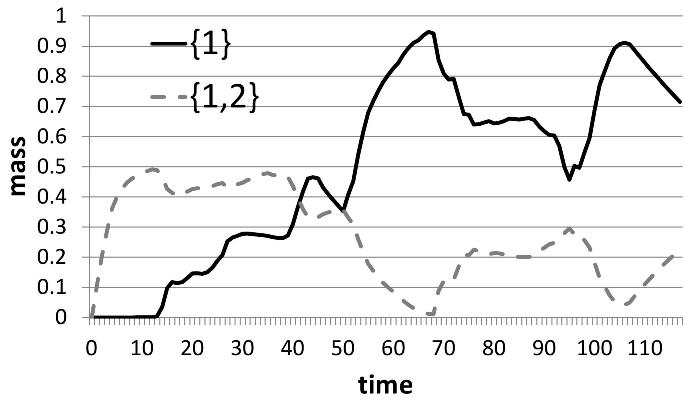
Next, we will present the descriptions and the results of the different tests performed with *Reasoning Engine II* (Section 7.2.2).



(a) Confidence on the mappings.



(b) Precision–Recall graph



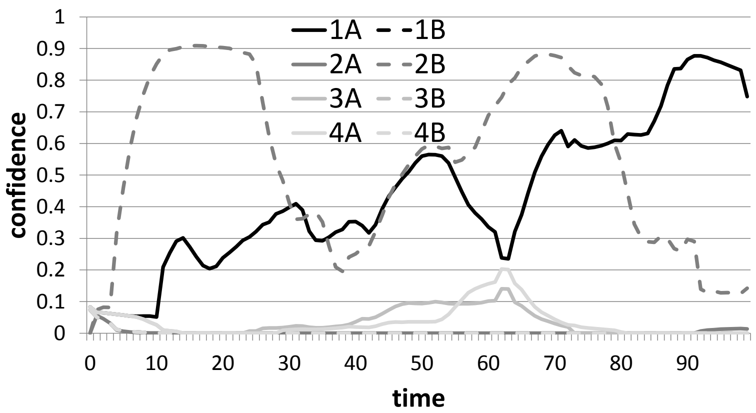
(c) Some focal elements of $bMap^{\tau_A}$; $1 \equiv \nu_1$ and $2 \equiv \nu_2$

Figure 7.6: *Test 1*: The correct mappings are: $\langle \nu_1, \tau_A \rangle \equiv$1A and $\langle \nu_2, \tau_B \rangle \equiv$2B

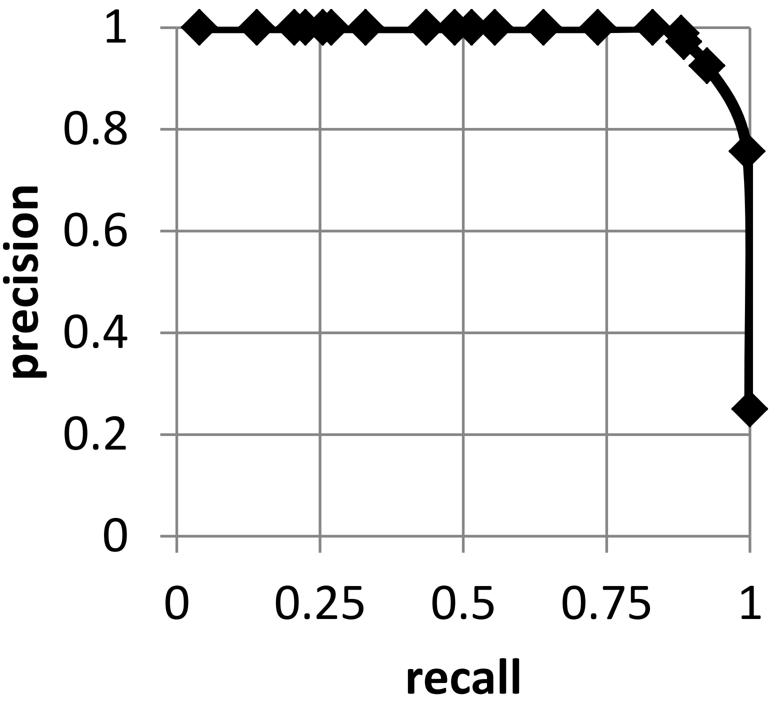
***Test 1*: two people, two authorized (tags τ_A and τ_B) (Fig. 7.6)**. The two authorized people ν_1 and ν_2, holding respectively τ_A and τ_B, walk side by side for a while and then split (around time 50). System is able to manage the group of people thanks to the nature of TBM. As long as the group moves together it is impossible to discover the correct mappings: System believes that every combination of people and tags could be a possible mapping. The data available after the splitting allow to refine System's knowledge, *transferring the belief* to more specific sets of hypothesis (Fig. 7.6(c), where around time 50 the belief is transferred from the set $\{1, 2\}$ to the correct set $\{1\}$) until a mapping is found. We also compute the average precision and recall by changing the threshold applied on the confidence of the mappings reported in Fig. 7.6(a). The different values are collected in the precision–recall graph shown in Fig. 7.6(b).

***Test 2*: four people, two authorized (tags τ_A and τ_B), and two intruders (Fig. 7.7)**. The authorized person ν_1 holding tag τ_A walks nearby a group of two intruders. Because of the short distance between ν_1 and the intruders, they are always detected in the same location, except when they cross an edge between two locations (around time 10). The history coded in the belief functions allows to keep the correct mapping $\langle \nu_1, \tau_A \rangle$ from that moment on. The two intruders ν_3 and ν_4 are correctly found as not holding any tag. The other authorized person (ν_2) holding tag τ_B, which is located far enough to not be confused with the others, is correctly mapped $\langle \nu_2, \tau_B \rangle$ for most of the time.

***Test 3*: three people, two authorized (tags τ_A and τ_B), and one intruder (Fig. 7.8)**. The two authorized people ν_1 and ν_2 walk side by side, while the intruder ν_3 follows them at some distance. System is able to manage the ambiguous situation keeping the confidences on the mappings very low when ν_1 and ν_2 are very close and their localization is the same.



(a) Confidence on the mappings

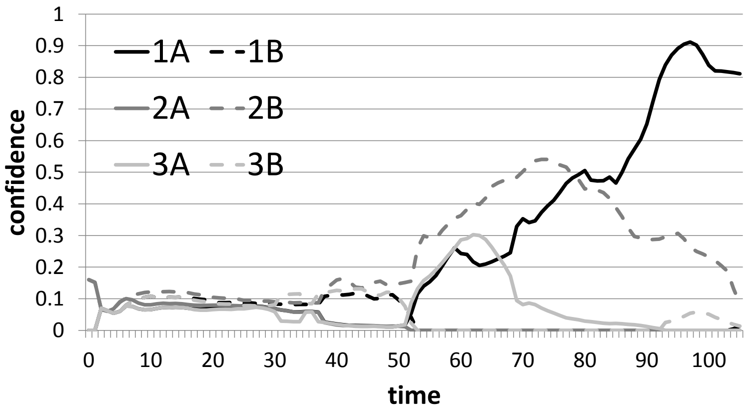


(b) Precision–Recall graph

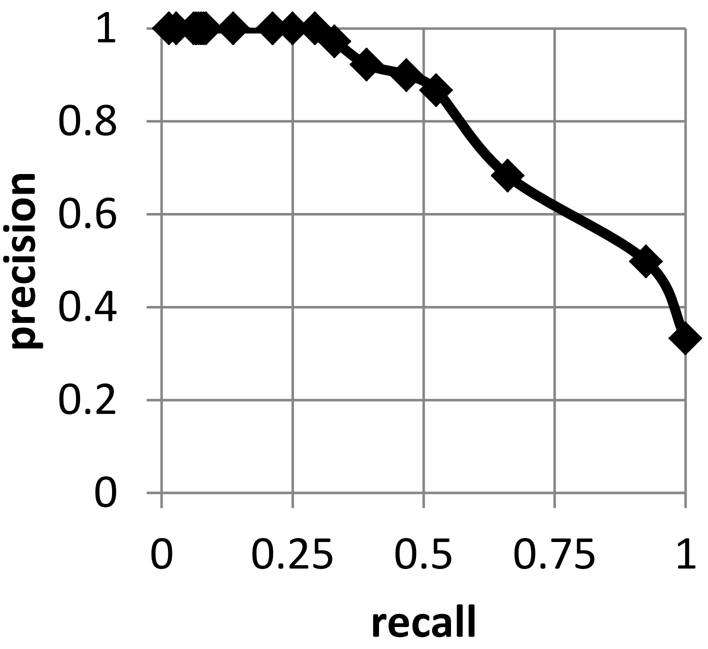
Figure 7.7: *Test 2*: The correct mappings are: $\langle \nu_1, \tau_A \rangle \equiv$ 1A and $\langle \nu_2, \tau_B \rangle \equiv$ 2B

When the two people split (around time 65), however, data on localization allow to recognize the correct mappings and their confidences increase quickly. The confidence on the mappings for the intruder ν_3 is correctly always very low, because his position is different from the one of both tags. The precision and recall values are lower than test 1 because of the long-lasting side-by-side walk of

ν_1 and ν_2.



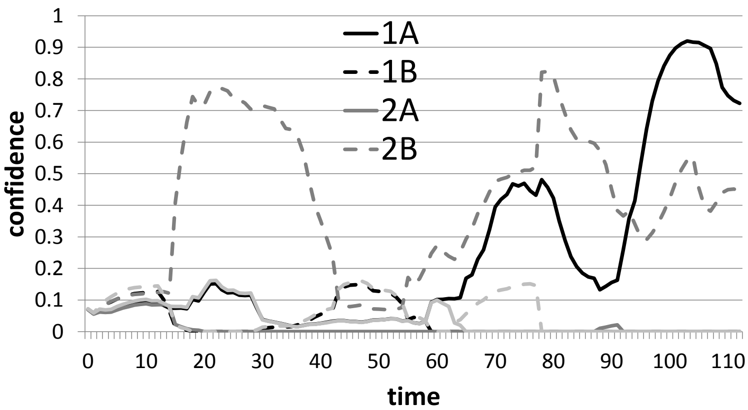
(a) Confidence on the mappings.



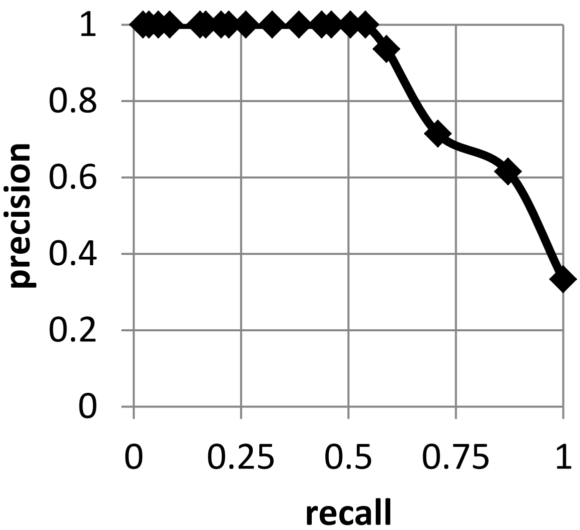
(b) Precision–Recall graph

Figure 7.8: *Test 3*: The correct mappings are: $\langle\nu_1, \tau_A\rangle \equiv$1A and $\langle\nu_2, \tau_B\rangle \equiv$2B

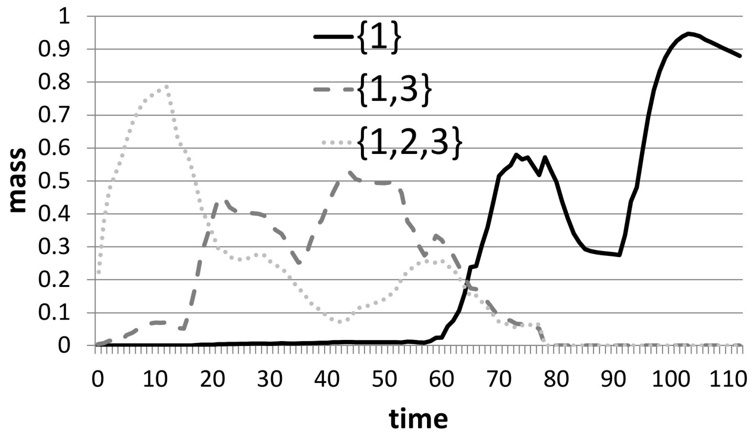
***Test 4*: three people, two authorized (tags τ_A and τ_B), and one intruder (Fig. 7.9)**. The authorized person ν_1 (tag τ_A) and the intruder ν_3 walk side by side, and the other authorized person ν_2 follows them at some distance. The correct mapping $\langle\nu_2, \tau_B\rangle$ is soon found, while $\langle\nu_1, \tau_A\rangle$ has low confidence until ν_1 and the intruder split (around time 60). Analyzing some focal elements of the belief



(a) Confidence on the mappings.



(b) Precision–Recall graph



(c) Some focal elements of $bMap^{\tau_A}$; $1 \equiv \nu_1$, $2 \equiv \nu_2$, $3 \equiv \nu_3$

Figure 7.9: *Test 4*: The correct mappings are: $\langle \nu_1, \tau_A \rangle \equiv 1\text{A}$ and $\langle \nu_2, \tau_B \rangle \equiv 2\text{B}$

function $bMap^{\tau_A}$ (Fig. 7.9(c)), is clearly visible how System first allocates most of the mass to the broader set of hypothesis (i.e. System believes that τ_A can be held by every person) and progressively transfers the mass to narrower sets, until only one hypothesis (ν_1) remains.

***Degeneration of the system*:** In some particular conditions, e.g. when an intruder ν_a is present in the scene, and, simultaneously, a tag τ_j is sensed but the relative authorized person is not detected, System is forced to map that tag with the intruder. The similarity values in 7.2.2.c would be very low, and the evidence generated in 7.2.2.d would assign correctly most of the mass to the $\mathcal{T}_t$ set. Nevertheless, because $\mathcal{T}_t$ contains τ_j only, the resulting mapping is $\langle \nu_a, \tau_j \rangle$, which is incorrect.

To avoid these wrong mappings, at every t a dummy person ν_x and a dummy tag τ_k (on which System has no belief on mapping) are added to *close the world* on the sets of hypotheses. Both of them are localized with a uniform distribution over the set of locations $\mathcal{L}$. Thus, $\mathcal{V}_t$ and $\mathcal{T}_t$ always contain, respectively, the identifiers ν_x and τ_k.

7.4 Related Works

In the last years, the surveillance of wide open areas has become an urgent matter for security reasons. In particular, the so-called "third generation" of intelligent video surveillance systems [Valera and Velastin, 2005] has been conceived to provide more accurate information by fusing more sensors, possibly belonging to different types (not only cameras). As stated in [Valera and Velastin, 2005], this requirement poses several challenges, summarized in: (i) distributed versus centralized intelligence, (ii) data fusion, (iii) probabilistic reasoning framework, (iv) multi-camera surveillance techniques.

Our work proposes an integrated framework with joint use of RFID sensors and cameras for detecting intruders. The following related works will basically focus on: (a) examples of joint use of RFIDs and cameras for surveillance applications; (b) management of RFID sensors; (c) data fusion using a probabilistic reasoning framework.

Though distributed video surveillance is not new by itself, the use of different sensors and advanced reasoning techniques is not so diffused in the literature. For instance, the combination of cameras and RFID sensors is proposed in [Wickramasuriya et al., 2004, Zhang et al., 2005] only to avoid to expose the privacy of authorized people in recording video streams. In this case, however, the scenario comprises buildings with doors and entrances where the user is forced to authenticate by means of RFID technology before entering in the monitored area: if the person is authorized the recorded video is protected by a watermarking algorithm. Therefore, this approach does not allow to identify those who are authorized and those who are not among several people.

Another example is provided by the work in [Nohara et al., 2008] where a robot simultaneously interacts with two or more people and has to identify them with a passive-type RFID reader and floor sensors. To solve the association problem, if two or more people are around the robot, hypotheses are modeled using Bayesian networks and validated using the observations. In [Cho et al., 2009] a sensor fusion method for an heterogeneous sensor environment with visual and identification sensors is proposed. The problem of the coverage uncertainty of the sensors is managed by grouping unassociated identifications. Despite these examples and to the best of our knowledge, a complete RFID/camera system for intruder detection in wide open area is still missing in the literature.

Coming to the use of RFID sensors, see Section 5.4 for details. It is worth noting that all such RFID systems define locations on the basis of actual places/areas which are of interest to the final users (e.g. a restricted-access room), as reflected also by the supported queries and the produced results (e.g. "Find out which rooms entered Paul today"). In this regard, the combined RFID/camera system we propose in this chapter, differs from this vision: in our case the subdivision of the open area in a number of locations is solely an internal parameter which can be fine-tuned by the system administrator so to allow the best possible (and effective) communication between the RFID sensors' and the cameras' processing engines. While our final users do not even need to be aware of the chosen locations, they will surely benefit from a "smart" location choice, allowing the system to better and faster identify/localize the people.

Finally, our approach makes use of a TBM for inferring the mapping between people and RFID tags. TBM has been used in the literature for different applications, such as for the classification of the camera motion [Guironnet et al., 2007] and for developing a system for advanced driver assistance [Clerentin et al., 2009]. The work in [Clerentin et al., 2009] is particularly interesting since it considers two heterogeneous sources of information (omnidirectional cameras and a laser scanner), but with similar objective (the localization of vehicles). Here, instead, the two heterogeneous sources also have heterogeneous purposes.

Chapter 8

Conclusions and Future work

This thesis has studied and described the challenges related to data management in an RFID system, and has presented the design, implementation and evaluation of realtime system named *RPDM*. *RPDM* addresses these issues in the context of location tracking. In particular, it focused on *Online Filtering & Uncertainty Management* of RFID data, efficient RFID data *Aggregation & Storing* and *Temporal Probabilistic Query Execution*. Particularly, the main contributions of this thesis are as follow:

- We studied RFID and RFID data management systems with their architectures and main applications. We also presented a description of the interesting research projects in this area. An overview of the most innovative solutions to the design issues arising in RFID data management systems is provided too.

- This thesis considered the problem of RFID data management in the context of location tracking, and identified the transformation of unreliable and imprecise RFID data streams into reliable probabilistic data streams and huge volume of data streams as fundamental issues.

- We presented an RFID *Data Acquisition, Online Filtering & Uncertainty Management* mechanism that manages unreliable and imprecise RFID data streams and transforms them into reliable probabilistic data streams. The developed method includes a specifically designed data model based on probabilistic graphical model *HMM*. The presented model combines prior domain knowledge about the system behavior in the form of its parameters with the actual observations to infer the hidden variables. Moreover, proposed model used a sample based sequential Monte Carlo algorithm, *Particle Filtering* to infer the locations of the people or objects in location tracking system. We presented a series of experiments performed under real cases showing the effectiveness of our approach.

- We proposed a simple *on-line summarization* mechanism for massive RFID probabilistic data streams that provides small space representation for them while preserving the meaningful information. The proposed mechanism is based on concept of clustering. In particular, the proposed approach keep on aggregating tuples in an incremental way until a state transition is detected, while avoiding redundant information produced in stable states. The approach correctly identify and highlight state transitions according to three boundary conditions: Maximum Probability Change (MPC), Diameter-oriented (DM) and Centroid vs Latest Reading (CLRC). Finally, we experimentally proved that a probabilistic database such as MayBMS can effectively answer a wide range of probabilistic queries on the summarized version of the data, which only take up a fraction of the original space.

- We proposed an innovative system for intruder detection relying on the joint use of cameras and RFIDs, in real noisy and complex wide open areas. The proposed system built on sophisticated filtering technique applied to RFID signals and evidential fusion architecture based on *TBM* used as reasoning engine, where these methods successfully handled the noise in the data and the uncertainty in the localization. The effectiveness of the achieved results has been experimentally proved through a series of real tests, where the proposed RFID/camera system shows excellent inference properties in localizing intruders in wide open areas, also in challenging cases where authorized people and intruders follow the same path in group.

Based on ideas and work presented in thesis, following main aspects of future work can be considered:

- We plan to further extend the application of proposed *RPDM System* in order to explore the people's behavior in an environment. To this end, on the basis of estimated positions of the persons or objects, we intend to find out the trajectory of the tracked person/object. These trajectories can be a good source of information in order to estimate the particular behavior or activities done by the people/object in a particular environment. In this context, we plan to investigate following main perspectives: in an environment such as university, which paths are most commonly used by the students to reach specific area e.g. classroom, hall, coffee room etc. The main idea behind this is to estimate the visiting patterns of the students. Moreover, which places are usually crowded/uncrowded to identify the space usage of different parts of an environment. We also believe that the extension of our idea is applicable in any environment such as hospital, shopping mall etc.

- Another relevant future work can be the use of other state of the art or available probabilistic data management systems other than MayBMS such as Orion, Trio, MystiQ or PrDB. We intend to assess the performance of these different probabilistic DBMSs on our generated probabilistic tuples in context of uncertainty handling and temporal probabilistic query execution.

Publications related to this thesis

- R. Haider, F. Mandreoli, R. Martoglia, S. Sassatelli and P. Tiberio: Toward a Flexible Data Management Middleware for Wireless Sensor Networks in Proceedings of the VI Conference of the Italian Chapter of AIS (ItAIS 2009), October 2-3 2009, Costa Smeralda, Italy.
- R. Haider, F. Mandreoli, R. Martoglia, S. Sassatelli and P. Tiberio: "Toward a Flexible Data Management Middleware for Wireless Sensor Networks", in Management of the Interconnected World: ItAIS: the Italian Association for Information Systems, D'Atri, A.; De Marco, M.; Braccini, A.M.; Cabiddu, F. (Eds.),(Book Chapter) pp: 157-165, ISBN: 978-3-7908-2403-2, Springer - Verlag Berlin Heidelberg ,2010.
- R. Cucchiara M. Fornaciari R. Haider F. Mandreoli R. Martoglia A. Prati S. Sassatelli: "A Reasoning Engine for Intruders' Localization in Wide Open Areas using a Network of Cameras and RFIDs , in Proceedings of the 1st IEEE Workshop on Camera Networks and Wide Area Scene Analysis (IEEE WCNWASA 2011) , June 2011, Colorado Springs, USA.
- R. Cucchiara, M. Fornaciari, R. Haider, F. Mandreoli and A. Prati:" Identification of Intruders in Groups of People using Cameras and RFIDs", in Proceedings of Fifth ACM/IEEE International Conference on Distributed Smart Cameras, August 22-25, 2011, Ghent, Belgium
- R. Haider, F. Mandreoli, R. Martoglia and S. Sassatelli: "Fast On-Line Summarization of RFID Probabilistic Data Streams", in International Conference on Information Systems, Technology & Management, March 28-30, 2012, Grenoble, France.

Bibliography

[may,] http://www.cs.cornell.edu/bigreddata/maybms/.

[pos,] http://www.postgresql.org/.

[gre,] http://gretl.sourceforge.net/.

[whi, 2002] (2002). Active and passive rfid: Two distinct, but complementary, technologies for real-time supply chain visibility:[white paper]. by Savi Technology. http://logmgt.nkmu.edu.tw/news/articles/White%20Paper-Active%20and%20Passive%20RFID.pdf.

[rfi, 2005] (2005). Rfid working group, radio frequency identification opportunities and challenges in implementation. Technical report, Department of Defence, Washington D.C. http://www.technology.gov/reports.htm.

[rfi, 2007] (2007). The rfid ecosystem. http://rfid.cs.washinton.edu/,2007.

[Abadi et al., 2003] Abadi, D., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., and Zdonik, S. (2003). Aurora: a new model and architecture for data stream management. *The VLDB Journal*, 12(2):120–139.

[Aggarwal and Yu, 2008] Aggarwal, C. and Yu, P. (2008). A framework for clustering uncertain data streams. In *Proceedings of the 24th international conference on Data Engineering*, pages 150–159. IEEE.

[Ahsan et al., 2009a] Ahsan, K., Kingston, P., and Shah, H. (2009a). Context based knowledge management in healthcare: An ea approached. *AMCIS 2009 Proceedings*, page 297.

[Ahsan et al., 2009b] Ahsan, K., Shah, H., and Kingston, P. (2009b). The role of enterprise architecture in healthcare-it. In *2009 Sixth International Conference on Information Technology: New Generations*, pages 1462–1467. IEEE.

[Antova et al., 2008] Antova, L., Jansen, T., Koch, C., and Olteanu, D. (2008). Fast and simple relational processing of uncertain data. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 983–992. IEEE.

[Arnaud et al., 2005] Arnaud, D., de Freitas, N., and Neil, G. (2005). *Sequential Monte Carlo Methods in Practice*. Springer.

[Bai et al., 2006] Bai, Y., Wang, F., and Liu, P. (2006). Efficiently filtering rfid data streams. In *CleanDB Workshop*, pages 50–57.

[Bishop and en ligne), 2006] Bishop, C. and en ligne), S. S. (2006). *Pattern recognition and machine learning*, volume 4. Springer New York.

[Blackman, 1986] Blackman, S. (1986). Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1.

[Bleco and Kotidis, 2009] Bleco, D. and Kotidis, Y. (2009). Rfid data aggregation. *GeoSensor Networks*, pages 87–101.

[Blythe, 1999] Blythe, P. (1999). Rfid for road tolling, road-use pricing and vehicle access control. In *RFID Technology (Ref. No. 1999/123), IEE Colloquium on*, pages 8–1. IET.

[Calderara et al., 2008] Calderara, S., Prati, A., and Cucchiara, R. (2008). Hecol: Homography and epipolar-based consistent labeling for outdoor park surveillance. *Computer Vision and Image Understanding*, 111(1):21–42.

[Chandrasekaran et al., 2003] Chandrasekaran, S., Cooper, O., Deshpande, A., Franklin, M., Hellerstein, J., Hong, W., Krishnamurthy, S., Madden, S., Raman, V., Reiss, F., et al. (2003). Telegraphcq: Continuous dataflow processing for an uncertain world. *CIDR*.

[Chawathe et al., 2004] Chawathe, S. S., Krishnamurthy, V., Ramachandran, S., and Sarma, S. (2004). Managing RFID data. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1189–1195.

[Chen et al., 2010] Chen, H., Ku, W., Wang, H., and Sun, M. (2010). Leveraging spatio-temporal redundancy for rfid data cleansing. In *Proceedings of the 2010 international conference on Management of data*, pages 51–62. ACM.

[Cho et al., 2009] Cho, S. H., Hong, S., and Nam, Y. (2009). Association and identification in heterogeneous sensors environment with coverage uncertainty.

In *AVSS '09: Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 553–558, Washington, DC, USA. IEEE Computer Society.

[Choi et al., 2006] Choi, J., Oh, D., and Song, I. (2006). R-lim: an affordable library search system based on rfid.

[Chow, 1990] Chow, Y. (1990). Maximum mutual information estimation of hmm parameters for continuous speech recognition using the n-best algorithm. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 701–704. IEEE.

[Clerentin et al., 2009] Clerentin, A., Delahoche, L., Marhic, B., Delafosse, M., and Allart, B. (2009). An evidential fusion architecture for advanced driver assistance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 327 –332.

[Cocci et al., 2007] Cocci, R., Diao, Y., and Shenoy, P. (2007). Spire: Scalable processing of rfid event streams. *Proceedings of the 5th RFID Academic Convocation, Brussels, Belgium*.

[Cocci et al., 2012] Cocci, R., Nie, Y., Diao, Y., and Shenoy, P. (2012). Spire: Efficient data inference and compression over rfid streams. *Knowledge and Data Engineering, IEEE Transactions on*, pages 141–155.

[Cocci et al., 2008] Cocci, R., Tran, T., Diao, Y., and Shenoy, P. (2008). Efficient data interpretation and compression over rfid streams. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 1445–1447. IEEE.

[Cooper et al., 2004] Cooper, O., Edakkunni, A., Franklin, M., Hong, W., Jeffery, S., Krishnamurthy, S., Reiss, F., Rizvi, S., and Wu, E. (2004). Hifi: A unified architecture for high fan-in systems. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1357–1360. VLDB Endowment.

[Cucchiara et al., 2011] Cucchiara, R., Fornaciari, M., Haider, R., Mandreoli, F., Martoglia, R., Prati, A., and Sassatelli, S. (2011). A Reasoning Engine for Intruders' Localization in Wide Open Areas using a Network of Cameras and RFIDs. In *Proceedings of 1st IEEE Workshop on Camera Networks and Wide Area Scene Analysis*. IEEE.

[Cucchiara et al., 2010] Cucchiara, R., Fornaciari, M., Prati, A., and Santinelli, P. (2010). Mutual calibration of camera motes and rfids for people localization and identification. In *Proceedings of the ACM/IEEE ICDSC 2010*.

[Cucchiara et al., 2003] Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts and shadows in video streams. 25(10):1337–1342.

[Darcy et al., 2010] Darcy, P., Stantic, B., and Sattar, A. (2010). Applying a neural network to recover missed rfid readings. In *Proceedings of the Thirty-Third Australasian Conferenc on Computer Science-Volume 102*, pages 133–142. Australian Computer Society, Inc.

[De Virgilio et al., 2009] De Virgilio, R., Sugamiele, P., and Torlone, R. (2009). Incremental aggregation of rfid data. In *Proceedings of the 2009 International Database Engineering & Applications Symposium*, pages 194–205. ACM.

[Dechter, 1998] Dechter, R. (1998). Bucket elimination: A unifying framework for probabilistic inference. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 89:75–104.

[Derakhshan et al., 2007] Derakhshan, R., Orlowska, M., and Li, X. (2007). Rfid data management: challenges and opportunities. In *RFID, 2007. IEEE International Conference on*, pages 175–182. IEEE.

[Deshpande et al., 2005a] Deshpande, A., Guestrin, C., and Madden, S. (2005a). Using probabilistic models for data management in acquisitional environments. In *Proc. CIDR*, pages 317–328.

[Deshpande et al., 2005b] Deshpande, A., Guestrin, C., Madden, S. R., Hellerstein, J. M., and Hong, W. (2005b). Model-based approximate querying in sensor networks. *The VLDB Journal*, 14(4):417–443.

[Diao et al., 2007] Diao, Y., Immerman, N., and Gyllstrom, D. (2007). Sase+: An agile language for kleene closure over event streams. Technical report, UMass Technical Report 07.

[Domdouzis et al., 2007] Domdouzis, K., Kumar, B., and Anumba, C. (2007). Radio-frequency identification (rfid) applications: A brief introduction. *Advanced Engineering Informatics*, 21(4):350–355.

[Doucet et al., 2001] Doucet, A., De Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer Verlag.

[Doucet et al., 2000] Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

[Ehrenberg et al., 2007] Ehrenberg, I., Floerkemeier, C., and Sarma, S. (2007). Inventory management with an rfid-equipped mobile robot. In *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, pages 1020–1026. IEEE.

[Ester et al., 1996] Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. Portland: AAAI Press.

[Fazzinga et al., 2009] Fazzinga, B., Flesca, S., Masciari, E., and Furfaro, F. (2009). Efficient and effective rfid data warehousing. In *Proceedings of the 2009 International Database Engineering & Applications Symposium*, pages 251–258. ACM.

[Floerkemeier and Lampe, 2004] Floerkemeier, C. and Lampe, M. (2004). Issues with RFID usage in ubiquitous computing applications. *Pervasive Computing*, pages 188–193.

[Franklin et al., 2005] Franklin, M. J., Jeffery, S. R., Krishnamurthy, S., Reiss, F., Rizvi, S., Wu, E., Cooper, O., Edakkunni, A., and Hong, W. (2005). Design considerations for high fan-in systems: The HiFi approach. In *Proc. of the CIDR Conf.*

[Garofalakis et al., 2006] Garofalakis, M. N., Brown, K. P., Franklin, M. J., Hellerstein, J. M., Wang, D. Z., Michelakis, E., Tancau, L., Wu, E., Jeffery, S. R., and Aipperspach, R. (2006). Probabilistic data management for pervasive computing: The data furnace project. *IEEE Data Eng. Bull*, 29(1):57–63.

[Gauvain and Lee, 1994] Gauvain, J. and Lee, C. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *Speech and Audio Processing, IEEE Transactions on*, 2(2):291–298.

[Gelb, 1999] Gelb, A. (1999). *Applied optimal estimation*. MIT press.

[Gonzalez et al., 2006a] Gonzalez, H., Han, J., and Li, X. (2006a). Flowcube: constructing rfid flowcubes for multi-dimensional analysis of commodity flows. In *Proceedings of the 32nd international conference on Very large data bases*, pages 834–845. VLDB Endowment.

[Gonzalez et al., 2006b] Gonzalez, H., Han, J., Li, X., and Klabjan, D. (2006b). Warehousing and analyzing massive RFID data sets. In *22nd International Conference on Data Engineering, ICDE'06*. IEEE Computer Society.

[Gonzalez et al., 2006c] Gonzalez, H., Han, J., Li, X., and Klabjan, D. (2006c). Warehousing and analyzing massive RFID data sets. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, page 83.

[Guha et al., 1998] Guha, S., Rastogi, R., and Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM.

[Guironnet et al., 2007] Guironnet, M., Pellerin, D., and Rombaut, M. (2007). A fusion architecture based on tbm for camera motion classification. *Image Vision Comput.*, 25:1737–1747.

[Günther et al., 2008] Günther, O., Kletti, W., and Kubach, U. (2008). *RFID in Manufacturing*. Springer Verlag.

[G:Welch and G.Bishop, 2002] G:Welch and G.Bishop (2002). An introduction to the kalman filter.

[Gyllstrom et al., 2006] Gyllstrom, D., Wu, E., Chae, H., Diao, Y., Stahlberg, P., and Anderson, G. (2006). Sase: complex event processing over streams. *Arxiv preprint cs/0612128*.

[Hakim et al., 2006] Hakim, H., Renouf, R., and Enderle, J. (2006). Passive rfid asset monitoring system in hospital environments. In *Bioengineering Conference, 2006. Proceedings of the IEEE 32nd Annual Northeast*, pages 217–218. IEEE.

[Hayashi et al., 2003] Hayashi, H., Tsubaki, T., Ogawa, T., and Shimizu, M. (2003). Asset tracking system using long-life active rfid tags. *NTT Technical Review*, 1(9):19–26.

[Hu and Cheng, 2010] Hu, W. C. and Cheng, Z. L. (2010). Clustering algorithm for probabilistic data streams over sliding window. In *Proceedings of the 9th International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 2065–2070. IEEE.

[Hu et al., 2005] Hu, Y., Sundara, S., Chorma, T., and Srinivasan, J. (2005). Supporting rfid-based item tracking applications in oracle dbms using a bitmap datatype. In *Proceedings of the 31st international conference on Very large data bases*, pages 1140–1151. VLDB Endowment.

[Huang et al., 2009] Huang, J., Antova, L., Koch, C., and Olteanu, D. (2009). MayBMS: a probabilistic database management system. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 1071–1074. ACM.

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Compututing Survey*, 31:264–323.

[Jeffery et al., 2006a] Jeffery, S., Alonso, G., Franklin, M., Hong, W., and Widom, J. (2006a). Declarative support for sensor data cleaning. *Pervasive Computing*, pages 83–100.

[Jeffery et al., 2008] Jeffery, S. R., Franklin, M. J., and Garofalakis, M. (2008). An adaptive RFID middleware for supporting metaphysical data independence. *The VLDB JournalThe International Journal on Very Large Data Bases*, 17(2):289.

[Jeffery et al., 2006b] Jeffery, S. R., Garofalakis, M., and Franklin, M. J. (2006b). Adaptive cleaning for RFID data streams. In *Proceedings of the 32nd international conference on Very large data bases*, pages 163–174.

[Jensen and Jensen, 1994] Jensen, F. and Jensen, F. (1994). *Optimal junction trees*. Citeseer.

[Jousselme et al., 2001] Jousselme, A.-L., Grenier, D., and Bosse, E. (2001). A new distance between two bodies of evidence. *Information Fusion*, 2(2):91 – 101.

[Kanagal and Deshpande, 2008] Kanagal, B. and Deshpande, A. (2008). Online filtering, smoothing and probabilistic modeling of streaming data. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 1160–1169.

[Kärkkäinen, 2003] Kärkkäinen, M. (2003). Increasing efficiency in the supply chain for short shelf life goods using rfid tagging. *International Journal of Retail & Distribution Management*, 31(10):529–536.

[Khoussainova et al., 2006] Khoussainova, N., Balazinska, M., and Suciu, D. (2006). Towards correcting input data errors probabilistically using integrity constraints. In *Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*, pages 43–50.

[Khoussainova et al., 2007a] Khoussainova, N., Balazinska, M., and Suciu, D. (2007a). Peex: Extracting probabilistic events from rfid data. In *Proceedings* *of the 22th International Conference on Data Engineering (ICDE 08)*, pages 1480–1482.

[Khoussainova et al., 2007b] Khoussainova, N., Balazinska, M., and Suciu, D. (2007b). Probabilistic RFID data management. *Technical Report TR2007-03-01, University of Washington, Seattle, Washington*.

[Khoussainova et al., 2008a] Khoussainova, N., Balazinska, M., and Suciu, D. (2008a). Probabilistic event extraction from RFID data. pages 1480–1482.

[Khoussainova et al., 2008b] Khoussainova, N., Welbourne, E., Balazinska, M., Borriello, G., Cole, G., Letchner, J., Li, Y., Ré, C., Suciu, D., and Walke, J. (2008b). A demonstration of cascadia through a digital diary application. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1319–1322. ACM.

[Kim et al., 2008] Kim, D., Kim, J., Kim, S., and Yoo, S. (2008). Design of RFID based the Patient Management and Tracking System in hospital. In *Engineering in Medicine and Biology Society, EMBS. 30th Annual International Conference of the IEEE*, pages 1459–1461. IEEE.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. The MIT Press.

[Kriegel and Pfeifle, 2005] Kriegel, H. and Pfeifle, M. (2005). Density-based clustering of uncertain data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 672–677. ACM.

[Kwon and Murphy, 2000] Kwon, J. and Murphy, K. (2000). Modeling freeway traffic with coupled hmms. *Unpublished manuscript*.

[Lampe and Strassner, 2003] Lampe, M. and Strassner, M. (2003). The potential of rfid for moveable asset management. In *Workshop on Ubiquitous Commerce at Ubicomp*, volume 2003. Citeseer.

[Lauritzen, 1996] Lauritzen, S. (1996). *Graphical models*, volume 17. Oxford University Press, USA.

[Lee and Chung, 2008] Lee, C. and Chung, C. (2008). Efficient storage scheme and query processing for supply chain management using rfid. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 291–302. ACM.

[Legány et al., 2006] Legány, C., Juhász, S., and Babos, A. (2006). Cluster validity measurement techniques. In *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 388–393.

[MacCormick and Blake, 2000] MacCormick, J. and Blake, A. (2000). A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71.

[Madsen and Jensen, 1999] Madsen, A. and Jensen, F. (1999). Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113(1):203–245.

[Mahdin and Abawajy, 2009] Mahdin, H. and Abawajy, J. (2009). An approach to filtering rfid data streams. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, pages 742–746. IEEE.

[Mihajlovic and Petkovic, 2001] Mihajlovic, V. and Petkovic, M. (2001). Dynamic bayesian networks: A state of the art.

[Murphy, 2002] Murphy, K. (2002). *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, Citeseer.

[Murphy et al., 1999] Murphy, K., Weiss, Y., and Jordan, M. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in AI*, volume 467475.

[Myung, 2003] Myung, I. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100.

[Nath et al., 2006] Nath, B., Reynolds, F., and Want, R. (2006). Rfid technology and applications. *Pervasive Computing, IEEE*, 5(1):22–24.

[Ngai et al., 2006] Ngai, W., Kao, B., Chui, C., Cheng, R., Chau, M., and Yip, K. (2006). Efficient clustering of uncertain data. In *Proceedings of the 6th International Conference on Data Mining(ICDM),*, pages 436–445. IEEE.

[Nie et al., 2009] Nie, Y., Cocci, R., Cao, Z., Diao, Y., and Shenoy, P. (2009). Spire: Efficient data interpretation and compression over rfid streams. *Knowledge and Data Engineering, IEEE Transactions on*, (99).

[Nohara et al., 2008] Nohara, K., Tajika, T., Shiomi, M., Kanda, T., Ishiguro, H., and Hagita, N. (2008). Integrating passive RFID tag and person tracking for social interaction in daily life. *Robot and Human Interactive Communication,* *2008. RO-MAN 2008. The 17th IEEE International Symposium on*, pages 545 –552.

[Peng et al., 2008] Peng, X., Ji, Z., Luo, Z., Wong, E., and Tan, C. (2008). A p2p collaborative rfid data cleaning model. In *Grid and Pervasive Computing Workshops, 2008. GPC Workshops' 08. The 3rd International Conference on*, pages 304–309. IEEE.

[Rabiner, 1990] Rabiner, L. R. (1990). Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296.

[Rao et al., 2006] Rao, J., Doraiswamy, S., Thakkar, H., and Colby, L. S. (2006). A deferred cleansing method for RFID data analytics. In *Proceedings of the 32nd international conference on Very large data bases*, pages 175–186.

[Ré et al., 2008] Ré, C., Letchner, J., Balazinksa, M., and Suciu, D. (2008). Event queries on correlated probabilistic streams. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 715–728.

[Rizvi et al., 2005] Rizvi, S., Jeffery, S., Krishnamurthy, S., Franklin, M., Burkhart, N., Edakkunni, A., and Liang, L. (2005). Events on the edge. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 885–887. ACM.

[SANTOSH and Smith, 2008] SANTOSH, B. and Smith, L. (2008). Rfid in the supply chain: panacea or pandora's box? *Communications of the ACM*, 51(10):127–131.

[Sarma, 2001] Sarma, S. (2001). Towards the five-cent tag.

[Sarma et al., 2003] Sarma, S., Weis, S., and Engels, D. (2003). Rfid systems and security and privacy implications. *Cryptographic Hardware and Embedded Systems-CHES 2002*, pages 1–19.

[Shepard, 2005] Shepard, S. (2005). *RFID: radio frequency identification*. McGraw-Hill Professional.

[Smets, 1994] Smets, P. (1994). The transferable belief model. *Artificial Intelligence*, 66(2):191–234.

[Smets, 2007] Smets, P. (2007). Analyzing the combination of conflicting belief functions. *Information Fusion*, 8(4):387–412.

[Strassner and Fleisch,] Strassner, M. and Fleisch, E. The promise of auto-id in the automotive industry. *Auto-ID Lab Working Paper. http://www. autoid. org/SC31/clr/200305_3826_Automotive% 20Prpsl. pdf.*

[Tran et al., 2009] Tran, T., Sutton, C., Cocci, R., Nie, Y., Diao, Y., and Shenoy, P. (2009). Probabilistic inference over RFID streams in mobile environments. In *IEEE International Conference on Data Engineering*, pages 1096–1107.

[Valera and Velastin, 2005] Valera, M. and Velastin, S. (2005). Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings -*, 152(2):192 – 204.

[Vezzani and Cucchiara, 2008] Vezzani, R. and Cucchiara, R. (2008). Ad-hoc: Appearance driven human tracking with occlusion handling. In *First International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS'2008), in conjunction with BMVC 2008.*

[Wadham, 2003] Wadham, R. (2003). Radio frequency identification. *Library Mosaics*, 14(5):22.

[Wang and Liu, 2005] Wang, F. and Liu, P. (2005). Temporal management of RFID data. In *Proceedings of the 31st international conference on Very large data bases*, pages 1128–1139.

[Ward et al., 2006] Ward, M., van Kranenburg, R., Platform, V., Backhouse, G., and TechWatch, J. (2006). Rfid: Frequency, standards, adoption and innovation. *JISC Technology and standards Watch*, 16.

[Welbourne et al., 2008] Welbourne, E., Khoussainova, N., Letchner, J., Li, Y., Balazinska, M., Borriello, G., and Suciu, D. (2008). Cascadia: a system for specifying, detecting, and managing rfid events. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 281–294. ACM.

[Wickramasuriya et al., 2004] Wickramasuriya, J., Datt, M., Mehrotra, S., and Venkatasubramanian, N. (2004). Privacy protecting data collection in media spaces. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 48–55, New York, NY, USA. ACM.

[Wu et al., 2006] Wu, E., Diao, Y., and Rizvi, S. (2006). High-performance complex event processing over streams. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 407–418. ACM.

[Xu and Li, 2008] Xu, H. and Li, G. (2008). Density-based probabilistic clustering of uncertain data. In *Proceedings of International Conference on Computer Science and Software Engineering*, pages 474–477. IEEE.

[Zhang et al., 2009] Zhang, C., Gao, M., and Zhou, A. (2009). Tracking high quality clusters over uncertain data streams. In *25th International Conference on Data Engineering, ICDE'09.*, pages 1641–1648. IEEE.

[Zhang and Poole, 1994] Zhang, N. and Poole, D. (1994). A simple approach to bayesian network computations.

[Zhang et al., 1996] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. In *ACM SIGMOD Record*, volume 25, pages 103–114. ACM.

[Zhang et al., 2005] Zhang, W., Cheung, S.-C. S., and Chen, M. (2005). Hiding privacy information in video surveillance system. pages 868–871.

[Zhou et al., 2007] Zhou, S., Ling, W., and Peng, Z. (2007). An rfid-based remote monitoring system for enterprise internal production management. *The International Journal of Advanced Manufacturing Technology*, 33(7):837–844.

[Zimmer and Unland, 1999] Zimmer, D. and Unland, R. (1999). On the semantics of complex events in active database management systems. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 392–399. IEEE.