Online Filtering and Uncertainty Management Techniques for RFID Data Processing

RAZIA HAIDER, FEDERICA MANDREOLI, RICCARDO MARTOGLIA
FIM - University of Modena and Reggio Emilia
Via Campi 213/b, 41125 Modena
ITALY
<name.surname>@unimo.it

Abstract: RFID is one of the emerging technologies for a wide-range of applications, including supply chain and asset management, healthcare and intruder localization. However, the nature of an RFID data stream is noisy, redundant and unreliable, making it unsuitable for direct use in applications. In this paper, we propose specific RFID Online Filtering and Uncertainty Management techniques that operate on unreliable and imprecise data streams in order to transform them into reliable probabilistic data that can be meaningful to the applications. Our proposal makes use of an Hidden Markov Model (HMM) that continuously infers hidden variables (locations, in case of above example) based on sensor readings. The resulting data can be directly stored in a probabilistic database table for further analysis. All the techniques presented in this paper are implemented in a complete framework and succesfully evaluated in real-world object tracking scenarios.

Key–Words: RFID data streams, Hidden Markov Model, Probabilistic Data Management, Object tracking

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 scenarios 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 technologies for a wide-range of applications, including supply chain and asset management [11, 30], healthcare [21], monitoring the location and status of patients in hospital environment [18], localizing intruders for alerting services [5] and so on. RFIDs offer a promising alternative to barcode identification systems. In an RFID system, an environment is deployed with the RFID readers and antennas while users and objects carry RFID tags. RFID readers detect the presence of tags in their vicinity and generate streams of low-level observations in the form of TREs (Tag Read Events): $(tag\_id, antenna\_id, time)$ that show 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 was in her office at 10:00".

Nevertheless, the management of RFID data in transforming low-level streams in to high-level events poses a number of challenges [4, 14]. 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 the following reasons: (a) *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 the technologies used in the construction of the devices; the incidence of this phenomenon is, however, high and not negligible: recent studies report that an RFID reader is usually able to detect only 60% -70% of tags that are in its vicinity [9, 15]; (b) *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 becomes unreliable for RFID applications and makes them not suitable to be directly used for

further analysis. To this end, in this paper we propose specific *RFID Online Filtering & Uncertainty Management* techniques that operate on unreliable and imprecise data streams in order to transform them into reliable probabilistic data that can be meaningful to the applications. A common way of dealing with such kind of imprecise data is to build a model of the data and use stream of raw readings as input to the model. Our proposal makes use of a temporal graphical model [19], a Hidden Markov Model (HMM) [27] that continuously infers hidden variables (locations, in case of above example) based on sensor readings. Such a relation, becomes a probabilistic relation `At(tagID,location,time,prob)` that can be directly stored, for instance, in a (probabilistic) database table and queried to detect complex events meaningful to applications [29]. 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.

All the techniques presented in this paper are implemented in a complete framework and evaluated under real-cases in the context of location tracking. However, they can be applicable in other contexts of RFID data management applications. The rest of the paper is organized in the following way: Section 2 describes some background notions about probabilistic graphical models, Section 3 describes the filtering and uncertainty management techniques we propose, Section 4 contextualizes the techniques in a complete RFID data acquisition and management framework, while in Section 5 we present extensive experiments in real object tracking scenarios, showing a very good reliability of the proposed techniques. Finally, Section 6 analyzes related works and gives some concluding remarks.

2 Background: Probabilistic Graphical Models

2.1 Representation

Graphical models [20] 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. Probabilistic graphical models [19] 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.

There are two types of probabilistic models: undirected and directed graphical models. DBNs are the example of directed graphical models of stochastic processes. They are used to compactly represent the stochastic evolution of a set of variables over time, where the graph structure captures the complex interdependencies between the variables of the process. DBNs generalize HMMs and linear dynamic systems (LDSs) [13] 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.

Since the solutions we present in this paper are based on HMMs, we will now focus on HMMs.

2.1.1 Hidden Markov Models

HMMs have one discrete hidden node variable and one discrete or continuous observed node variable per slice. HMMs are an often used model for time series data. They are used in various applications such as image recognition, pattern recognition, data compression and speech recognition. They represent probability distributions over sequences of observations.

Definition: An HMM is formally defined as a finite set of discrete states (it can be multidimensional), each of which is associated with a probability distribution. Since the states are discrete, transitions among 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 observations can be generated according to the associated probability distributions. Only the observed value, not the state, is visible to an external observer; hence, states are hidden.

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

- N, the number of states in model $S = \{S_1, S_2, \ldots, S_N\}$;
- M, the number of possible observations;
- The initial state distribution $\Pi = \{\Pi_i\}$ where $\Pi = P\{q_0 = S_i\}, 1 \leq i \leq N$;
- 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;
- The observation/emission probabilities $B = b_i(j)$:

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

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

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

- **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 to 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);
- **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 to adjust the model parameters in order to maximize $P(O|\lambda)$.

2.2 Inference

In various real-world data streams, the elements of interest may not be directly observable (e.g. location information in a raw data stream coming from RFID tracking application [29, 16, 31]), 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".

While "exact inference" algorithms can be effectively used in simple cases, such as linear dynamic systems (LDSs), most of them face severe challenges for large, densely connected models with high update rates. In order to handle the intractability in real-world scenarios "approximate inference" algorithms have been developed. In particular, recursive estimate techniques such as *Particle Filtering* [8] are memoryless inferencing techniques which are particularly effective in such contexts. They are Monte Carlo sampling based techniques implementing recursive Bayesian filters. 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.

2.3 Learning

A probabilistic graphical model is 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) [26] is one of the most widely used statistical techniques to learn the parameters of a CPD. From the training data, this provides an estimate 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.

In particular, the MLE method allows us 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 $P(O_t|S_t)$.

3 Online Filtering & Uncertainty Management

In this section, we will present in detail the techniques we exploit in order to provide filtering and uncertainty management to RFID data. The reference scenario will be location tracking. The techniques will then be put in context in Section 4, where they will be shown as being at the heart of a complete RFID data management framework.

3.1 Representation

The detailed block diagram of the involved process is shown in Figure 1. In the reference scenario, the interest of the application is to infer the positions of people and/or objects over time on the basis of the RFID readings collected by the reader. Positions are not being directly observable and are considered as the hidden variables, while the readings are our observable events, or simply our "observations". Thus, this process uses an HMM to produce, at each timestamp, a distribution over each tag location (i.e. the hidden variables or states) based on observations, i.e. sensor readings. These observations include four types of information: 1) the identifier of the tag the reading concerns to; 2) the identifier of the antenna(s) the tag is seen by; 3) the Received Signal Strength Indicator(s) (RSSIs) of the reading; 4) the timestamp of the reading. The employed model allows us to combine prior domain knowledge about the system behavior with the actual observations, so to compute the most

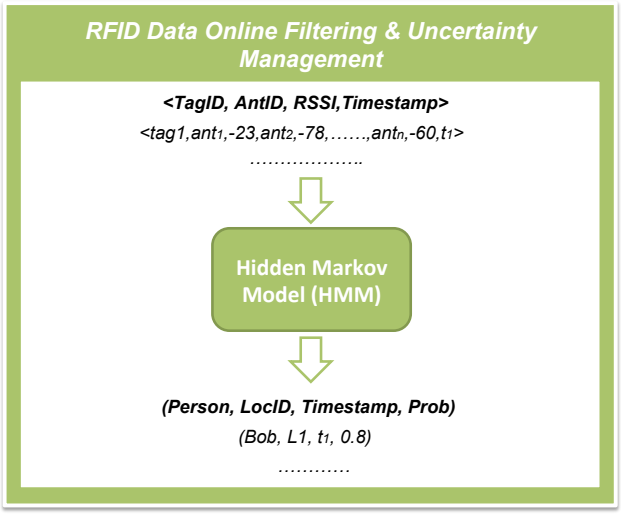


Figure 1: Block Diagram of the Online Filtering and Uncertainty Management process

likely values of the hidden variables. While observations are directly evaluable, the prior knowledge about the system is represented by Conditional Probability Distributions (CPDs) which are referenced as the parameters of the HMM.

A graphical representation of designed HMM is shown in Figure 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, square nodes in the graph represent the observations O and thus correspond to measurements collected by RFID antennas, while round nodes represent states and thus coincide with the location of people (for these reason, in the following we will denote each of these states as L). It is noted that, according to the well-known Markov principle, the model assumes 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 between the variables that are represented as directional arcs in Figure 2, are listed below:

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

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;

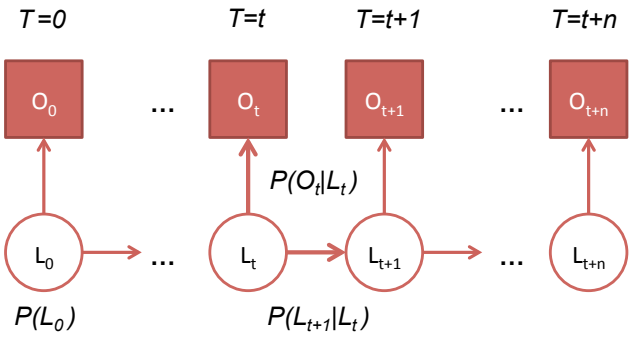


Figure 2: Graphical Representation of the Hidden Markov Model Used

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.

3.2 Learning

In order to maximize the effectiveness of the filtering and uncertainty management techniques in the consideed location tracking context, the above discussed CPDs are modeled as follows:

1. the *initial states probability* $P(L_0)$: it is assumed 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 will contain 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 a *Maximum Likelihood Estimation (MLE)* approach: given learning data, we estimate 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.

3.3 Inference

Our final aim of modeling a stochastic process with an HMM is to obtain the *posterior probability dis-*

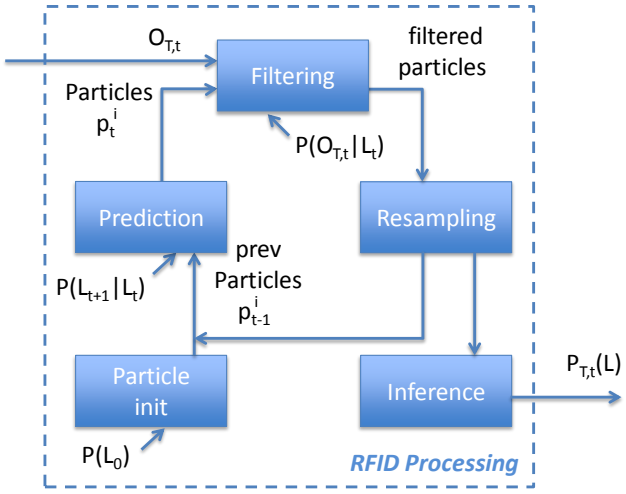


Figure 3: Detail of the steps performed for RFID Online Filtering & Uncertainty Management inference

tribution $P\left(L_t^{\tau_i}\right)$ over the hidden variable $L_t^{\tau_i}$ (location of tag τ_i at time instant t) given the observed measurements ("inference" task). Among the others, we decided to exploit the popular Monte Carlo algorithm called *Particle Filtering* [3], 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 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 $P\left(L_0\right)$;

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 $P\left(L_t|L_{t-1}\right)$;

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 $P\left(O_t|L_t\right)$. This weight is proportional to $P\left(O_t = o_t^{\tau_i}|L_t = \lambda\right)$ 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 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 just need to count the number of particles in each location and divide it by the total number.

4 Filtering Techniques in Context: the Complete RFID Data Management Framework

In this section, we will contextualize the filtering and uncertainty management techniques presented in this paper in a complete RFID Data Management Framework, the one we exploited in order to verify their effectiveness in a location tracking application.

- At the lowest part of the framework, RFID readers and tags, managed in a *Data Acquisition Layer* (see Section 4.1), provide raw RFID data;
- Raw data is the input to a *Data Filtering Layer*, at the heart of the framework, which implements the techniques discussed in the previous sections. The results of their application is the transformation of the raw data into a stream of "filtered" tuples, according to the schema $(Person, Location, Time, Probability)$;
- such filtered probabilistic tuples can then be managed, queried and stored in a standard probabilistic database, as will be briefly discuss in Section 4.2.

4.1 RFID Data Acquisition

At the lowest level of the framework is the *Data Acquisition Layer*, which is populated by RFID devices including RFID tags and readers. RFID tags are attached to the objects and people that have to be tracked, while RFID readers receive data from these tags in the form of radio signals and convert them in digital form to pass it to the upper levels of the framework. In the following, we will give some details on the hardware configuration that we exploited (and to which the results presented in Section 5 will refer).

Figures 4, 5 and 6 present the RFID reader, antennas and tags we employed in instantiating our framework. These are some specifications that can be useful in order to better understand how the proposed techniques work and perform:

- ***Reader***: we used a fixed reader that can interrogate tags at distances of up to 300 feet (100 me-

Figure 4: A fixed reader of our framework

Figure 5: An Elliptical Polarized Antenna of our framework

ters) (Figure 4). The reader establishes the connection to the host system by using the RS422 interface. For data exchange, a simple master/slave protocol is used by the reader. The protocol also gives us some additional information such as time of data reception, signal strength and number of times the tag has been read by the reader;

- ***Antennas***: the choice of antennas depended on the type and requirement of the application. An *Elliptical Polarized Antenna* (Figure 5) has a 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 at fast speeds. The orientation of the tags relative to the antenna is not important. On the other hand, a *Linear Polarized Antenna* is more suitable for applications in which read zones are restricted and data collection must be selective.

Figure 6: An active tag exploited in our framework

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 tag to have the same orientation. Elliptical antennas are the ones most suited to our purposes and have been used for final experimentation;

- ***Tags***: we employed active RFID tags based on UHF radio frequency (Figure 6). The tags are capable of providing long range for wireless applications and can transmit data at distances of up to 300 feet (100 meters) to readers. The tags continuously send static data written in their memory at pre-programmed intervals known as ping rate. Ping rate can be one second to four minutes (one second in our setup). Due to the ultra-low power consumption of the active tags, an operational lifetime of up to 6 years can be expected making them suitable for identification and tracking applications.

4.2 RFID Data Storage and Querying

.

Even if not at the focus of this paper, we will complete the description of the framework by providing a short description of how the tuples produced by the data filtering layer can be stored and queried in the context of a full RFID location tracking application. Since the output of our filtering and uncertainty management techniques are filtered probabilistic tuples, they can be directly and effectively stored in a probabilistic database management system (an example is MayBMS [1]). A probabilistic database stores data by means of special U-relational tables, providing a complete and concise representation of the large number of possible worlds that are generated in the presence of probabilistic tuples [2]. 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

```
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 7: Example of probabilistic queries

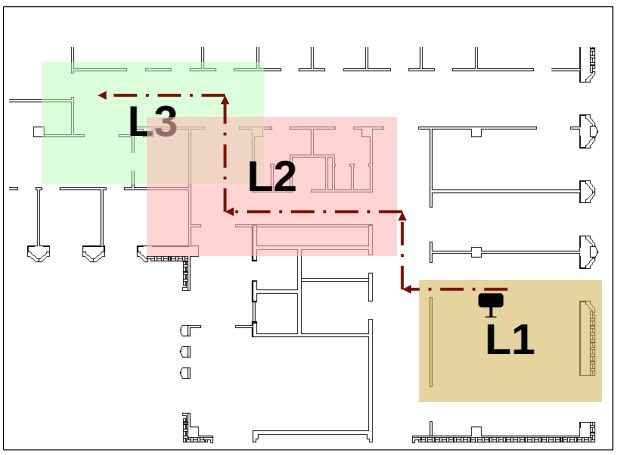


Figure 8: An Overview of the testbed used detailing the mapped locations

with uncertainty. Due to its compatibility with the relational algebra and standard SQL, a comprehensive set of constructs for data transformation can be easily exploited. Figure 7 shows some examples of possible temporal probabilistic queries that could be issued on the probabilistic data we generate.

5 Experimental Evaluation

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

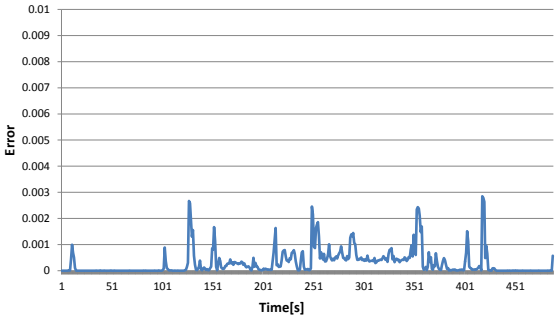
5.1 Experimental Setup

We performed experiments in different scenarios, collecting data from people wearing RFID tags. The experimental scenarios are set in three indoor locations (denoted by $L1$, $L2$, and $L3$) and capture different possible movement behaviors. Figure 8 shows the overview of the testbed, where locations are represented by bounded areas and the antenna by a black box. In this setup, we have collected data from RFID tags in two different scenarios: 1) "Stay", where people move between locations and spend some time on each of them; 2) "No Stay", where people rapidly move between locations without staying on any specific one; Both types of scenarios have been tested with one/multiple tags.

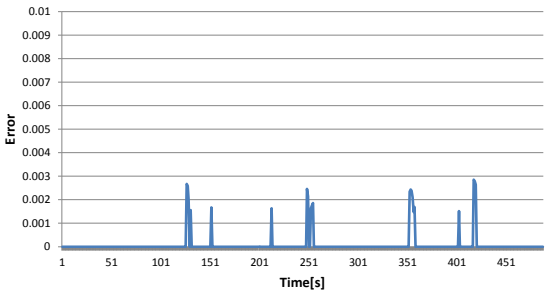
During the training phase, we used a single person as a probe to collect RSSI samples for each of three locations ($L1$, $L2$, and $L3$). Then, we performed MLE on them in order to map the locations and to learn the observation probability. During the testing phase, instead, we applied the proposed techniques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

5.2 Experimental Results

For each experiment, we evaluated the results on the basis of a location error criteria. In order to estimate the location error, we proceeded in two steps: first of all we computed a value we call the "estimated vs ground truth error": it is calculated at each time instant by means of an Euclidean distance between the ground truth and the estimated value. Then, we computed the actual location error, which is devised to ultimately quantify what we care about in a location tracking application: how long and how much the estimated value differs from the ground truth, when the actual estimated location is wrong. More specifically, it coincides with the estimated vs ground truth error but only for those time instants when a "wrong" location is reported; in the other instants, location error is 0. The values of location error (and estimated vs ground truth error) are between 0 and 1, where the lower the value the better the estimate. In the following, we will show for each experimental case a graph of the trend of the location error over the whole time span of the experiment; for completeness, we will also present the associated estimated vs ground truth error graphs. Moreover, we will complement this data with a single summarizing value, i.e. the average precision, computed as the percentage of time for which the esti-

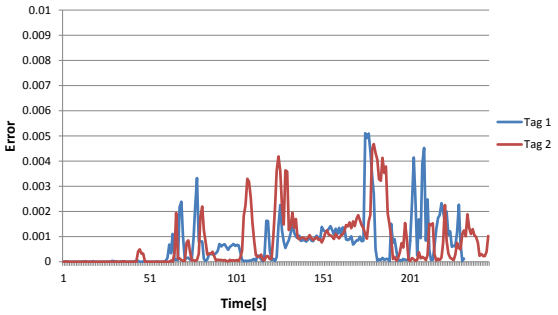


(a) Estimated vs Ground truth Error

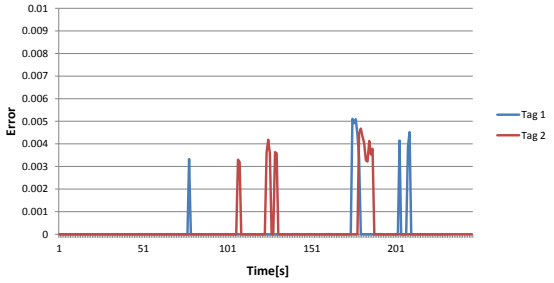


(b) Location Error

Figure 9: Case 1: Stay with 1 Tag



(a) Estimated vs Ground truth Error



(b) Location Error

Figure 10: Case 2: Stay with 2 Tags

mated answer reports the same location as the ground truth (the higher the value the better).

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

Case 1: Stay with 1 Tag: this case considers one person with an RFID tag moving between locations but staying for some time on each of the location. Figures 9 (a,b) show the achieved results for this case. The average precision is 96.95%, which is a very satisfying figure.

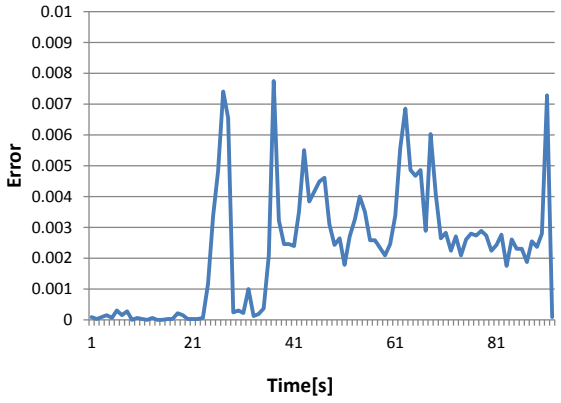
Case 2: Stay with 2 Tags: in this case, two people wearing RFID tags walk side by side and stay on each location. Figures 10 (a,b) show the results of experiments done in this case. Both persons were walking together and change their locations on the same time instants and, again, this behavior is shown by all graphs in overlapping results for both tags. The average precision is 95.39%.

Case 3: No Stay with 1 Tag: in this case, a person wearing an RFID tag that transmits every second rapidly moves between $L1$, $L2$ and $L3$ and does not stay at any of them. Please note that the movement scenario of this case (and case 4) could potentially be a difficult situation for capturing the exact locations, due to the fact that people move rapidly and do not stay on one particular point. Therefore, it could not be easy to produce stable RSSI values from the RFID antennas. Figures 11 (a,b) show the estimated vs ground truth error and location error for this case, respectively. As we can see from Figure 11 (b), our techniques reported a wrong estimated location at only one second. Similarly, if we consider estimated vs ground truth error, it is clear from Figure 11 (a) that the highest peak of error is nearly 0.008 which is a wrong location according to ground truth, while all other values are lower and correspond to correct locations. The resulting average precision for this case is 99%, again a very satisfying result.

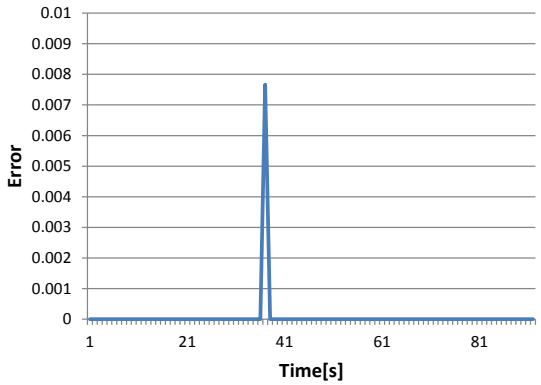
Case 4: No Stay with 2 Tags: this case considers the same movement scenario of case 3 but the number of involved people is two, holding RFID tags and walking side by side. Figure 12 (a,b) show the obtained results for this case. Both persons were walking side by side and changing their locations together and this behavior of movement is very clear from the resulting graphs. In this case, the average precision is 87.05%.

6 Related Works and Concluding Remarks

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, data management in these RFID applications poses a number of challenges [4]. Among the issues that need to be effectively faced in most RFID deployments,

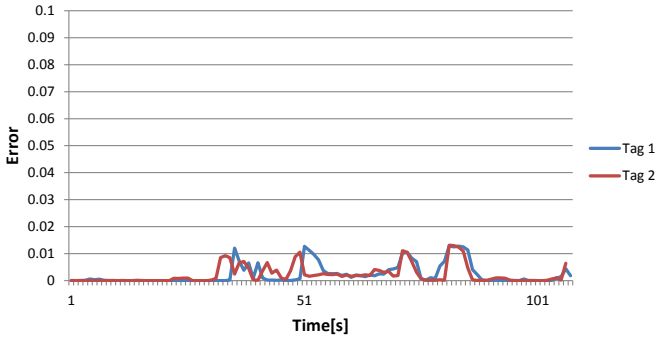


(a) Estimated vs Ground truth Error

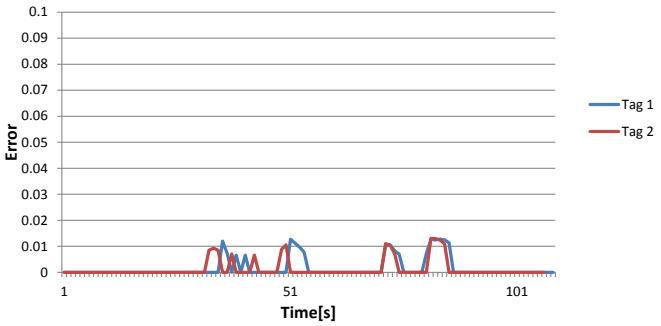


(b) Location Error

Figure 11: Case 3: No Stay with 1 Tag



(a) Estimated vs Ground truth Error



(b) Location Error

Figure 12: Case 4: No Stay with 2 Tags

avoiding missing/wrong readings and being able to extract high-level complex events from the huge volumes of low-level atomic events acquired by the sensors are particularly critical and challenging tasks.

Several techniques have been proposed for the analysis and processing of raw noisy RFID data [28]. A number of techniques propose to clean data streams deterministically. For instance, [10] 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.

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 [29, 31, 17]. For instance, [29, 16] 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 [29, 17]. 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").

In [6, 7], Deshpande et al. discuss 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, to identify outliers and to approximate answers to queries. Most of them mainly deal with inaccuracy errors present in raw RFID data, filtering and smoothing operations before feeding into higher level applications, thus not dealing (and not exploiting) the "meaning" of the managed information.

A number of probabilistic techniques have also 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 [17, 29, 31, 32].

For instance [16, 29] 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 the observed low level sensor data is translated into precise and more reliable estimates about the location of these objects by implementing an HMM [29, 31].

In this paper, we presented filtering and uncertainty management techniques for RFID probabilistic data management which are able to convey the RFID data to higher level data information modules, filtering inaccuracy errors and smoothing the raw RFID streams.

The proposed techniques, also in the light of the successful experimental evaluation we performed in real-world object tracking scenarios: (a) differently from most of the techniques available in the literature, work effectively without knowing in advance any specific information characterizing data uncertainty, such as the entire probability density function or standard error data available; (b) achieve the ultimate goal of transforming raw RFID data into reliable meaningful probabilistic data streams.

In the future, we will continue our work in making RFID data available to high level data management modules. In particular, by extending the techniques developed in complementary research fields, such as semantic data sharing and querying [12, 22, 23, 24, 25], toward RFID data, we will contemplate the feasibility of querying in a uniform way multiple RFID streams together with other kinds of heterogeneous data sources.

References:

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

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

[3] Doucet Arnaud, Nando de Freitas, and Gordon Neil. *Sequential Monte Carlo Methods in Practice*. Springer, 2005.

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

[5] R. Cucchiara, M. Fornaciari, R. Haider, F. Mandreoli, R. Martoglia, A. Prati, and S. Sassatelli. 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, 2011.

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

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

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

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

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

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

[12] F. Grandi, F. Mandreoli, R. Martoglia, E. Ronchetti, M. R. Scalas, and P. Tiberio. Ontology-based personalization of e-government services. In *Intelligent User Interfaces: Adaptation and Personalization Systems and Technologies, Constantinos Mourlas and Panagiotis Germanakos (Ed.), IGI Global*, pages 167–187. 2008.

[13] G:Welch and G.Bishop. An introduction to the kalman filter. 2002.

[14] Y. Hu, S. Sundara, T. Chorma, and J. Srinivasan. 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, 2005.

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

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

[17] N. Khoussainova, M. Balazinska, and D. Suciu. Probabilistic event extraction from RFID data. pages 1480–1482, 2008.

[18] D.S. Kim, J. Kim, S.H. Kim, and S.K. Yoo. 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, 2008.

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

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

[21] A. Anny Leema and M. Hemalatha. An effective and adaptive data cleaning technique for colossal rfid data sets in healthcare. *WSEAS Trans. Info. Sci. and App.*, 8(6):243–252, 2011.

[22] F. Mandreoli and R. Martoglia. Knowledge-based sense disambiguation (almost) for all structures. *Information Systems (Information)*, 36(2):406–430, 2011.

[23] F. Mandreoli, R. Martoglia, W. Penzo, and S. Sassatelli. Data-sharing p2p networks with semantic approximation capabilities. *IEEE Internet Computing (IEEE)*, 13(5):60–70, 2009.

[24] F. Mandreoli, R. Martoglia, W. Penzo, S. Sassatelli, and G. Villani. Sri@work: Efficient and effective routing strategies in a pdms. In *Proceedings of the 8th International Conference on Web Information Systems Engineering, December 2007 (WISE 2007)*, pages 285–297, 2007.

[25] F. Mandreoli, R. Martoglia, and E. Ronchetti. Versatile structural disambiguation for semantic-aware applications. In *Proceedings of the 14th ACM International Conference on Information Knowledge and Management, November 2005 (ACM CIKM 2005)*, pages 209–216, Bremen, Germany, 2005.

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

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

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

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

[30] Chien-Yuan Su and Jinsheng Roan. Investigating the impacts of rfid application on supply chain dynamics with chaos theory. *WSEAS Trans. Info. Sci. and App.*, 8(1):1–17, 2011.

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

[32] E. Welbourne, N. Khoussainova, J. Letchner, Y. Li, M. Balazinska, G. Borriello, and D. Suciu. 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, 2008.