Using integrity constraints to guide the interpretation of RFID-trajectory data

Bettina Fazzinga[1], Sergio Flesca[2], Filippo Furfaro[3], Francesco Parisi[4]
[1] ICAR, National Research Council of Italy (CNR), Italy
[2 3 4]DIMES, University of Calabria, Italy

Abstract

We discuss an approach for interpreting RFID data in the context of object tracking. It consists in translating the readings generated by RFID-tracked moving objects into semantic locations over a map, by exploiting some integrity constraints. Our approach performs a probabilistic conditioning: it starts from an a-priori probability assigned to the possible trajectories, discards the trajectories that are inconsistent with the constraints, and assigns to the others a suitable probability of being the actual one.

1 Introduction

RFID-based applications. In the last years, RFID technology has gained more and more attention as an effective tool for object tracking. In fact, monitoring of people, animals, and objects inside buildings, such as museums, schools, hospitals, office buildings, factories, farms, has become essential in several scenarios, with the aim of finding out trajectories of moving assets for behavior- and security- analyses. For instance, determining people trajectories can help prevent or look into crimes, and detect dangerous or suspicious situations. Similarly, knowing the trajectory followed by a visitor in a museum can help provide her with context-aware information, personalized on the basis of the artworks seen in previously visited rooms.

RFID technology relies on *tags* (which can emit radio signals encoding identifying information), and *readers* (which detect the signals emitted by tags). Thus, moving objects can be tracked by attaching RFID tags to them and properly placing RFID readers in the locations. Data collected by RFID-tracking systems in indoor spaces need to be properly managed to make them suitable for analysis purposes. In particular, data need to be cleaned to reduce their ambiguity.

Ambiguity of RFID data. The RFID data collected for an object o over a time interval $[0..\tau_f]$ form a sequence $\Theta = R_0, \ldots, R_{\tau_f}$, where each R_τ is called "*reading*" and is the (possibly empty) set of readers that detected o at τ. Analysis tools typically reason on an interpretation of these data: they require each reading R_τ to be translated into the location where o was at τ. Unfortunately, in general, there is no way to deterministically decide this translation. In fact, a one-to-one correspondence between locations and readers is infrequent (the same location may contain zones "covered" by different readers, and the same reader may detect objects at different locations), and things are made harder by false negative readings (an object close to a reader is not detected, owing to interferences or malfunctions). For instance, consider Figure $1(a, b)$. If o was detected at some τ by both $r1$ and $r5$, both $l1$ or $l4$ should be considered as possible positions of o at τ. Analogously, if o was detected only by $r3$, we could not conclude that it was in $l3$: it could be that some malfunction made $r2$ not detect o, thus a possible location is also $l2$ (which has a portion covered by both $r2$ and $r3$).

This means that any sequence Θ of readings can be interpreted in different ways. That is, there are different *trajectories* (i.e., sequences of locations) that may have been followed by o and that can have generated Θ, and

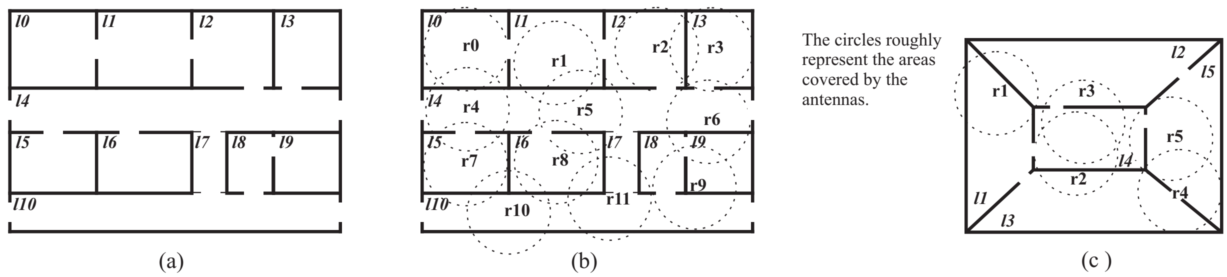

Figure 1: (a) A floor of a building; (b) Positions of the readers in the floor; (c) a map used in the examples

a crucial problem is determining how likely these trajectories are.

From RFID data to probabilistic trajectories. The above-introduced problem of interpreting a sequence Θ of readings by translating it into a trajectory can be naturally addressed in probabilistic terms. A naive approach is as follows. Preliminarily, a probabilistic model for interpreting a *single* reading is constructed. This model is a *probability distribution function* (pdf) $p^a(l|R)$ (where "*a*" stands for "*a-priori*", and l and R range over the locations and the sets of readers, respectively) encoding the probability that an object detected by the readers in R is in l. This $p^a(l|R)$ is called "*a-priori*" since it is defined without looking at the readings to be interpreted, but only on the basis of the readers' positions and physical parameters (such as the correlation between the reading rate and the distance from the reader's antenna). After being obtained, $p^a(l|R)$ is exploited to reason on each time point *independently* from the others: for each $\tau \in [0..\tau_f]$ and each location l, the probability that l is the actual position at τ is set equal to $p^a(l|R_\tau)$. In turn, any trajectory $t = l_0, \dots, l_{\tau_f}$ (representing the interpretation of Θ meaning that o was at location l_τ at each $\tau \in [0..\tau_f]$) is assigned $p^a(t|\Theta) = \Pi_{\tau=0}^{\tau_f} p^a(l_\tau|R_\tau)$ as the probability of being the actual trajectory of o.

Example 1: Consider Figure $1(a, b)$ and assume $\Theta = R_0, R_1, R_2$, with $R_0 = R_1 = \{r1, r5\}$ and $R_2 = \{r0\}$, meaning that, at $\tau = 0$ and $\tau = 1$, object o was detected by both $r1$ and $r5$, while, at $\tau = 2$, by $r0$. Assume also that $p^a(l|R)$ is such that: $p^a(l0|\{r0\}) = 1$ and $p^a(l1|\{r1, r5\}) = p^a(l4|\{r1, r5\}) = 0.5$. Hence, the trajectories compatible with Θ are: t_1: $l1, l1, l0$; t_2: $l1, l4, l0$; t_3: $l4, l1, l0$; t_4: $l4, l4, l0$, where $p^a(t_1|\Theta) = p^a(t_2|\Theta) = p^a(t_3|\Theta) = p^a(t_4|\Theta) = 0.25$ $(= 0.5 \cdot 0.5 \cdot 1)$.

Unfortunately, relying on the independence assumption and the probabilities returned by $p^a(t|\Theta)$ is not always correct: $p^a(t|\Theta)$ may assign non-zero probabilities to trajectories violating some integrity constraint implied by the domain, thus making unreasonable interpretations look reasonable.

Example 2: (cont. Example 1) Although t_1, t_2, t_3, t_4 are equi-probable according to $p^a(t|\Theta)$, the structure of the floor in Figure $1(a)$ implies that only t_1 is a correct interpretation, since $l0$ and $l4$ have no direct connection, and $l1$ is directly connected to $l0$ but not to $l4$ (we are also assuming that o's speed does not allow a room to be reached in one time point from a room not directly connected to it). Thus, a correct pdf over t_1, t_2, t_3, t_4 is: $\Pr(t_1|\Theta) = 1$, $\Pr(t_2|\Theta) = \Pr(t_3|\Theta) = \Pr(t_4|\Theta) = 0$.

The point is that while $p^a(l|R)$ and, in turn, $p^a(t|\Theta)$, are easy to obtain (as discussed above), it is very hard to find a formulation of a pdf over the alternative interpretations of the readings that takes into account the correlations between the possible positions over time.

The trajectory cleaning problem. In this work, we address this problem: given a sequence Θ of readings and the a-priori pdf $p^a(l|R)$ (and thus $p^a(t|\Theta)$), revise $p^a(t|\Theta)$ (which relies on the independence assumption) to properly take into account the known correlations between time points, thus assigning more "reasonable" probabilities to the trajectories. Intuitively, this can be seen as a cleaning problem: the data to be cleaned are the probabilistic trajectories representing the interpretations of Θ, and the cleaning task consists in revising the probabilities assigned by $p^a(t|\Theta)$.

Exploiting integrity constraints and probabilistic conditioning. The main idea underlying our approach, originally proposed in [11], is to address the above-defined cleaning problem exploiting:

$a)$ *specific forms of integrity constraints*: they will be used to find trajectories that, although pointwise compatible with Θ, are wrong interpretations (as they are inconsistent with the constraints);
$b)$ *probabilistic conditioning*: it will be used to revise the probabilities of the trajectories.

As regards $a)$, it is natural to assume that some knowledge of the domain, that can be naturally encoded in terms of integrity constraints, is available when the cleaning task starts. In fact, in several cleaning frameworks [3, 17, 18, 23, 24], the map of locations is assumed to be known, as well as the maximum speed of the objects being monitored. From this knowledge, constraints can be easily derived on the connectivity between pairs of locations (*direct unreachability* constraints) and/or on the time needed for reaching a location starting from another one (*traveling-time* constraints).

Example 3: (continuing examples 1, 2). The map implies a set of *direct unreachability* constraints, one per pair of rooms not directly connected by a door (such as $l0$, $l4$, and $l1$, $l4$). These constraints are those used in Example 2 to infer that t_1 is the only consistent interpretation of Θ. The map implies further constraints. For instance, it says that $l0$ and $l5$ are connected by a "long" path, namely 18*m*-long. If we know that the monitored tag is attached to a person whose maximum speed is 3*m/s*, then we have the constraint that 6 secs are required to walk this path (this will be called "*traveling-time* constraint"). This constraint implies that the interpretations corresponding to trajectories where $l5$ was reached from $l0$ in less than 6 secs should be discarded.

As regards $b)$, probabilistic conditioning is a rigorous approach commonly adopted in probabilistic databases to enforce constraints over probabilistic data [13, 19]). In our scenario, performing the conditioning means revising the probabilities assigned by the a-priori pdf $p^a(t|\Theta)$ (which does not take into account the constraints) by re-evaluating them as *conditioned* to the event that the constraints are satisfied. That is, given a set IC of constraints, $p^a(t|\Theta)$ is revised into $p^a(t|\Theta \wedge \text{IC})$: the probability of the invalid trajectories becomes 0, while that of each valid trajectory becomes the ratio of its a-priori probability to the overall a-priori probability of the valid trajectories. For instance, in the case of examples 1, 2, 3, each $p^a(t_i|\Theta)$ is revised into $p^a(t_i|\Theta \wedge \text{IC})$, where $p^a(t_2|\Theta \wedge \text{IC}) = p^a(t_3|\Theta \wedge \text{IC}) = p^a(t_4|\Theta \wedge \text{IC}) = 0$, while $p^a(t_1|\Theta \wedge \text{IC}) = \frac{0.25}{0.25} = 1$. In general, constraints reduce the number of valid trajectories, and the conditioning assigns "new" probabilities to them by keeping, for each pair of trajectories, the same probability ratios as between their a-priori probabilities, as shown in the following example.

Example 4: Let t_1, t_2, t_3, t_4 be trajectories with a-priori probabilities $p_1 = 0.5$, $p_2 = 0.25$, $p_3 = 0.2$, $p_4 = 0.05$, respectively. If t_3 and t_4 are inconsistent with the constraints, then they will be discarded, while t_1 and t_2 will be assigned the (conditioned) probabilities $\frac{0.5}{0.75} = \frac{2}{3}$ and $\frac{0.25}{0.75} = \frac{1}{3}$, respectively. This reflects the fact that, before conditioning, t_1 was twice as probable as t_2.

2 Cleaning through conditioning: the challenges.

The revision problem of evaluating $p^a(t|\Theta \wedge \text{IC})$ starting from $p^a(t|\Theta)$ is generally complex. The naive approach of enumerating the trajectories compatible with Θ, discarding those not satisfying the constraints, and revising the probabilities of the remaining ones, is often infeasible, as the trajectories to deal with are too many. For instance, if $\tau_f = 100$ and, for each time point, two locations are compatible with the readings, we have to consider 2^{100} ($\cong 10^{30}$) trajectories.

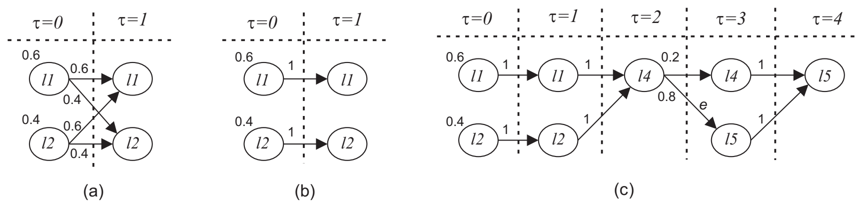

Figure 2: Graphs G' (a), G'' (b), and G (c)

Since separately representing the trajectories yields inefficiency, a promising direction for addressing our problem is devising a compact data structure to represent the interpretations and their a-priori probabilities. This data structure should be also prone to be revised to take into account the constraints and perform the conditioning. A naive way to do this is starting from a graph where:

$i.$ for each time point $\tau \in \mathcal{I}$, there is a node for each location l compatible with R_τ;

$ii.$ every node over $\tau = 0$ is labeled with $p^a(l|R_0)$, where l is the location of the node;

$iii.$ for each $\tau \in [0..\tau_f - 1]$ and each node n over τ, there is an edge from n to every node n' over $\tau + 1$, labeled with $p^a(l|R_{\tau+1})$.

It is easy to see that this graph represents all the interpretations of Θ along with their a-priori probabilities: every trajectory corresponds to a path from a node over $\tau = 0$ to a node over τ_f, and its a-priori probability is the product of the probabilities associated with the starting node and the edges of the path. This can be verified over the graph G' in Figure 2(a), that corresponds to the case that $\Theta = R_0, R_1$, where $R_0 = R_1 = \{r1\}$ and $p^a(l1|\{r1\}) = 0.6$ and $p^a(l2|\{r1\}) = 0.4$. Starting from this graph, the integrity constraints could be taken into account by performing edge removals making inconsistent trajectories no longer represented, and by conditioning the probabilities of the remaining edges. For instance, if the direct unreachability constraints implied by the map in Figure $1(c)$ are considered, the graph G' is revised into the graph G'' in Figure 2(b). It is easy to see that G'' represents the only consistent interpretations of Θ along with their conditioned probabilities, i.e., the trajectories $l1, l1$ and $l2, l2$ with their conditioned probabilities 0.6 and 0.4, respectively.

However, this naive approach does not work in the general case. In fact, assume that Θ is prolonged and becomes $\Theta = R_0, R_1, R_2, R_3, R_4$, where $R_2 = \{r2, r3\}$, $R_3 = \{r5\}$, $R_4 = \{r4, r5\}$, and that $p^a(l4|\{r2, r3\}) = 1$, $p^a(l4|\{r5\}) = 0.2$, $p^a(l5|\{r5\}) = 0.8$, $p^a(l5|\{r4, r5\}) = 1$. In order to take into account the new readings, G' would be extended into the graph G in Figure 2(c). Now, take this graph G as a starting point and try to consider also the traveling time constraint imposing that 3 time points are required to reach $l5$ from $l1$. As is, G also represents the trajectory $l1, l1, l4, l5, l5$, which is inconsistent with IC (as $l5$ cannot be reached in less than 3 time points from $l1$). The point is that there is no way to properly revise G by performing edge removals, as this would result in discarding also valid trajectories. For instance, if we remove the edge from $l4$ to $l5$, denoted as e in the figure, we discard also the consistent trajectory $l2, l2, l4, l5, l5$. As a matter of fact, in order to represent all and only the valid trajectories, location $l4$ cannot be encoded as a single node at time point 2, since the transitions allowed from this location depend on the locations visited before $l4$.

3 Our approach

Our approach cleans RFID data by exploiting *direct unreachability*, *traveling time* and *latency* constraints, where the last ones impose a duration for the stays in a certain locations (for instance, it is possible to specify the requirement that, if the monitored object goes in l_1, then it must stay there for at least two time points). Our approach returns a compact representation (*ct-graph*) of the valid trajectories and their conditioned probabilities. In the case of $\Theta = R_0, R_1, R_2, R_3, R_4$ discussed above, our approach builds a ct-graph whose shape is shown

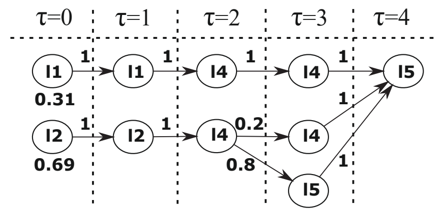

Figure 3: Graph encoding all and only the consistent trajectories

in Figure 3, where all and only the three consistent trajectories are represented. This compact representation is obtained by an iterative algorithm that builds a graph whose nodes correspond to pairs ⟨*location*, *timestamp*⟩ and where paths from source to target nodes one-to-one correspond to valid trajectories. This graph is built incrementally, aiming at: 1) creating more than one node for the same location l at the same time point if different transitions are allowed from l depending on the locations visited before; 2) preventing the creation of nodes and edges that would yield paths corresponding to invalid trajectories. The same algorithm assigns to each node/edge a probability obtained by suitably revising the a-priori probability of the corresponding pair ⟨*location*, *timestamp*⟩, so that the overall probability of a source-to-target path is the conditioned probability of the corresponding trajectory. For instance, the pdf $p^a(t|\Theta \wedge \textsc{ic})$ encoded by the ct-graph in Figure 3 assigns probability 0.31 to trajectory $l1, l1, l4, l4, l5$, 0.14 to $l2, l2, l4, l4, l5$, and 0.55 to $l2, l2, l4, l5, l5$.

3.1 The algorithm in detail

Given a sequence Θ of readings and a set of integrity constraints, our algorithm builds a ct-graph in two phases: a *forward* phase and a *backward* phase.

In the forward phase, Θ is scanned from the first to the last time point, and, for each time point, the possible interpretations of R_i are considered, on the basis of $p^a(l|R)$. For the first time point, all the locations l such that $p^a(l|R_0) \neq 0$ are considered, and for each of them a node (called *source* node) of the ct-graph is built. From the second time point on, for each node n built at the previous time point, a set of *successor* nodes is built as follows. A node n' over time point i is a successor of a node n over time point $i-1$, iff the location l specified in n' is such that $p^a(l|R_i) \neq 0$ and the trajectory represented by the locations contained in the path from the source node to n', passing through n, does not violate any constraint. Thus, successor nodes n' of n are added to the ct-graph (avoiding the addition of identical nodes for the sake of efficiency) along with edges $\langle n, n' \rangle$. The probabilities of the edges $\langle n, n' \rangle$ are set according to $p^a(l|R_i)$. Observe that it can happen that the sum of the probabilities of the outgoing edges of n is less than 1: this happens when there is l s.t. $p^a(l|R_i) \neq 0$ for which no successor of n can be built, meaning that there is no trajectory (consistent with the constraints) passing through n that can be prolonged with a stay of in l at time point i. As a special case, n may have no successor, so this sum is 0. Obviously, these cases must undergo revision of the probabilities of the outgoing edges, that will be performed in the backward phase. For the case of $\Theta = R_0, R_1, R_2, R_3, R_4$ discussed above, the ct-graph under construction at the end of the forward phase is shown in Figure 4. At both time points 2 and 3, the ct-graph contains two nodes over location l_4: this is due to the traveling time constraint from l_1 to l_5 that makes that the transition from l_4 to l_5 not possible at $\tau = 2$ for the trajectory having l_1 at time points 1 and 2, in order to avoid the encoding of inconsistent trajectories as explained at the end of Section 2. At time point 4, instead, since the traveling time constraint is expired, all the trajectories merge into the unique node over location l_5.

The backward phase performs two actions: 1) the revision of the probabilities, and 2) the removal of the non-target nodes with no successor. These tasks are deeply interwoven, since removing a node (and its ingoing edges) alters the sum of the outgoing edges of its predecessors, so that these nodes, in turn, will have to be revised or even removed. As regards 2), intuitively enough, removing a node n with no successor means removing a

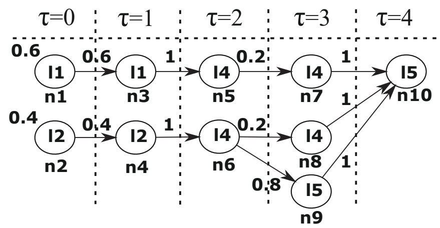

Figure 4: The ct-graph under construction at the end of the forward phase

useless node, since n belongs to no source-to-target path, thus it encodes no information on valid trajectories. As regards $1)$, in order to perform the revision, the algorithm performs a backward propagation of a quantity, called *loss*, summarizing the trajectories recognized as invalid. For the ct-graph of Figure 4, the probabilities of the edges $\langle n_1, n_3\rangle$, $\langle n_2, n_4\rangle$, and $\langle n_5, n_7\rangle$ need to be revised, since for each of the nodes n_1, n_2 and n_5, the sum of the probabilities of the outgoing edges is not 1. Specifically, all n_1, n_2 and n_5 suffered some "loss" during the forward phase: n_1 (resp., n_2) has lost 0.4 (resp., 0.6) as the transition from l_1 to l_2 (resp., from l_2 to l_1) is not possible, while n_5 has lost 0.8 since l_5 is not a valid location at time point 3, for the trajectory having l_1 at time point 1. These losses are propagated backward during the backward phase, up to the source nodes, and have as effect the normalization and the redistribution of the probabilities, leading to obtain the ct-graph depicted in Figure 3.

4 Related Work

The management of RFID data has been studied from different perspectives. The definition of models for suitably representing RFID data has been addressed in [5, 20], while the problem of defining efficient warehousing models and of summarizing and indexing RFID data has been investigated in [15, 14], that can be seen as lossless compression mechanisms. Lossy compression techniques for RFID-data are instead proposed in [6, 12, 8], where compression can be also seen as a form of cleaning.

One of the first cleaning approaches for RFID trajectory data in indoor spaces is [18], where the position of the tracked object during an interval I of no detection is decided as the set of locations that are directly connected with both the positions of the object before and after I. However, differently from our approach, that technique does not work in the case of overlapping readers. The scenario of non-overlapping readers has been also addressed in [2], where a distance-aware deployment graph (which encodes the topology of the map, the assumed speed of the object, the position and some physical parameters of the readers) is used to fill missing detections. However, in [2], the cleaning of missing detections is addressed at the level of raw RFID readings, rather than that of "semantic" locations. This means that the result of their cleaning task on an r-sequence Θ is another r-sequence (and not an l-sequence), where the empty readings of Θ are replaced with sets of readers that should have detected the object. Also that technique reasons at the level of raw RFID readings, but in the absence of missing detections, and its cleaning task consists in deciding, for each $R_\tau \in \Theta$ consisting of multiple detections, which subset of R_τ better represents the actual position.

Besides the above-mentioned [18], other cleaning approaches working at our abstraction level of locations are those based on *particle filtering* [7, 1], such as [24] (that was used in our experiments in [11] as the core of some terms of comparison), [23] and [17]. The first two works mainly differ in the motion model and in the positions that are considered as possible at each time point ([24] allows free movements, while [23] assumes that the positions are laid onto a Voronoi graph over the map of locations). The main contribution of [17] is instead a unified model of outdoor and indoor spaces, where the need of incorporating cleaned RFID data arises to support the analysis of potential points of traffic overload.

The above mentioned techniques based on particle filtering were devised to tackle the online tracking problem, thus they do not exploit the correlations between the current time point and the future ones. The idea of exploiting the correlations with both the past and the future time points is instead targeted at the offline cleaning problem, that we have recently addressed in [10, 9]. In particular, in [10], we presented a smoothing technique following a two-way-filtering scheme that, differently from the approach proposed in this paper, does not assign probabilities to cleaned trajectories, but associates, for each time point, each candidate location with a probability, that is uncorrelated with the future and past time points. In [9], we first proposed the use of probabilistic conditioning for supporting trajectory cleaning. In [11], we built on that work by: $1)$ elaborating the formal proof of the correctness of the cleaning framework, $2)$ devising the look-ahead mechanism (that, once embedded into the ct-graph construction, has been experimentally shown to yield a significant performance improvement), $3)$ extending the framework to deal with the online tracking problem, and $4)$ adapting to our scenario several paradigms for Bayesian inference used in the literature (i.e., Metropolis Hastings, Particle Filtering, Hidden Markov Models) and using them in a comparative experimental validation of the framework.

5 Conclusions

We discussed a probabilistic cleaning framework for RFID data, where the set of trajectories that are possible interpretations of a given sequence of readings is cleaned by exploiting the knowledge of integrity constraints to guide the probabilistic conditioning paradigm.

Although our technique has been motivated and explained by considering indoor RFID-tracking systems as the main application scenario, it is worth noting that its applicability is not limited to the contexts where RFID technology is used. In fact, our approach can be used whenever it is possible to determine an a-priori pdf p^a describing the position at each time point: p^a can be conditioned by our technique w.r.t. our forms of constraints independently from the way p^a was obtained. For instance, in the case of the tracking frameworks exploiting RF signals (such as WiFi [4], iBeacon [22], etc. [16, 21]), the same “fingerprinting” procedure as that performed in our case for obtaining $p^a(l|R)$ can be used to associate each position with a probability of establishing a connection with one or more WiFi access points, or receiving the signal from one or more iBeacons.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, 2002.

[2] A. I. Baba, H. Lu, T. B. Pedersen, and X. Xie. Handling false negatives in indoor RFID data. In *Proc. Int. Conf. on Mobile Data Management (MDM)*, pages 117–126, 2014.

[3] A. I. Baba, H. Lu, X. Xie, and T. B. Pedersen. Spatiotemporal data cleansing for indoor rfid tracking data. In *Proc. Int. Conf. on Mobile Data Management (MDM)*, pages 187–196, 2013.

[4] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proc. Conf. IEEE Computer and Communications Societies (INFOCOM)*, pages 775–784, 2000.

[5] Y. Bai, F. Wang, P. Liu, C. Zaniolo, and S. Liu. RFID data processing with a data stream query language. In *Proc. Int. Conf. on Data Engineering (ICDE)*, pages 1184–1193, 2007.

[6] D. Bleco and Y. Kotidis. RFID data aggregation. In *Proc. Int. Conf. on GeoSensor Networks (GSN)*, pages 87–101, 2009.

[7] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001.

[8] B. Fazzinga, S. Flesca, F. Furfaro, and E. Masciari. Rfid-data compression for supporting aggregate queries. *ACM Trans. Database Syst. (TODS)*, 38(2):11, 2013.

[9] B. Fazzinga, S. Flesca, F. Furfaro, and F. Parisi. Cleaning trajectory data of rfid-monitored objects through conditioning under integrity constraints. In *Proc. Int. Conf. on Extending Database Technology (EDBT)*, pages 379–390, 2014.

[10] B. Fazzinga, S. Flesca, F. Furfaro, and F. Parisi. Offline cleaning of RFID trajectory data. In *Proc. Int. Conf. on Statistical and Scientific Database Management (SSDBM)*, page 5, 2014.

[11] B. Fazzinga, S. Flesca, F. Furfaro, and F. Parisi. Exploiting integrity constraints for cleaning trajectories of rfid-monitored objects. *ACM Trans. Database Syst.*, 41(4):24:1–24:52, 2016.

[12] B. Fazzinga, S. Flesca, E. Masciari, and F. Furfaro. Efficient and effective RFID data warehousing. In *International Database Engineering and Applications Symposium (IDEAS)*, pages 251–258, 2009.

[13] S. Flesca, F. Furfaro, and F. Parisi. Consistency checking and querying in probabilistic databases under integrity constraints. *J. Comput. Syst. Sci. (JCSS)*, 80(7):1448–1489, 2014.

[14] H. Gonzalez, J. Han, H. Cheng, X. Li, D. Klabjan, and T. Wu. Modeling massive rfid data sets: A gateway-based movement graph approach. *IEEE Trans. Knowl. Data Eng. (TKDE)*, 22(1):90–104, 2010.

[15] H. Gonzalez, J. Han, X. Li, and D. Klabjan. Warehousing and Analyzing Massive RFID Data Sets. In *Proc. Int. Conf. on Data Engineering (ICDE)*, pages 83–88, 2006.

[16] Y. Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *Commun. Surveys Tuts.*, 11:13–32, 2009.

[17] S. H. Hussein, H. Lu, and T. B. Pedersen. Reasoning about rfid-tracked moving objects in symbolic indoor spaces. In *Proc. Int. Conf. on Statistical and Scientific Database Management (SSDBM)*, page 9, 2013.

[18] C. S. Jensen, H. Lu, and B. Yang. Graph model based indoor tracking. In *Proc. Int. Conf. on Mobile Data Management (MDM)*, pages 122–131, 2009.

[19] C. Koch and D. Olteanu. Conditioning probabilistic databases. *PVLDB*, 1(1):313–325, 2008.

[20] C. H. Lee and C. W. Chung. Efficient storage scheme and query processing for supply chain management using rfid. In *Proc. Int. Conf. on Management of Data (SIGMOD)*, pages 291–302, 2008.

[21] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *Trans. Sys. Man Cyber Part C*, 37(6):1067–1080, 2007.

[22] D. Stojanovic and N. Stojanovic. Indoor localization and tracking: methods, technologies and research challenges. *Facta Universitatis, Series: Automatic Control and Robotics*, 13(1), 2014.

[23] J. Yu, W. S. Ku, M. T. Sun, and H. Lu. An RFID and particle filter-based indoor spatial query evaluation system. In *Proc. Int. Conf. on Extending Database Technology (EDBT)*, pages 263–274, 2013.

[24] Z. Zhao and W. Ng. A model-based approach for RFID data stream cleansing. In *Proc. Int. Conf. Information and Knowledge Management (CIKM)*, pages 862–871, 2012.