

ABSTRACT

In recent years, indoor spatial data management has started to attract attention, partly due to the increasing use of receptor devices (e.g., RFID readers, and wireless sensor networks) in indoor, as well as outdoor spaces. There is thus a great need for a model that captures such spaces, their receptors, and provides powerful reasoning techniques on top. This paper reviews and extends a recent unified model of outdoor and indoor spaces and receptor deployments in these spaces. The extended model enables modelers to capture various information pieces from the physical world. On top of the extended model, this paper proposes and formalizes the route observability concept, and demonstrates its usefulness in enhancing the reading environment. The extended model also enables incorporating receptor data through a probabilistic trajectory-to-route translator. This translator first facilitates the tracking of moving objects enabling the search for them to be optimized, and second supports high-level reasoning about points of potential traffic (over)load, so-called bottleneck points. The functional analysis illustrates the behavior of the route observability function. The experimental evaluation shows the accuracy of the translator, and the quality of the inference and reasoning. The experiments are conducted on both synthetic data and uncleansed, real-world data obtained from RFID-tagged flight baggage.

Categories and Subject Descriptors

H.1.1 [Systems and Information Theory]: Information theory; H.2.8 [Database Applications]: Spatial databases and GIS, Statistical databases; I.2.3 [Deduction and Theorem Proving]: Inference Engine, Uncertainty, fuzzy, and probabilistic reasoning

General Terms

Algorithms, Design, Experimentation, Theory

Keywords

Indoor space, modeling, reasoning, RFID, moving objects tracking

1. INTRODUCTION

Ubiquitous receptor devices are increasingly deployed in both outdoor and indoor spaces (OI-spaces) to enable new classes of applications that enhance human awareness about the physical world. A myriad of examples exist, including supply chain and product life cycle management, and asset and personnel tracking. In order to support these emerging applications, so-called receptor-based systems are being built with a focus on managing and analyzing the data collected by receptors. A common assumption made in spatial data management systems is that the spaces under consideration are outdoor spaces (O-spaces). However, most of our time is spent indoors – thus creating a need to support complex indoor spaces (I-spaces), and, in turn, combined OI-spaces. A variety of applications, facilitated by receptor-based systems, need to seamlessly span both O- and I-spaces. One application is tracking, i.e., determining the location of moving objects in OI-spaces. Another application is deciding the parts of OI-spaces that are covered by receptors. A third application is determining the locations of heavy traffic in OI-spaces. Supporting these applications and others (at various levels in OI-spaces) motivates this paper which makes the following contributions:

- The paper reviews and extends a recent unified model of OI-spaces and receptor deployments in these spaces.
- Based on the extension, the study investigates the route observability concept, with the aim of enhancing the reading environment.
- The paper then proposes a probabilistic translator of receptor data that enhances and complements the knowledge about the locations of RFID-tracked moving objects in OI-spaces.
- Furthermore, the paper uses the translated data in order to perform high-level reasoning about points of potential traffic load in OI-spaces, so-called bottleneck points (BPs).
- Last, the paper evaluates the proposals via functional analysis and experimentation with a real-world RFID dataset.

The remainder of this paper is organized as follows. Section reviews the authors’ recent unified model of OI-spaces and receptor deployments in these spaces. Section extends this model with various properties from the physical OI-space environment. Using the coverage weight property introduced in Section and building on solid information-theoretic foundations, the paper proposes the...
route observability concept in Section 2 derives its bounded function from the ground up, and establishes its lower and upper bound. The notion of a BP is proposed in Section 2 alongside static reasoning about this notion. Probabilistic incorporation of RFID data is carried out in Section 3 using the probabilistic trajectory-to-route translator, which is based on the extended model. This translator constitutes the foundation for an over-time upgrade, performed in Section 4 of the static reasoning done in Section 2. Section 5 follows with a comprehensive functional analysis and experimental evaluation that analyze the route observability function, the probabilistic trajectory-to-route translator, and the dynamic BP estimate algorithms under a variety of settings. Related work is reviewed in Section 6 and the paper concludes in Section 7. The proofs are given in Appendix A and Table 1 offers a summary of the notation used throughout the paper.

Table 1: Summary of notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>l, l_i, l_j</td>
<td>A semantic location (a location for short) and a connection point</td>
</tr>
<tr>
<td>(l_i, l_j), r</td>
<td>A binary sub-route (a sub-route for short) and an RFID reader</td>
</tr>
<tr>
<td>W_l, W_l_r</td>
<td>The sets of locations and connection points</td>
</tr>
<tr>
<td>W_m, W_m_r</td>
<td>The sets of moving objects and sub-routes</td>
</tr>
<tr>
<td>D_o-space, D_f</td>
<td>The OI-space and RFID deployment pseudographs</td>
</tr>
<tr>
<td>c</td>
<td>The edge label mapping in D_o-space</td>
</tr>
<tr>
<td>c_l, c_m, c_r</td>
<td>The vertex label mapping, the edge label mapping, and the coverage weight mapping in D_f</td>
</tr>
<tr>
<td>R = (l_1, l_k), obs(R)</td>
<td>A route in D_f and its observability</td>
</tr>
<tr>
<td>obj</td>
<td>A moving object</td>
</tr>
<tr>
<td>TR(obj, T)</td>
<td>The trajectory of obj over T</td>
</tr>
<tr>
<td>appear-ds</td>
<td>The data structure of appearance records</td>
</tr>
<tr>
<td>inter-ds</td>
<td>The data structure of intermediate records</td>
</tr>
<tr>
<td>prob-ds</td>
<td>The data structure of probabilistic records</td>
</tr>
<tr>
<td>infer-ds</td>
<td>The data structure of inferred records</td>
</tr>
<tr>
<td>synth-ds</td>
<td>The data structure of synthetic records</td>
</tr>
<tr>
<td>E_{GT}(l)</td>
<td>The static estimate that l is a BP</td>
</tr>
<tr>
<td>E_{GT}(l)</td>
<td>The dynamic estimate that l is a BP over T</td>
</tr>
<tr>
<td>BP/MIQ</td>
<td>A BP monitoring query over T</td>
</tr>
</tbody>
</table>

2. MODEL REVIEW

The authors’ recent work [8] proposes a unified model of OI-spaces and receptor deployments in these spaces. The work focuses on the partially constrained outdoor and indoor motion common in receptor-based systems. The model is shown to be expressive, flexible, and invariant to the segmentation of the space plan and the receptor deployment policy. The viability of this model is demonstrated via applying it to the real-world baggage handling plan in Aalborg Airport. This plan comprises two sub-plans; the I-space and O-space plans of Aalborg Airport hall and apron respectively. The I-space plan is shown in Figure 1.

To create the OI-space pseudograph model of the baggage handling example, the sets of locations (W_l), connection points (W_l_r), moving objects (W_m), and sub-routes (W_m_r) are identified. To give examples in Figure 1 the check-in desks (CD) and check-in conveyor (CC) are two locations and (CD/CC) is their connection point. The moving objects are bags with attached RFID tags. An example bag route is CD → CC → MC → SMC → TTS (repeatedly in general) → CH. Two sub-routes along this route are (CD/CC) and (TTS, TTS). Next, the locations are converted into vertices and sub-routes into edges (an edge direction matches the motion direction and the order of the sub-route). Furthermore, the edges are labeled using sets taken from the power set of the connection points. For instance in Figure 1 the locations CD and CC are converted into vertices, and the sub-route (CD/CC) is converted into an edge connecting these two vertices. The edge (CD, CC) is directed from CD to CC and labeled (CD/CC). The same identification, conversion, and labeling steps are carried out for Aalborg Airport apron which yields D_o-space = (W_l, W_m, c) shown in Figure 2 where c is the edge label mapping. In this figure, CGS1-GS4 are the apron geometric segments, BL1-BL3 are the belt loaders, and AP1-AP3 are the airplanes.

In the same work [8] Aalborg Airport RFID deployment (Figure 1) is modeled. An algorithm is applied in order to transform D_o-space = (W_l, W_m, c) (Figure 2) into D_f = (W_l, W_m, c_l, c_m) (Figure 3). For instance, c_l and c_m are the vertex and edge label mappings respectively. To give examples on vertex and edge labeling in Figure 3 the reader r_1 is positioned inside MC away from any connection point. Therefore r_1 ∈ c_l(MC). On the other hand, r_2 and r_3 are adjacent to positioned at SMC/TTS, and r_2 reads before r_3 when moving from SMC to TTS across SMC/TTS. Thus, (r_2, r_3) ∈ c_m(SMC, TTS).

3. MODEL EXTENSION

The graph D_f is now extended into a property pseudograph [21] by allowing the vertices and edges to have various properties (key/value pairs) from the physical hall and apron environments. The advantage of this extension is threefold. First, a property pseudograph gives the modelers freedom to express their
awareness of various information pieces from the physical world. Intuitively, the more the information gathered about the physical world, the wider the scope of the questions that can be asked about it. Second, a property pseudograph is a malleable structure that can be easily transformed into other common graph structures. Third, a property pseudograph is the typical data model used in graph databases. Therefore, the extension to this type of graph enables benefiting from the proven efficiency of graph databases in processing dense and interrelated datasets and quickly traversing along the edges between vertices [21]. The extended model and reasoning techniques proposed in this paper have been implemented in the UniModeling tool. This tool stores D_{rfid} in an OrientDB graph database (http://www.orientdb.org/). It then issues graph queries in order to complete the probabilistic incorporation of RFID data (Section 4). UniModeling is demonstrated in a demo paper [9].

Various properties can be obtained from the hall (Figure 1) and apron environments and subsequently added to the vertices and edges of D_{rfid} (Figure 3). Two example properties are conveyor throughput (measured in bags/hour), and speed limit on the apron geometric segments (measured in m/s). An important property for the route observability concept (Section 4) is the coverage weight of RFID readers among locations. This vertex property can be captured in a mapping c_r : \mathcal{W}_l \rightarrow (\forall r \rightarrow w(r) = \frac{ZONE(r) \cap AREA(l)}{ZONE(r)} : ZONE(r) \cap AREA(l) \neq \emptyset)

It is also convenient to use the simplifying notation:

\[ c_r(l) = \sum_{w(r) \in c_r(l)} w(r) \]  

(1)

The coverage weights of the locations seen in Figure 3 are approximated based on the RFID deployment (Figure 1) and listed in Table 2. For instance, c_r(SMC) = \{r_2 \rightarrow .8, r_3 \rightarrow .2\} means that roughly 80% of r_2 and 20% of r_3 reading zones overlap with AREA(SMC). The notation of the RFID pseudograph becomes D_{rfid} = (\forall r \in \mathcal{W}_l, c_r, c_m, c_e), augmented by c_r, the new coverage weight mapping.

<table>
<thead>
<tr>
<th>l</th>
<th>c_r(l) = (r \rightarrow w(r))</th>
<th>l</th>
<th>c_r(l) = (r \rightarrow w(r))</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD, CC, GS4, AP1</td>
<td>\emptyset</td>
<td>GS1</td>
<td>{r_6 \rightarrow 1}</td>
</tr>
<tr>
<td>OB, RB, AP2, AP3</td>
<td>\emptyset</td>
<td>GS2</td>
<td>{r_7 \rightarrow 1}</td>
</tr>
<tr>
<td>MC</td>
<td>{r_1 \rightarrow 1}</td>
<td>GS3</td>
<td>{r_8 \rightarrow 1}</td>
</tr>
<tr>
<td>SMC</td>
<td>{r_2 \rightarrow .8, r_3 \rightarrow .2}</td>
<td>BL1</td>
<td>{r_6 \rightarrow 9}</td>
</tr>
<tr>
<td>TTS</td>
<td>{r_2 \rightarrow 2, r_3 \rightarrow 8}</td>
<td>BL2</td>
<td>{r_7 \rightarrow 9}</td>
</tr>
<tr>
<td>OC</td>
<td>{r_4 \rightarrow .95, r_5 \rightarrow .95}</td>
<td>BL3</td>
<td>{r_8 \rightarrow 8}</td>
</tr>
<tr>
<td>CH</td>
<td>{r_4 \rightarrow .05, r_5 \rightarrow .05}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGS</td>
<td>{r_4 \rightarrow .05, r_5 \rightarrow .05}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4. ROUTE OBSERVABILITY

The route observability is a measure of the extent to which a given route is covered by RFID readers. The study of route observability is motivated as follows. Some physical approaches to RFID deployment in OI-spaces attempt to correct RFID anomalies by enhancing the reading environment. This can be attained through either installing additional readers or (more economically) adjusting the positioning of already-installed readers. In both cases, the aim is to cover a more substantial amount of the OI-space \[I\].

The number of RFID readers positioned along a route does not affect the observability of RFID deployment in OI-spaces attempt to correct RFID anomalies, because the reading environment can be enhanced. This can be attained by adjusting the positioning of already-installed readers. In both cases, the aim is to cover a more substantial amount of the OI-space \[I\].

A precise route observability measure is thus needed.

The route definition can be made more formal by saying that a route \( R = (l_1 \ldots l_k) \) in \( D_{rfid} \) is an alternating sequence of locations and sub-routes from \( D_{rfid} \) that starts in \( l_1 \) and ends in \( l_k \). The sets of locations and sub-routes in \( R \) are denoted as \( \forall R \) and \( \forall A(R) \) respectively. Alternatively, one may write \( R = (\forall R, \forall A(R)) \).

The starting point in deriving an observability measure is to consider the observability of a single location and then generalize it into routes with an arbitrary number of locations. The observability of a location \( l \) can be denoted as follows:

\[ \Gamma : [0, 1] \rightarrow \mathbb{R}_{\geq 0} \]

To measure the observability in a meaningful way, the function \( \Gamma \) should satisfy the following properties:

P1. Nonnegativity: \( \forall w(r) \in c_r(l) : \Gamma(w(r)) \geq 0 \).

P2. Increasing monotonicity: \( \forall w(r_1), w(r_2) \in c_r(l) : w(r_1) \leq w(r_2) \implies \Gamma(w(r_1)) \leq \Gamma(w(r_2)) \).

P3. Normalization: If the whole coverage of a reader \( r \) is contained within the location \( l \), then the observability should be 1, that is \( \forall w(r) \in c_r(l) : w(r) = 1 \implies \Gamma(w(r)) = 1 \).

Intuitively, a location observability should be expressed by an increasing function of the coverage weights: the higher these weights, the higher the observability. This justifies P2, and makes P1 convenient to have. The property P3 is a requirement for the measurement unit and it can be modified. One class of functions that satisfy
P1-P3 is defined for each \( w(r) \in [0, 1] \) by the formula:

\[
\Gamma(w(r)) = a \log_b (w(r) + 1)
\]

where \( a \) is an arbitrary constant and \( b \) is a nonnegative constant different from 1. Adding 1 to \( w(r) \) satisfies P1. Since the logarithmic function is increasing, satisfying P2 entails a nonnegative \( a \). P3 can be formally expressed by the equation:

\[
a \log_b(1 + 1) = 1
\]

This equation can be conveniently satisfied by choosing \( a = 1 \) and \( b = 2 \), making bits the measurement unit of location observability. The function becomes: (Here and hereafter, all logarithms are to the base 2).

\[
\Gamma(w(r)) = \log(w(r) + 1)
\]

Due to the choice of \( a \) and \( b \), the co-domain of \( \Gamma \) become \([0, 1]\). One more desirable property is the finite additivity which can be expressed as follows:

P4. Finite additivity: For every finite sequence of pairwise disjoint routes, the observability of a union of these routes equals the sum of the individual observabilities.

This property enables measuring the observability of routes with an arbitrary number of locations. It also enables the concatenation of routes. To satisfy P4, one sums for all \( w(r) \in c_v(l) \) and then for all \( l \in \mathcal{V}(R) \), which yields the route observability function:

\[
obs(R) = \sum_{l \in \mathcal{V}(R)} \sum_{w(r) \in c_v(l)} \log(w(r) + 1) \tag{2}
\]

The obs function has solid bounds that are explored in Theorem 1. These bounds are important in that they delimit the optimization that can be introduced into an RFID readers deployment.

**Theorem 1.** Given an RFID readers deployment pseudograph \( D_{\text{rfid}} = (\mathcal{W}_l, \mathcal{W}_m, c_v, c_m, c_r) \), the observability of any route \( R \) in \( D_{\text{rfid}} \) has the bounds:

\[
0 \leq obs(R) \leq \sum_{l \in \mathcal{V}(R)} \log \left( c_v(l) + |c_r(l)| \right)
\]

Revisiting \( D_{\text{rfid}} \) in Figure 3 and the coverage weights in Table 2, some baggage routes, their observabilities, and bounds are listed in Table 3.

Table 3: Some baggage routes in Figure 3, their observabilities, and bounds

<table>
<thead>
<tr>
<th>Route</th>
<th>(CD) · · · AP1</th>
<th>(RB) · · · AP2</th>
<th>(OB) · · · AP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs(R)</td>
<td>6.3533</td>
<td>6.3533</td>
<td>3.1312</td>
</tr>
<tr>
<td>bounds(R)</td>
<td>[0.8, 2673]</td>
<td>[0.8, 2673]</td>
<td>[0.4, 0974]</td>
</tr>
</tbody>
</table>

Notice in this table that the observabilities of the chosen routes are less than their maximum attainable values. This suggests the possibility to adjust Alborg Airport RFID deployment in order to improve the reading environment and thereby reduce or even eliminate the occurrence of RFID anomalies. A modeler can experiment with different scenarios of RFID reader positioning, and monitor the change in the observabilities until optimum values (those closer to the upper bound) are obtained. Three such scenarios are chosen and analyzed in Section 5.1. As a matter of fact, the adequacy of the observabilities obtained using the obs function is dependent on the purpose of building the RFID-based system. For instance, higher route observabilities (and less uncertainty in tracking moving objects) are more crucial in safety- and business-critical RFID-based systems. Indeed, the relation between the route observability and the uncertainty in tracking moving objects along this route can be characterized as follows: the higher a route observability, the less the uncertainty in tracking moving objects along this route. This relation is formalized in Appendix A.2.

5. Static Model-Based Reasoning

The model-based reasoning described in this section deals with the BP concept in a static, time-independent fashion (i.e., independently of timestamped RFID data streams). This kind of reasoning is hence important at the planning stages that precede the actual RFID deployment. A BP is a location in an OI-space where there is potentially a lot of traffic. Definition 1 characterizes the static estimate about BPs, while Lemma 1 explores its nature.

**Definition 1.** (Static BP Estimate) Given an OI-space pseudograph \( D_{\text{oi-space}} = (\mathcal{W}_l, \mathcal{W}_m, c) \), the static support degree that \( l \in \mathcal{W}_l \) is a BP is estimated by the ratio of \( l \)'s pseudodegree to the number of edges in \( D_{\text{oi-space}} \). Formally speaking:

\[
\forall l \in \mathcal{W}_l : E_{BP}(l) = \frac{d(l)}{2|\mathcal{W}_m|}
\]

where \( l \)'s pseudodegree is the number of all directed edges (including loops) whose head or tail is \( l \).

**Lemma 1.** Given an OI-space pseudograph \( D_{\text{oi-space}} = (\mathcal{W}_l, \mathcal{W}_m, c) \), \( E_{BP} \) is consistent as a probability distribution on a random variable whose alphabet is \( \mathcal{W}_l \).

Revisiting \( D_{\text{oi-space}} \) (Figure 2), the locations, their pseudodegrees, and the static estimates that they are BPs are listed in Table 4. Notice that the sorter loop is counted twice when deciding \( d(TTS) \). Thus, CGS has the highest static support degree of being a BP in Alborg Airport hall and apron, followed equally by GS2 and GS3, then by GS1, and after it by GS4 and TTS with equal static support. The relatively high likelihood (4/60) of suffocation by baggage in TTS suggests a need for careful deployment of RFID readers in this location.

Table 4: The locations, their pseudodegrees, and the static estimates that they are BPs in \( D_{\text{oi-space}} \) (Figure 2)

<table>
<thead>
<tr>
<th>( l )</th>
<th>CD</th>
<th>CC</th>
<th>MC</th>
<th>SM</th>
<th>SMCC</th>
<th>TTS</th>
<th>CH</th>
<th>RB</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d )</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( E_{BP} )</td>
<td>1/60</td>
<td>2/60</td>
<td>3/60</td>
<td>2/60</td>
<td>4/60</td>
<td>2/60</td>
<td>1/60</td>
<td>1/60</td>
</tr>
</tbody>
</table>

6. Probabilistic Incorporation of RFID Data

A probabilistic account of RFID data is crucial to compensate for the missing information that is inherent in this data. The probabilistic incorporation attained in this section offers complete and...
more informative knowledge about the locations of moving objects in OI-spaces. This knowledge facilitates the tracking of these objects and enables the search for them to be optimized. Preliminaries are given in Section 6.1 and the novel probabilistic translator follows in Section 6.2.

Algorithm 1 Probabilistic Trajectory-to-Route Translator

Input: \( D_{fid} = (W_l, W_m, c_l, c_m, c_r) \), \( TR(obj, T) \), a dynamic Bayesian network (DBN), the interval between time slices \( inv \) (a positive integer), and the data structures of intermediate, probabilistic, and inferred records:

- inter-ds \( \equiv \) (ar-id, obj-id, loc, s-time, e-time)
- prob-ds \( \equiv \) (obj-id, prob-loc, s-time, e-time)
- infer-ds \( \equiv \) (obj-id, infer-loc, s-time, e-time)

where loc is a location, prob-loc and infer-loc are probability distributions on a random variable whose alphabet is \( W_l \).

Output: Records in the inter-ds, prob-ds, and infer-ds.

1: \( \text{inter-ds} \leftarrow \emptyset; \text{prob-ds} \leftarrow \emptyset; \text{infer-ds} \leftarrow \emptyset; \)
2: for each \( arr \in TR(obj, T) \) do
   // Stage 1. Translation based on \( D_{fid} \).
   3: for each \( l \in W_l \) do
      4: if \( arr\_reader\_id \in c_l(l) \) then
         5: insert \( (arr, arr\_obj\_id, l, arr\_s\_time, arr\_e\_time) \) into inter-ds
      6: break
   7: for each \( m = (l_1, l_2) \in W_m \) \( l_1, l_2 \in W_l \) do
      8: if \( arr\_reader\_id \in c_m(m) \) or \( (arr\_reader\_id, arr\_obj\_id, reader\_id) \in c_r(m) \) then
         9: insert \( (arr, arr\_obj\_id, l_1, arr\_s\_time, arr\_e\_time) \) and \( (arr, arr\_obj\_id, l_2, arr\_s\_time, arr\_e\_time) \) into inter-ds
   10: Transform inter-ds into prob-ds.
   // Stage 3. Inferring the information gaps.
   11: for each \( p\_rec \in \text{prob-ds} \) do
      12: inject \( p\_rec\_prob\_loc \) and \( p\_rec\_probs \) into evidence into DBN
   13: update DBN beliefs using EPIS-BN
   14: beln = first-DBN-belief
   15: beln = last-DBN-belief
   16: insert \( (p\_rec\_obj\_id, beln, p\_rec\_s\_time, p\_rec\_e\_time) \) and \( (p\_rec\_obj\_id, beln, p\_rec\_s\_time, p\_rec\_e\_time) \) into infer-ds
   17: start \( \leftarrow p\_rec\_s\_time + inv \)
   18: end \( \leftarrow p\_rec\_e\_time - 1 \)
   19: if start \( \leq \) end then
      20: evolve infer-loc from DBN
   21: insert \( (p\_rec\_obj\_id, infer-loc, start, end) \) into infer-ds

6.1 Preliminaries

A raw RFID reading can be denoted as a triple of the form \( rd \equiv (obj\_id, reader\_id, \text{time}) \) which indicates that the tag affixed to \( obj\_id \) was detected by reader\_id at timestamp \text{time}. An RFID data stream produced by all the readers in a deployment is then a stream of triples \( S \equiv \{rd_1, rd_2, \ldots \} \). Minding the efficiency of query processing, one does not want to store and persistently manipulate raw RFID readings at the timestamp level. Instead, one would like to store the first and last detection of a tag by a reader, i.e., the appearance of a moving object in a reader’s reading zone over a closed time period. Thus, the level of raw RFID readings is lifted by employing a pre-processing module (the details of which can be found elsewhere) that condenses these readings into so-called appearance records. Each appearance record has the form \( ar \equiv (ar\_id, obj\_id, reader\_id, s\_time, e\_time) \) where \( s\_time \) and \( e\_time \) are the start and end time of an appearance. These appearance records are stored in the data structure appear-ds. A moving object trajectory can now be defined.

Definition 2. (Moving Object Trajectory) A trajectory \( TR(obj, T) \) of a moving object \( obj \) inside an RFID data stream \( S \) over a time period \( T \) is the sequence of appearance records whose detected object is \( obj \) and detection time is in \( T \).

\[ TR(obj, T) = ar_1, ar_2, \ldots, ar_n : ar\_obj\_id = obj \land [ar\_s\_time, ar\_e\_time] \subseteq T \]

For instance, the trajectory of \( bag_1 \) in Figure 3 during \( [t_1, t_{37}] \) is \( TR(bag_1, [t_1, t_{37}]) = ar_1, ar_2, \ldots, ar_7 \). Table 5 lists the corresponding appearance records.

<table>
<thead>
<tr>
<th>ar-id</th>
<th>obj-id</th>
<th>reader-id</th>
<th>s-time</th>
<th>e-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar_1</td>
<td>bag_1</td>
<td>r_1</td>
<td>t_1</td>
<td>t_2</td>
</tr>
<tr>
<td>ar_2</td>
<td>bag_2</td>
<td>r_2</td>
<td>t_5</td>
<td>t_6</td>
</tr>
<tr>
<td>ar_3</td>
<td>bag_3</td>
<td>r_3</td>
<td>t_7</td>
<td>t_8</td>
</tr>
<tr>
<td>ar_4</td>
<td>bag_4</td>
<td>r_4</td>
<td>t_{11}</td>
<td>t_{12}</td>
</tr>
<tr>
<td>ar_5</td>
<td>bag_5</td>
<td>r_5</td>
<td>t_{13}</td>
<td>t_{14}</td>
</tr>
<tr>
<td>ar_6</td>
<td>bag_6</td>
<td>r_6</td>
<td>t_{19}</td>
<td>t_{20}</td>
</tr>
<tr>
<td>ar_7</td>
<td>bag_7</td>
<td>r_7</td>
<td>t_{32}</td>
<td>t_{37}</td>
</tr>
</tbody>
</table>

6.2 Probabilistic Trajectory-to-Route Translator

The precise definition of the records used below can be seen at the top of Algorithm 1. Given a trajectory \( TR(obj, T) \) of a moving object \( obj \) over a time period \( T \) (Table 5), one would like to infer the route that this object followed (or was carried over) between locations over the same period. This is achieved through the probabilistic trajectory-to-route translator in Algorithm 1 which comprises three stages:

Stage 1. Translation based on \( D_{fid} \): Based on the vertex and edge labels in \( D_{fid} \), this stage translates the appearance records in \( TR(obj, T) \) into intermediate records in the inter-ds. Notice that the loop in line 3 terminates after one insertion, since a reader cannot be positioned inside more than one location, neither can it be simultaneously positioned inside a location and at a connection point. On the contrary, the loop in line 7 does not terminate after one insertion, since more than one sub-route can have shared elements in their labels (Figure 4).

Stage 2. Transformation: This stage condenses the intermediate records in the inter-ds into a smaller number of probabilistic records that are pushed into the prob-ds. The transformation is done using a simple SQL query (this query effect will be shown later in an example).

Stage 3. Inferring the information gaps: Gaps in RFID data streams are unavoidable due to RFID anomalies. Stage 3 aims at inferring the information gaps in the prob-ds by borrowing from the prior knowledge available about the RFID-based system. The inference is based on the dynamic Bayesian network (DBN) in Figure 5a. In this DBN, the location \( L_t \) and reader \( R_t \) are two random variables whose alphabets are \( W_l \) and \( W_r \) respectively. \( L_t \) denotes \( obj \)’s location, whereas \( R_t \) denotes the reader detecting \( obj \)’s tag. In probabilistic reasoning terms, \( L_t \) and \( R_t \) are referred to as the state and evidence variables respectively. Additionally, \( R_t \) is observable while \( L_t \) is not. The DBN world in Figure 5a is viewed as a series of time slices each of which contains \( L_t \) and \( R_t \). The interval between these slices depends on the problem considered, and it is parameterized as \( inv \) in the input to Algorithm 1. Three kinds of information specify the DBN in Figure 5a:

1. The transition model \( P(L_{t+1}|L_t) \): It describes the likelihood of \( obj \)’s location at the next slice given its location at the current slice.
2. The sensor model \( P(R_t|L_t) \): It describes the likelihood of detecting \( obj \)’s tag given its location at the current slice.
3. The prior model $P(L_0)$: It describes the likelihood of obj’s location at slice 0.

The complete joint distribution can be determined based on these models as follows:

$$P(L_{0\rightarrow t}, R_{1\rightarrow t}) = P(L_0) \prod_{i=1}^{t} P(L_{i+1} | L_i) P(R_i | L_i)$$

Furthermore, a complete DBN with an unbounded number of slices can be constructed as needed by copying the first slice. The unrolled DBN over five slices is shown in Figure 5(a). Observe in Figure 5(b) that the current state depends only on the previous state and not on any earlier states. This is due to the first-order Markov process assumption which is commonly made when reasoning over time. Another important assumption that is made in Figure 5(b) is that the changes in the DBN world are caused by a process whose laws are static over time. With this assumption in place, only one $P(L_{t+1}|L_t)$ and $P(R_t | L_t)$ has to be specified albeit the unrolled DBN may have infinitely many slices.

![DBN and Unrolled DBN](Image)

Figure 5: Figure (a) shows the DBN used for inferring the information gaps; and Figure (b) shows the unrolled DBN over five slices.

The distributions $P(L_0)$ and $P(L_{t+1}|L_t)$ can be specified based on $D_{fid}$ as follows. Regarding $P(L_0)$, uniform probabilities are ascribed to entry locations and zero probabilities are ascribed to the rest. For instance in Figure 3, CD, RB, and OB are given a probability $1/3$ each, while the rest of the locations are given a probability $0$. Turning to $P(L_{t+1}|L_t)$, it is sufficient to consider the location at slice $t$ and the two locations that follow it along a route in Figure 3. If any of these three locations has a loop, then a higher probability is ascribed to the loop location while lower, equal probabilities are ascribed to the remaining two locations. If none of these locations has a loop, then uniform probabilities are ascribed to all of them. For instance in Figure 3 if the location at $t$ is SMC, then the location at $t+1$ is SMC, TTS, or CH with probabilities $.25$, $.25$, and $.25$ respectively. The distribution $P(R_t | L_t)$, on the other hand, can be specified based on the coverage weights in Table 2. For instance, if the location at $t$ is SMC, then $r_2$ or $r_3$ detects obj’s tag at $t$ with probabilities $.8$ and $.2$ respectively.

The pieces of evidence available in the prob-ds (outcome of stage 2) are injected into the DBN’s $L_t$ nodes in line 12 of Algorithm 1. The purpose of evidence injection is to incorporate the prior knowledge about moving objects available in the prob-ds. Incorporation of prior knowledge has the desirable impact of amplifying the inference quality. Following the evidence injection, the DBN beliefs can be updated using the formula:

$$P(L_{t+1}|R_{1\rightarrow t+1}) = \frac{P(L_{t+1}|R_{1\rightarrow t+1}) P(R_{t+1} | L_{t+1})}{P(R_{t+1} | R_{1\rightarrow t})}$$

DBN belief updating is computationally complex so that several algorithms were developed to cope with this complexity. These algorithms fall into two categories: exact and approximate belief updating. Algorithms for exact belief updating (e.g., variable elimination, polytree [19], and clustering [12]) have exponential space and time complexities in the number of state variables in a DBN. Therefore, we must fall back to approximate algorithms. The most widely used algorithm in the database literature [20, 26, 29] for approximate belief updating is particle filtering [4]. The approximate algorithm used in line 13 of Algorithm I is the Estimated Posterior Importance Sampling algorithm for Bayesian Networks (EPIS-BN) [30]. EPIS-BN uses loopy belief propagation to compute an estimate of the posterior probability over all DBN nodes and then refines this estimate via importance sampling. EPIS-BN is quite likely the best approximate algorithm available to date. In addition to being faster, it produces results that are an order of magnitude more precise than other algorithms. The beliefs that evolve from applying EPIS-BN are used to populate the infer-ds in lines 14-21 of Algorithm I. It is relevant to stress that a DBN is quite likely the best Bayesian filtering method for location estimation [3]. A DBN, for instance, surpasses a Hidden Markov Model (HMM) in its ability to model domains with many state variables. A DBN also outperforms a Kalman filter [23] in which very strong Gaussian assumptions are made. These assumptions limit the applicability of a Kalman filter to location estimation using accurate RFID reader4.

Applying stage 1 of Algorithm I to bag1 trajectory in Table 5 yields the intermediate records in Table 6. Stage 2 transforms Table 6 into the probabilistic records in Table 7. To demonstrate the effect of the SQL query applied in stage 2, note that TTS appears three times in records 3-5 in Table 6 and therefore TTS probability is $.75$ in record 2 of Table 7.

Next, stage 3 is applied in order to infer the information gaps $[t_3, t_4], [t_6, t_{10}], [t_{15}, t_{18}]$, and $[t_{30}, t_{31}]$ in Table 6. Parameterizing stage 3 with $i_{new} = 1$ and proceeding with the computations yield the inferred records in Table 8.

These records correspond to the inferred route of bag1. The temporal evolution of the probabilities in this route is plotted in Figure 6. Note that the decreasing/increasing probabilities associated with SMC/TTS indicate baggage movement from SMC to TTS.

<table>
<thead>
<tr>
<th>ar-id</th>
<th>obj-id</th>
<th>loc</th>
<th>s-time</th>
<th>e-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar1</td>
<td>bag1</td>
<td>MC</td>
<td>t1</td>
<td>t2</td>
</tr>
<tr>
<td>ar2</td>
<td>bag2</td>
<td>SMC</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>ar2</td>
<td>bag1</td>
<td>TTS</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>ar3</td>
<td>bag2</td>
<td>TTS</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>ar3</td>
<td>bag1</td>
<td>SMC</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>ar3</td>
<td>bag2</td>
<td>TTS</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>ar3</td>
<td>bag2</td>
<td>TTS</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>ar4</td>
<td>bag1</td>
<td>SMC</td>
<td>t11</td>
<td>t12</td>
</tr>
<tr>
<td>ar4</td>
<td>bag2</td>
<td>TTS</td>
<td>t11</td>
<td>t12</td>
</tr>
<tr>
<td>ar5</td>
<td>bag1</td>
<td>SMC</td>
<td>t11</td>
<td>t12</td>
</tr>
<tr>
<td>ar5</td>
<td>bag2</td>
<td>TTS</td>
<td>t11</td>
<td>t12</td>
</tr>
<tr>
<td>ar5</td>
<td>bag2</td>
<td>TTS</td>
<td>t11</td>
<td>t12</td>
</tr>
<tr>
<td>ar6</td>
<td>bag1</td>
<td>CH</td>
<td>t10</td>
<td>t20</td>
</tr>
<tr>
<td>ar6</td>
<td>bag1</td>
<td>CGS</td>
<td>t10</td>
<td>t20</td>
</tr>
<tr>
<td>ar7</td>
<td>bag1</td>
<td>GS2</td>
<td>t32</td>
<td>t37</td>
</tr>
<tr>
<td>ar7</td>
<td>bag1</td>
<td>BL2</td>
<td>t32</td>
<td>t37</td>
</tr>
</tbody>
</table>

Contemplating Table 8 in comparison to Table 5 shows that the knowledge obtained from the translator is (1) complete and (2) more informative about the locations of baggage in transit. To clar-

4Readers that do not produce many RFID anomalies in the reported data.
ify (1), note that Table 8 communicates full observability of bag1 during \([t_1, t_{37}]\), whereas the observability delivered is only partial in Table 5 during the same period (note the information gaps \([t_1, t_3], [t_9, t_{10}], [t_{15}, t_{16}]\), and \([t_{30}, t_{31}]\). To give an example on (2), a bag in Table 5 tells that bag1 passed under r2 during \([t_{13}, t_{14}]\). Due to the adjacent positioning of r2 and r3 (Figure 1), this information piece is deficient and possibly inaccurate. Contrary to this, record 7 in Table 5 tells that bag1 is highly likely to be at TTS and less likely to be at SMC during \([t_{13}, t_{14}]\). All in all, the translator better facilitates the tracking of baggage in Aalborg Airport and enables the search for lost baggage to be optimized.

Table 7: The probabilistic records of bag1 during \([t_1, t_{37}]\)

<table>
<thead>
<tr>
<th>obj-id</th>
<th>prob-loc</th>
<th>s-time</th>
<th>e-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>bag1</td>
<td>MC : 1</td>
<td>t1</td>
<td>t2</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t7</td>
<td>t8</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t11</td>
<td>t12</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t13</td>
<td>t14</td>
</tr>
<tr>
<td>bag1</td>
<td>CH : .25, OC : .25, CGS : .5</td>
<td>t19</td>
<td>t20</td>
</tr>
<tr>
<td>bag1</td>
<td>GS2 : .5, BL2 : .5</td>
<td>t32</td>
<td>t37</td>
</tr>
</tbody>
</table>

Table 8: The inferred route of bag1 during \([t_1, t_{37}]\)

<table>
<thead>
<tr>
<th>obj-id</th>
<th>infer-loc</th>
<th>s-time</th>
<th>e-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>bag1</td>
<td>MC : 1</td>
<td>t1</td>
<td>t2</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t5</td>
<td>t6</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t7</td>
<td>t8</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t11</td>
<td>t12</td>
</tr>
<tr>
<td>bag1</td>
<td>SMC : .25, TTS : .75</td>
<td>t13</td>
<td>t14</td>
</tr>
<tr>
<td>bag1</td>
<td>CH : .25, OC : .25, CGS : .5</td>
<td>t19</td>
<td>t20</td>
</tr>
<tr>
<td>bag1</td>
<td>GS2 : .5, BL2 : .5</td>
<td>t32</td>
<td>t37</td>
</tr>
</tbody>
</table>

Figure 6: The temporal evolution of the probabilities in the inferred route of bag1, which is given in Table 8.

7. DYNAMIC MODEL-BASED REASONING

The dynamic nature of moving objects in RFID-based systems that evolves over time makes it useful to model time explicitly. Having incorporated RFID data streams and inferred the routes of moving objects in Section 6 dynamic, time-dependent reasoning about BPs can be done in this section.

Definition 3. (Dynamic BP Estimate) Given an RFID readers deployment pseudograph \(D_{\text{rfid}} = (W_l, W_m, c_l, c_m, c_r)\), the infer-ds, and a monitoring period \(T\) of a location \(l \in W_l\), the dynamic support degree over \(T\) that \(l\) is a BP is estimated by the joint probability distribution on all the random variables of the inferred records in the infer-ds whose detection time is joint (overlapping/nested) with \(T\). Formally speaking:

\[
\forall l \in W_l : E^T_{Dm}(l) = Pr(\text{obj}_1 \text{ at } l, \ldots, \text{obj}_n \text{ at } l) : \text{obj}_j \in W_o
\]

The joint distribution in Definition 3 indicates that the marginal distributions \(Pr(\text{obj}_j \text{ at } l)\) occur with a certain probability. This means concentration at \(l\).

Definition 4. (Dynamic BP Monitoring Query) A dynamic BP monitoring query (BPMQ) takes as input an RFID readers deployment pseudograph \(D_{\text{rfid}} = (W_l, W_m, c_l, c_m, c_r)\), the infer-ds, and a monitoring period \(T\). It then reports \(E^T_{Dm}(l)\) for all \(l \in W_l\).

To put in symbols:

\[
\text{BPMQ}^T = \{ E^T_{Dm}(l) : l \in W_l \}
\]

It is known that inference using joint distributions has prohibitive time complexity [22]. This complexity is coped with in two ways. First, the specification of a monitoring period \(T\) limits the inference to only a subset \(I-REC\) of the infer-ds:

\[
I-REC(T) = \{ i\text{-rec } \in \text{infer-ds} : \text{[i-rec-s-time, i-rec-e-time]} \cap T \neq \emptyset \}
\]

Second, the absolute independence assertion (imposed on random variables) radically reduces the amount of information necessary to encode the joint distribution by enabling the factoring of this distribution into separate, smaller distributions.

An algorithm for answering a BPMQ is given in Algorithm 2.

The probability tweaking parameter \(\eta\) (in the input to this algorithm) is fed as a percentage, and it specifies the quotient by which all \(E^T_{Dm}\) are (de)concentrated in accordance with the detection time. The normalization function \(\psi\) normalizes \(E^T_{Dm}\) generated by the algorithm by dividing each \(E^T_{Dm}\) by the sum of all \(E^T_{Dm}\). This normalization transforms \(E^T_{Dm}\) and ensures its consistency as a probability distribution on a random variable whose alphabet is \(W_l\). Capturing the dynamic estimates in a probability distribution facilitates comparing them with the static estimates that were also represented as a probability distribution in Section 6.

Algorithm 2 Answering a BPMQ

Input: \(D_{\text{rfid}} = (W_l, W_m, c_l, c_m, c_r)\), the infer-ds, a monitoring period \(T\), a probability tweaking parameter \(\eta\), and a normalization function \(\psi\) to \([0, 1]\).

Output: \(\psi(E^T_{Dm})\).

1: extract \(I-REC(T)\) from the infer-ds
2: increase \(i = 1.0 + \eta/100.0\)
3: decrease \(i = 1.0 - \eta/100.0\)
4: for each \(l \in W_l\) do
5: \(E^T_{Dm}(l) = \{i\text{-rec } \in I-REC(T) : l \in i\text{-rec}\}\)
6: for each \(i\text{-rec} \in I-REC(T)\) do
7: \(l = i\text{-rec-e-time} - i\text{-rec-s-time}\)
8: if \(i\text{-rec-pr}(\text{obj}_j \text{ at } l) > 0\) then
9: \(E^T_{Dm}(l) = E^T_{Dm}(l) \times \text{i-rec-pr}(\text{obj}_j \text{ at } l)\)
10: repeat \(t\) times
11: \(E^T_{Dm}(l) = E^T_{Dm}(l) \times \text{increase}\)
13: else
14: repeat \(t\) times
15: \(E^T_{Dm}(l) = E^T_{Dm}(l) \times \text{decrease}\)
16: return \(\psi(E^T_{Dm})\)

Suppose that the inferred route of bag2 during \([t_3, t_{28}]\) is as given in Table 8. Imagine further that bag1 (whose inferred route is given in Table 6) and bag2 are the only bags that are handled during \([t_1, t_{37}] \cup [t_3, t_{28}] = [t_1, t_{37}]\). To answer BPMQ\([t_1, t_{37}]\), one follows Algorithm 2 steps extracting \(I-REC([t_1, t_2])\) from the
infer-ds to get the records in Table 10 and then proceeding with the calculations given \( \eta = 10\% \) to obtain:

\[
\begin{align*}
E^{v_1 \rightarrow v_2}_{BP} & (MC) = 4 \times 0.39 \times 0.32 \times 0.70 \times 1.1^5 = 0.6512 \\
E^{v_1 \rightarrow v_2}_{BP} & (SMC) = 4 \times 0.40 \times 0.30 \times 0.45 \times 0.9^2 \times 1.1^5 = 0.0394 \\
E^{v_1 \rightarrow v_2}_{BP} & (TTS) = 4 \times 0.21 \times 0.70 \times 0.85 \times 0.3 \times 1.1^5 = 0.1517 \\
E^{v_1 \rightarrow v_2}_{BP} & (rest of locations) = 0
\end{align*}
\]

A final normalization of \( E^{v_1 \rightarrow v_2}_{BP} \) yields respectively the values: 

\[
(7731, 0.468, 0.1801, 0)
\]

Table 9: The inferred route of \( b_{ag_2} \) during \([t_3, t_{28}]\)

<table>
<thead>
<tr>
<th>obj-id</th>
<th>infer-loc</th>
<th>s-time</th>
<th>e-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_{ag_2} )</td>
<td>( MC : 1 )</td>
<td>( t_3 )</td>
<td>( t_4 )</td>
</tr>
<tr>
<td>( b_{ag_2} )</td>
<td>( SMC : 0.32, SMC : 0.45, TTS : 0.23 )</td>
<td>( t_5 )</td>
<td>( t_7 )</td>
</tr>
<tr>
<td>( b_{ag_2} )</td>
<td>( CH : 0.31, CGS : 0.69 )</td>
<td>( t_{11} )</td>
<td>( t_{25} )</td>
</tr>
<tr>
<td>( b_{ag_2} )</td>
<td>( GS1 : 0.41, BL1 : 0.59 )</td>
<td>( t_{26} )</td>
<td>( t_{28} )</td>
</tr>
</tbody>
</table>

Table 10: \( I-REC([t_1, t_2]) \) extracted from the infer-ds

<table>
<thead>
<tr>
<th>obj-id</th>
<th>infer-loc</th>
<th>s-time</th>
<th>e-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_{ag_3} )</td>
<td>( MC : 1 )</td>
<td>( t_1 )</td>
<td>( t_2 )</td>
</tr>
<tr>
<td>( b_{ag_3} )</td>
<td>( SMC : 0.39, SMC : 0.40, TTS : 0.21 )</td>
<td>( t_3 )</td>
<td>( t_4 )</td>
</tr>
<tr>
<td>( b_{ag_3} )</td>
<td>( SMC : 0.30, TTS : 0.70 )</td>
<td>( t_5 )</td>
<td>( t_6 )</td>
</tr>
<tr>
<td>( b_{ag_3} )</td>
<td>( SMC : 0.14, TTS : 0.86 )</td>
<td>( t_7 )</td>
<td>( t_8 )</td>
</tr>
<tr>
<td>( b_{ag_3} )</td>
<td>( MC : 1 )</td>
<td>( t_3 )</td>
<td>( t_4 )</td>
</tr>
<tr>
<td>( b_{ag_3} )</td>
<td>( SMC : 0.32, SMC : 0.45, TTS : 0.23 )</td>
<td>( t_5 )</td>
<td>( t_7 )</td>
</tr>
</tbody>
</table>

In this example, \( MC \) has the highest dynamic support degree of being a BP in Aalborg Airport hall and apron, followed by TTS, and then by SMC. The dynamic support for the rest of the locations is zero, due to the complete absence of inferred records in these locations as seen in Table 11.

8. FUNCTIONAL ANALYSIS AND EXPERIMENTAL EVALUATION

This section offers an analysis of the \( obs \) function (Formula 2) in addition to two series of experiments that evaluate the accuracy and performance of Algorithm 1 and 2. The experiments are conducted on actual, uncleaned data that is gathered from Aalborg Airport RFID deployment over the period between 2011-08-10 and 2012-09-17. The dataset size is 2, 189 MBs. The actual deployment in the hall differs from the one shown in Figure 1 in that a single reader is deployed at \((SMC|TTS)\). Reader deployment in the apron is at present planned. Therefore, RFID readings from the apron are currently unavailable, and the outdoor locations and readers are excluded from \( D_{RFID}(\text{Figure } 3) \). The pre-processing module (mentioned in Section 5.1) reduces the number of raw RFID readings from around 3.3 million down to 845,000 that are stored in the infer-ds. The overall number of RFID-tagged bags for which these readings are reported is around 270,000. The implementation is done in Java SE version 1.7.0_10 and MATLAB version 7.14. The DBMS used is Oracle 11g Release 2 version 11.2.0.2.0. The desktop machine on which the experiments are conducted has an Intel(R) Core(TM) i7-2600 processor with clock speed 3.40 GHz and 8.00 GB memory, and runs a 64-bit installation of Microsoft Windows 7 Enterprise version 6.1.7601.

8.1 Analysis of the \( obs \) Function

In order to analyze and discern the behavior of the \( obs \) function, the baggage route \( (MC \rightarrow BL1) \equiv MC \rightarrow SMC \rightarrow TTS \rightarrow CH \rightarrow CGS \rightarrow GS1 \rightarrow BL1 \) is chosen (Figure 3) due to the considerable number of readers positioned along it when compared to other routes. The readers positioning and the number of locations along this route are varied following three scenarios. In the first scenario (Figure 7a), the readers positions are varied along a line parallel to \((MC \cdots BL1)\). The initial coverage weights (before the variation starts) are listed in Table 11. This scenario is used to understand the impact of varying the distribution of coverage weights along a route on this route’s observability. In the second scenario (Figure 7b), the readers positions are varied along a line at a right angle to \((MC \cdots BL1)\). The initial coverage weights (before the variation starts) are listed in Table 12. This scenario is used to understand the impact of decreasing the coverage weights along a route on this route’s observability. In the third and final scenario, the number of locations along \((MC \cdots BL1)\) is increased as shown in Table 13. Together with this increase, the reader positions along \((MC \cdots BL1)\) are varied at a right angle to this route (in a similar fashion to Figure 7). This scenario is used to understand the impact of increasing a route length and simultaneously decreasing the coverage weights on this route’s observability. In all the aforementioned scenarios, the variation in readers positioning changes (increases/decreases) the coverage weights of the locations along \((MC \cdots BL1)\) by 5% at a time.

Figure 7: Figure (a) shows the variation in readers positioning along a line parallel to the route \((MC \cdots BL1)\); and Figure (b) shows the variation along a line at a right angle to the same route. Dashed arrows depict the variation, and solid ones depict the route.

Table 11: The initial coverage weights of the locations along \((MC \cdots BL1)\) in Figure 7a

| l | \( c_r(l) = \{r \rightarrow w(r)\} \)
|---|---|
| MC \( \{r_1 \rightarrow 1\} \) | CGS \( \{r_4 \rightarrow 0, r_5 \rightarrow 0\} \)
| TTS \( \{r_2 \rightarrow 0, r_3 \rightarrow 0\} \) | BL1 \( \{r_6 \rightarrow 0\} \)
| CH \( \{r_4 \rightarrow 1, r_5 \rightarrow 1\} \) |

Table 12: The initial coverage weights of the locations along \((MC \cdots BL1)\) in Figure 7b

| l | \( c_r(l) = \{r \rightarrow w(r)\} \)
|---|---|
| MC \( \{r_1 \rightarrow 1\} \) | CGS \( \{r_4 \rightarrow 0.5, r_5 \rightarrow 0.5\} \)
| SMC \( \{r_2 \rightarrow 1, r_3 \rightarrow 0\} \) | GS1 \( \{r_6 \rightarrow 0.5\} \)
| TTS \( \{r_2 \rightarrow 0, r_3 \rightarrow 1\} \) | BL1 \( \{r_6 \rightarrow 0.5\} \)
| CH \( \{r_4 \rightarrow 0.5, r_5 \rightarrow 0.5\} \) |

The impact of the first, second, and third variation scenarios on \( obs(MC \cdots BL1) \) is depicted in Figures 8a, 8b, and 8c respectively. Figure 8a tells that \( obs(MC \cdots BL1) \) is maximized when
Table 13: Increasing the number of locations along (MC ... BL1)

<table>
<thead>
<tr>
<th>(MC ... BL1)</th>
<th>Coverage Weights (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td></td>
</tr>
<tr>
<td>MC → SMC</td>
<td></td>
</tr>
<tr>
<td>MC → SMC → TTS</td>
<td></td>
</tr>
<tr>
<td>MC → SMC → TTS → CH</td>
<td></td>
</tr>
<tr>
<td>MC → SMC → TTS → CH → CGS</td>
<td></td>
</tr>
<tr>
<td>MC → SMC → TTS → CH → CGS → GS1</td>
<td></td>
</tr>
<tr>
<td>MC → SMC → TTS → CH → CGS → GS1 → BL1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8: Figures(a), (b) and (c) respectively show the effect of the first, second, and third variation scenarios on obs(MC ... BL1)

the coverage weights along (MC ... BL1) are evenly distributed (i.e., when the reading zones of readers positioned at connection points uniformly cover adjacent locations). The short term even positioning can be used to refer to this kind of readers positioning in an RFID deployment. The interpretation of the opposite term uneven positioning follows in a similar fashion. Uneven positioning yields the undesirable impact of minimizing obs(MC ... BL1). Notice again in Figure 5 that obs(MC ... BL1) attains its maximum value (6.8496 bits) under the following even positioning:

\[ c_r(SMC) = c_r(TTS) = \{r_2 \rightarrow 0, r_3 \rightarrow 0 \} \]
\[ c_r(CH) = c_r(CG) = \{r_4 \rightarrow 0, r_5 \rightarrow 0 \} \]
\[ c_r(GS1) = c_r(BL1) = \{r_6 \rightarrow 0 \} \]

On the other hand, obs(MC ... BL1) attains its minimum value (6 bits) in Figure 5a under the following uneven positioning:

\[ c_r(SMC) = \{r_2 \rightarrow 1, r_3 \rightarrow 0 \} \]
\[ c_r(TTS) = \{r_2 \rightarrow 0, r_3 \rightarrow 1 \} \]
\[ c_r(CH) = \{r_4 \rightarrow 1, r_5 \rightarrow 1 \} \]
\[ c_r(CG) = \{r_4 \rightarrow 0, r_5 \rightarrow 0 \} \]
\[ c_r(GS1) = \{r_6 \rightarrow 0 \} \]

and the following uneven positioning:

\[ c_r(SMC) = \{r_2 \rightarrow 0, r_3 \rightarrow 1 \} \]
\[ c_r(TTS) = \{r_2 \rightarrow 1, r_3 \rightarrow 0 \} \]
\[ c_r(CH) = \{r_4 \rightarrow 0, r_5 \rightarrow 0 \} \]
\[ c_r(CG) = \{r_4 \rightarrow 1, r_5 \rightarrow 0 \} \]
\[ c_r(GS1) = \{r_6 \rightarrow 0 \} \]

Figure 5b exhibits that obs(MC ... BL1) decreases with the decrease in the coverage weights along (MC ... BL1) (i.e., with the decrease in the overlap between readers reading zones and the areas of locations). This observation is in line with the obs function definition given in Formula 2. Last, the fluctuation in Figure 5c between the last four obs values 5.2293, 4.9577, 5.8471, and 5.4592 tells that extending (MC ... BL1) (by adding locations observed by readers) does not necessarily increase obs(MC ... BL1) if the readers along (MC ... BL1) are improperly positioned. Put equivalently, the number of readers and their positioning are equally important to achieve a desirable route observability (recall from Section 4 that the number of readers positioned along a route does not accurately reflect this route observability).

8.2 Evaluation of Algorithm 1

Accuracy: In order to evaluate the accuracy of the translation done by Algorithm 1 one should decide how far the translated distribution of baggage is from the real one (from the ground truth). This distance is measured via Jensen-Shannon (JS) divergence measure 16, which is defined by the formula:

\[ JS(p_1, p_2) = \sum_{x \in X} p_1(x) \cdot \log \frac{p_1(x)}{\frac{p_1(x) + p_2(x)}{2}} \]

where \( p_1 \) and \( p_2 \) are two probability distribution functions on a random variable \( X \) whose alphabet is \( \mathcal{X} \). \( JS \) enjoys a number of salient properties that the commonly-used Kullback-Leibler (KL) divergence lacks 28. Among these properties are the finiteness and boundness (\( 0 \leq JS \leq 1 \)). The reader is referred to 6 for a detailed comparison between these two divergence measures. Another work 7 replaces KL with \( JS \) to offer a seminal refinement of an information flow metric. Returning to the accuracy evaluation, the translated distribution of each bag can be easily identified by looking at the infer-ds. The real distribution has to be constructed recalling the intuition that reality occurs with certainty, i.e., with a probability of 1. Minding the fairness of this evaluation, synthetic RFID readings are generated for all the 270,000 bags for which actual data is available. Naturally, the synthetic data is generated under the virtual assumptions of optimal coverage and read rates of RFID readers, and optimal baggage handling in Aalborg Airport hall. This implies that RFID anomalies are not expected. Moreover, it means that baggage delivered at the check-in desks is properly handled until it is loaded into the designated airplanes. The synthetic data is stored in the synth-ds whose records have the same form as those of the infer-ds. One then proceeds to identify the real distribution of each bag (in the synth-ds) and compute \( JS \) divergence between it and the corresponding translated distribution (in the infer-ds). Thus, there is one \( JS \) divergence value per bag, i.e., 270,000 \( JS \) values in total. Due to this large number of \( JS \) values, the known range \([0, 1]\) of \( JS \) is partitioned into 10 smaller ranges, the length of each being 0.1. Then the distribution of the 270,000 bags across these ranges is reported in Figure 5a. As seen in this figure, the translated distribution is generally no further than \([0, 0.4]\) (99.14% of the bags) from reality, which demonstrates the competitive accuracy of the translator.

Inference Quality: Another aspect of evaluating Algorithm 1 is to judge the quality of the inference carried out in stage 3. For this, one should compare between two distances: the distance between the DBN-based infer-ds and real synth-ds distributions, and the distance between the naive prob-ds and real synth-ds distributions. Figure 5b demonstrates the former of these two distances. The latter distance can be similarly determined using \( JS \) divergence: it is plotted in Figure 5c. Figure 5a tells that for the DBN-based distribution, only 0.86% of the bags falls outside the range \([0, 0.4]\), as opposed to 7.42% for the naive distribution. Thus, the DBN-based inference of Algorithm 1 improves by around 8.6 times
over the naive translation that does not utilize a DBN.

**Performance:** In order to evaluate the performance of Algorithm 1, the number of appearance records is varied in the input, and then the time needed by the algorithm to populate the infer-ds is plotted in Figure 9c. Clearly, the execution time increases with the increase of the processed records. Roughly speaking, 94% of the execution time is spent on stage 3, whereas 6% is spent on stages 1 and 2.

### 8.3 Evaluation of Algorithm 2

First, the translator (Section 6.2) is utilized on the full content of the appear-ds in order to populate the infer-ds, input to Algorithm 2. Then the input parameters to this algorithm are varied as shown in Table 14 (the number of inferred records corresponding to each setting appears within parentheses).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
<td>2012-02-01 (375), 2012-04-15 (164), 2012-06-01 (602)</td>
</tr>
<tr>
<td>$T$ in hours</td>
<td>1 (358), 2 (462), 3 (551), 4 (796)</td>
</tr>
<tr>
<td>$\eta$</td>
<td>1% (233), 2% (223), 3% (223), 4% (223)</td>
</tr>
</tbody>
</table>

**Effect of varying the day:** Figures 10a, 10b, and 10c show $E_{BP}^T$ for 30 minutes in the morning, afternoon, and evening of the three days listed in Table 14. Notice in these figures that MC, SMC, and TTS have high dynamic support degrees of being BPs in Aalborg Airport hall in most of the day periods. Occasionally, CGS witnesses high dynamic support of being a BP. Looking closely tells that the results obtained answer 9 BPMQs posed in different day periods, and targeting baggage handling quality at Aalborg Airport. Thus, the dynamic estimation constitutes a good model that not only highlights the points of potential suffocation by baggage, but also ranks the standard of each responsibility area in Aalborg Airport. According to a statement from Aalborg Airport, the actual congestion problems mostly occur in MC, SMC, and TTS. This is consistent with the BPs identified by our model. Figure 10a shows that the execution time decreases, for all days, in the order morning, afternoon, and evening, which reflects a corresponding decrease in Aalborg Airport hall traffic. Recall that the execution time is dependent on the number of processed inferred records.

**Effect of varying $T$:** Results for varying $T$ for one day, according to the values in Table 14 are shown in Figure 10c. The variation of $E_{BP}^T$ in this figure does not follow a noticeable pattern. One can however reason about the histograms by saying that expanding $T$ does not necessarily lead to an increase in the dynamic support for a specific location at the expense of others. Instead, this expansion may introduce support for locations that were not BPs prior to the expansion. Figure 10c tells that the execution time increases proportionally to $T$, primarily due to the corresponding increase in the number of inferred records that Algorithm 2 processes.

**Effect of varying $\eta$:** The impact of varying $\eta$ (as prescribed in Table 14) on $E_{BP}^T$, and the execution time is depicted in Figures 11a and 11c respectively. The slight fluctuation in these figures is due to the time the Java Virtual Machine needs, and the manner it handles the multiplication and rounding of floating-point numbers.

**Performance:** It is also interesting to study the execution time of Algorithm 2 when varying the number of inferred records in the input. This is shown in Figure 11a.

### 9. RELATED WORK

Although it falls into several categories, related work has so far focused on the modeling of indoor spaces. An integrated indoor model [5] covers different information dimensions of indoor models including thematic, geometric, and routing-related information. A lattice-based location model for indoor navigation [14] is capable of preserving semantic relationships and distances, e.g., the nearest neighbor relationship among indoor entities. A grid graph-based model for indoor environments [15] combines the structural properties of these environments with the continuous metric properties that might be of interest to some applications. Another work [24] employs this grid model for evacuation planning. A distance-aware indoor space model [17] accompanies a set of indoor distance computation algorithms and an indexing framework in order to enable the processing of indoor distance-aware queries over indoor spatial objects. This work distinguishes itself from those aforementioned by capturing both O- and I-spaces in a unified model. A model of built environments [25] uses bigraphs in order to understand the relationships between entities in these environments. The authors’ development of inference tools on top of their bigraphs is however ongoing. In contrast, the present paper offers a self-contained set of modeling and reasoning techniques for O- and I-spaces.

The development of a navigation ontology for outdoor and indoor environments [29] is ongoing. This ontology is based on so-called shared microworlds between these two environments. These microworlds are learnt through the application of the Affordance Theory, which enables the identification of functions that entities in outdoor and indoor spaces have or have not in common. A framework for otological reasoning (PERSONAF) [18] is composed of two parts, a three-layered ontology (PECO) and a reasoning engine (ONCOR). The third layer in PECO involves continually changing knowledge about people locations in a building (detected via Bluetooth sensors). ONCOR interprets all three layers of PECO using the accretion-resolution approach. This approach resolves the evidence set to determine which propositions in PECO are relevant for a user. The present paper’s model differs from [29] [18] as fol-
Evening
CGS
SMC
OB
RB
4 hours
CH
TTS
OB
OC
CH
TTS
RB
Afternoon
MC
CH
TTS
OB
OC
RB
3 hours
CGS
RB
SMC
MC
TTS
TTS
RB

about moving objects rather than navigation which is the theme of
extended model. This translator performs probabilistic incorporation of RFID
data that enables the search for moving objects in OI-spaces to be optimized. The translated RFID data permits reasoning about BPs in a dynamic, time-dependent fashion. The functional analysis and experimental evaluation (conducted on both synthetic and uncleansed, real-world data) validate the proposals made in this paper. In particular, they recognize the behavior of the route observability function, and they demonstrate the competitive accuracy of the translator. Furthermore, they show the high quality of the inference and reasoning about the BPs.

10. CONCLUSIONS
This paper reviews and extends a recent unified model of OI-spaces and receptor deployments in these spaces. The proposed extension enables modelers to express their awareness of various information pieces from the physical world. Based on the extended model, the paper derives the bounded route observability function, and demonstrates its potential for enhancing the reading environments. The paper defines the notion of a BP. It then performs static reasoning about this notion using the extended model. The paper describes a trajectory-to-route translator that utilizes the extended model. This translator performs probabilistic incorporation of RFID data that enables the search for moving objects in OI-spaces to be optimized. The translated RFID data permits reasoning about BPs in a dynamic, time-dependent fashion. The functional analysis and experimental evaluation (conducted on both synthetic and uncleansed, real-world data) validate the proposals made in this paper. In particular, they recognize the behavior of the route observability function, and they demonstrate the competitive accuracy of the translator. Furthermore, they show the high quality of the inference and reasoning about the BPs.

Acknowledgment
This work was supported by the BagTrack project sponsored by the Danish National Advanced Technology Foundation under grant 010-2011-1.

11. REFERENCES
The absolute lower bound

A. PROOFS

APPENDIX

A. PROOFS

A.1 Proof of Theorem [1]

The absolute lower bound \( \text{obs}(R) \geq 0 \) is a natural result of \( w(r) \) falling in the range \([0, 1]\). As for the dynamic upper bound, it is noted that for a concave function \( f \) and a random variable \( X \), Jensen’s inequality gives:

\[
E f(X) \leq f(E(X))
\]

The function \( \log(w(r)+1) \) is concave (i.e., it lies above any chord), therefore:

\[
\text{obs}(R) = \sum_{l \in \mathbb{V}(R)} \log \left( \sum_{r \in c_r(l)} w(r) + 1 \right) \quad \text{(Formula 2)}
\]

This gives the bounds of \( \text{obs}(R) \):

\[
0 \leq \text{obs}(R) \leq \sum_{l \in \mathbb{V}(R)} \log \left( c_r(l) + |c_r(l)| \right)
\]

and proves the theorem.

A.2 Observability and Uncertainty

Suppose that one is tracking an object \( obj \) moving along an arbitrary route \( R \) in an IO-space monitored by an RFID-based system. The appearance of \( obj \) at an arbitrary location along \( R \) denotes an event \( x \) which occurs with a probability \( p(x) \). The information conveyed by this event (self-information) is defined as follows:

\[
SI(x) = - \log p(x)
\]

Intuitively, the higher \( \text{obs}(R) \), the higher \( p(x) \), and so the lower \( SI(x) \). In information theory terms, \( SI(x) \) also summarizes the uncertainty about the occurrence of \( x \). Therefore, the higher \( \text{obs}(R) \), the lower the uncertainty about the occurrence of \( x \).

A.3 Proof of Lemma [1]

Proving the consistency of \( E_{BP} \) as a probability distribution is carried out in two steps. In the first step, \( E_{BP} \) is shown to have proper bounds as follows:

\[
\sum_{l \in \mathbb{W}_m} d(l) = 2 |\mathbb{W}_m| \quad \text{(Basic result in graph theory)}
\]

\[
\sum_{l \in \mathbb{W}_m} d(l) = 1 \quad \text{(Division properties, \( \mathbb{W}_m \) is nonempty)}
\]

\[
0 \leq \frac{d(l)}{2 |\mathbb{W}_m|} \leq 1 : \forall l \in \mathbb{W}_l \quad \text{(Inequality properties)}
\]

\[
E_{BP}(l) \in [0, 1] : \forall l \in \mathbb{W}_l \quad \text{(Definition 1)}
\]

In the second step, it is ensured that no intermediate value of \( E_{BP} \) falls outside the range \([0, 1]\) by showing that \( E_{BP} \) is a monotonically increasing function, i.e., it is shown that:

\[
\forall l_1, l_2 \in \mathbb{W}_l : d(l_1) \leq d(l_2) \Rightarrow E_{BP}(l_1) \leq E_{BP}(l_2)
\]

as follows:

\[
d(l_1) \leq d(l_2) \quad \text{(Assumption)}
\]

\[
\frac{d(l_1)}{2 |\mathbb{W}_m|} \leq \frac{d(l_2)}{2 |\mathbb{W}_m|} \quad \text{(Division properties, \( \mathbb{W}_m \) is nonempty)}
\]

\[
E_{BP}(l_1) \leq E_{BP}(l_2) \quad \text{(Definition 1)}
\]

Thus, \( E_{BP} \) is invariably consistent as a probability distribution.