Unified Modeling and Reasoning in Constrained Outdoor and Indoor Spaces

Sari Haj Hussein, Hua Lu, and Torben Bach Pedersen

July, 2012

TR-30

A DB Technical Report

Title	Unified Modeling and Reasoning in Constrained Outdoor and Indoor Spaces
	Copyright © 2012 Sari Haj Hussein, Hua Lu, and Torben Bach Pedersen. All rights reserved.
Author(s)	Sari Haj Hussein, Hua Lu, and Torben Bach Pedersen
Publication History	July 2012. A DB Technical Report

Abstract

Geographic information systems traditionally dealt with only outdoor spaces. In recent years, indoor spatial information systems have started to attract attention partly due to the increasing use of receptor devices (e.g., RFID readers or wireless sensor networks) in both outdoor and indoor spaces. Applications that employ these devices are expected to span uniformly and supply seamless functionality in both outdoor and indoor spaces. What makes this impossible is the current absence of a unified account of these two types of spaces both in terms of modeling and reasoning about the models. This paper presents a unified model of outdoor and indoor spaces and receptor deployments in these spaces. The model is expressive, flexible, and invariant to the segmentation of a space plan, and the receptor deployment policy. It is focused on partially constrained outdoor and indoor motion. On top of this model, clear representation of routes is honed, and a powerful route observability function is derived. The model facilitates probabilistic incorporation of receptor data through a probabilistic trajectory-to-route translator that enables performing high-level reasoning about points of potential traffic (over)load in outdoor and indoor spaces, so-called bottleneck points. An experimental evaluation corroborates the accuracy of the translator and the sensibleness of the reasoning, when applied to synthetic data, and importantly, to uncleansed, real-world receptor data obtained from tracking RFID-tagged flight baggage.

Keywords: outdoor space, indoor space, modeling, RFID, route observability, probabilistic translator, uncertainty, reasoning, bottleneck point.

1 Introduction

Ubiquitous receptor devices such as RFID readers, wireless sensor networks (WSNs), and motion detectors are increasingly deployed in outdoor and indoor spaces (OI-spaces) [19] to enable new classes of applications that enhance our ambient awareness about the physical world. A myriad of examples exist, of which we mention supply chain and product lifecycle management, asset and personnel tracking, environmental monitoring, and intelligent buildings. In order to support these emerging applications, so-called *receptor-based systems* [8] are being built with a focus on managing and analyzing the data collected by receptors. In most of these systems, outdoor and indoor motion is partially constrained, primarily due to the presence of obstacles in outdoor spaces (O-spaces) and floor plans in indoor spaces (I-spaces).

As a motivating example, imagine a scenario in which an airport's personnel are notified about the baggage loss of a specific passenger. The proliferation of mobile technologies should equip these personnel with positioning assistance that enables them to locate the lost baggage. One can imagine that mobile access to online geographic database services could provide the apron[1] map and the floor plan of the airport. Additionally, the analysis of receptor data, obtained from RFID deployments or WSNs, could provide real-time information on the personnel smartphones that tells the approximate location of the lost baggage. However, our ability to analyze and reason about receptor data has lagged behind its acquisition technologies. Indeed, we are faced with the challenges of coping with incomplete information, dealing with incompatible models of OI-spaces, in addition to taming the uncertainty about which features are the most salient in these spaces.

A common assumption made in geographic information systems research is that geographic spaces under consideration are O-spaces. As a matter of fact, a considerable portion of our lives is spent indoors - what increases the size and complexity of I-spaces. Nonetheless, indoor spatial information systems are less developed than their outdoor counterparts that have GIS at their core. The unification of these two types of spaces, both in terms of modeling and reasoning about the models, is lacked so far. A variety of applications, facilitated by receptor-based systems, need to span seamlessly both O- and I- spaces. The most fundamental of these applications is positioning, i.e., determining the location of a moving object in OI-spaces. Supporting this application and others, at various levels in OI-spaces, motivates this study

[1]An open part of an airport in which airplanes are parked, fueled, boarded by passengers, and loaded with baggage.

which reports a unified modeling of OI-spaces and receptor deployments in these spaces. This modeling seamlessly integrates the topology and dynamics of OI-spaces. It enables probabilistic incorporation of receptor data that minds the innate anomalies in this data. Furthermore, it permits performing high-level reasoning about points of potential traffic load in OI-spaces, so-called bottleneck points. Another purpose of this study lies in providing clear representation of routes, and deciding how much of these routes is covered by receptors.

The remainder of this paper is organized as follows. Section 2 presents the unified pseudograph-based model of OI-spaces. The modeling of a case study of receptor-based systems, namely an RFID readers deployment, is chosen and dealt with in Section 3. Using a route formalism we introduce in Section 4, and building on some solid information-theoretic foundations, we derive our route observability function from the ground up, and establish its lower and upper bounds. The notion of bottleneck points is realized in Section 5. Static reasoning about this notion is performed in the same section on top of the OI-space model built in Section 2. Probabilistic incorporation of RFID data is carried out in Section 6 using our probabilistic trajectory-to-route translator. This incorporation paves the way for an over-time upgrade, we perform in Section 7, of the static model-based reasoning done in Section 5. Section 8 follows with a comprehensive experimental evaluation that analyzes the probabilistic trajectory-to-route translator and the dynamic bottleneck point estimate algorithms under a variety of settings. Related work is discussed in brief in Section 9, and the paper concludes in Section 10.

2 Modeling an OI-Space

This section presents a unified pseudograph model that captures two essential elements of an OI-space; the topology (i.e., the geometric properties) and the dynamics (i.e., the changes in motion). Given an OI-space plan, we proceed through four steps to identify semantic locations, connection points, moving objects, and routes. Then we construct our model, to finally show how to control its granularity and permit alternative interpretations. To illustrate our definitions, we adopt a concrete example throughout this study while emphasizing that our choice is by no means curtailing of the generality of the proposed modeling. Our example is the real-world baggage handling plan in Aalborg Airport. This plan comprises two essential sub-plans; the O-space and I-space plans that offer a graphical representation of baggage handling in an apron and a hall in Aalborg Airport respectively. The O-space and I-space plans are respectively shown in Figures 1 and 2. Notice in these figures that the gateways are meeting points for baggage handling that span both the hall and the apron. Notice further that the motion of baggage is partially constrained in the hall (due to the presence of conveyors) and the apron (due to the parked airplanes).

Identifying Semantic Locations Receptor-based systems are typically not interested in the latitude and longitude of a point, neither are they interested in a location named after a specific receptor. Instead, applications are interested in the notion of a *"semantic location"*, which is a location that has a meaningful interpretation to the application. In the I-space plan shown in Figure 2, some meaningful semantic locations are the screening machine conveyor[2] (SMC), the tilt-tray sorter [3] (TTS), the chutes[4] (CH) that are numbered from 1 to 12, the re-induction baggage[5] (RB) and the odd-size baggage (OB) collection points. As for the O-space plan shown in Figure 1, the meaningful semantic locations are the airplanes AP1-AP3, the belt loaders[6] BL1-BL3, in addition to the geometric segments CGS, and GS1-GS4. Observe that the seg-

[2] A machine that performs X-ray screening of baggage as a security measure against weapon infiltration.

[3] A high-speed, continuous-loop sortation conveyor used to sort baggage to chutes.

[4] A narrow, steep slope from which baggage is loaded into wagons that ultimately transport the baggage to airplanes.

[5] A point at which baggage that underwent a manual security check and found safe are re-inducted into the baggage handling system.

[6] A vehicle with a movable belt for loading/unloading baggage into/from an airplane

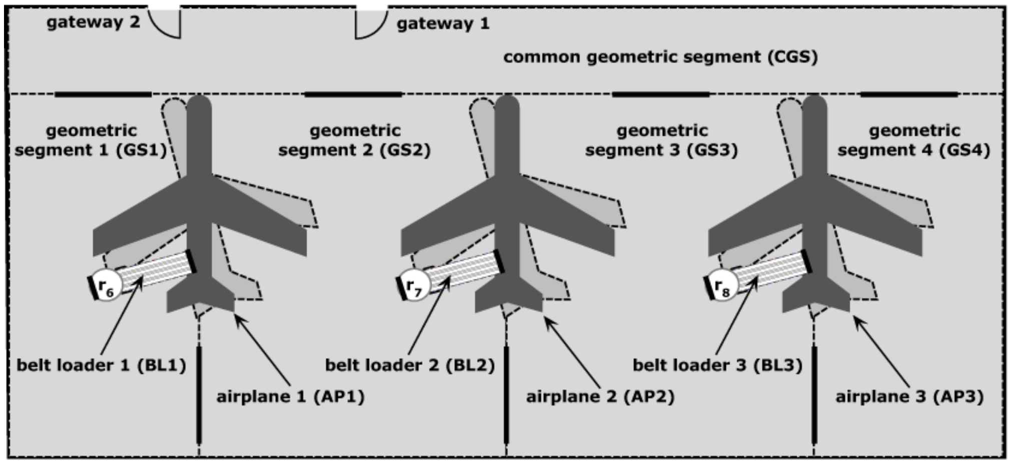


Figure 1: The O-space plan in Aalborg Airport apron

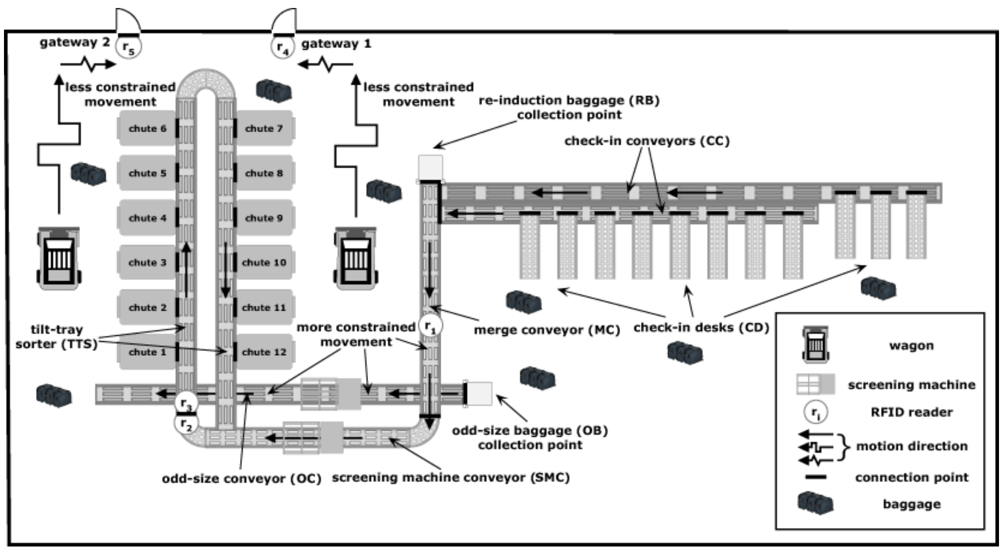


Figure 2: The I-space plan in Aalborg Airport hall

mentation carried out of the apron surface can be done in a variety of other ways. Moreover, the geometric segments themselves need not be line segments. Indeed and in contrast to [16], they can take any geometric shape as long as they are part of the apron surface, and that satisfactory interpretations are attached to them. These geometry-friendliness and invariance to the space plan segmentation add both depth and ample expressiveness to our modeling of O-spaces.

Identifying Connection Points We proceed in the second step to identify *"connection points"*. A connection point is simply an actual (movable/immovable) or virtual structure at which two or more semantic locations meet one another. The connection points are shown as bold line segments in Figures 1 and 2. We note that in reality the connection points between CD and CC are actual movable shutters. The two connection points between CH and CGS are the physical gateways 1 and 2. However, the connection point between SMC and TTS is virtual. Speaking of the notation of connection points, we differentiate between a single connection point between n semantic locations l_1,l_2, …,l_n that we denote as $l_1|l_2|\dots|l_n$, and n connection points between two semantic locations l_1 and l_2 that we denote as $(l_1|l_2)_1$, $(l_1|l_2)_2$, …,$(l_1|l_2)_n$.

In both cases, we do not mind the order in which the location symbols are listed. As a special case, a connection point can be referred to using a given name. For instance, the connection point $(\mathrm{CH}|\mathrm{CGS})_1$ is named $\mathrm{gateway}_1$. The identification of semantic locations and their connection points completes our grasp of the topology of an OI-space.

Identifying Moving Objects Receptor-based systems need to realize meaningful objects, not transponders such as RFID tags, or transducers such as motes, and motion detectors. Therefore, the third step in modeling an OI-space is to identify *"moving objects"*. A moving object is a living/nonliving mobile entity, to which a transponder/transducer is affixed, and whose motion reflection is crucial to the application. In the OI-space plans shown in Figures 1 and 2, moving objects are bags to which RFID tags are attached, and the RFID-based application is concerned with the identification and location of these bags.

Identifying Routes Having an OI-space plan imposes routes on objects as they move within semantic locations. Thus, the fourth step in modeling an OI-space is to identify *"routes"*. A route is a particular way moving objects follow (or are carried over) between semantic locations. Revisiting Figures 1 and 2, a route of bags is; $\mathrm{CD} \rightarrow \mathrm{CC} \rightarrow \mathrm{MC} \rightarrow \mathrm{SMC} \rightarrow \mathrm{TTS}$ (repeatedly in the general case) $\rightarrow \mathrm{CH} \rightarrow \mathrm{CGS} \rightarrow \mathrm{GS1} \rightarrow \mathrm{BL1} \rightarrow \mathrm{AP1}$. For convenience of modeling, we cut up routes into binary sub-routes, each of which constitutes an ordered pair, such as $(\mathrm{CD}, \mathrm{CC})$ in the baggage handling example. The identification of moving objects and routes completes our grasp of the dynamics of an OI-space.

Constructing the OI-Space Pseudograph Let $\mathcal{W}_l$, $\mathcal{W}_c$, $\mathcal{W}_o$, and $\mathcal{W}_m$ be the sets of semantic locations, connection points, moving objects, and binary sub-routes. Given that both sets $\mathcal{W}_l$ and $\mathcal{W}_m$ are finite and assuming that $\mathcal{W}_l$ is further nonempty. We use a directed graph $\mathcal{D}_{\textit{oi-space}} = (\mathcal{W}_l, \mathcal{W}_m, c)$ to model an OI-space [3] adding $\mathcal{W}_o$ as an explanatory symbol on its pictorial drawing. The sets $\mathcal{W}_l$ and $\mathcal{W}_m$ are respectively called the vertex and arc sets. The mapping $c : \mathcal{W}_m \rightarrow \mathcal{P}(\mathcal{W}_c)$ assigns labels to the arcs in $\mathcal{D}_{\textit{oi-space}}$ where $\mathcal{P}(\mathcal{W}_c)$ is the first-order power set of $\mathcal{W}_c$. The direction of an arc in $\mathcal{D}_{\textit{oi-space}}$ indicates the order of the corresponding binary sub-route. We allow $\mathcal{D}_{\textit{oi-space}}$ to include looping arcs whose head and tail coincide (such as (l_1, l_1)) and arcs with the same end-vertices (such as (l_1, l_2) and (l_2, l_1)); however, we do not allow multiple arcs with the same tail and head (such as (l_1, l_2) twice). Multiple arcs are avoided because there is no point in modeling the same binary sub-routes multiple times. The imposed restrictions characterize $\mathcal{D}_{\textit{oi-space}}$ as a labeled directed pseudograph without multiple arcs. Figure 3 shows the OI-space pseudograph of the OI-space plans shown earlier in Figures 1 and 2.

Modeling Granularity and Alternative Interpretations The OI-space pseudograph we built is sufficiently flexible to enable us to control the granularity or the extent to which we capture the details of the physical world. Our modeling can be made finer using the essential operations of splitting a vertex and subdividing an arc, whereas it can be made coarser using the opposite operations of set- and path- contractions [3]. Consider for example the OI-space pseudograph shown in Figure 3. If we are interested in including more details about the chutes, we split the vertex CH into 12 vertices, and subdivide the arc $(\mathrm{TTS}, \mathrm{CH})$ into 12 arcs to obtain the sub-pseudograph shown in Figure 4a.

The flexibility of our pseudograph goes beyond granularity control to enable us to interpret a connection point as a semantic location. This could be beneficial if we wish to extend probabilistic reasoning to that point (see Section 5). Returning to the OI-space pseudograph shown in Figure 3, we may wish to think of the gateways 1 and 2 as semantic locations and this is possible. We simply convert them into vertices and assume virtual connection points between them and the surrounding semantic locations. The sub-pseudograph in Figure 4b illustrates the outcome.

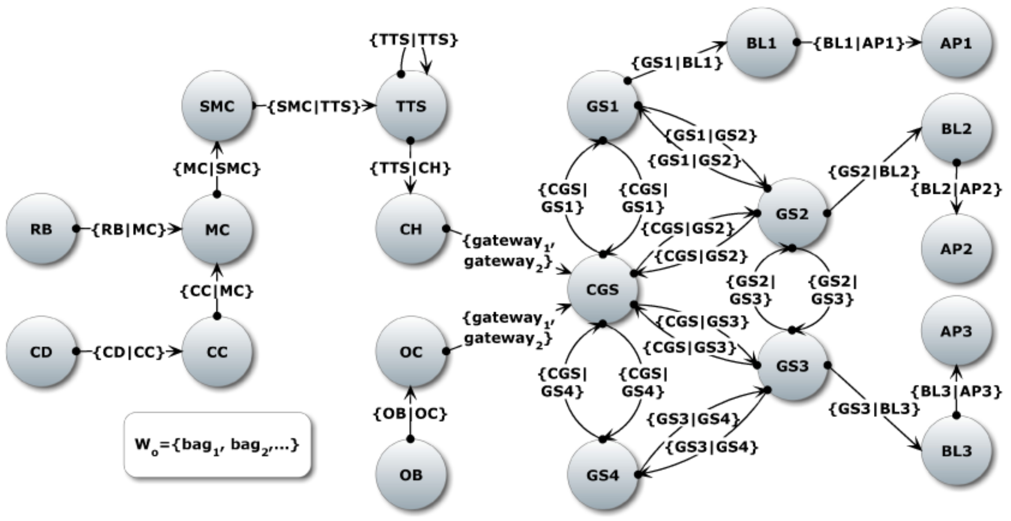
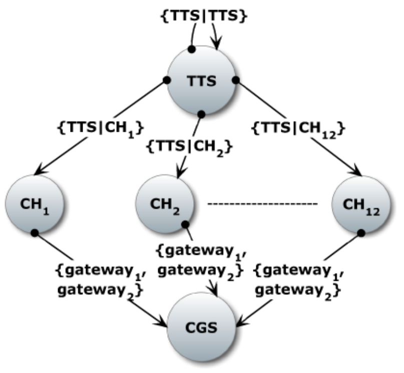
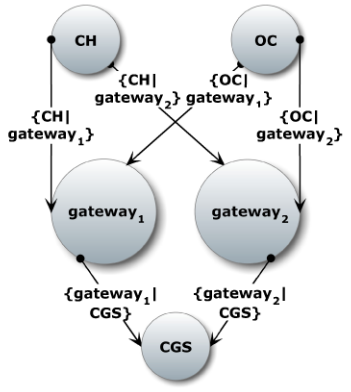


Figure 3: The OI-space pseudograph of the OI-space plans shown in Figures 1 and 2


{TTS|TTS}
TTS
{TTS|CH$_1$}
{TTS|CH$_2$}
{TTS|CH$_{12}$}
CH$_1$
CH$_2$
CH$_{12}$
{gateway$_1$, gateway$_2$}
{gateway$_1$, gateway$_2$}
{gateway$_1$, gateway$_2$}
CGS

(a)



(b)

Figure 4: Figure (a) shows how we can increase the fineness of the pseudograph in Figure 3; and Figure (b) shows alternative interpretation of the gateways in Figure 3 too

3 Modeling an RFID Readers Deployment

As a case study on receptor-based systems, we address the modeling of an RFID readers deployment in an OI-space. Modeling the deployments of WSNs and motion detectors in OI-spaces should not differ from that. In fact, several schemes for integrating the RFID technology into WSNs are being used [5].

Two components are always present in any RFID-based system; the tag, which is located on the object we wish to identify, and the reader. Each reader has a reading zone that varies between 0-5 meters [5]. An RFID tag is only activated when it is within the reading zone of a reader. In the RFID readers deployment under consideration, we assume that RFID tags are passive, that is they do not have any power supply.

A possible RFID readers deployment for our running baggage handling example is shown in Figures 1 and 2. It is worth pointing out in these figures that reader positioning varies with respect to semantic locations and their connection points. However, we do not impose any influence on reader positioning in order to avoid the loss of modeling generality. More importantly, we do not assume that reader positioning yields any division of the OI-space in which the deployment is done. Partition readers divide an I-space in [9] into what the authors call "cells" where each cell is simply a combination of a number of semantic locations. The tracking of moving objects, that is based on this division, reports object locations in terms of cells, which does not make much sense to RFID-based applications as we emphasized in Section 2. For RFID-based applications to attain the maximum usefulness of the tracking results, these results are better reported in terms of semantic locations.

To model our example RFID readers deployment, we transform $\mathcal{D}_{oi\text{-}space}$, we constructed in Section 2, by modifying the arc labels and introducing labels and weights into vertices. Let $\mathcal{W}_l$, $\mathcal{W}_c$, $\mathcal{W}_o$, and $\mathcal{W}_m$ be as annotated in Section 2. We drop $\mathcal{W}_c$ from the RFID modeling, and add the set $\mathcal{W}_r$ of RFID readers in the deployment instead. Additionally, we drop the mapping $c : \mathcal{W}_m \to \mathcal{P}(\mathcal{W}_c)$ adding instead three new mappings $c_l : \mathcal{W}_l \to \mathcal{P}(\mathcal{W}_r)$, $c_m : \mathcal{W}_m \to \mathcal{P}(\mathcal{W}_r) \cup \mathcal{P}(\mathcal{W}_r \cdot \mathcal{W}_r)$, and $c_r : \mathcal{W}_l \to (\mathcal{W}_r \to [0, 1])$ for labeling the vertices and arcs, and specifying the coverage weights of RFID readers among semantic locations respectively. Notice that the coverage weight mapping is effectively a mapping to a mapping in the sense that it maps any semantic location $l \in \mathcal{W}_l$ to a set of assignments each of which specifies the coverage weight of a reader $r \in \mathcal{W}_r$ whose reading zone is joint (overlapping/nested) with l. Making this concrete, we write:

$$c_r : \mathcal{W}_l \to w; w : \mathcal{W}_r \to [0, 1]$$
$$c_r(l) = \{r \to w(r) : r \cap l \neq \emptyset\}$$

We also find it convenient to use the simplifying notation:

$$\overline{c_r(l)} = \sum_{w(r) \in c_r(l)} w(r) \tag{1}$$

These nearly effortless readjustments transform $\mathcal{D}_{oi\text{-}space}$ into $\mathcal{D}_{rfid} = (\mathcal{W}_l, \mathcal{W}_m, c_l, c_m, c_r)$ with $\mathcal{W}_o$ similarly added as an explanatory symbol on $\mathcal{D}_{rfid}$'s pictorial drawing. $\mathcal{D}_{rfid}$ is characterized as a labeled, vertex-labeled, and vertex-weighted directed pseudograph without multiple arcs. The assignment of c_l and c_m labels is controlled by the readers positioning in the deployment with respect to semantic locations and their connection points as specified in the following guidelines:

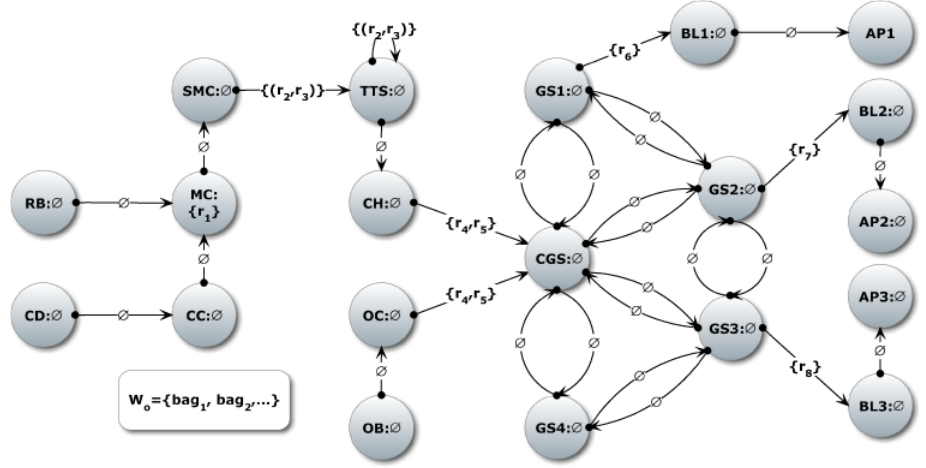
G1. If a reader is positioned inside a semantic location away from any connection point, then add this reader to the label set of this semantic location.

G2. If a reader is positioned at a connection point between semantic locations, then add this reader to the label set of the arcs connecting these locations.

G3. If two readers are adjacently positioned at a connection point between semantic locations, then add these two readers as an ordered pair to the label set of the arcs connecting these locations.

Observe the generality and completeness of G2 and G3 in that they permit the handling of a single connection point between any number of semantic locations, and any number of connection points between two semantic locations. The reader can easily verify this fact. Another important thing to notice in G3 is that labeling an arc via an ordered pair enables the capturing of the motion direction across the connection point by merely looking at the RFID readings sequence. One final thing we accentuate in G3 is that it permits and equally handles joint (overlapping/nested) and disjoint reading zones begotten by adjacent positioning of two readers. The coverage weights c_r can be easily approximated based on the readers deployment in the OI-space.

Let us return to Figures 1 and 2 and experiment with c_l, c_m, and c_r assignments. The reader r_1 is positioned inside MC away from any connection point. Therefore r_1 is added to the label set of MC i.e., $r_1 \in c_l(\mathrm{MC})$. Moreover, $c_r(\mathrm{MC}) = (r_1 \to 1)$. On the other hand, r_4 is positioned at $\mathrm{gateway}_1$ which leads to adding r_4 to the label sets of $(\mathrm{CH}, \mathrm{CGS})$ and $(\mathrm{OC}, \mathrm{CGS})$ i.e., $r_4 \in c_m(\mathrm{CH}, \mathrm{CGS})$ and



l	$c_r(l)$	l	$c_r(l)$
CD, CC, CC, RB, OB, AP1, AP2, AP3	$\emptyset$	GS1	$\{r_6 \rightarrow 0.1\}$
MC	$\{r_1 \rightarrow 1\}$	GS2	$\{r_7 \rightarrow 0.1\}$
SMC	$\{r_2 \rightarrow 0.8, r_3 \rightarrow 0.2\}$	GS3	$\{r_8 \rightarrow 0.1\}$
TTS	$\{r_2 \rightarrow 0.2, r_3 \rightarrow 0.8\}$	BL1	$\{r_6 \rightarrow 0.9\}$
OC	$\{r_4 \rightarrow 0.95, r_5 \rightarrow 0.95\}$	BL2	$\{r_7 \rightarrow 0.9\}$
CH	$\{r_4 \rightarrow 0.95, r_5 \rightarrow 0.95\}$	BL3	$\{r_8 \rightarrow 0.9\}$
CGS	$\{r_4 \rightarrow 0.05, r_5 \rightarrow 0.05\}$		

Figure 5: The RFID readers deployment pseudograph of the OI-space pseudograph shown in Figure 3

$r_4 \in c_m(\mathrm{OC}, \mathrm{CGS})$. Furthermore, $c_r(\mathrm{CH}) = c_r(\mathrm{OC}) = (r_4 \rightarrow 0.95, r_5 \rightarrow 0.95)$ and $c_r(\mathrm{CGS}) = (r_4 \rightarrow 0.05, r_5 \rightarrow 0.05)$. Finally, r_2 and r_3 are adjacently positioned at $\mathrm{SMC}|\mathrm{TTS}$, and r_2 reads before r_3 when moving from SMC to TTS across $\mathrm{SMC}|\mathrm{TTS}$. Thus, $(r_2, r_3) \in c_m(\mathrm{SMC}, \mathrm{TTS})$, $c_r(\mathrm{SMC}) = (r_2 \rightarrow 0.8, r_3 \rightarrow 0.2)$, and $c_r(\mathrm{TTS}) = (r_2 \rightarrow 0.2, r_3 \rightarrow 0.8)$.

Figure 5 shows the RFID readers deployment pseudograph of the OI-space pseudograph shown in Figure 3. For clarity, we do not include the coverage weights in the vertices, listing them instead in a table underneath the pseudograph. A generic algorithm for constructing the RFID readers deployment pseudograph is given in Algorithm 1. Algorithm 1 is divided into five stages. In stage 1, all the vertices and arcs are copied verbatim from $\mathcal{D}_{oi\text{-}space}$ to $\mathcal{D}_{rfid}$. The mappings c_l and c_m are initialized in stage 2 by assigning $\emptyset$ to the labels of all vertices and arcs. In stage 3, labels are assigned to all the vertices following the reasoning prescribe in G1. G2 and G3 are used to assign labels to all the arcs in stage 4. Finally, the coverage weights of RFID readers are approximated in stage 5. Notice how we preserve the efficiency of Algorithm 1 in stage 3 by removing the readers that were successfully processed by G1 from $\mathcal{W}_r$. This removal is justified since we really do not expect a reader to be positioned inside more than one semantic location, neither do we expect it to be simultaneously positioned inside a semantic location and at a connection point. However, we do not do the same in stage 4 since it is possible for more than one binary sub-route to have shared elements in their labels (refer to Figure 3).

4 Route Observability

Some physical approaches to RFID deployment in OI-spaces attempt to correct RFID anomalies (wrong, duplicate, and missed RFID readings) by enhancing the environment wherein the reading is conducted, which can be attained through installing multiple readers in order to cover a more substantial amount of an OI-space [1]. These approaches necessitate the study of route observability in OI-spaces which is the focus of this section.

Algorithm 1 Constructing the RFID readers deployment pseudograph

```
    *Input:* An OI-space pseudograph $\mathcal{D}_{\textit{oi-space}} = (\mathcal{W}_l, \mathcal{W}_m, c)$ where $c : \mathcal{W}_m \to \mathcal{P}(\mathcal{W}_c)$, and the set $\mathcal{W}_r$ of RFID readers in a possible deployment.
    *Output:* The RFID readers deployment pseudograph $\mathcal{D}_{\textit{rfid}} = (\mathcal{W}_l, \mathcal{W}_m, c_l, c_m, c_r)$ where $c_l : \mathcal{W}_l \to \mathcal{P}(\mathcal{W}_r)$, $c_m : \mathcal{W}_m \to \mathcal{P}(\mathcal{W}_r.\mathcal{W}_r)$ and $c_r : \mathcal{W}_l \to (\mathcal{W}_r \to [0,1])$.
    // Stage 1. Copying stage.
1:  copy $\mathcal{W}_l$ from $\mathcal{D}_{\textit{oi-space}}$ to $\mathcal{D}_{\textit{rfid}}$
2:  copy $\mathcal{W}_m$ from $\mathcal{D}_{\textit{oi-space}}$ to $\mathcal{D}_{\textit{rfid}}$
    // Stage 2. Initialization stage.
3:  **for all** $l \in \mathcal{W}_l$ **do**
4:      $c_l(l) = \emptyset$
5:  **for all** $m \in \mathcal{W}_m$ **do**
6:      $c_m(m) = \emptyset$
    // Stage 3. Vertex labeling stage.
7:  **for** each $l \in \mathcal{W}_l$ **do**
8:      **for** each $r \in \mathcal{W}_r$ **do**
9:          **if** $r$ is positioned in $l$ away from any connection point
            **then**                                                        ▷ G1
10:             $c_l(l) = c_l(l) \cup \{r\}$
11:             $\mathcal{W}_r = \mathcal{W}_r \backslash \{r\}$
    // Stage 4. Arc labeling stage.
12: **for** each $cp = l_1|l_2|\ldots|l_n \in \mathcal{W}_c.l_1, l_2, \ldots l_n \in \mathcal{W}_l$ **do**
13:     **for** each $r, r' \in \mathcal{W}_r$ **do**
14:         **if** $r$ is positioned at $cp$ **then**                        ▷ G2
15:             **for** each $m = (l_i, l_j) \in \mathcal{W}_m.i, j \in [1, n]$ **do**
16:                 $c_m(m) = c_m(m) \cup \{r\}$
17:         **if** $r, r'$ are adjacently positioned at $cp$ **then**        ▷ G3
18:             **for** each $m = (l_i, l_j) \in \mathcal{W}_m.i, j \in [1, n]$ **do**
19:                 **if** $r$ reads before $r'$ when moving from $l_i$ to $l_j$
                    across $cp$ **then**
20:                     $c_m(m) = c_m(m) \cup \{(r, r')\}$
21:                 **else**
22:                     $c_m(m) = c_m(m) \cup \{(r', r)\}$
    // Stage 5. Vertex weighing stage.
23: **for** each $l \in \mathcal{W}_l$ **do**
24:     approximate $c_r(l)$
```

Equipped with the graph terminology we have introduced to this point, we can make the definition of a route, given in Section 2, more formal by saying that a route $R = (l_1 \dots l_k)$ in $\mathcal{D}_{rfid}$ is an alternating sequence of semantic locations and binary sub-routes from $\mathcal{D}_{rfid}$. We denote the sets of semantic locations and binary sub-routes in R by $\mathcal{V}(R)$ and $\mathcal{A}(R)$ respectively. Subsequently we may write $R = (\mathcal{V}(R), \mathcal{A}(R))$. Let us now denote the route observability function as follows:

$$\Gamma : [0, 1] \to [0, \infty)$$

To measure the observability in a meaningful way, we would like the function Γ to satisfy the following properties:

P1. Nonnegativity: $\forall l \in \mathcal{V}(R) : \forall w(r) \in c_r(l) : \Gamma(w(r)) \geq 0$.

P2. Increasing monotonicity: $\forall l \in \mathcal{V}(R) : \forall w(r_1), w(r_2) \in c_r(l) : w(r_1) \leq w(r_2) \Rightarrow \Gamma(w(r_1)) \leq \Gamma(w(r_2))$.

P3. Normalization: If the whole coverage of a reader r is contained within a semantic location l, then the observability should be 1, that is $\forall l \in \mathcal{V}(R) : \forall w(r) \in c_r(l) : w(r) = 1 \Rightarrow \Gamma(w(r)) = 1$.

Intuitively, a route observability should be expressed by an increasing function of the coverage weights: the higher these weights, the higher the observability. This justifies including P2, and makes P1 convenient to have. The property P3 is a requirement for the measurement unit and it can be modified accordingly. 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 a route observability. Our function becomes (here and hereafter, all logarithms are to the base 2):

$$\Gamma(w(r)) = \log(w(r) + 1)$$

One more desirable property is 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.

In order to satisfy P4, we sum for all $w(r) \in c_r(l)$, and then we wrap the result in the expected value function for all $l \in \mathcal{V}(R)$, which yields our novel route observability function:

$$obs(R) = \sum_{l \in \mathcal{V}(R)} \sum_{w(r) \in c_r(l)} w(r) \log(w(r) + 1)$$

The obs function has solid bounds that we seek in Theorem 1.

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

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

Proof 1 *The proof is given in Appendix A.1.*

Revisiting the RFID readers deployment pseudograph (shown in Figure 5) for our baggage handling example, some routes, their observabilities, and bounds are as shown in Table 1. Based on the observability values, we can adjust the RFID readers deployment in order to better the reading environment thus reducing or even eliminating the occurrence of RFID anomalies.

Table 1: Some routes, their observabilities, and bounds in the baggage handling example

R	(CD...AP1)	(RB...AP2)	(OB...AP3)
$obs(R)$	5.1468	5.1468	2.6848
$bounds(R)$	$[0, 8.2673]$	$[0, 8.2673]$	$[0, 4.0974]$

5 Static Model-Based Reasoning

Since our models of an OI-space is graph-based, we can rely on the fundamentals of graph theory to perform high-level reasoning and gain a better insight into the physical world. The model-based reasoning, given in this section, deals, for the first time, with the concepts of bottleneck point in a static time-independent fashion. However, the dynamic nature of moving objects in RFID-based systems that evolves over time makes it useful to model time explicitly. We will therefore upgrade our static model-based reasoning in Section 7 by incorporating timestamped RFID data streams. Whether static or dynamic, our reasoning is uncertain in that it relies on probability theory to capture a support degree (between 0 and 1) summarizing our uncertainty that a semantic location is a bottleneck point [15].

A Bottleneck Point (BP) is a semantic location in an OI-space where there is potentially a lot of traffic. The estimation of these locations has strategic value in various RFID-based applications - baggage handling, production line monitoring, car body construction, and mail and shipping are a few to mention. Given an OI-space pseudograph, we can postulate a static estimate about the BPs using the concept of a vertex pseudodegree. A vertex pseudodegree is the number of all arcs (including loops) whose head or tail is that vertex. We denote the pseudodegree of a vertex v as $d(v)$. A very basic result in graph theory tells us that the sum of pseudodegrees in a directed pseudograph $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ is twice the number of arcs in $\mathcal{D}$. Strictly speaking:

$$\sum_{v \in \mathcal{V}(\mathcal{D})} d(v) = 2|\mathcal{A}(\mathcal{D})| \tag{2}$$

Definition 1 formalizes our static estimate about the BPs.

Definition 1 *(Static Bottleneck Point Estimate) Given an OI-space pseudograph $\mathcal{D}_{\textit{oi-space}} = (\mathcal{W}_l, \mathcal{W}_m, c)$, we estimate the static support degree that $l \in \mathcal{W}_l$ is a BP by the ratio of l's pseudodegree to twice the number of arcs in $\mathcal{D}_{\textit{oi-space}}$. Formally speaking:*

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

The nature of our static BP estimate is revealed in Lemma 1.

Lemma 1 *Given an OI-space pseudograph $\mathcal{D}_{\textit{oi-space}} = (\mathcal{W}_l, \mathcal{W}_m, c)$, the static bottleneck point estimate E_{BP} is consistent as a probability distribution on a random variable with alphabet $\mathcal{W}_l$.*

Proof 2 *The proof is given in Appendix A.2.*

Revisiting the OI-space pseudograph (shown in Figure 3) for our baggage handling example, the semantic locations, their pseudodegrees, and our static estimates that they are BPs are as shown in Table 2. It is important to notice that the sorter loop is counted twice when deciding $d(\mathrm{TTS})$. Thus, CGS has the highest static support degree of being a BP, followed equally by $\mathrm{GS2}$ and $\mathrm{GS3}$, then by $\mathrm{GS1}$, and after it by $\mathrm{GS4}$ and TTS with equal static support.

Table 2: The semantic locations, their pseudodegrees, and our static estimates that they are BPs in the baggage handling example

l	CD	CC	MC	SMC	TTS	CH	RB
d	1	2	3	2	4	3	1
$60E_{BP}$	1	2	3	2	4	3	1
l	OB	OC	CGS	GS1	GS2	GS3	GS4
d	1	3	12	5	7	7	4
$60E_{BP}$	1	3	12	12	5	7	4
l	BL1	BL2	BL3	AP1	AP2	AP3	
d	2	2	2	1	1	1	
$60E_{BP}$	7	2	2	1	1	1	

6 Probabilistic Incorporation of RFID Data

This section deals with the probabilistic incorporation of timestamped RFID data streams which is essential to the upgrade we perform in Section 7 of our model-based reasoning. A probabilistic account of RFID data streams is crucial to compensate for missing information that is inherent in these streams. We start by giving a denotation for a raw RFID reading, and we follow that by describing an efficient method for storing RFID streams. A clear definition of a trajectory of a moving object is given thereafter, and is followed by our novel probabilistic trajectory-to-route translator.

RFID Reading Denotation Each reader in an RFID readers deployment detects RFID tags within its reading zone, and reports raw RFID readings at a specific read rate [5] which is influenced by the technology used and the configuration of the reader. A raw RFID reading is a tuple of the form $\mathrm{rd} \equiv \langle \text{tag-id}, \text{reader-id}, \text{time} \rangle$ which indicates that tag-id was detected by reader-id at timestamp time. The reader-id is a unique identifier for the RFID reader that detected tag-id, and the timestamp time is recorded on a discrete time axis, thereby following the common assumption on spatio-temporal indexing [4]. To further clarify our findings, we assume a one-to-one correspondence between tags and the moving objects to which they are affixed. This assumption enables us to denote a raw reading as $\mathrm{rd} \equiv \langle \text{obj-id}, \text{reader-id}, \text{time} \rangle$. An RFID data stream produced by all the readers in a deployment is then a stream of tuples $\mathrm{S} \equiv \langle \mathrm{rd}_1, \mathrm{rd}_2, \dots \rangle$.

Appearance of a Moving Object Minding the efficiency of our storage and query processing, we do not want to store and persistently manipulate raw RFID readings at the timestamp level. Instead we would like to record 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 we lift the level of raw RFID readings by employing a pre-processing module (the details of which can be found elsewhere [9]) that condenses these readings into appearance records each of which has the form $\mathrm{ar} \equiv \langle \text{obj-id}, \text{reader-id}, \text{s-time}, \text{e-time} \rangle$ where s-time and $\mathrm{e\text{-}time}$ are the start and end time of an appearance. We store these appearance records in a table $\mathrm{appear\text{-}table}$ that has the schema $\langle \mathrm{ar\text{-}id}, \mathrm{obj\text{-}id}, \mathrm{reader\text{-}\ id}, \mathrm{s\text{-}time}, \mathrm{e\text{-}time}\rangle$.

Trajectory of a Moving Object A trajectory [2] of a moving object o inside an RFID data stream S over a time period T is the sequence of appearance records whose detected object is o and detection time is in T. Put formally:

$$TR(o,\,T) = ar_1, ar_2, \ldots, ar_n : ar_i.obj\text{-}id = o \wedge [ar_i.s\text{-}time, ar_i.e\text{-}time] \subseteq T$$

For instance, the trajectory of bag_1 in our baggage handling example over the time period $[t_1, t_{65}]$ is $TR(bag_1, [t_1, t_{65}]) = ar_1, ar_2, \ldots, ar_8$. Table 3 lists the corresponding appearance records.

Table 3: The trajectory of bag_1 over the period $[t_1, t_{65}]$

ar-id	obj-id	reader-id	s-time	e-time
ar_1	bag_1	r_1	t_1	t_2
ar_2	bag_1	r_2	t_5	t_6
ar_3	bag_1	r_3	t_7	t_8
ar_4	bag_1	r_2	t_{11}	t_{12}
ar_5	bag_1	r_3	t_{13}	t_{14}
ar_6	bag_1	r_4	t_{25}	t_{35}
ar_7	bag_1	r_7	t_{61}	t_{65}

Probabilistic Trajectory-to-Route Translator Given a trajectory $TR(o,T)$ of a moving object o over a time period T (see Table 3), we would like to construct the probabilistic route that this object followed (or were carried over) between semantic locations over the same period. That we achieve through a probabilistic trajectory-to-route translator whose base is Algorithm 2. This algorithm assumes the existence of an intermediate table $\mathrm{inter\text{-}table}$ with the schema $\langle \mathrm{ar\text{-}id}, \mathrm{obj\text{-}id}, \mathrm{loc}, \mathrm{s\text{-}time}, \mathrm{e\text{-}time}\rangle$ where loc is a semantic location. It proceeds to populate the mentioned table with intermediate records corresponding to the appearance records in $TR(o,T)$. For each appearance record, the algorithm executes two translation stages. In stage 1, translation is carried out based on vertex labels whereas it is carried out based on arc labels in stage 2. Notice that the loop in stage 1 terminates at the completion of one insertion. However, the loop in stage 2 does not terminate under the same condition. The reason for this was given earlier, near the end of Section 3.

Applying Algorithm 2 to the trajectory of bag_1, given in Table 3, yields the records listed in Table 4. Noticeably the number of records in this table can be condensed into a smaller number of records (each of which defines a probability distribution on a random variable whose alphabet is $\mathcal{W}_l$) through the execution of a simple SQL query that updates the final probabilistic route table $\mathrm{prob\text{-}table}$ whose schema is $\langle \mathrm{obj\text{-}id}, \mathrm{prob\text{-}loc}, \mathrm{s\text{-}time}, \mathrm{e\text{-}time}\rangle$. The outcome is shown in Table 5.

7 Dynamic Model-Based Reasoning

Having incorporated RFID data and obtained the probabilistic routes of moving objects in Section 6, we can now perform an over-time upgrade of the static reasoning, presented earlier in Section 5, in order to obtain our dynamic BP estimate in Definition 2.

Algorithm 2 Probabilistic Trajectory-to-Route Translator

```
    Input: $\mathcal{D}_{rfid} = (\mathcal{W}_l, \mathcal{W}_m, c_l, c_m, c_r)$, and $TR(o, T)$.
    Output: Records in the inter-table.
1:  for each $ar_i \in TR(o, T)$ do
      // Stage 1. Translation based on vertex labels.
2:    while ($l \in \mathcal{W}_l$ and $ar_i.reader\text{-}id \notin c_l(l)$) do
3:      insert into inter-table values
        $(ar_i, ar_i.obj\text{-}id, l, ar_i.s\text{-}time, ar_i.e\text{-}time)$
      // Stage 2. Translation based on arc labels.
4:    while ($m = (l_i, l_j) \in \mathcal{W}_m : l_i, l_j \in \mathcal{W}_l$ and
          ($ar_i.reader\text{-}id \in c_m(m)$ or
          $(ar_i.reader\text{-}id, ar_{i-1}.reader\text{-}id) \in c_m(m)$)) do
5:      insert into inter-table values
        $(ar_i, ar_i.obj\text{-}id, l_i, ar_i.s\text{-}time, ar_i.e\text{-}time)$,
        $(ar_i, ar_i.obj\text{-}id, l_j, ar_i.s\text{-}time, ar_i.e\text{-}time)$
```

Table 4: The intermediate route of bag_1 over the period $[t_1, t_{65}]$

ar-id	obj-id	loc	s-time	e-time
ar_1	bag_1	MC	t_1	t_2
ar_2	bag_1	SMC	t_5	t_6
ar_2	bag_1	TTS	t_5	t_6
ar_2	bag_1	TTS	t_5	t_6
ar_2	bag_1	TTS	t_5	t_6
ar_3	bag_1	SMC	t_7	t_8
ar_3	bag_1	TTS	t_7	t_8
ar_3	bag_1	TTS	t_7	t_8
ar_3	bag_1	TTS	t_7	t_8
ar_4	bag_1	SMC	t_{11}	t_{12}
ar_4	bag_1	TTS	t_{11}	t_{12}
ar_4	bag_1	TTS	t_{11}	t_{12}
ar_4	bag_1	TTS	t_{11}	t_{12}
ar_5	bag_1	SMC	t_{13}	t_{14}
ar_5	bag_1	TTS	t_{13}	t_{14}
ar_5	bag_1	TTS	t_{13}	t_{14}
ar_5	bag_1	TTS	t_{13}	t_{14}
ar_6	bag_1	CH	t_{25}	t_{35}
ar_6	bag_1	CGS	t_{25}	t_{35}
ar_6	bag_1	OC	t_{25}	t_{35}
ar_6	bag_1	CGS	t_{25}	t_{35}
ar_7	bag_1	GS2	t_{61}	t_{65}
ar_7	bag_1	BL2	t_{61}	t_{65}

Definition 2 *(Dynamic Bottleneck Point Estimate) Given an R-FID readers deployment pseudograph $\mathcal{D}_{rfid} = (\mathcal{W}_l, \mathcal{W}_m, c_l, c_m)$, the* prob-table*, and a monitoring period T of a semantic location $l \in \mathcal{W}_l$, we estimate the dynamic support degree over T that l is a BP by the*

Table 5: The probabilistic route of bag_1 over the period $[t_1, t_{65}]$

obj-id	prob-loc	s-time	e-time
bag_1	$[\mathrm{MC} : 1]$	t_1	t_2
bag_1	$[\mathrm{SMC} : .25; \mathrm{TTS} : .75]$	t_5	t_6
bag_1	$[\mathrm{SMC} : .25; \mathrm{TTS} : .75]$	t_7	t_8
bag_1	$[\mathrm{SMC} : .25; \mathrm{TTS} : .75]$	t_{11}	t_{12}
bag_1	$[\mathrm{SMC} : .25; \mathrm{TTS} : .75]$	t_{13}	t_{14}
bag_1	$[\mathrm{CH} : .25; \mathrm{OC} : .25; \mathrm{CGS} : .5]$	t_{25}	t_{35}
bag_1	$[\mathrm{GS2} : .5; \mathrm{BL2} : .5]$	t_{61}	t_{65}

full joint probability distribution on all the random variables of the probabilistic records in the prob-table *whose detection time is joint (overlapping/nested) with T. Formally speaking:*

$$\forall l \in \mathcal{W}_l : E_{BP}^{T}(l) = Pr(o_1 \; at \; l, \ldots, o_n \; at \; l) : o_i \in \mathcal{W}_o$$

It is known that inference using full joint distributions has prohibitive time complexity [15]. This complexity can be coped with in two ways. First, the specification of a monitoring period T enables us to limit our inference to only a subset $P\text{-}REC(T)$ of the prob-table:

$$P\text{-}REC(T) = \{p\text{-}rec \in \text{prob-table} : [p\text{-}rec.s\text{-}time, p\text{-}rec.e\text{-}time] \cap T \neq \emptyset\}$$

Second, the absolute independence assertion, we impose on random variables, dramatically reduces the amount of information necessary to encode the full joint distribution by enabling the factoring of that distribution into separate, smaller distributions.

An algorithm for computing the dynamic BP estimates of all semantic locations over a monitoring period T is given in Algorithm 3. The probability tweaking parameter η, in the input to this algorithm, is fed as a percentage, and it specifies the quotient by which E_{BP}^{T} are modified in accordance with the detection time. The purpose of this modification is to (de)concentrate the dynamic support degree that a semantic location is a BP. The normalization function ψ normalizes E_{BP}^{T} generated by the algorithm from iterating over all the semantic locations by dividing each $E_{BP}^{T}(l)$ by the sum of all E_{BP}^{T}. This normalization transforms E_{BP}^{T} and ensures its consistency as a probability distribution on a random variable whose alphabet is $\mathcal{W}_l$. Capturing the dynamic BP estimates in a probability distribution facilitates comparing them with the static estimates that were also represented as a probability distribution in Section 5. Ensuring efficiency, Algorithm 3 starts by extracting $P\text{-}REC(T)$ from the prob-table (line 1). Then using η, the algorithm determines the amount of increase and decrease by which to modify $E_{BP}^{T}(l)$ (lines 2-3). Afterwards, and for each semantic location l in $\mathcal{W}_l$ (line 4), the algorithm initializes the corresponding $E_{BP}^{T}(l)$ to the number of records in $P\text{-}REC(T)$ in which l appears (line 5). Thereafter, and for each probabilistic record $p\text{-}rec$ in $P\text{-}REC(T)$ (line 6), the algorithm determines the detection time (line 7), and then modifies $E_{BP}^{T}(l)$ by the scalar probability in $p\text{-}rec$ while increasing/decreasing it by η proportionally to the detection time (lines 8-14). This is sensible - the longer the time period during which a moving object stays at a semantic location, the higher the likelihood that this object contributes to the traffic at that location and vice vera. A final normalization of E_{BP}^{T} using ψ transforms it into a probability distribution (line 15).

Suppose that the probabilistic route of bag_2 over the time period $[t_3, t_{28}]$ is as given in Table 6. Imagine further that bag_1 (whose probabilistic route is given in Table 5) and bag_2 are the only bags that are handled over the time period $[t_1, t_{65}] \cup [t_3, t_{28}] = [t_1, t_{65}]$. If we would like to estimate $E_{BP}^{[t_1,t_{11}]}$, we follow the steps in Algorithm 3 extracting $P\text{-}REC([t_1, t_{11}])$ from the prob-table to get the records shown in Table 7, and

Algorithm 3 Dynamic Bottleneck Point Estimate

```
    Input: $\mathcal{D}_{rfid} = (\mathcal{W}_l, \mathcal{W}_m, c_l, c_m, c_r)$, the prob-table, a monitoring period $T$, a probability tweaking parameter $\eta$, and a normalization function $\psi$ to $[0,1]$.
    Output: $\psi(E_{BP}^{T})$.
 1: extract $P\text{-}REC(T)$ from the prob-table
 2: $increase = 1.0 + \eta/100.0$
 3: $decrease = 1.0 - \eta/100.0$
 4: for each $l \in \mathcal{W}_l$ do
 5:     $E_{BP}^{T}(l) = |\{p\text{-}rec \in P\text{-}REC(T) : l \in p\text{-}rec\}|$
 6:     for each $p\text{-}rec \in P\text{-}REC(T)$ do
 7:         $t = p\text{-}rec.e\text{-}time - p\text{-}rec.s\text{-}time$
 8:         if $p\text{-}rec.prob(o\ at\ l) > 0$ then
 9:             $E_{BP}^{T}(l) = E_{BP}^{T}(l) \times p\text{-}rec.prob(o\ at\ l)$
10:             repeat $t$ times
11:                 $E_{BP}^{T}(l) = E_{BP}^{T}(l) \times increase$
12:         else
13:             repeat $t$ times
14:                 $E_{BP}^{T}(l) = E_{BP}^{T}(l) \times decrease$
15: return $\psi(E_{BP}^{T})$
```

then proceeding with the calculations given $\eta = 10\%$ to obtain:

$$\begin{aligned}
&E_{BP}^{[t_1,t_{11}]}(\text{MC}) = 2 \times .9^{10} \times 1.1^{2} = .8438\\
&E_{BP}^{[t_1,t_{11}]}(\text{SMC}) = 5 \times .25^{5} \times .9^{7} \times 1.1^{5} = .0038\\
&E_{BP}^{[t_1,t_{11}]}(\text{TTS}) = 5 \times .75^{5} \times .9^{7} \times 1.1^{5} = .9140\\
&E_{BP}^{[t_1,t_{11}]}(\text{CH or OC}) = .25 \times .9^{7} \times 1.1^{5} = .1926\\
&E_{BP}^{[t_1,t_{11}]}(\text{CGS}) = .5 \times .9^{7} \times 1.1^{5} = .3852\\
&E_{BP}^{[t_1,t_{11}]}(\text{rest of locations}) = 0 \times .9^{12} = 0
\end{aligned}$$

A final normalization of $E_{BP}^{[t_1,t_{11}]}$ yields respectively the values:

$$\langle .3333, .0015, .3610, .0761, .0761, .1521, 0\rangle$$

Table 6: The probabilistic route of bag_2 over the period $[t_3, t_{28}]$

obj-id	prob-loc	s-time	e-time
bag_2	[MC : 1]	t_3	t_4
bag_2	[SMC : .25; TTS : .75]	t_5	t_6
bag_2	[SMC : .25; TTS : .75]	t_7	t_8
bag_2	[CH : .25; OC : .25; CGS : .5]	t_{11}	t_{16}
bag_2	[GS1 : .5; BL1 : .5]	t_{26}	t_{28}

In this example, TTS has the highest dynamic support degree of being a BP in the OI-space, followed by MC, then by CGS, and after it by CH and OC with equal dynamic support. SMC comes last with the least dynamic support. The dynamic support for the rest of the semantic locations is zero, due to the complete absence of probabilistic records in those locations as seen in Table 7.

Table 7: $P\text{-}REC([t_1, t_{11}])$ extracted from the prob-table

obj-id	prob-loc	s-time	e-time
bag_1	[MC : 1]	t_1	t_2
bag_1	[SMC : .25; TTS : .75]	t_5	t_6
bag_1	[SMC : .25; TTS : .75]	t_7	t_8
bag_1	[SMC : .25; TTS : .75]	t_{11}	t_{12}
bag_2	[MC : 1]	t_3	t_4
bag_2	[SMC : .25; TTS : .75]	t_5	t_6
bag_2	[SMC : .25; TTS : .75]	t_7	t_8
bag_2	[CH : .25; OC : .25; CGS : .5]	t_{11}	t_{16}

8 Experimental Evaluation

We experiment with actual uncleansed data gathered from the RFID readers deployment in the hall of Aalborg Airport over the period between 2011-12-16 and 2012-04-17. The actual deployment in the hall differs from the one shown in Figure 2 in that a single reader is deployed at the connection point $(\mathrm{SMC}|\mathrm{TTS})$. The deployment of readers in the O-space of Aalborg Airport (shown in Figure 1) is planned, therefore RFID readings from that space are unavailable, and the outdoor semantic locations and readers are accordingly excluded from $\mathcal{D}_{rfid}$ shown in Figure 5. The pre-processing module, we mentioned in Section 6, enables us to reduce the number of raw RFID readings from around 6.75 million down to 180 thousand that we store in the appear-table. The overall number of RFID-tagged bags for which these readings are reported is around 71 thousand. All experiments are implemented in Java SE version 1.6.0_32 and MATLAB version 7.14, and the database 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, supporting a 64-bit installation of Microsoft Windows 7 Enterprise version 6.1.7601. We present two groups of experiments; the first concerns Algorithm 2, whereas the second analyzes Algorithm 3.

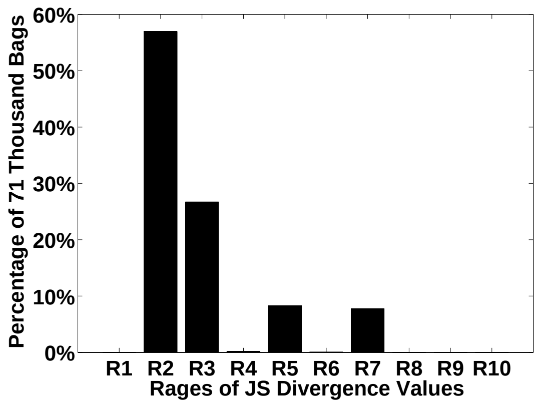
8.1 Analysis of Algorithm 2

Accuracy Evaluation In order to evaluate the accuracy of the translation done by Algorithm 2, we need to decide how far the translated distribution of moving objects is from the real one. We measure this distance via Jensen-Shannon (JS) divergence measure [13], which is defined by the formula:

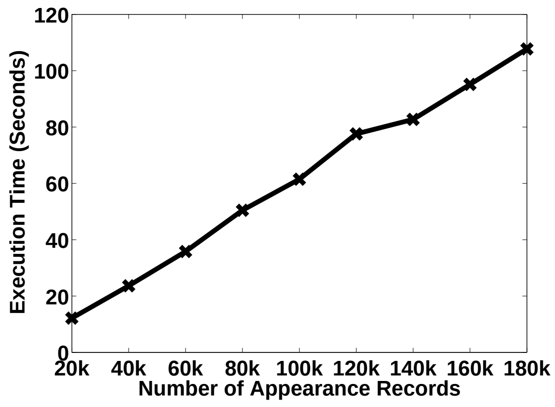
$$JS(p_1, p_2) = \sum_{x \in \mathcal{X}} p_1(x) \cdot \log \frac{p_1(x)}{\frac{p_1(x)+p_2(x)}{2}}$$

where p_1 and p_2 are two probability distribution functions on a discrete random variable X whose alphabet is $\mathcal{X}$. JS enjoys a number of salient properties that the commonly-used Kullback-Leibler (KL) divergence lacks [10, 20]. Of these properties, we mention the finiteness and boundness ($0 \leq JS \leq 1$). The reader is referred to [7] for a detailed comparison between these two divergence measures. Returning to our accuracy evaluation, the translated distribution of each bag can be easily identified by looking at the prob-table. The real distribution on the other hand 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, we generate synthetic RFID readings for all the 71 thousand 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 the hall of Aalborg Airport. This implies that wrong, duplicate, and missed RFID readings are not expected. Moreover, it means that each bag delivered at the check-in desks is properly handled until it is loaded into the designated airplane. We store this synthetic data in the $\mathrm{synth\text{-}prob\text{-}table}$ whose schema is identical to that of the $\mathrm{prob\text{-}table}$. We can then proceed to identify the real distribution of each bag, based on the synthetic data, and compute the JS divergence between it and the corresponding translated distribution. Thus, we have one JS divergence value per bag i.e., 71 thousand JS values in total. Due to this large number of JS values, we partition the known range $[0, 1]$ of JS into 10 smaller ranges, the length of each is 0.1, and then report the distribution of the 71 thousand bags across these ranges in Figure 6a. As seen in this figure, the translated distribution is generally no further than $[0.1, 0.2]$ (57% of the bags) to $[0.2, 0.3]$ (27% of the bags) from reality, which substantiates the accuracy of our probabilistic translator. The few and unusually high JS values for some bags (16% of them) is the result of severe corruption in their trajectories as prescribed in the actual uncleansed data. These values are by no means related to the accuracy of our translator.

Performance Evaluation In order to evaluate the performance of Algorithm 2, we vary the number of appearance records, input to this algorithm, and then plot the time the algorithm needs to populate the $\mathrm{inter\text{-}table}$ in Figure 6b. Clearly, the execution time increases with the increase of the processed records.



(a) JS



(b) Execution time

Figure 6: Figure (a) demonstrates the accuracy evaluation of Algorithm 2; and Figure (b) demonstrates the performance evaluation of the same algorithm

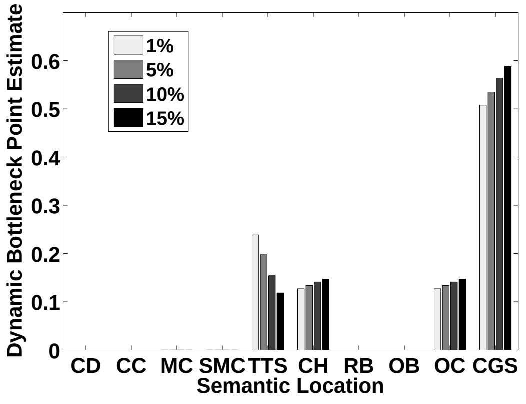
8.2 Analysis of Algorithm 3

First, we utilize our probabilistic translator, described in Section 6, on the full content of the appear -table in order to populate the $\mathrm{prob\text{-}table}$, input to Algorithm 3. Then we vary the input parameters to this algorithm as shown in Table 8. The number of probabilistic records corresponding to each setting appears within parentheses in the same table.

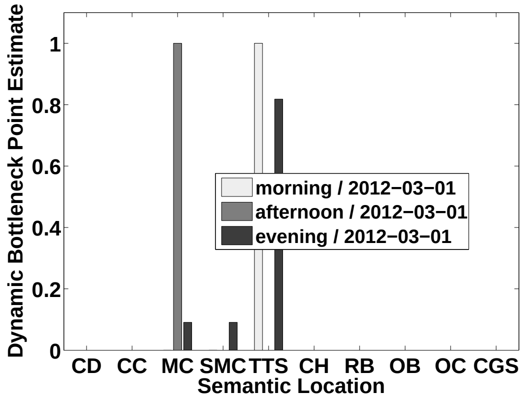
Table 8: The input parameters to Algorithm 3 and their values

Parameter	Values
Day	2012-01-15 (96), 2012-03-01 (85), 2012-04-15 (31)
T in minutes	10 (17), 20 (33), 30 (55), 40 (73)
η	1% (33), 5% (33), 10% (33), 15% (33)

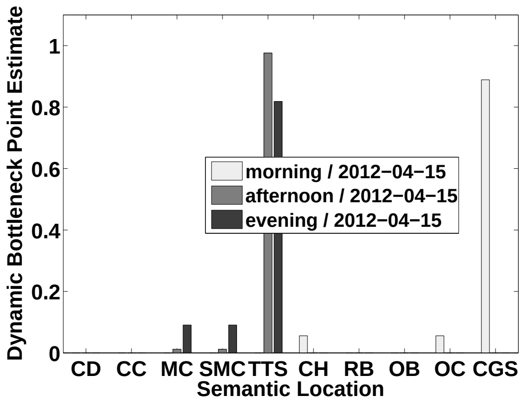
Effect of varying the day Figures 7a, 7b, and 7c show E_{BP}^{T} for 20 minutes in the morning, afternoon, and evening of the three days listed in Table 8. Notice in these figures that both MC and TTS have high dynamic support degrees of being BPs in the hall of Aalborg Airport irrespective of the day and its period. Occasionally, OC and CGS witness high dynamic support of being BPs. This seems to be in line with the results obtained earlier in Section 5 for the static BP estimates. Figure 7d shows that the execution time decreases, for all days, in the order morning, afternoon, and evening, which reflects a corresponding decrease in the hall traffic in Aalborg Airport. Recall that the execution time is mostly dependent on the number of probabilistic records over T.



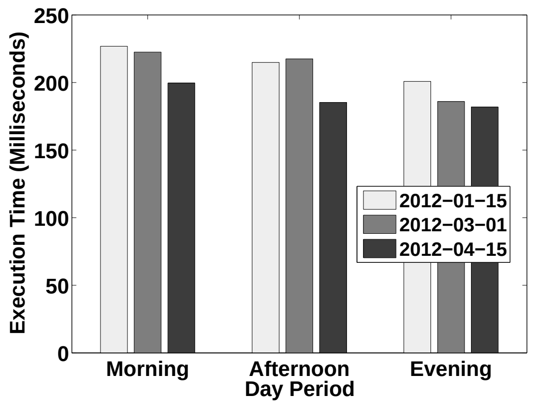
(a) E_{BP}^{T}



(b) E_{BP}^{T}



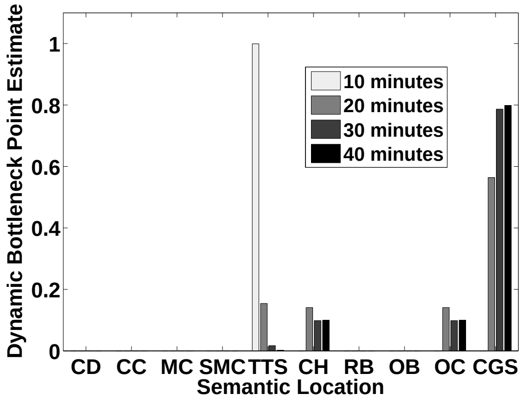
(c) E_{BP}^{T}



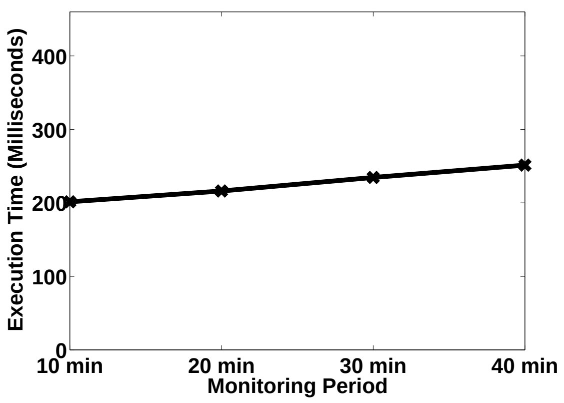
(d) Execution time

Figure 7: Figures (a), (b), and (c) show the effect of varying the day with $T = 20$ minutes, and $\eta = 10\%$; and Figure (d) shows the execution time for the former

Effect of varying T Results for varying T for one day, according to the values listed in Table 8, are shown in Figure 8a. The variation of E_{BP}^{T} in this figure does not follow a noticeable pattern. We 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 semantic location at the expense of others. Instead, this expansion might introduce support for semantic locations that were not supported as BPs prior to the expansion. Figure 8b tells us that the execution time increases proportionally to T, primarily due to the corresponding increase in the number of probabilistic records that Algorithm 3 processes.



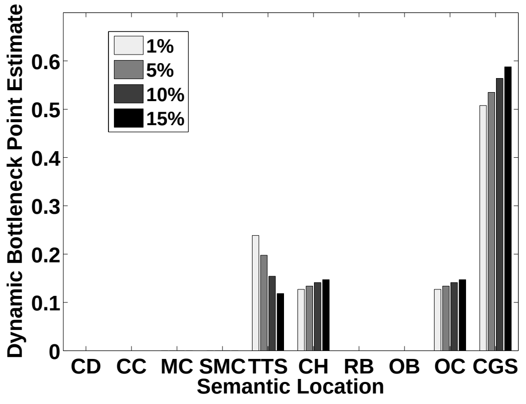
(a) E_{BP}^{T}



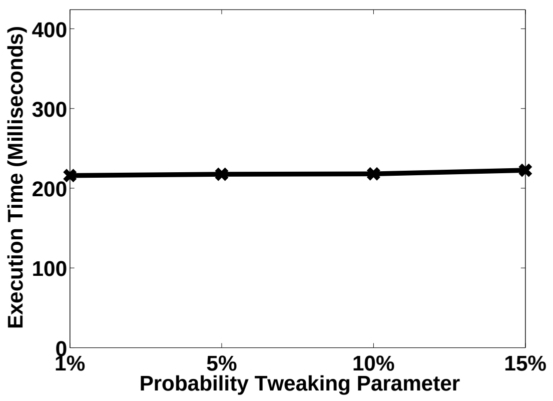
(b) E_{BP}^{T}

Figure 8: Figure (a) shows the effect of varying T with $\mathrm{day} = $ 2012-01-15, and $\eta = 10\%$; and Figure (b) shows the execution time for the former

Effect of varying η The impact of varying η (as prescribed in Table 8) on E_{BP}^{T} and the execution time is depicted in Figures 9a and 9b respectively. The slight fluctuation seen in these figures is only connected to the time the Java Virtual Machine needs, and the manner it handles the multiplication and rounding of floating-point numbers.



(a) E_{BP}^{T}



(b) Execution time

Figure 9: Figure (a) shows the effect of varying η with $\mathrm{day} = $ 2012-01-15, and $T = 20$ minutes; and Figure (b) shows the execution time for the former

Performance Evaluation It is also interesting to consider the execution time of Algorithm 3 when varying the number of probabilistic records in the input. That we show in Figure 10.

9 Related Work

Although it falls into several categories, related work has by far focused on the modeling of indoor spaces. An integrated indoor model is proposed in [6]. This model covers different information dimensions of

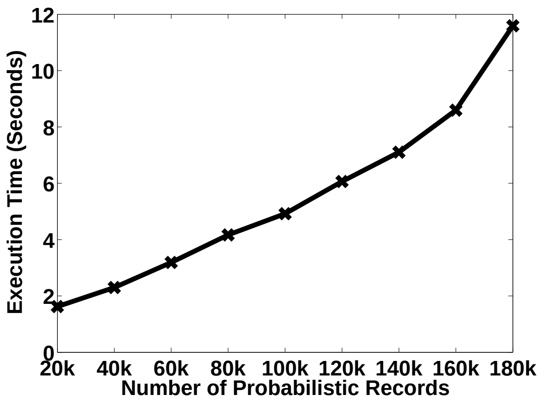


Figure 10: The performance evaluation of Algorithm 3

indoor models including thematic, geometric, and routing-related information. It is based on classifying indoor objects and structures while taking geometry, appearance, and semantics into account. A lattice-based location model for indoor navigation is introduced in [11]. The lattice construction is based on the theory of ”formal concept analysis”, and the model 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 is presented in [12]. The advantage of this model is that it combines the structural properties of indoor environments with the continuous metric properties that might be of interest to some applications. The model presented in [12] is employed in [17] for route analysis and evacuation planning. A distance-aware indoor space model is suggested in [14]. A set of indoor distance computation algorithms and an indexing framework accompany this model in order to enable the processing of indoor distance-aware queries over indoor spatial objects. Our work distinguishes itself from those aforementioned by capturing both O- and I-spaces in a unified model.

Bigraphs are used in [18] in order to model image schemas of built environments, which is claimed to be helpful in understanding the relationships between entities in these environments. The authors’ development of spatial reasoning and inference tools on top of their bigraphs is however ongoing. In contrast, we offer a self-contained set of modeling and reasoning techniques for O- and I-spaces in this paper.

The authors of [21] report the progress of their work on the development of a navigation ontology for outdoor and indoor environments, which 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. Our model differs from [21] as follows. First, our model is designed for reasoning about indoor moving objects rather than navigation which is the theme of [21]. Second, our model accommodates receptor deployments (such as RFID readers) which are not considered in [21].

10 Conclusions

We propose a unified model of outdoor and indoor spaces, and show that it is expressive, flexible, and invariant to the segmentation of a space plan. The model is focused on partially constrained outdoor and indoor motion. Adopting the RFID technology, we transform this model into an RFID readers deployment model that not only incorporates readers in a deployment but also the distribution of their coverage. We introduce clear representation of routes in our model, and rely on it and on some solid information-theoretic foundations in order to derive an important route observability function and establish its lower and upper bounds. We define the notion of a bottleneck point (BP), and rely on our models, and the fundamentals of graph theory in order to perform static reasoning about BPs in OI-spaces. Probabilistic incorporation of RFID data, that compensates for missing information pieces in this data, is carried out on top of our models using a trajectory-to-route translator. This incorporation enables reasoning about BPs in OI-spaces, under uncertainty, and in the presence of timestamped RFID data. The experimental evaluation, conducted on synthetic and uncleansed real-world data, validates our proposals. In particular, it substantiates the accuracy of the probabilistic incorporation done, and the sensibleness of the reasoning made about BPs.

References

[1] Y. Bai, F. Wang, and P. Liu. Efficiently filtering rfid data streams. In *CleanDB*, 2006.

[2] P. Bakalov and V. Tsotras. A generic framework for continuous motion pattern query evaluation. In *ICDE*, 2008.

[3] J. Bang-Jensen and G. Gutin. *Digraphs : theory, algorithms, and applications*. Springer, 2. ed. edition, 2009.

[4] S. Chen, B. C. Ooi, and Z. Zhang. An adaptive updating protocol for reducing moving object database workload. *PVLDB*, 3(1-2), 2010.

[5] K. Finkenzeller and D. Müller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. Wiley, 3rd ed. edition, 2010.

[6] B. Hagedorn, M. Trapp, T. Glander, and J. Dollner. Towards an indoor level-of-detail model for route visualization. In *MDM*, 2009.

[7] S. H. Hussein. A precise information flow measure from imprecise probabilities. In *SERE*, 2012. http://arxiv.org/abs/1206.5487.

[8] S. Jeffery, G. Alonso, M. Franklin, W. Hong, and J. Widom. A pipelined framework for online cleaning of sensor data streams. In *ICDE*, 2006.

[9] C. S. Jensen, H. Lu, and B. Yang. Graph model based indoor tracking. In *MDM*, 2009.

[10] S. Kullback and R. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 1951.

[11] D. Li and D. L. Lee. A lattice-based semantic location model for indoor navigation. In *MDM*, 2008.

[12] X. Li, C. Claramunt, and C. Ray. A grid graph-based model for the analysis of 2d indoor spaces. *Computers, Environment and Urban Systems*, 34(6), 2010.

[13] J. Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1), 1991.

[14] H. Lu, X. Cao, and C. S. Jensen. A foundation for efficient indoor distance-aware query processing. In *ICDE*, 2012.

[15] S. J. Russell and P. Norvig. *Artificial intelligence : a modern approach*. Pearson Education, 3. ed. edition, 2010.

[16] L. Speičys, C. S. Jensen, and A. Kligys. Computational data modeling for network-constrained moving objects. In *ACM GIS*, 2003.

[17] J. Sun and X. Li. Indoor evacuation routes planning with a grid graph-based model. In *Geoinformatics*, 2011.

[18] L. Walton and M. Worboys. An algebraic approach to image schemas for geographic space. In *Spatial Information Theory*, volume 5756. 2009.

[19] M. Worboys. Modeling indoor space. In *ISA*, 2011.

[20] J. Xie, J. Yang, Y. Chen, H. Wang, and P. Yu. A sampling-based approach to information recovery. In *ICDE*, 2008.

[21] L. Yang and M. Worboys. A navigation ontology for outdoor-indoor space: (work-in-progress). In *ISA*, 2011.

A Proofs

A.1 Proof of Theorem 1

The *absolute* lower bound $obs(R) \geq 0$ is a natural result of $w(r)$ falling in the range $[0,1]$. As for the *dynamic* upper bound, we note that for a convex function f and a random variable X, Jensen's inequality gives us:

$$Ef(X) \leq f(EX)$$

The function $x\log(x+1)$ is convex (i.e., it lies below any chord), therefore:

$$\begin{aligned} obs(R) &= \textstyle\sum_{l\in\mathcal{V}(R)} \sum_{w(r)\in c_r(l)} w(r)\log(w(r)+1) \\ &\leq \textstyle\sum_{l\in\mathcal{V}(R)} \log \sum_{w(r)\in c_r(l)} w(r)(w(r)+1) \\ &\leq \textstyle\sum_{l\in\mathcal{V}(R)} \log \sum_{w(r)\in c_r(l)} (w(r)+1) \quad (w(r)\in[0,1]) \\ &= \textstyle\sum_{l\in\mathcal{V}(R)} \log \left(\sum_{w(r)\in c_r(l)} w(r) + \sum_{w(r)\in c_r(l)} 1\right) \\ &= \textstyle\sum_{l\in\mathcal{V}(R)} \log \left(\sum_{w(r)\in c_r(l)} w(r) + |c_r(l)|\right) \\ &= \textstyle\sum_{l\in\mathcal{V}(R)} \log \left(\overline{c_r(l)} + |c_r(l)|\right) \quad \text{(Formula 1)} \end{aligned}$$

This gives us the bounds of $obs(R)$:

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

and proves the theorem.

A.2 Proof of Lemma 1

Proving the consistency of E_{BP} as a probability distribution is carried out in two steps. In the first step, we show that E_{BP} has proper bounds as follows.

$$\begin{aligned} &\textstyle\sum_{l\in\mathcal{W}_l} d(l) = 2|\mathcal{W}_m| \quad \text{(Formula 2)} \\ &\textstyle\sum_{l\in\mathcal{W}_l} \frac{d(l)}{2|\mathcal{W}_m|} = 1 \quad \text{(Division properties, } \mathcal{W}_\mathrm{m} \text{ is nonempty)} \\ &0 \leq \tfrac{d(l)}{2|\mathcal{W}_m|} \leq 1 : \forall l \in \mathcal{W}_l \quad \text{(Inequality properties)} \\ &E_{BP}(l) \in [0,1] : \forall l \in \mathcal{W}_l \quad \text{(Definition 1)} \end{aligned}$$

In the second step, we ensure that no intermediate value of E_{BP} falls outside the range $[0, 1]$ by showing that E_{BP} is a monotonically increasing function. In other words, we show that:

$$\forall l_1, l_2 \in \mathcal{W}_l : d(l_1) \leq d(l_2) \Rightarrow E_{BP}(l_1) \leq E_{BP}(l_2)$$

as follows:

$$\begin{array}{ll} d(l_1) \leq d(l_2) & \text{(Assumption)} \\ \frac{d(l_1)}{2|\mathcal{W}_m|} \leq \frac{d(l)}{2|\mathcal{W}_m|} & \text{(Division properties, } \mathcal{W}_\mathrm{m} \text{ is nonempty)} \\ E_{BP}(l_1) \leq E_{BP}(l_2) & \text{(Definition 1)} \end{array}$$

Thus, the static bottleneck point estimate E_{BP} is invariably consistent as a probability distribution.