The SIGSPATIAL Special

Newsletter of the Association for Computing Machinery
Special Interest Group on Spatial Information

Volume 9 Number 2 July 2017

The SIGSPATIAL Special

The SIGSPATIAL Special is the newsletter of the Association for Computing Machinery (ACM) Special Interest Group on Spatial Information (SIGSPATIAL).

ACM SIGSPATIAL addresses issues related to the acquisition, management, and processing of spatially-related information with a focus on algorithmic, geometric, and visual considerations. The scope includes, but is not limited to, geographic information systems.

For more details and membership information for ACM SIGSPATIAL as well as for accessing the newsletters please visit http://www.sigspatial.org.

The SIGSPATIAL Special serves the community by publishing short contributions such as SIGSPATIAL conferences' highlights, calls and announcements for conferences and journals that are of interest to the community, as well as short technical notes on current topics. The newsletter has three issues every year, i.e., March, July, and November. For more detailed information regarding the newsletter or suggestions please contact the editor via email at chiychow@cityu.edu.hk.

The SIGSPATIAL Special (ISSN 1946-7729) Volume 9, Number 2, July 2017.

Table of Contents

Message from the Editor

Chi-Yin Chow
Department of Computer Science, City University of Hong Kong, Hong Kong
Email: chiychow@cityu.edu.hk

In the first section, we have a special issue of some topic of interest to the SIGSPATIAL community. The topic of this issue is "Indoor Spatial Awareness (Part 2)" which is edited by our editor Dr. Chi-Yin Chow (Ted). Dr. Chow is currently an Assistant Professor in the Department of Computer Science, City University of Hong Kong.

The second section consists of one event report from:

1. The 10th ACM SIGSPATIAL Workshop on Geographic Information Retrieval (ACM SIGSPATIAL GIR 2016)

I would like to sincerely thank all the newsletter authors and event organizers for their generous contributions of time and effort that made this issue possible. I hope that you will find the newsletters interesting and informative and that you will enjoy this issue.

You can download all Special issues from:

`http://www.sigspatial.org/sigspatial-special`

.

The SIGSPATIAL Special

Section 1: Indoor Spatial Awareness (Part 2)

ACM SIGSPATIAL
http://www.sigspatial.org

Introduction to this Special Issue: Indoor Spatial Awareness (Part 2)

Chi-Yin Chow
Department of Computer Science, City University of Hong Kong
Email: chiychow@cityu.edu.hk

People spend large part of their time indoors, such as school buildings, office buildings, shopping malls, and public transportation centers. As indoor space is different from outdoor space, it is difficult to just employ location-based technologies and services designed for outdoor space to indoor space. Thus, it is necessary for scientists and researchers to develop new spatial and spatio-temporal data management and geographic information systems (GIS) theories, technologies and applications for indoor space. The mission of this special issue "Indoor Spatial Awareness (Part 2)" is to bring together scientists and researchers who work on different topics of indoor spatial awareness and to provide a venue for inspiring new research directions in all relevant aspects.

In the first article, Demetrios Zeinalipour-Yazti and Christos Laoudias give an overall of Anyplace, an indoor navigation service based on an open, modular, extensible and scalable navigation architecture and crowdsourced Wi-Fi data.

The next three contributions are related to indoor Radio Frequency Identification (RFID) tracking data. Tanvir Ahmed et al. design a data mining methodology for detecting risk factors to identify potential issues in the baggage management from RFID baggage tracking data. Shan-Yun Teng et al. propose a new mining framework to discover user visited behavior from indoor RFID data in mall environments. Bettina Fazzinga et al. present an approach to exploit integrity constraints from semantic RFID trajectory data for interpreting RFID data in the context of object tracking.

Last but not least, Moustafa Elhamshary and Moustafa Youssef envision a ubiquitous indoor spatial awareness system with two main components, namely, that can be deployed anywhere around the world, with minimum overhead, and that works with the heterogeneous Internet of Things (IoT) devices.

I hope the readers will enjoy reading this issue and find it useful in their research work.

The Anatomy of the Anyplace Indoor Navigation Service

Demetrios Zeinalipour-Yazti*‡ and Christos Laoudias*

* University of Cyprus, 1678 Nicosia, Cyprus
‡ Max Planck Institute for Informatics, Saarland Informatics Campus, 66123 Saarbrücken, Germany
{dzeina, laoudias}@ucy.ac.cy; dzeinali@mpi-inf.mpg.de

Abstract

The pervasiveness of smartphones is leading to the uptake of a new class of Internet-based Indoor Navigation (IIN) services, which might soon diminish the need of Satellite-based localization technologies in urban environments. These services rely on geo-location databases that store spatial models along with wireless, light and magnetic signals used to localize users and provide better power efficiency and wider coverage than predominant approaches. In this article we overview Anyplace, an open, modular, extensible and scalable navigation architecture that exploits crowdsourced Wi-Fi data to develop a novel navigation service that won several international research awards for its utility and accuracy (i.e., less than 2 meters). Our MIT-licenced open-source software stack has to this date been used by thaousands of researchers and practitioners around the globe, with the public Anyplace service reaching over 100,000 real user interactions.

1 Introduction

The omni-present availability of sensor-rich smartphones along with the fact that people spend 80-90% of their time in indoor environments has recently boosted an interest around indoor location-based services, such as, in-building guidance and navigation, inventory management, marketing and elderly support through Ambient and Assisted Living [3, 2]. The key enablers for the uptake of such indoor applications are nowadays, what we call the *Internet-based Indoor Navigation (IIN)* [13] services. These comprise of indoor models, such as floor-maps and *Points-of-Interest (POIs)*, along with wireless, light and magnetic signals used to localize users. As shown in [13], there is a rich spectrum of different types of services, but none of them provides infrastructure-free localization combined with rich modeling, crowdsourcing and privacy elements under the same hood. More importantly, none of them is freely available as an open source project limiting in this way the wide adoption of important scientific findings but also limiting transparency of happens behind the service.

In this overview paper we summarize the current developments around Anyplace[1], our open, modular, scalable and extensible architecture that collects indoor information using crowdsourcing. We follow a multi-tier architecture that allows to plug-n-play additional modules, either for extending system capabilities by implementing new features, or for enhancing user-experience by improving existing functionalities (e.g., map-matching and sophisticated data fusion to increase localization accuracy). Regarding scalability, Anyplace operates on top of a NoSQL data management back-end service, a JSON Application Protocol Interface (API), mobile clients for Web, Android and Windows Phone and a seamless integration with the Google Maps API for outdoor navigation and search, tiles and sattelite view. Anyplace has become an open-source project and is openly distributed

[1] Anyplace. `https://anyplace.cs.ucy.ac.cy/`

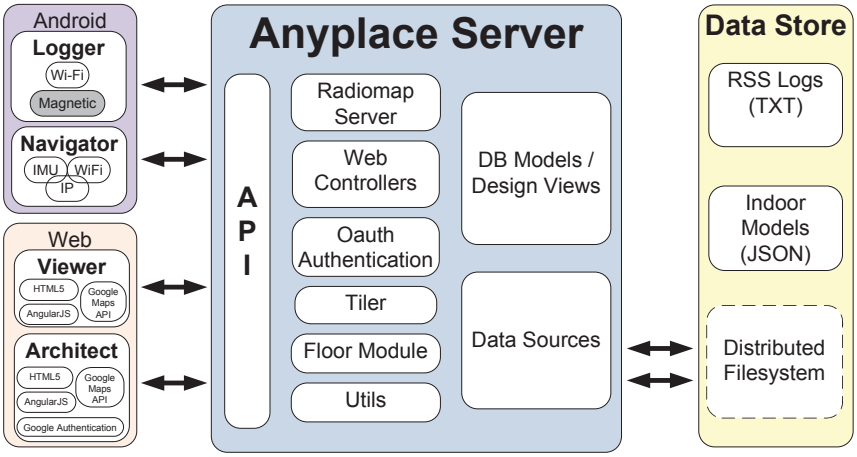


Figure 1: **The *Anyplace* Internet-based Indoor Information (IIN) Service Architecture.**

on Github using an MIT licence[2]. The public Anyplace service has to this date been obtained more than 100,000 real user interactions, with many more users using its standalone installations on the Web.

2 Overview of Anyplace

The Anyplace software stack consists of five main modules, including the *Server*, the *Data Store*, the *Architect*, the *Viewer* and two client applications running on Android smartphones, namely the *Logger* and the *Navigator*. A native Navigator is also available for Windows Phone. The Anyplace system architecture is shown in Figure 1.

The Anyplace *Server* contains the complete backend application logic of the service, including the modeling, crowdsourcing and API functionality. It is implemented in the Play Framework 2.2.x., which provides a lightweight, stateless and web-friendly architecture to build web applications. The *Server* delivers indoor navigation directions and information search and exploration queries through the JSON API. In addition, the *Server* features several modules that facilitate the crowdsourcing functionality, tiling of images uploaded, authentication of users and the interface to the design views of the data store. The Anyplace *Data Store* stores the indoor models and the collected Wi-Fi and other signals on storage.

The Anyplace *Architect* is a *Web App* (*HTML5, CSS3, JS*) that enables users to design and upload building structures to Anyplace. The Anyplace *Viewer* is a respective Web App that allows search and navigation off-the-shelf, without installation or logistical challenges. Both the *Architect* and the *Viewer* are built with the *AngularJS* framework and utilize the *Google API* (*Maps, Directions, Heatmaps and URL shortener*) to present and process data on a map along with the *HTML5* Geolocation for localization. The Viewer/Architect codebase can be encapsulated directly into native mobile apps using the Ionic Framework, which we have already tested with satisfactory results in another open-source project we developed, named Rayzit [1].

The combined Anyplace *Navigator* and *Logger* is a native Android application, which can benefit from Wi-Fi fingerprinting [13, 9] available under this platform. The *Navigator* allows users to see their current location on top of the floorplan map and navigate between POIs inside the building, similarly to the *Viewer* (iOS, Android, Windows). The main difference is that the Navigator offers superb accuracy, as it uses Wi-Fi Radiomap localization and the on-board smartphone sensors (i.e., accelerometer, gyroscope and digital compass), which are seamlessly integrated in our tracking module to smooth the Wi-Fi locations and enhance the navigation experience. The *Logger* application enables users to record Wi-Fi readings from nearby Wi-Fi access points

[2]Anyplace Github. `https://github.com/dmsl/anyplace`

and upload them to our *Server* through a Web 2.0 API (in JSON). It is used by volunteers for contributing Wi-Fi data and for crowdsourcing the Radiomaps of buildings. In order to facilitate the collection of quality Radiomaps, we present a heat-map of previously collected fingerprints in the building. Components for massive processing of Wi-Fi signals in Apache Hadoop (e.g., filter incorrect contributions and exploit readings collected by heterogeneous devices), have been developed but not integrated in the latest open release yet.

3 Localization in Anyplace

The localization literature is very broad and diverse as it exploits several technologies. GPS is obviously ubiquitously available but has an expensive energy tag and is also negatively affected from the environment (e.g., cloudy days, forests, downtown areas). Besides GPS, the localization community proposed numerous proprietary solutions including: *Infrared, Bluetooth, visual or acoustic analysis, RFID, Inertial Measurement Units, Ultra-Wide-Band, Sensor Networks, Wireless LANs, etc.*; including their combinations into hybrid systems [2].

Anyplace uses Wi-Fi Radiomap-based indoor localization, which stores radio signals from Wi-Fi APs in a database at a high density. The localization subsystem of Anyplace utilizes the following routine: in an offline phase, a logging application records the so called *Wi-Fi fingerprints*, which comprise of *Received Signal Strength (RSS)* indicators of Wi-Fi Access Points (APs) at certain locations (x,y) pin-pointed on a building floor map (e.g., every few meters). Subsequently, in a second offline phase, the Wi-Fi fingerprints are joint into a NxM matrix, coined the *Wi-Fi RadioMap*, where N is the number of unique (x,y) fingerprints and M the total number of APs. Finally, a user can compare its currently observed RSS fingerprint against the RadioMap in order to find the best match, using known algorithms such as KNN or WKNN [8]. A similar methodology can be applied to other types of signals, for instance, we are experimenting with magnetic fingerprints [10]. Both are considered infrastructure-free approaches, as Wi-Fi APs are ubiquitously available in urban and indoor spaces [9].

3.1 Crowdsourcing the Radiomap: The Anyplace Logger

Crowdsourcing has recently emerged as a promising solution for collecting the high volume of location-tagged data, e.g., the WiFi RSS radiomap of a multi-storey building, which are required to support indoor localization systems. In this context, volunteers engage in participatory sensing campaigns to collect location-dependent RSS samples. This is an attractive approach, because it splits the cumbersome and time consuming data collection task among the crowd. For example, it required 15 collectors for 2 weeks to collect point-by-point 200,000 Wi-Fi signal strength readings at 10,000 unique locations to cover the 450,000 m^2 COEX underground shopping mall area in S. Korea [13]. Another benefit from crowdsourcing is also the cost factor (e.g., the measurement survey upon the *Ekahau* system installation can cost 10,000 USD for a large office building with no maintenance included [7]). At the same time, however, it raises new challenges such as filtering incorrect contributions (trustworthiness), managing the radiomap size and fusing data from heterogeneous mobile devices [6].

The Anyplace *Logger*[3] is an Android application that allows volunteers to freely obtain RSS data and contribute it to Anyplace for improvement of the location quality. Crowdsourcers can select the desired building and floor, as well as modify the number of samples to be recorded and other settings, through the preferences screen. Subsequently, the users indicate their current location by clicking on the map and then click the on-screen buttons to initiate and end the logging process. In order to facilitate the collection of quality Radiomaps, we present a heat-map of previously collected fingerprints in the building. A crowdsourcer seeing this heat-map can easily identify areas where additional samples have to be collected. If a crowdsourcer is outside a given building no Wi-Fi signals can be collected. Upon finishing the collection of data, a user can upload this data to our or his cloud service through a Web 2.0 API (in JSON). The post-processed measurements are then available to other users that aim to localize accurately in the same building. At the University of Cyprus, 27 students crowdsourced

[3] Anyplace Logger. Video: `https://youtu.be/8EvioLZ6hvg`

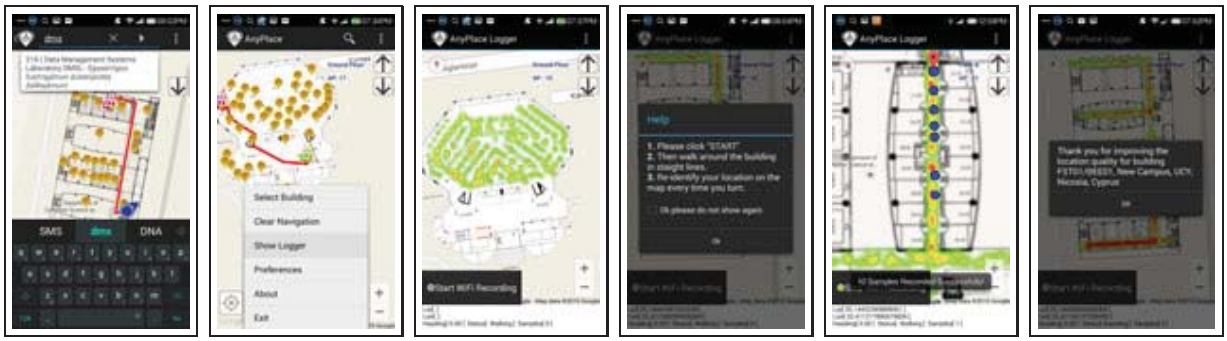


Figure 2: **Anyplace Logger and Navigator.** This is a native Android application that enables fine-grain indoor localization (up to 1.96m accuracy [9]) through the use of RSS fingerprints contributed by the crowd.

12 buildings (36,000 m^2) in a few hours while the rest of the 52 buildings were mapped by a single student. Similar efforts have already been observed on Anyplace for other campuses around the world.

3.2 Fine-grain Localization and Navigation: The Anyplace Navigator

The Anyplace *Navigator*[4] allows users to see their current location on top of the floorplan map and navigate between POIs inside the building with high accuracy. Particularly, the Navigator localization subsystem achieved an accuracy of 1.96 meters at the Microsoft Indoor Localization Competition at ACM/IEEE IPSN'14 [9] and was awarded the second position in its (infrastructure-free) category and third position overall. A user installing Anyplace from the Google Play market can use the *Navigator* in any public building listed on Anyplace. There is also a notion of private buildings whose access is limited to users having the unique URL. When the *Navigator* is launched, the building map and the associated POIs are automatically loaded by using the rough user location provided by the Google Geolocation API (see Figure 2). Then, the application downloads the RSS Radiomap of the relevant floor (subsequently the complete building) and displays the user location on top of the map. Moreover, users may search for POIs and get navigation directions from their current location. The *Navigator* also uses the onboard smartphone sensors (i.e., accelerometer, gyroscope and digital compass), which are seamlessly integrated in our tracking module to smooth the Wi-Fi locations and enhance the navigation experience.

3.3 Location Privacy

Location privacy refers to the the ability of an individual to move in public space with the reasonable expectation that their location will not be systematically and secretly recorded for later use [11]. The Anyplace was designed up-front with the aim to offer absolute location privacy. As such, localization structures (e.g., Radiomaps, POIs, connectors) are downloaded to the hand-held of a mobile user (u) from the Anyplace service (s) through the secure Anyplace API. All localization requests are then carried out in local mode on u, as opposed to s, which could fundamentally be compromised (e.g., by hackers or operators that install Anyplace on their servers). One downside of this approach, was that the localization structures would many times be outdated (i.e., these would not capture the latest crowdsourced data). As such, we got interested in investigating alternative hybrid localization strategies that would on the one hand exploit the utility of Anyplace, but on the other hand also offer controllable location privacy to the user. In [4] we devise the Temporal Vector Map (TVM) algorithm, where a user u camouflages its location from s, by requesting a subset of k entries from s, where k is a user-defined constant. Access the localization structures efficiently, is another complementary problem that we model and address using the Preloc framework [5].

[4]Anyplace Navigator. Video: `https://youtu.be/MO-473oWSfE`

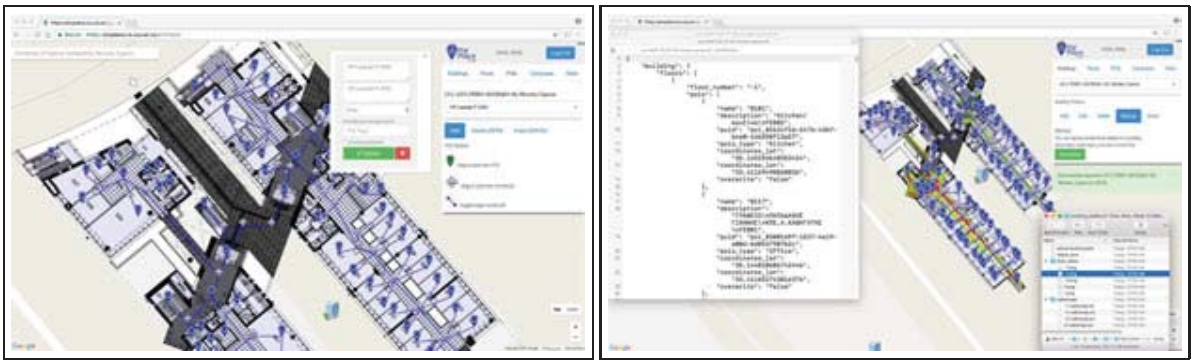


Figure 3: **Anyplace Architect. (Left)** Managing a campus of buildings through the architect Web app (cross-platform HTML5 interface). The architect allows a user to add floor-maps, POIs and connectors using drag-n-drop both interactively and in batch form. **(Right)** Import/export functionality in Anyplace simplifies data entry and promotes openness and data portability among different indoor management platforms.

4 Modeling in Anyplace

Unlike outdoor environments, indoor spaces are characterized by complex topologies and are composed of entities that are unique to indoor settings, such as multiple floors, rooms and hallways connected by doors, walls, stairs, escalators, and elevators [3]. To make things worse, doors may be one-directional (e.g., in security control in airports), while temporal variations may occur (e.g., a room may be temporarily inaccessible due to its opening hours). In this section we explain how buildings are managed in Anyplace and also summarize the predominant directions in this domain.

4.1 Managing Buildings and Campuses: The Anyplace Architect

The Anyplace *Architect*[5] is a web application that offers a feature-rich, user-friendly and account-based interface for managing indoor models in Anyplace. Particularly, it can be used to log-in with a Google account and place the blueprint of a building on top of Google Maps with multi-floor support. Using the floor editor, the user can upload, scale and rotate the desired blueprints to fit them properly, as shown in Figure 3. The user can later add, annotate and geo-tag POIs inside the building and connect them to indicate feasible paths for enabling the delivery of navigation directions. This interaction is carried out with drag-n-drop functionality that is cross-browser compatible and even operational on tablets and smartphones used in field deployments (e.g., while moving around with a tablet and correcting the indoor model).

The Architect also provides a range of other functionality, namely: i) *monitoring crowdsourcing progress* to collect Wi-Fi Radiomaps using color heat-maps (see Figure 3, right). An assigner can easily identify whether a given collection is satisfactory or not and thus define quantitative acceptance criteria for the output of crowd-sourcers; ii) *making a building public or private*, which automatically shares a building on the Anyplace *Viewer* interface (given that there are no collisions). Alternatively, a building can remain private and be shared among users through a URL (e.g., a person mapping a building for a specific event publicizes a private building to its audience by email or social media); and iii) *export and import of indoor models and Radiomaps*, which allows somebody to backup/restore a building, expedite user input of POIs, but also create a new model for a different purpose (i.e., template-based generation of a new building id for a new purpose).

[5]Anyplace Architect. Video: `https://youtu.be/dIVxcQ_5Wbg`

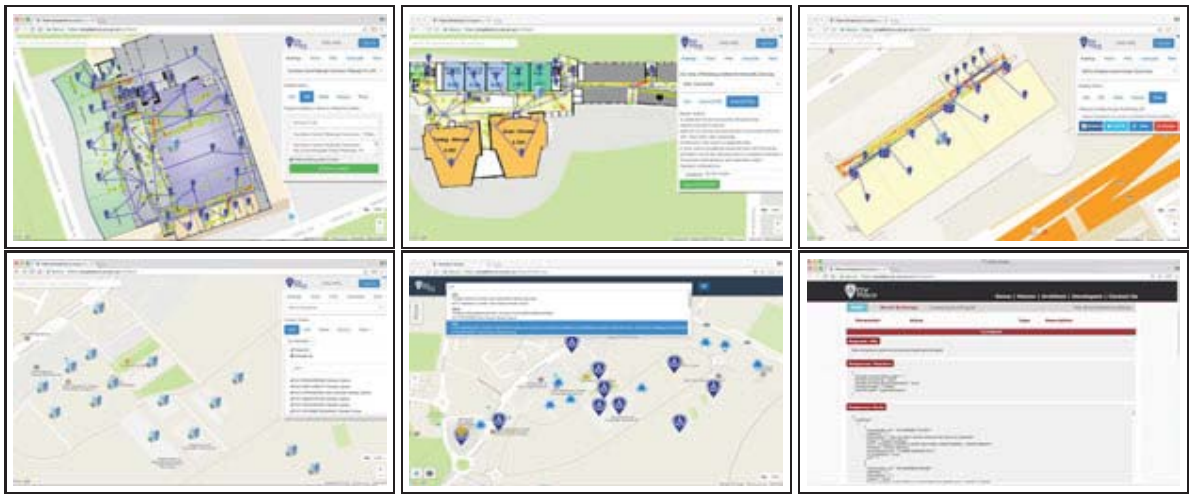


Figure 4: **Anyplace Information Layers. (Top)** Three buildings mapped on Anyplace (left-to-right): Hotel mapped in Pittsburgh, USA; Building at the University of Würzburg, Germany and a Convention Center, South Korea. Heatmaps facilitate architects to monitor the progress of the Radiomap collection. **(Bottom)** Campus layer allows information search and exploration across several buildings (left-to-right): Subset of the 52 buildings mapped at the University of Cyprus, the campus search engine that became available after the mapping, and the JSON Web 2.0 API that streamlines all data accesses from mobile apps and the web.

4.2 Modeling Directions

The technology road-map is towards indoor GIS integration where `indoorgml.net`, `geojson.org` or any other standard that may appear in the future and become fully inter-operable. Having the right modeling primitives will give rise to a variety of data management and query processing challenges in the future. Another direction is towards automated indoor model checkers (e.g., graph connectivity, automatically connect overlapping stairs and elevators) as these will make the mapping of an indoor space more straightforward. There already competing industrial systems, like `mazemap.com` and `micello.com`, which work directly with Autodesk's Industry Foundation Classes (IFC) data model that is used for the description of building and construction data. One final direction is the provisioning of libraries for managing specific types of objects in indoor spaces (e.g., office equipment, industrial appliances) but also libraries for representing indoor spaces more richly either using 3D models (e.g., `indoor.io`) or augmented reality.

5 Indoor Information Search in Anyplace

Indoor information search and exploration is among the most important aspects that complements indoor navigation. This happens as users will typically start their navigation out by first issuing a spatio-textual search that will return some Points-of-Interest (POIs) upon which navigation instructions to a particular target can be obtained. In this section, we describe how indoor information is modeled in Anyplace and how this helps in bringing forward a seamless navigation experience through the Anyplace Viewer.

5.1 Information Layers

In Anyplace, indoor data is organized in three logical layers (see Figure 4): i) Floor Layer; ii) Building Layer and iii) the Campus Layer. A floor layer comprises of a floor-map (i.e., JPG image), a set of POIs with anno-

Figure 5: **Anyplace Viewer.** This is a cross-platform web application optimized for mobiles, which supports indoor search, exploration and navigation over buildings mapped on Anyplace. It complements the native Anyplace apps with an alternative to open Anyplace URLs without the installation of a dedicated mobile app.

tations (e.g., door, entrance, office), edges connecting the POIs and sensor readings (i.e., Radiomaps). A floor is anchored in the WGS84 coordinate system, which is compatible with all tile layer providers such as Google Maps and Openstreetmaps. A building represents several floors logically linked together by POI edges (i.e., by connecting stairs or elevators of two floors). Every building has to feature at least one "building door" to which outdoor navigation instructions will be linked (this provides outdoor-to-indoor linkage). A building in Anyplace is identified globally uniquely through a *Building ID (BUID)*. Several BUIDs can be logically organized together through the Campus Layer to generate a globally unique *Campus ID (CUID)*. Both the BUIDs and the CUIDs can be obtained through the user interface and shared with a URL[6] through popular social media, email, SMS, embedded in websites as an HTML iframe or as a field in the respective API calls.

In the current release, we use the Google maps API for provisioning of the underlying map tiles but this doesn't restrict the utility of Anyplace to the particular provider (i.e., it could be Openstreetmaps, Bing Maps or any other service). The objective of linking to an outdoor web mapping service was to obtain search and navigation instructions for the outdoor world, while Anyplace then only focuses on navigation and information search in the indoor spaces it represents (e.g., Figure 2 and Figure 5). Google Maps also provides us with sattelite, birds-eye, street-map views and in certain occasions even floor maps and floor selectors (from Google Indoor), which can nicely complement the Anyplace experience. Google Indoor has a similar scope to Anyplace Architect, but it does so in a very centralized manner that eventually will only lead to a single mapping of some public building. On the other hand, Anyplace Architect allows anybody to add any type of building in as many representations as necessary. For example, one might create a general-purpose indoor navigation model of a hospital that is inherited and refined by some other user into a model for inventory management in that building or a new model that arranges a set of additional POIs on a map for a specific event. The import and export functionality of Anyplace provides a true opportunity for achieving these scenarios.

5.2 Viewing Indoor Mappings: The Anyplace Viewer

The Anyplace *Viewer*[7] is again a Web App that enables the quick visualization of buildings modeled in Anyplace. It is ideal for a first-time user that doesn't want to invest considerable time before launching the service through an app downloaded from a popular mobile market. The viewer enables off-the-shelf usage without installation or logistical challenges, which is many times an overhead when users aim to get to their destination quickly, as it only requires a web browser. A more involved user can download the Anyplace Navigator from the various markets and enjoy advanced functionality (e.g., superb accuracy, caching, etc.) The UX/UI of the Viewer has

[6]UCY Campus on Anyplace. `https://anyplace.cs.ucy.ac.cy/viewer/?cuid=ucy`
[7]Anyplace Viewer. Video: `https://youtu.be/uMFnxXnm1yc`

been implemented with a mobile user on-the-go in mind (i.e., thumb-based user interface, large buttons, less clutter) and is extremely straightforward to use.

Acknowledgments

Anyplace has been implemented by researchers and students at the Data Management Systems Laboratory of the Dept. of Computer Science at the Univ. of Cyprus. This work was supported in part by the University of Cyprus. The first author's research is currently supported by the Alexander von Humboldt-Foundation, Germany.

References

[1] G. Chatzimilioudis, C. Costa, D. Zeinalipour-Yazti, W.-C. Lee and E. Pitoura *"Distributed In-Memory Processing of All k Nearest Neighbor Queries"*, *IEEE TKDE*, vol. 28, iss. 4, pp. 925-938, 2016.

[2] Y. Gu, A. Lo, I. Niemegeers, *"A survey of indoor positioning systems for wireless personal networks"*, in IEEE Comm. Surv. Tutor., vol. 11, no. 1, pp. 13–32, 2009.

[3] C.S. Jensen, H. Lu, B. Yang, "Indoor - A New Data Management Frontier," in *IEEE Data Eng. Bull* vol. 33, iss. 2, pp. 12–17, 2010.

[4] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, Y. Theodoridis, "Privacy-Preserving Indoor Localization on Smartphones", *IEEE TKDE*, vol. 27, iss. 11, pp. 3042-3055, 2015.

[5] A. Konstantinidis, G. Nikolaides, G. Chatzimilioudis, G. Evagorou, D. Zeinalipour-Yazti, P. K. Chrysanthis "Radiomap Prefetching for Indoor Navigation in Intermittently Connected Wi-Fi Networks," in *IEEE MDM*, pp. 34–43, 2015.

[6] C. Laoudias, D. Zeinalipour-Yazti, C.G. Panayiotou, "Crowdsourced Indoor Localization for Diverse Devices through Radiomap Fusion", in *IPIN*, pp. 1–7, 2013.

[7] J. Ledlie, et. al., *"Molé: a scalable, user-generated WiFi positioning engine"*, Journal of Loc. Based Serv., vol. 6, no. 2, pp. 55–80, 2012.

[8] B. Li, J. Salter, A. G. Dempster, C. Rizos, "Indoor positioning techniques based on wireless lan," in *1st International Conference on Wireless Broadband and Ultra Wideband Communications*, pp. 13–16, 2006.

[9] D. Lymberopoulos et. al., "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *ACM/IEEE IPSN*, pp. 178–189, 2015.

[10] A. Nikitin, C. Laoudias, G. Chatzimilioudis, P. Karras, D. Zeinalipour-Yazti "Indoor Localization Accuracy Estimation from Fingerprint Data," in *IEEE MDM*, 12 pages, Daejeon, South Korea, 2017 (accepted).

[11] R.A. Popa, H. Balakrishnan, A.J. Blumberg, "VPriv: protecting privacy in location-based vehicular services," in *USENIX SSYM*, pp. 335–350, 2009.

[12] D. Zeinalipour-Yazti, C. Laoudias, C. Costa, M. Vlachos, M.I. Andreou, D. Gunopulos, "Crowdsourced Trace Similarity with Smartphones", in *IEEE TKDE*, vol. 25, iss. 6, pp. 1240–1253, 2013.

[13] D. Zeinalipour-Yazti, C. Laoudias, K. Georgiou, G. Chatzimiloudis, "Internet-based Indoor Navigation Services", in *IEEE Internet Computing*, DOI: 10.1109/MIC.2016.21, 2017 (in-press).

Risk Detection and Prediction from Indoor Tracking Data

Tanvir Ahmed[1], Toon Calders[2,3], Hua Lu[4], Torben Bach Pedersen[4]
[1]RadioAnalyzer ApS, Denmark
[2]University of Antwerp, [3]Université Libre de Bruxelles, Belgium
[4]Aalborg University, Denmark

Abstract

Technologies such as RFID and Bluetooth have received considerable attention for tracking indoor moving objects. In a time-critical indoor tracking scenario such as airport baggage handling, a bag has to move through a sequence of locations until it is loaded into the aircraft. Inefficiency or inaccuracy at any step can make the bag risky, i.e., the bag may be delayed at the airport or sent to a wrong airport. In this paper, we discuss a risk detection and a risk prediction method for such kinds of indoor moving objects. We propose a data mining methodology for detecting risk factors from RFID baggage tracking data. The factors should identify potential issues in the baggage management. The paper presents the essential steps for pre-processing the unprocessed raw tracking data and discusses how to deal with the class imbalance problem present in the data set. Next, we propose an online risk prediction system for time constrained indoor moving objects, e.g., baggage in an airport. The target is to predict the risk of an object in real-time during its operation so that it can be saved before being mishandled. We build a probabilistic flow graph that captures object flow and transition times using least duration probability histograms, which in turn is used to obtain a risk score of an online object in risk prediction.

1 Introduction

Technologies such as RFID and Bluetooth enable a variety of indoor, outdoor, and mixed indoor-outdoor tracking applications. Examples of such applications include tracking people's movement in large indoor spaces (e.g., airports, office buildings, and shopping malls), airport baggage tracking, item movement tracking in supply chains, and package tracking in logistics systems. The research work presented in this paper in particular attempts to solve the airport baggage mishandling problem as the aviation industry suffers from enormous loss due to baggage mishandling. A recent report[1] shows that in 2013, 3.13 billion passengers traveled by airlines and among them around 21 million passengers and 21.8 million bags were affected by baggage mishandling that costs 2.09 billion USD to the airline industry. Common baggage mishandlings are: left behind at the origin or connecting airport (i.e., failed to catch the intended flight), bag loss, wrong bag destination, etc.

In airport baggage management a bag has to go through different steps in each airport from origin to final destination. Suppose that Nadia needs to travel from Aalborg Airport (AAL) to Brussels Airport (BRU) via Copenhagen Airport (CPH). First, Nadia has to check-in and hand over her bag to the check-in desk staff at AAL. Then the staff puts the bag on the conveyor belt for the automatic baggage sortation system. After passing all the steps inside AAL, the bag is loaded into the aircraft using the belt loader for the targeted flight. Upon arrival at CPH it is shifted to the transfer system. After all the required stages at CPH, the bag is loaded into the aircraft for BRU. After arriving at BRU, Nadia collects the bag from the arrival belt. During this journey,

[1]SITA Baggage Report 2014 `www.sita.aero/content/baggage-report-2014`

the bag has to go through up to 11 stages and there can be as many baggage handlers handling the bag at the different stages. Fig. 1 visualizes this airport baggage tracking scenario. The upper part of the figure shows the top level path of a bag that travels from AAL to BRU via CPH. The bottom part of the figure shows the baggage processing stages inside AAL. The circles represent the baggage tracking locations where RFID readers are deployed for baggage tracking.

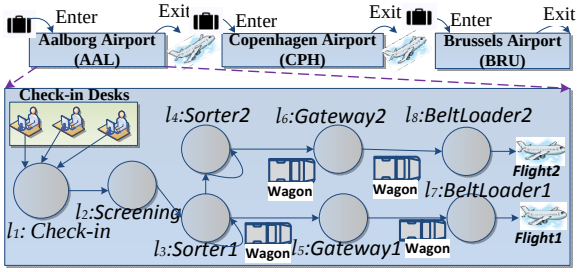


Figure 1: Baggage tracking in airport [2]

At check-in, an RFID tag is attached to the bag. The memory inside the tag stores bag information including the bag identifier, flights, route legs, date of departure, etc. While passing different stages, whenever a bag enters into a reader's activation range, it is continuously detected by the reader with a sampling rate which generates raw reading records of the form: $\langle BagID,\ Location,\ Time,\ \{Info\}\rangle$, meaning that a reader at location *Location* detects a bag with ID *BagID* at timestamp *Time* and the tag stores the information *Info*. Considering only location and time related information, some examples of raw reading records are shown in Fig. 2a. In the table, *RID* represents the reading identifier. The massive baggage tracking data can be very useful for analyzing and finding interesting patterns. Combining the tracking data with other dimensions like route information, flights and punctuality, day hours, week day, transit duration, etc., can reveal risk factors that are responsible for baggage mishandling. As seen, a bag can have several readings at the same location and due to the rawness, it is also difficult to do further analysis. To overcome these problems and prepare the data for further analysis, we convert the raw reading records into stay records [3], described below.

(a) Raw Reading Table

RID	BagID	Location	Time
R1	B1	AAL.Chkin1	1
R2	B1	AAL.Screen	4
R3	B1	AAL.Screen	5
R4	B1	AAL.sorter1	8
R5	B1	AAL.sorter1	9
R6	B1	AAL.Gate2	15
R7	B1	AAL.BltLd1	21
R8	B1	CPH.Trans1	70
R9	B1	CPH.Tran1	72
R10	B1	CPH.Sorter2	80
R11	B1	CPH.Sorter2	90
R12	B1	CPH.Sorter2	100

(b) Stay Records

Rec ID	Bag ID	FromLoc	ToLoc	t_{start}	t_{end}	Dur.
Rec1	B1	AAL.Chkin1	AAL.Screen	1	4	3
Rec2	B1	AAL.Screen	AAL.sorter1	4	8	4
Rec3	B1	AAL.sorter1	AAL.Gate2	8	15	7
Rec4	B1	AAL.Gate2	AAL.BeltLd1	15	21	6
Rec5	B1	AAL.BltLd1	CPH.Tran1	21	70	49
Rec6	B1	CPH.Trans1	CPH.Sorter2	70	80	10
Rec7	B1	CPH.Sorter2	CPH.Sorter2	80	100	20

(c) FlightLeg Records

Bag id	From Airpt.	To Airpt.	Is Tran.	Weekday	Flight Time	DurBef. Flight	IsLongStayFound	Delay InArr.	TotBag ThatHr	Status
B1	AAL	CPH	0	Monday	9-10	25	0	NULL	86	OK
B1	CPH	ARN	1	Monday	10-11	30	1	-5	70	Mishan.

Figure 2: Raw reading and relevant records [1]

Stay Records A stay record is of the form: $StayRecord\langle BagID,\ FromLocation,\ ToLocation,\ t_{start},\ t_{end},\ Duration,\ \{StayInfo\}\rangle$ which represents that a bag with *BagID* first appeared at *FromLocation* at time t_{start} and then first appeared at the next location *ToLocation* at time t_{end}. It took *Duration* time to go from the reader at *FromLocation* to the reader at *ToLocation*. The $\{StayInfo\}$ represents a set of other dimensional information related to the bag and to the transition (e.g., bag status, next flight schedule, origin and destination airports, and other flight-related information). The stay records compress the huge data volume of raw readings and also enable to find abnormally long time spans between locations that may lead to baggage mishandling. Fig. 2b shows the stay records for the raw records of Fig. 2a. In the table, *RecID* represents the stay record identifier. In Fig. 2b, *Rec1* represents that bag *B1* had a transition from *AAL.Checkin-1* to *AAL.Screening* and it took 3 time units for the transition. The stay records introduce a very important feature which is the *duration* information.

This paper addresses the baggage mishandling problem in two phases. First, it analyzes the problem in an offline scenario where it presents a data mining methodology for automatic extraction of risk factors from RFID baggage tracking data. Second, it addresses the issue in an online scenario where it develops an online risk prediction system for predicting the risk of a bag in real-time so that it can be saved from being mishandled. For example, from the data, one might be interested to know, *what is the probability that a bag will be mishandled if it has 35 minutes transit time at Copenhagen Airport on Sunday morning*? A real-time analysis request can be: *notify the baggage manager whenever a bag becomes risky during its processing time at Aalborg Airport*.

The rest of this paper is organized as follows. Sections 2 and 3 describe the mining risk factors from offline baggage tracking data and the online risk prediction system, respectively. Section 4 concludes and points to future work.

2 Risk Detection

In this section, we will discuss a methodology for building the best predictive model which will give us a set of rules or patterns that are highly correlated to the mishandled bags.

FlightLeg Records For finding risk factors, we are interested to find the baggage management performance at airport level rather than at the reader level. We take the duration feature including some other dimensions from the stay records and create a table called *FlightLeg Records*. For each flight of a bag we have one instance in the *FlightLegRecords* table. It captures some important features extracted from the *Stay Records* like {*IsTransit, DurationBeforeFlight, IsLongerStayFound, TotalBagInThatHour, Status*}. The attributes are discussed below.

(i) *FromAirport*: Departure airport of the corresponding flight. (ii) *ToAirport*: Destination airport of the corresponding flight. (iii) *IsTransit*: A Boolean value representing whether the record belongs to a transit bag or not. (iv) *Weekday*: Weekday of the corresponding flight. (v) *FlightTimeHour*: Departure hour of the corresponding flight. (vi) *DurationBeforeFlight*: Available time (in minutes) for the bag to catch the flight. For a non-transit record, it is calculated from the first reading time of the bag at check-in and the actual departure time of the flight. Conversely, for a transit record, it is calculated from the actual flight arrival time at the *FromAirport* and actual departure time of the next flight to the *ToAirport*. (vii) *IsLongerStayFound*: If any stay duration between readers at the *FromAirport* is longer than expected then it is *true*, otherwise it is *false*. Its value is determined by comparing the movements of baggage between readers at *FromAirport*. For each distinct transition between a pair of locations in the *Stay Records*, the bags that had followed the top 5% longest durations are considered as *longer than expected*. (viii) *DelayInArrival*: Delay in arrival (in minutes) of the arrival flight for the transit bag. Its value is calculated from the actual and scheduled arrival times of the flight in the transit airport (i.e., *FromAirport* is a transit airport). For the non-transit records *DelayInArrival* is *NULL*. (ix) *TotalBagInThatHour*: Number of bags read during the departure hour of the flight at *FromAirport*. (x) *Status*: Status of the bag i.e., 'OK' or 'Mishandled'. The status of the bag indicates whether the bag was mishandled or not in the *FromAirport*. The status of a bag is extracted from the reading records of the bag at the readers at *FromAirport*, flight timing, route information, etc. If a bag has any reading in the *FromAirport* after the corresponding flight departure time, then the bag is considered as left behind. Conversely, if a bag has any reading from an airport which is not in its planned route, then it is considered as wrong destination.

Fig. 2c shows an example content of *FlightLegRecords*. We use the *FlightLegRecords* for further analysis.

2.1 Solution Steps for Risk Detection

For detecting risks, we take the help of a classification algorithm. We use the *Status* attribute of the *FlightLegRecords* as the class column. However, baggage tracking data are highly imbalanced as the rate of mishandled bags is quite small ($<1\%$) compared to the correctly handled bags [1]. This imbalance presents difficulties to most data mining techniques. As a result, this *imbalance problem* should be handled wisely to overcome poor quality results otherwise produced by the classifier. The solution steps are given in Fig. 3. At first, the raw records are converted into the *FlightLegRecords*. The other steps are discussed next.

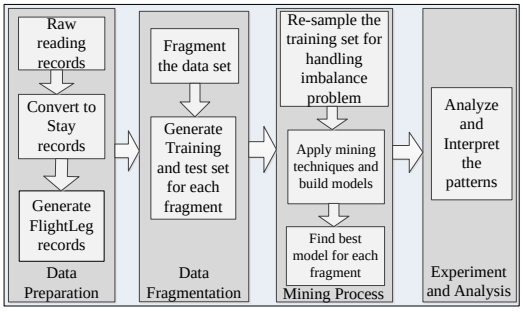


Figure 3: Outline of the steps

2.2 Data Fragmentation

Fragments The baggage management problem varies based on different important factors like whether the bag is in the transit airport or not, the duration of the transit, etc. Based on that, we have divided the data set into 5 fragments and applied data mining algorithms on each of the fragments for finding patterns specific to each fragment. Fig. 4 shows the different fragments of our data set, and the numbers inside the square bracket show the number of records and mishandling rate for the corresponding fragment in our experimental data [1].

Training and test set For each of the discussed fragments, we have one partition for training (P1), and another for testing (P2). In our experiments, all the records on or before a certain flight date are included in the training set and the rest is included in the test set. We did not rely on a standard cross-validation approach because bags that were on the same plane are more likely to have similar properties, as well as a similar class label. Therefore, spreading bags of the same flight over both the training and test set may cause a biased estimation of the performance due to overfitting. By dividing the data based on date, we can guarantee that the training set and the test set are independent, and we get an unbiased estimate of the performance of the mined models.

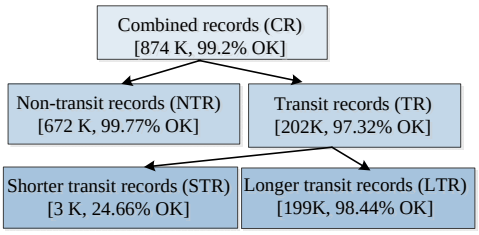


Figure 4: Fragments of the data set

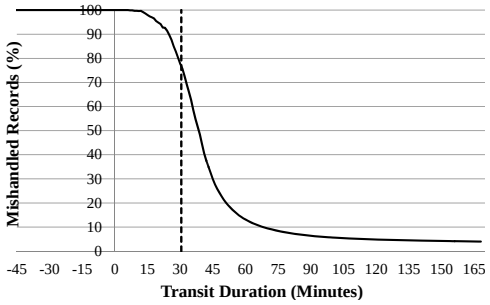


Figure 5: Mishandling vs. transit duration

2.3 Mining Process

Handling imbalance data. To remedy the imbalance problems, we use 2 different kinds of re-sampling for the training data(P1): ***Undersampling*** **(US)**: In this technique, a subset of *P1* is created by randomly deleting *OK* records until we reach equal number of records with class *OK* and class *MH*. ***Oversampling*** **(OS)**: In this technique, a superset of *P1* is created by copying some instances or generating new instances of *MH* records until we obtain an equal number of records for class *OK* and *MH*. We use Synthetic Minority Over-sampling Technique (SMOTE) [4] for getting *OS* data.

Mining Techniques We apply *Decision Tree (DT), Naive Bayes classifier (NB), K NN c lassifier (KNN), Linear regression (LIR), Logistics regression (LOR)*, and *Support vector machine (SVM)* on the training set *P1* of the combined records *CR* with the sampling strategies discussed above. We also do the same directly to P1 without re-sampling *(WS)*. Then we use different types of measures (discussed in the next paragraph) for finding the classification and sampling techniques that provide the best model for our data set. Subsequently, the chosen techniques are used for generating models for the remaining fragments. Before applying KNN, linear regression, logistics regression, and SVM, the structure of the input data table is changed as these algorithms do not work with categorical attributes. In these cases, we convert each value of a categorical attribute into a separate column and put Boolean 0 or 1 accordingly. An example of such conversion for Fig. 2c is shown in Table 1. For linear regression and logistic regression, the attributes with continuous values are normalized into [0,1]. Moreover, in our data set all the *FromAirport*s are within the *Schengen territory* (http://en.wikipedia.org/wiki/Schengen_Area). Unlike *FromAirport*s we have too many values in the *ToAirport* column which creates many branches in the decision tree and for other classification algorithms this column become useless. To make the *ToAirport* column useful and make the learned pattern interesting, we categorized the *ToAirport*s into three types: *Domestic, Schengen*, and *Others*.

Table 1: String values into columns of Fig. 2c

AA	L	CP	H	Is	Transit	Mo	nday	9-	10	10	-11	....
1		0		0		1		1		0		...
0		1		1		1		0		1		...

Finding the best model Typically the performance of a classifier is evaluated by its predictive accuracy. However, for an imbalanced data set the accuracy is not an appropriate measure, e.g., in our case an accuracy of 99% does not make sense, as it may misclassify all the examples as *OK (negative)* regardless of whether a record belongs to *Mishandled (positive)* or not. In our scenario, misclassifying a *Mishandled* bag as *OK* is more severe than misclassifying an *OK* bag as *Mishandled*. As such we are specifically interested in algorithms with a high recall on the *Mishandled* bags rather than in merely optimizing the accuracy of the classifier. We use the AUC of the ROC curve as the main measure for choosing the model that provides the best ranking. We also use precision-recall curves for finding which threshold provides higher precision for a good amount of recall. We perform a comprehensive experiment which can be found in [1]. Some results were presented in Section 4.

3 Online Risk Prediction (ORP)

Given a set of stay records R and a set of online moving objects O, we are interested in building a predictive model from R that can predict, as early as possible, whether an object $o_i \in O$ is at risk. For example, the model should be able to predict as early as possible whether a bag going through the baggage handling stages is at risk of being delayed.

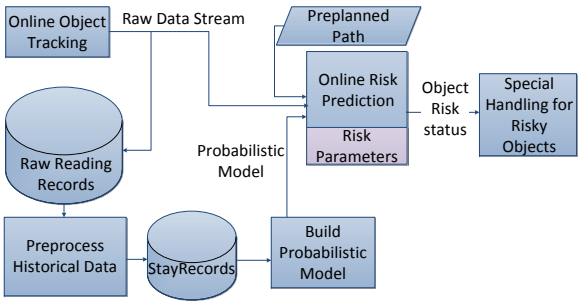


Figure 6: Overview of the ORP system [2]

3.1 Solution

The overall outline of the ORP system is shown in Fig. 6. The online tracking data stream is passed into two modules. One stores the data offline for future analysis and model building purpose and another uses it during the ORP process. The offline/historical records are converted into *StayRecords*, which are used for building the probabilistic model. The model, the raw data stream, and the preplanned paths of the objects are used by the ORP for deciding which objects are at risk. Finally, risky objects are notified by the ORP for special handling.

Let L = $\{l_1, l_2\ l_3, ..., l_n\}$ be the set of locations available in the data set and the set of durations taken by the transitions from location l_i to l_j be $D_{i,j}$=$\{d_1, d_2, d_3, \ldots, d_n\}$.

Definition 1: ***Least Duration Probability (LDP).*** An LDP for a movement from l_i to l_j with threshold duration $d_k \in D_{i,j}$ is defined as

$$LDP(l_i, l_j, d_k^{\geq}) = \frac{Count(l_i, l_j, d_k^{\geq})}{Count(l_i, l_j)} \quad (1)$$

Here, $Count(l_i, l_j, d_k^{\geq})$ is the number of objects that took at least d_k duration from l_i to l_j and $Count(l_i, l_j)$ is the number of objects that have a transition from l_i to l_j.

Definition 2: ***Least Duration Probability Histogram (LDPH).*** An LDPH for transitions from l_i to l_j is a histogram with transition durations $D_{i,j}$ on the X-axis, and LDPs for the transitions on the Y-axis.

Transition $tr(l_i \rightarrow l_j)$	Dur (d_k)	$C(tr)$	$C(tr, d_k^{\geq})$	$LDP(tr, d_k^{\geq})$
$l_1 \rightarrow l_2$	13	1000	1000	1
	16		700	0.7
	20		300	0.3
	28		20	0.02
$l_2 \rightarrow l_3$	8	1000	1000	1
	10		580	0.58
	17		50	0.05
$l_3 \rightarrow l_5$	60	470	470	1
	70		250	0.53
	98		50	0.11
$l_5 \rightarrow l_7$	40	460	460	1
	50		250	0.54

Table 2: Transition summary (C stands for *Count*)

Table 2 shows an example summary of transitions for the path l_1 to l_7 generated from stay records. Fig 7 shows the LDPH for transition l_1 to l_2 shown in Table 2. In Fig. 7, LDP=0.7 represents that the probability of transition from l_1 to l_2 with a duration ≥ 16 is 0.7. It also shows that the LDP for duration 28 is very low (0.02).

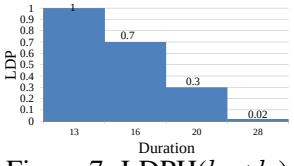


Figure 7: LDPH($l_1 \rightarrow l_2$)

Probabilistic Flow Graph (PFG). We build a probabilistic flow graph (PFG) from the data set, where each location is represented by a node and transitions between locations are represented by edges. The edges are labeled by their corresponding LDPHs. Fig. 8 shows the PFG constructed from the transition summary shown in Table 2. One LDPH of Fig. 8 is shown in Fig 7. The data for the rest of the LDPHs are available in the $LDP(tr, d_k^{\geq})$ column of Table 2.

3.2 Online Risk Prediction Steps

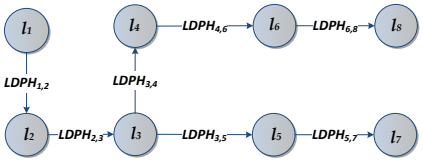


Figure 8: PFG

We consider a scenario, where the path of a given online object is predefined, e.g., in baggage tracking, all the bags intended for a particular flight *SK123* should follow the same path sequence starting from the check-in desk up to the belt loader to the aircraft (e.g., $l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_5 \rightarrow l_7$). If the object does not follow its preplanned path, it is triggered as risky. However, an object following its preplanned path, but taking an unusually longer duration for a transition, can also become risky. Moreover, the risk of an

object being delayed not only depends on the duration at a single location, but also at the subsequent locations. For example, an object might take a long duration from l_1 to l_2 that makes it risky. However, it might be handled very quickly in its next transition l_2 to l_3 that recovers the object. To capture this, we aggregate the paths in the PFG into an *aggregate probability flow graph (APFG)*. Here, we additionally maintain an *aggregate LDPH (ALDPH)* for each path sequence S=$l_i l_{i+1} l_{i+3} ... l_n$, where l_i must be the first tracking location of at least one object in the data set and we have $i<n<=p$ (the length of the path sequence). An *ALDPH (S)* contains all the *aggregate LDPs (ALDP)* for S. An *ALDP(S,$d^{\geqslant}$)* represents the probability of taking at least a duration of d by an object for completing the path sequence S. The value of an ALDP for the path sequence S with a total duration d is computed by Eq. (2). In Eq. (2), $Count(S, d^{\geqslant})$ is the number of objects taking at least a d duration to complete the path sequence S and $Count(S)$ is the number of objects traveling through path sequence S.

Table 3 shows the ALDPs and data for ALDPHs for the path from l_1 to l_7 in our example scenario.

$$ALDP(S, d^{\geqslant}) = \frac{Count(S, d^{\geqslant})}{Count(S)} \quad (2)$$

For risk prediction, we use an ALDP threshold *$ALDP_{th}(s_i)$* for each path sequence s_i for each of the preplanned paths. The $ALDP_{th}$ is converted into a *risk score threshold* RS_{th}, where $RS_{th}(s_i)$=*1-$ALDP_{th}(s_i)$*. After each transition of an object o, we get the total duration spent d by o to travel the path path sequence s_i and then we extract the *ALDP(s_i, d)* from the corresponding ALDPH. If d is not directly available in the ALDPH, we use linear interpolation to obtain the corresponding ALDP. After getting ALDP, we calculate the *risk score* of the object by *RS=1-ALDP*. If *RS*$\geq$*RS_{th}*, then o is marked as in risk. Also note that for the first transition, the values of ALDP and LDP are the same.

Path(S)	Dur(d)	*Count(S)*	*Count(S,$d^{\geqslant}$)*	*ALDP(S,$d^{\geqslant}$)*
$l_1 \rightarrow l_2 \rightarrow l_3$	21	600	600	1
	23		500	0.83
	24		300	0.5
	30		150	0.25
	33		30	0.05
$l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_5$	81	470	470	1
	83		415	0.88
	84		245	0.52
	94		195	0.41
	100		120	0.26
	121		50	0.11
	128		20	0.04
$l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_5 \rightarrow l_7$	123	460	460	1
	131		290	0.63
	134		235	0.51
	144		145	0.32
	150		110	0.24
	171		40	0.09
	178		10	0.02

Table 3: Path summary

In our example scenario, for the path from l_1 to l_7, there will be ALDP thresholds for each of the path sequences mentioned in Table 3. In the described scenario, o has to complete a transition to determine whether the object is at risk or not. To improve this scenario, we take the help of *$ALDP_{th}(s_i)$* to find the maximum acceptable stay duration ($Dur_{max}(s_i)$) of o to complete the path sequence s_i. Based on Dur_{max} a *time trigger (TT)* is set for o after each transition, which will trigger an alarm if o is not being read by the next planned location within the timestamp. The time trigger of o is set by, $TT[o]$ = t_{start}+Dur_{max}, where t_{start} is the timestamp when o was first tracked in the system and Dur_{max} is the maximum allowable duration for an object to complete its path sequence up to the next planned location. The overall processing steps of the ORP are shown in Fig. 9.

For example, consider an object o following a path: $l_1 \xrightarrow{17} l_2 \xrightarrow{17} l_3 \xrightarrow{60} l_5 \xrightarrow{40} l_7$. The labels on the arrows represent the duration taken for the transitions. Let us consider *$ALDP_{th}$* for any path sequence is 0.2, thus RS_{th} = 1 - 0.2 = 0.8. Also, $Dur_{max}(l_1, l_2)$ by linear interpolation=$\lceil 20 + (0.3 - 0.2)/(0.3 - 0.02) \times (28 - 20) \rceil$=23. Consider that the object followed its preplanned path. For the first transition, LDP (l_1,l_2,$17^{\geqslant}$) by linear interpolation = 0.7-(0.7-0.3)/(20-16)$\times$(17-16) = 0.6, thus RS=1-0.6=0.4. As 0.4$\ngeq$0.8, o is not risky at this stage. Also, in terms of Dur_{max}, o is safe in this step as 17$\ngtr$23. For the second transition (i.e., up to l_3), the ALDP is 0.05 (as the total duration=17+17=34), thus RS = 0.95. As As 0.95$\geq$0.8, o is at risk at that point. After the next transition (i.e., up to l_5), the ALDP is 0.41, thus RS = 0.59$\ngeq$$RS_{th}$=0.8 (as total duration=17+17+60=94). The new score shows that the object recovered from its risky state as it was processed quickly between l_3 and l_5. When o moves further, the time trigger for o is also updated based on the traversed path sequence, preplanned path, and $ALDP_{th}$.

Time Constrained ORP. Generally, a slow processing of an object could result in a dense location or traffic jam. However, there are many applications where an object has to reach a particular location by a given timestamp. For example, in an airport a bag has to be loaded in the aircraft before the scheduled flight departure. So, the duration before flight departure is an important factor for baggage risk prediction. If a bag starts its processing well in advance before the flight departure, it is less risky, even if it stays longer for a transition, and vice versa. So, the stay duration should be normalized with the available processing time and use the normalized duration for taking the corresponding ALDP to reflect the actual riskiness of the object.

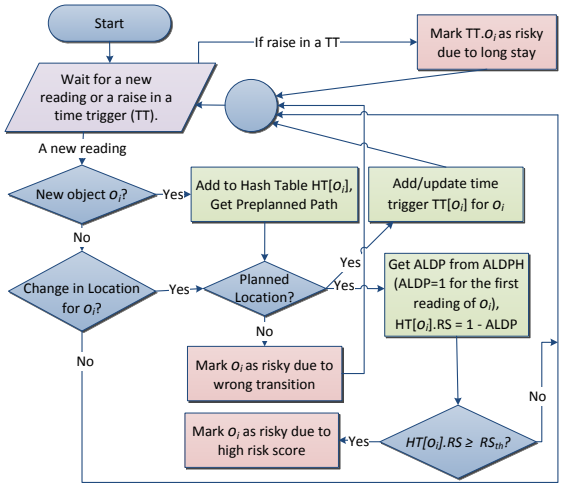


Figure 9: Online risk prediction steps

Let t_{enter} be the first time an object o was detected in the system and t_{final} be the maximum timestamp when o should reach its final reading point. So, the total available duration for o is d_a=t_{final}-t_{enter}. The expected average duration of travel of an object is extracted from the *ALDPH* for the full preplanned path. Let d_e be that expected duration extracted from the *ALDPH* with *ALDP* = 0.5. After each transition of o, its normalized total stay duration for the so far traversed path is computed by Eq. (3), where dur_i is the stay duration of o for its i^{th} transition. In the equation, the value of *offset* is computed initially by subtracting the value of d_a from d_e. Then, after the k^{th} transition of o, its total travel time up to that transition is added to the offset for obtaining the normalized duration. So, instead of taking the ALDP directly for the total duration d_t, we take the ALDP for d_n. Depending on the value of d_a and d_e, the value of *offset* as well as d_n can be negative.

$$d_n(o) = \textit{Offset} + \sum_{i=1}^{k} dur_i, \textit{where offset} = (d_e - d_a) \tag{3}$$

For example, consider an object o_1 following its preplanned path: $l_1 \xrightarrow{17} l_2 \xrightarrow{17} l_3 \xrightarrow{94} l_5 \xrightarrow{50} l_7$. o_1 has a total of 200 seconds to reach l_7 from l_1. So, d_a=200 sec. From Table 3, d_e=134 (as ALDP for 134 is 0.51). Hence, *offset*=134-200 = -66. Now, for the first transition $l_1 \xrightarrow{17} l_2$, d_n=-66+17=-49. Therefore, from Table 2, the value of ALDP or LDP for the transition is 1. So, instead of taking the actual LDP for duration 17 (which was 0.6 as computed earlier), we take the LDP for normalized duration. As o has plenty of time to reach l_7, the normalization makes the object less risky. After the next transition to l_3, d_n=-66+17+17 =-32. So, the ALDP after normalization is 1. Before the normalization, the ALDP was 0.05. However, after normalization the score says that the object is completely safe until that transition. Similarly, when o_1 reaches at l_7, the total stay duration is 178 and the normalized duration is 112. Thus, without normalization the ALDP is 0.02 and with normalization ALDP is 1. It shows that even if o_1 takes long for its transitions, the normalization marks it as a safe object as it has a long available time to reach its destination.

Adjusting Dur_{max} and Time Trigger. During processing of the ORP, $Dur_{max}(s_i)$ is adjusted to the concept of normalization. The normalized maximum allowable duration of an object o for completing its path sequence s_i is computed by $Dur_{maxN}(s_i, o)$ = $Dur_{max}(s_i)$−*offset*. As seen, if the value of *offset* is negative, then Dur_{maxN} allows more time for o_i. Besides, a higher value of *offset* will reduce the value of Dur_{maxN} for adjusting the riskiness of o. Finally, the corresponding time trigger is computed by $TT[o]_N$= t_{start}+Dur_{maxN} and is used for the risk prediction.

Finding the best thresholds. The optimal threshold depends on the particular goal of the system. We consider mishandled as the positive class for classification. A prediction system giving too many false positives (FP) or false negatives (FN) can make the system useless or not interesting. So, there should be a defined

acceptable metric for deciding the optimal operational threshold. We define a benefit function, where the costs for the different kinds of errors are used for finding the threshold that maximizes the benefit. For example, if a bag is predicted as mishandled, it requires a special manual handling so that the bag can reach the aircraft before the flight. If an FP occurs, there will be a waste of human resources for the mistake. However, if an FN occurs, there will be a significant cost for delivery, insurance, and other operating costs. So, in the baggage tracking scenario, the cost for an FN is much larger as compared to that for an FP. During model building and testing, we use Eq. (4) for obtaining the total benefit for each of the generated thresholds and use the threshold that provides the maximum benefit. In Eq. (4), x is the cost for handling a mishandled object (i.e., positive case (P)), y=cost for handling a predicted mishandled object (i.e., TP and FP), and #P is the total number of positive cases in the data set. So, Eq. (4) can provide an idea how much money can be saved by using the ORP system.

$$\textit{Benefit}(x, y) = x \times \#\text{P} - (x \times \#\text{FN} + y \times (\#\text{TP} + \#\text{FP})) \quad (4)$$

We report a comprehensive experimental study in [2]. The results show that the proposed method can identify the risky objects very accurately when they approach the bottleneck locations on their paths and can significantly reduce the operation cost. The risky objects are predicted early enough such that they can be saved before being mishandled.

4 Conclusion and Future Work

In this paper, we presented two approaches for analyzing the risk of indoor moving objects, primarily focusing on an RFID baggage tracking scenario. The first approach focused on the offline scenario where we proposed a data mining methodology for finding risk factors from the class imbalanced baggage tracking data. Our experiments [1] showed that the decision tree with under sampling provides the best model in our data set. The extracted patterns show that duration before flight is a critical factor and a bag is considered to be a high risk if it has less than 54 minutes in the transit airport. Moreover, departure airport, a longer stay at a location, and the busyness of the airport are also important factors. The second approach of this paper focused on the online scenario where we proposed an online risk prediction system for predicting risk of a bag in real-time so that it can be saved from being mishandled. Our experiments [2] showed that the proposed method can predict risk of a bag with more than 99% accuracy when it reaches in its bottleneck location (e.g., sorter). The proposed approach is also useful for other time critical multi-site based indoor tracking scenarios.

Several directions for future work exist. First, a more thorough study of the root causes for baggage mishandling, which are non-trivial, given the low probability of mishandled events. Second, the proposed online risk prediction technique can be extended to more general scenarios such as mixed indoor-outdoor object tracking. Third, it is useful to predict risks for the objects in nondeterministic scenarios, where the pre-planned paths are not available.

Acknowledgments

This work is supported by the BagTrack project funded by the Danish National Advanced Technology Foundation under grant no. 010-2011-1.

References

[1] T. Ahmed, T. Calders, and T. B. Pedersen. Mining risk factors in RFID baggage tracking data. In *MDM*, pages 235–242, 2015.

[2] T. Ahmed, T. B. Pedersen, T. Calders, and H. Lu. Online risk prediction for indoor moving objects. In *MDM*, pages 102–111, 2016.

[3] T. Ahmed, T. B. Pedersen, and H. Lu. A data warehouse solution for analyzing RFID-based baggage tracking data. In *MDM (1)*, pages 283–292, 2013.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 2002.

Toward Mining User Movement Behaviors in Indoor Environments

Shan-Yun Teng[1], Wei-Shinn Ku[2], Kun-Ta Chuang[1]
[1]Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
[2]Dept. of Computer Science and Software Engineering, Auburn University, USA
Email: syteng@netdb.csie.ncku.edu.tw, weishinn@auburn.edu, ktchuang@mail.ncku.edu.tw

Abstract

In this paper, we explore a new mining paradigm, called User Visited Patterns (abbreviated as UVP), to discover user visited behavior in the mall-like indoor environment. It is a highly challenging issue, in the indoor environment, to retrieve the frequent UVP, especially when the concern of user privacy is highlighted nowadays. The mining of UVP will face the critical challenge from spatial uncertainty. In this paper, the proposed system framework utilizes the probabilistic mining to identify top-k UVP over uncertain dataset collected from the RFID-based sensing result. Moreover, we redesign the indoor symbolic model to enhance the accuracy and efficiency. Our experimental studies show that the proposed system framework can overcome the impact from location uncertainty and efficiently discover high-quality UVP, to provide insightful observation for marketing collaborations.

1 Introduction

With the evolution of modern cities, the time duration staying in indoor spaces becomes longer for people living in metropolises. Indoor activities, such as window shopping in malls or indoor sports in gymnasiums, also become increasingly popular as leisure-time doings. The trend has led to an interest in identifying hidden patterns describing human behaviors in indoor environments [6, 7].

Motivated by this, we explore in this paper a practicably interesting task, named mining *User Visited Patterns*, to identify patterns which characterize the common sequences of visited spaces among users in indoor environments. For example, a *User Visited Pattern* p of the trajectory shown in Figure 1 is $\{S_7, S_1\}$. Specifically, the application need comes from the observation that people tend to linger away the whole evening at several locations (e.g., an underground market, a bookstore, and a coffee shop) after they get off work. On the weekend, people may spend the whole day at a mall or an outlet. The discovery of *User Visited Patterns* can enable new marketing collaborations among vendors in the same indoor space (e.g., a shopping center), such as a joint coupon promotion. Despite its increasing demand, mining *User Visited Patterns* is left unexplored thus far.

Essentially, the design of location-aware mining highly relies on the availability of precise user location information which can be transformed to the place/point of interest [5, 10]. However, it is not appropriate to deploy all devices within stores to retrieve precise user location as the concern of business security and personal privacy [2, 3]. In this work, we especially consider devices in the indoor spaces to be deployed as the illustration shown in Figure 1. It is, unfortunately, highly challenging to achieve the precise positioning in an indoor space due to hardware limitations and privacy concerns. The factor of location uncertainty will critically impact the effectiveness of traditional location-aware mining algorithms, causing the incorrect conclusions for marketing

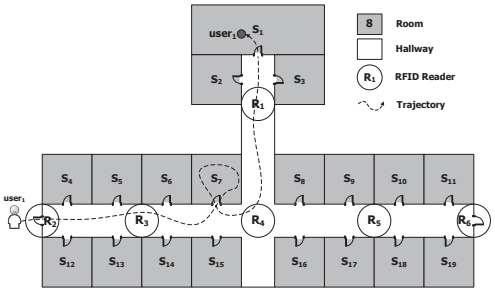


Figure 1: An illustrative example of indoor space and RFID reader deployment.

decisions. To remedy this, recent research advances in the literature have thus devoted to resolving the nature of indoor uncertainty, proposing methodologies of tracking user movements or identifying group moving behaviors in the indoor space [8, 9].

However, it is believed that maintaining the uncertain nature of user location is the key ingredient to the success of user-acceptable marketing strategies. The removal of location uncertainty will benefit the mining quality at the cost of user privacy. For mining *User Visited Patterns*, it is not a viable strategy to employ advanced indoor positioning media, which can precisely detect the location of a user. In this paper, we turn to employ the RFID-based framework as the indoor positioning media, accommodating to its error-prone characteristics in the algorithm design. To resolve the uncertainty problems, we propose a novel symbolic model to characterize the possible location of users, and design a probabilistic based mining algorithm to retrieve *UVP* in the indoor dataset.

The main contributions of this paper are three-fold:

- We present the concept to explore *User Visited Patterns* in the indoor environment.
- We comprehensively process the uncertainty of RFID raw data by proposing a novel symbolic model and probabilistic based mining method to enhance the performance of our framework.
- Empirical studies with synthetic data show the efficiency and accuracy of our framework.

In this work, we discuss the algorithm of mining UVP from the RFID-based uncertain data set in the indoor environment. The remainder of this paper is organized as follows. In section 2 and Section 3, we give the preliminaries and the system framework. The experimental results are conducted in Section 4. Finally, this paper concludes with Section 5.

2 Preliminaries

2.1 Indoor RFID Data

We describe indoor RFID raw data using an example shown in Figure 1. The indoor space is partitioned into several rooms and hallways. In addition, the numbered circles represent the reader detection ranges. Moving objects (human users in this paper) are attached with RFID tags. When a moving object u_i is within the sensing range of an RFID reader, its presence is detected and reported by the reader. Specifically, each raw RFID reading is in the format of $(deviceID, userID, t)$, which means the user represented by $userID$ is detected by the device identified by $deviceID$ at time t. For example, as shown in Table 1(a), user u_1 is detected by device R_2 at times t_1, which forms a raw reading r_1. For the trajectory of moving object u_1 illustrated in Figure 1, its indoor RFID data is shown in Table 1(a), where $readingID$ identifies a reading of an RFID reader.

Table 1: Illustrative examples of RFID raw reading data and User Visited Table.

(a) RFID raw reading data

readingID	deviceID	userID	t
r_1	R_2	u_1	t_1
r_2	R_2	u_1	t_2
r_3	R_3	u_1	t_6
r_4	R_3	u_1	t_7
r_5	R_4	u_1	t_{31}
r_6	R_4	u_1	t_{32}
r_7	R_1	u_1	t_{40}
r_8	R_1	u_1	t_{41}
r_9	R_1	u_1	t_{42}

(b) User Visited Table (UTT)

recordID	deviceID	userID	t_s	t_e
rd_1	R_2	u_1	t_1	t_2
rd_2	R_3	u_1	t_6	t_7
rd_3	R_4	u_1	t_{31}	t_{32}
rd_4	R_1	u_1	t_{40}	t_{42}

2.2 User Visited Table

We consolidate the raw reading data and represent it as a *User Visited Table* with schema ($recordID$, $deviceID$, $userID$, t_s, t_e). Each record in the table states that a moving object $userID$ is continuously detected by the device $deciceID$ in the period from time t_s to time t_e. In addition, $recordID$ identifies a record. For example, as shown in Table 1(b), user u_1 is detected by device R_2 starting from time t_1 to t_2. The *User Visited Table* transformed from Table 1(a) is represented in Table 1(b). Finally, with the records in the *User Visited Table*, a trajectory $T_{u_i} = \{rd_1, rd_2, ..., rd_x\}$ can be formed for a particular moving object u_i.

2.3 Problem Formulation

We give the necessary definitions as follows.

Definition 1 (User Visited Event). *Suppose that a moving object u_i is detected by the RFID reader R_j starting from time t_{sj}, and R_j continues sensing the appearance of u_i until time t_{ej}. Afterwards, u_i keeps being detected by reader R_k, starting from time t_{sk} and ending with time t_{ek}. Thus, we define a User Visited Event e_n of u_i, which is denoted by the 3-tuple $(R_j, R_k, t(e_n))$. And $t(e_n)$ denotes the time interval $[t_{ej}, t_{sk}]$ from R_j to R_k. In this paper, we call User Visited Event as event for short.*

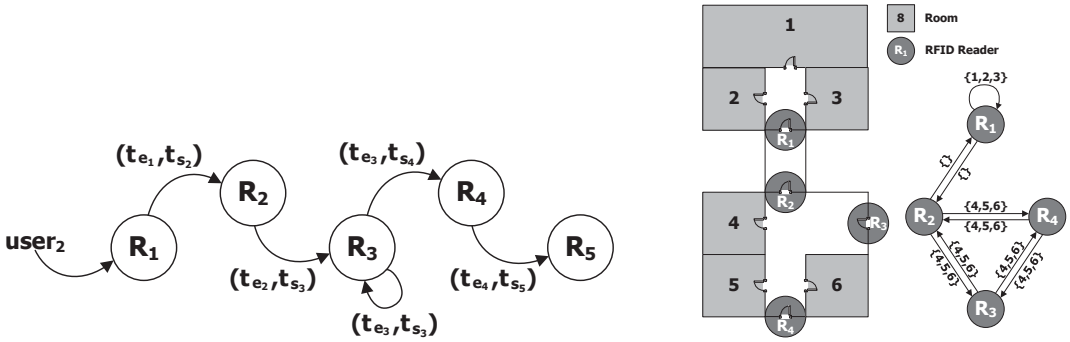
Definition 2 (User Visited Path). *Suppose that u_i is successively detected by R_1, R_2, ..., R_m, where R_m is the last reader in the system that senses the appearance of u_i. The corresponding User Visited Path, abbreviated as path, is denoted by $a_i = \{e_1, e_2, ..., e_n\}$, where e_k is an event, for $1 \leq k \leq n$.*

The illustrative example of the path of user u_2 is shown in Figure 2(a). In addition, instead of following the same principle which is generally used in previous works, we assume no readers are deployed in the room space due to the privacy issue. It is relatively easy to recognize if a user has a valid visit (or window shopping) in a room. The manager of a store can tell the visited time by their experience. For example, customers staying within the store longer than 3 minutes can be identified as valid. As such, we give the following definition.

Definition 3 (Valid State in Rooms). *For a room S_l, we can define the time stayed in S_l as valid, i.e., $t_{stay}(S_l) = [t_{min}(S_l), t_{max}(S_l)]$, where $t_{min}(S_l)$ and $t_{max}(S_l)$ are the minimum time and maximum time that we can state a user staying in room S_l, respectively. In general, $t_{max}(S_l)$ can be defined as infinite.*

Definition 4 (Uncertain Visited Rooms). *Given an event $e_i = (R_j, R_k, t(e_n))$, Uncertain Visited Rooms $uvr_{j,k}$ can be defined as the set of rooms which are placed in the area between readers R_j and R_k. As such, $uvr_{j,k} = \{S_{i,1}, S_{i,2}, ..., S_{i,n}\}$, where each room $S_{i,n}$ is a possible space that a user may get into when the event e_i occurs. In this paper, we state Uncertain Visited Rooms as U-room.*

Definition 5 (Uncertain Visited Transaction). *Given the U-room set of any event in the path a_i, we can*



(a) A user visited path (b) A symbolic model

Figure 2: Illustrative examples of a user visited path and a symbolic model.

completely transform the RFID sensing data to the transaction tr_i consisting of U-room sets, i.e.,

$$tr_i =< \{S_{1,1}, S_{1,2}...S_{1,k_1}\}, ..., \{S_{m,1}, S_{m,2}...S_{m,k_m}\} > .$$

In this paper, we state the Uncertain Visited Transaction as U-transaction.
Problem Formulation (Top-k User Visited Patterns Discovery): Suppose that a *User Visited Pattern* (abbreviated as UVP in the sequel) is defined as the form:

$$p_i = \{S_{i,1}, S_{i,2}, ..., S_{i,k}\},$$

and its support value $f(p_i)$ equals to its expected occurrence count in U-transactions. Given the uncertain database D and the desired number k, the goal of our framework is to discover top-k *UVP* from U-transactions according to the expected support of each *UVP*.

3 System Framework

3.1 Symbolic Model Design

Symbolic model is a graph that describes the topology of indoor spaces in which each separating space such as a room or a hallway is represented as a vertex [4]. In addition, edges capture the connectivity (undirected graph) or accessibility (directed graph) between two vertices. In our work, all readers are placed in the hallways, and the specific room for a user getting into is invalid. As such, our symbolic model can not be the same as the traditional one [4]. To solve the uncertainty issue, our symbolic model is designed as follows: each node represents a reader R_i in the indoor environment, the edge connecting nodes R_i and R_j is labeled as the U-room set $uvr_{i,j}$ between readers R_i and R_j.

A small indoor deployment and its corresponding symbolic model are illustrated in Figure 2(b). Our symbolic model is built as a simple digraph in which one loop may be presented at each vertex, and each ordered pair is the head and the tail of at most one edge.

3.2 User Visited Event Filtering

With the reading records in the user visited table provided from indoor RFID readers, a path $a_i = \{e_1, e_2, ..., e_n\}$ can be formed for a particular moving object u_i. However, not all of the events in this path are useful and meaningful in this work. For some particular cases, the events can be filtered before executing the mining process.

No-Stay Event: As shown in Figure 1, u_1 only goes through the RFID readers R_2 and R_3 without walking into any room. Since our goal is to discover the top-k *UVP*, an event is useful only when any space information can be retrieved. However, for a few events, the users only go through the RFID readers without staying in rooms. Therefore, given an event $e_i = (R_j, R_k, t(e_i))$, we define a $DT_{max}(R_j, R_k)$ as the maximum time of walking from R_j to R_k. If the dwell time $t(e_i)$ of an event e_i is not longer than $DT_{max}(R_j, R_k)$, we can remove this event from the user path.

No-Space Event: In Figure 1, the user u_1 passes through the readers R_4 and R_1. However, the set of U-room set between readers R_4 and R_1 is empty. Therefore, we can also remove such events from the user path as no space information can be retrieved.

Path Split to Transactions: For a path of a moving object u_i, we define a time period t_{split} to obtain a set of transactions. As can be expected, a moving object u_i may leave the indoor environment, and return another day. Occasionally, its presence keeps being detected by the indoor readers and is recognized as in the same path as last time it stayed in the indoor environment, since the RFID tag of u_i is not different. However, if u_i leaves the indoor environment for a long time period (e.g., half a day), the path should be cut off. As a result, we use t_{split} to identify a breaking point to split a path into transactions. If the dwell time $t(e_n)$ of an event e_i is longer than t_{split}, the event e_i is removed from the path, and events e_{i-1} and e_{i+1} are separated into two different transactions.

Finally, based on the proposed symbolic model, we transform each event to its U-room set and generate the U-transactions of users for mining process.

Algorithm 1 $\mathcal{P}$-Apriori

Desc.: X_l:a candidate itemset of size l; L_l:an itemset of size l; x_i: a candidate x_i; $p(x_i)$: expected support of a candidate x_i;

Input: D: U-transactions tr_m; k: top-k; f_{min}: support threshold;

Output: UVP: top-k UVP;

1: **procedure** $\mathcal{P}$-APRIORI(D,k,f_{min})
2: $L_1 := \{items\}$;
3: **for** $(l = 1; L_l! = \emptyset; l$++**)** **do**
4: X_{l+1}:= candidates generated from L_l;
5: **for** $(m = 1; m <= |D|; m$++**)** **do**
6: **for** each $e_i \in tr_m$ **do**
7: **for** each $x_i \in C$ **do**
8: **if** $e_i.uvr$ contains $x_j \in X_{l+1}$ **then**
9: compute $\mathcal{P}(e_i, x_j)$;
10: $f(x_j) = f(x_j) + \mathcal{P}(e_i, x_j)$;
11: remove x_i with $f(x_i) \leq f_{min}$ from X_{l+1}
12: L_{l+1}:= candidates in X_{l+1};
13: P:= $P \cup X_{l+1}$;
14: P:=sort(P);
15: UVP:=$\{p_1, p_2, ..., p_k\} \in P$;
16: **return** UVP

3.3 UVP Discovery

To retrieve top-k *UVP* from U-transactions of users, we extend the Apriori algorithm [1]. As aforementioned, U-room set $uvr_{j,k} = \{S_{i,1}, S_{i,2}, ..., S_{i,n}\}$ of event $e_i = (R_j, R_k, t(e_n))$ containing possible spaces that a user

Table 2: Simulator parameters.

Parameters	Settings
Number of objects	[2,000-20,000]
Radius of RFID detection range	1.5 meter
Moving objects' speed distribution	μ=1 m/s and σ=0.1
Reader accuracy	95%

may get into when event e_i happens. Therefore, we define a possible combination set C consisting of room sets $c_x = \{S_1, S_2, ..., S_y\}$, for each $S_y \in uvr_{j,k}$. Each combination $c_x \in C$ has a probability presenting in event e_i for the real world, and the probability can be defined as

$$\mathcal{P}(e_i, c_x) = \begin{cases} 1 - \frac{|t(e_i) - (DT_{max}(R_j, R_k) + \sum_{S_y \in c_x} t_{min}(S_y))|}{t(e_i)}, & \mathcal{P}(e_i, c_x) \geq 0. \\ 0, & otherwise. \end{cases} \quad (1)$$

With this probabilistic model, the mining process is described in Algorithm 1. It takes U-transactions and the top-k as input and returns top-k *UVP*. The 1-item set is firstly computed (line 2). Then, all candidates of X_{l+1} (the candidate itemset of size $l+1$) produced from L_l (the itemset of size l) are generated, and only candidates with expected support values exceeding the threshold are maintained (lines 3-13). The expected support of each candidate is to sum up all probabilities of its occurrence count in U-transactions (lines 8-10). Finally, all pattern candidates are sorted based on support values, and the patterns with top-k support values are returned as *UVP* (lines 14-16).

4 Experimental Results

This section presents experimental studies of this research. The system framework is implemented in Java. All of the experiments are executed on a 3.40 GHz Core i7 machine with 4 GB of main memory, running on the Windows 7 operating system.

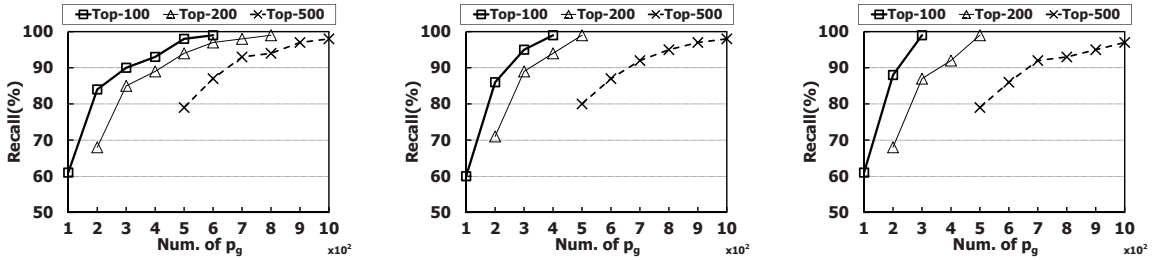
4.1 Experimental Setup

4.1.1 Simulator Implementation

In this paper, the synthetic data of raw RFID reader is generated by applying an indoor raw data generator. This generator, running on the Linux operating system, can simulate the user walking behavior in the indoor environment with the given room layout. The whole simulator consists of true trace generator and raw reading generator. The true trace generator module is used for generating the ground truth traces of moving objects and records of the precise location of each object in every second. It also simulates the objects' speeds using the Gaussian distribution, and the parameters μ and σ of this Gaussian distribution are set to 1 m/s and 0.1, respectively.

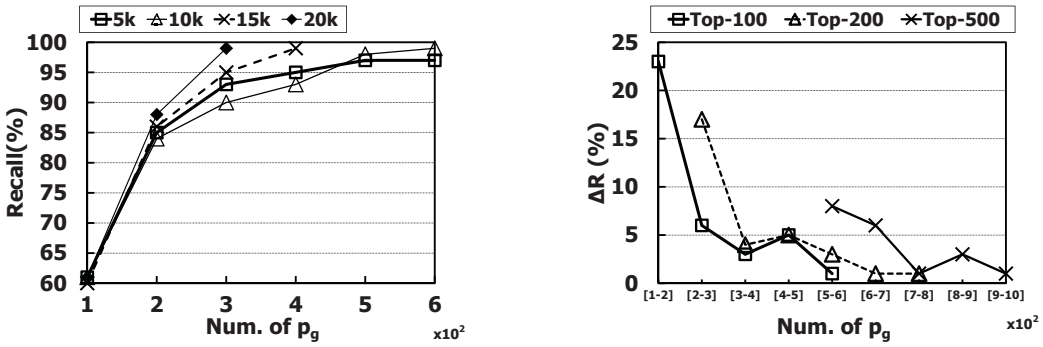
In this indoor spatial simulator, we apply a layout of a shopping mall, which is a real MRT-based underground market[1]. In this mall, there are 27 entrances/exits and 168 stores that sell various items (e.g., food, drinks, clothes, shoes, and so on). The generator simulates normal customer purchasing behavior in the indoor environment with 50 RFID readers evenly placed in the hallways. In addition, the dwell time is randomly assigned to simulate the stop-by period spent in a store for a user. The parameters used by this simulator are described in Table 2. Finally, the simulator generates two datasets of user trajectories, that are treated as the

[1]We refer to http://www.datong.taipei.gov.tw/public/Attachment/13811321925.jpg for the layout of Taipei mall.



(a) $|D| = 10,000$ (b) $|D| = 15,000$ (c) $|D| = 20,000$

Figure 3: The recall of the top-k UVP.



(a) The recall of top-k UVP (b) The recall difference of top-k UVP

Figure 4: The recall and the recall difference of top-k UVP.

raw reader data and the ground truth data with the precise locations of moving objects. Therefore, we have both ground truth and raw reader data for performance evaluation.

4.1.2 Evaluation Metrics

Recall: In this paper, the fraction of the top-k ground truth *UVP* p_t that is relevant to the generated *UVP* p_g is successfully retrieved. It cares about the probability that a relevant pattern is retrieved, so as to our work, the recall is defined as $\frac{|p_g \cap p_t|}{|p_t|}$.

4.2 Evaluating Our Method

4.2.1 Accuracy performance

We fix the number of trajectories ranging from 100,000 to 200,000 and vary the number of top-k to see the accuracy. The results on the recall of top-k *UVP* are shown in Figures 3(a), 3(b), and 3(c). Obviously, the recall rate is monotonically increasing as the number of generated patterns raises. In addition, two important points are drawn: (1) When the same number of p_g are generated as p_t, the recall value remains within a narrow range of a specific percentage value. (2) These curves stated as the same number of p_t have similar growing trends. It is clear that the number of transactions does not have an obvious impact on the recall rate.

Furthermore, we fix the number of p_t to 100, and then vary the size of trajectories $|D|$ from 5,000 to 20,000. The results on the recall of frequent *UVP* are shown in Figure 4(a). As it can be observed, we have to generate

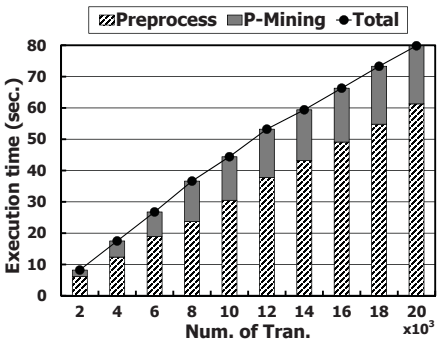


Figure 5: The execution time of preprocess, $\mathcal{P}$-Apriori, and total process.

more *UVP* to reach the recall value of 99% as the size of trajectories decreases. The reason comes from the fact that $\mathcal{P}$-Apriori process could compute the wrong expected support of items in the whole transactions, which would directly affect the support ranking of items and the consequence of top-k *UVP*.

Third, we fix the number of trajectories to 10,000, and then vary the number of p_t ranging from 100 to 500. The difference of recalls with respect to Figure 3(a) are shown in Figure 4(b). It is clear that these recall difference lines decrease faster at first than in the end, which states that the recall curve lines increase faster at first than in the end. As shown in Figure 3, the recalls of curves' starting points are all over 60%. It shows that most of the p_t will be retrieved at the beginning. Then as the number of p_g increases, the fewer p_t we can get. Clearly, the proposed system framework can retrieve high-quality results in the uncertain environment.

4.2.2 Execution time analysis

In this section, we try to analyze the execution time of our system framework. We vary the size of $|D|$ from 2,000 to 20,000, and demonstrate the execution time of the preprocess (including symbolic model construction and event filtering), the $\mathcal{P}$-Apriori, and the entire processing time. As we can observe in Figure 5, the execution time of preprocess is presented as a perfectly linear increasing curve. With regards to the mining efficiency of $\mathcal{P}$-Apriori, it is clear that for a small number of transactions, the execution time is increasing with the significant difference. However, when the number of transactions is huge, the execution time is increasing in an ignorable difference, which shows the stability of our system framework. Finally, the curve of total execution time is linear as that of preprocess. Because the time of preprocess ranging from 6 to 61 is much bigger than that of $\mathcal{P}$-apriori ranging from 2 to 18. Obviously, the time of preprocess dominates the execution time of our system framework.

5 Conclusions

In this paper, we explore the *User Visited Patterns* to identify the indoor mining challenges over uncertain data. Due to the concern of the user privacy, the placement of readers in hallways instead of rooms is considered. We devise a novel system framework to discover the top-k *UVP* from user paths. In addition, we also explore a novel symbolic model and a probabilistic based mining algorithm to efficiently discover the frequent *UVP*. Finally, the framework is studied with empirical observations in order to gain insight into the recall of generated *UVP*. The results show that the proposed framework is effective and efficient to retrieve high-quality patterns.

Acknowledgments

This paper was supported in part by Ministry of Science and Technology, R.O.C., under Contract 105-2221-E-006-140-MY2. The research has also been funded in part by the U.S. National Science Foundation grants IIS-1618669 (III) and ACI-1642133 (CICI).

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *International Conference on Very Large Data Bases*, 1994.

[2] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-aware location sensor networks. In *Hot Topics in Operating Systems*, 2003.

[3] J. P. Hansen, A. Alapetite, H. B. Andersen, L. Malmborg, and J. Thommesen. Location-based services and privacy in airports. In *International Conference on Human-Computer Interaction*, 2009.

[4] C. S. Jensen, H. Lu, and B. Yang. Graph model based indoor tracking. In *IEEE International Conference on Mobile Data Management*, 2009.

[5] Y. Liu, Y. Zhao, L. Chen, J. Pei, and J. Han. Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays. *IEEE Transactions on Parallel and Distributed Systems*, 2012.

[6] H. Lu, C. Guo, B. Yang, and C. S. Jensen. Finding frequently visited indoor pois using symbolic indoor tracking data. In *International Conference on Extending Database Technology*, 2016.

[7] H. Lu, B. Yang, and C. S. Jensen. Spatio-temporal joins on symbolic indoor tracking data. In *IEEE International Conference on Data Engineering*, 2011.

[8] L. Radaelli, D. Sabonis, H. Lu, and C. S. Jensen. Identifying typical movements among indoor objects - concepts and empirical study. In *IEEE International Conference on Mobile Data Management*, 2013.

[9] J. Yu, W.-S. Ku, M.-T. Sun, and H. Lu. An rfid and particle filter-based indoor spatial query evaluation system. In *International Conference on Extending Database Technology*, 2013.

[10] R. Zhang, Y. Liu, Y. Zhang, and J. Sun. Fast identification of the missing tags in a large rfid system. In *IEEE International Conference on Sensing, Communication and Networking*, 2011.

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

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

Abstract

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

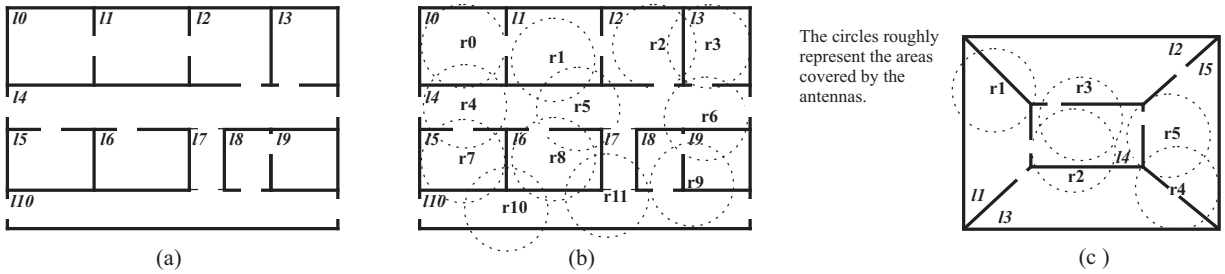
1 Introduction

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

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

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

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



(a) (b) (c)

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

a crucial problem is determining how likely these trajectories are.

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

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

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

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

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

The trajectory cleaning problem. In this work, we address this problem: given a sequence Θ of readings and the a-priori pdf $p^a(l|R)$ (and thus $p^a(t|\Theta)$), revise $p^a(t|\Theta)$ (which relies on the independence assumption) to properly take into account the known correlations between time points, thus assigning more "reasonable"

probabilities to the trajectories. Intuitively, this can be seen as a cleaning problem: the data to be cleaned are the probabilistic trajectories representing the interpretations of Θ, and the cleaning task consists in revising the probabilities assigned by $p^a(t|\Theta)$.

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

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

$b)$ *probabilistic conditioning*: it will be used to revise the probabilities of the trajectories.

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

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

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

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

2 Cleaning through conditioning: the challenges.

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

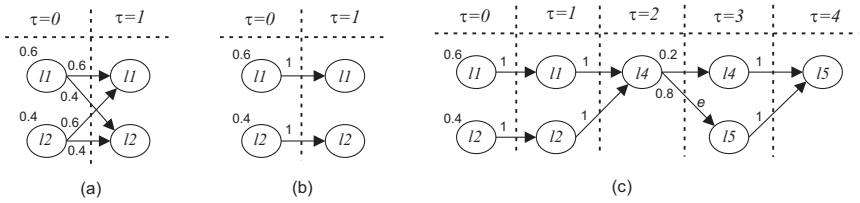


Figure 2: Graphs G' (a), G'' (b), and G (c)

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

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

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

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

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

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

3 Our approach

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

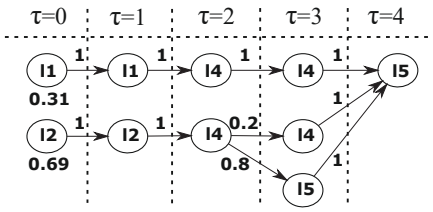


Figure 3: Graph encoding all and only the consistent trajectories

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

3.1 The algorithm in detail

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

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

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

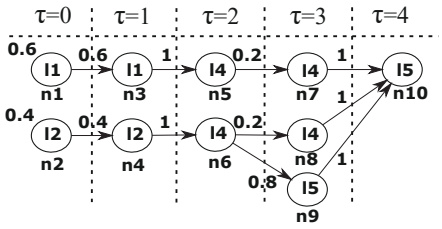


Figure 4: The ct-graph under construction at the end of the forward phase

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

4 Related Work

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

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

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

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

5 Conclusions

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

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Towards Ubiquitous Indoor Spatial Awareness on a Worldwide Scale

Moustafa Elhamshary and Moustafa Youssef
The Wireless Research Center & Department of Computer Science and Engineering
Egypt-Japan University of Science and Technology (E-JUST)

Abstract

While a remarkable effort has been put in developing indoor spatial awareness systems, they are still isolated efforts that are tailored to specific deployments. A truly ubiquitous indoor spatial awareness system is envisioned to be deployed on a large scale worldwide, with minimum overhead, and to work with the heterogeneous IoT devices. Such a system will enable a wide set of new applications including worldwide seamless direction finding between indoor locations, anywhere anytime health monitoring, enhanced first responders' safety, and providing richer context for indoor mobile computing applications.

In this paper, we describe our vision and work towards achieving ubiquitous indoor spatial awareness systems as well as the open challenges that need to be addressed to materialize this dream.

1 Introduction

The advancements in mobile devices coupled with wireless communication have opened the door for a myriad of context aware applications for indoor spaces including activity recognition, vital signs monitoring, mood detection, among many others. Of these, location is considered one of the main context information which, not only provides valuable information about the user herself, but also tags all other extracted context information with where they occurred. Such context information enriches the user spatial awareness and enables a new set of applications such as seamless navigation between indoors and outdoors, indoor analytics, indoor geographic Information Systems (GISs), first responders' safety, evacuation planning, indoor E-911, among many others.

Current efforts for indoor spatial awareness usually target designing systems for a specific purpose or a specific environment, which limits their scalability and ubiquitous deployment on a worldwide scale. In this paper, building on the pioneering work in the area, we envision a truly ubiquitous indoor spatial awareness system that can be deployed anywhere around the world, with minimum overhead, and that works with the heterogeneous Internet of Things (IoT) devices. For this, we provide a vision and possible implementation for two main components for enabling such a system. In particular, we describe systems for enabling indoor localization as well as device-free sensor-less sensing, both on a worldwide scale. For the indoor localization component (Section 2), we argue that the lack of a worldwide indoor floorplan database is one of the main hurdles facing worldwide indoor localization. To address this, we propose to leverage the ubiquitous sensor-rich mobile devices that are available with their users 24/7 to collect motion traces and analyze them to estimate the building floorplan shape as well as higher-level semantic information.

For the device-free sensor-less sensing (Section 3), we propose leveraging the already-installed wireless infrastructure to estimate different context information, *without using any special hardware*. Specifically, we show how one can identify different context information based on analyzing the changes of the RF signals due to

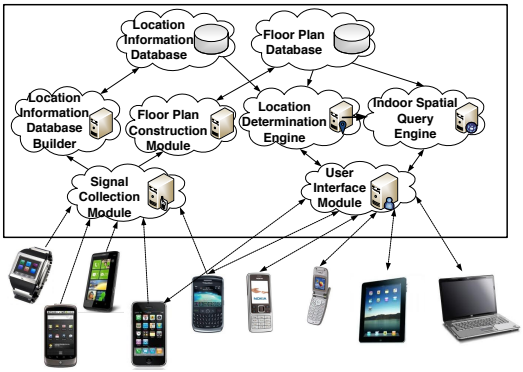


Figure 1: Architecture of a ubiquitous indoor spatial awareness system (Figure from [38], included here by permission).

the presence of humans or objects in an area of interest. We finally conclude the paper with different directions for extensions and open research challenges in Section 4.

2 Indoor Localization on a Worldwide Scale

One of the most commonly used context information is the user location. The user location provides not only where she is, but combined with other sensors can also provide information about different aspects of her preferences or conditions. A number of systems have been proposed over the years to provide indoor localization using different technologies such as infrared [36], ultrasound [29], various RF technologies [8, 37, 40, 41], magnetic field [3, 11], and light [27], etc. However, these systems usually target specific deployment areas, require special hardware, and/or require a tedious calibration process. We are still missing a ubiquitous indoor localization technologies, similar to GPS outdoors, that works virtually in any building worldwide, with minimal overhead, and supports the heterogeneity of IoT devices. This is reflected on the commercial domain, where indoor localization from the big players are limited, mainly based on manual efforts and focused on specific buildings. For example, in late 2011, Google Maps started to provide detailed floorplans for a few malls and airports in the US and Japan. In an independent effort, Esri offered their solutions for indoors for a number of case studies. Nevertheless, all these systems depend on manually building the floorplan and collecting the point of interest (PoIs). Manual addition/editing of all buildings floorplans in a large university campus or an entire city requires an enormous cost and effort which may be unaffordable. In addition, keeping these floorplans up to date is another challenge.

We argue that, in order to realize a truly ubiquitous indoor localization system, there are two main challenges: (1) providing indoor floorplans and a rich PoIs database on a world-wide scale and (2) providing accurate and ubiquitous indoor localization. The lack of indoor floorplans on a world-wide scale can be attributed in part to the manual effort required to populate this PoI database and upload the floorplans (if they exist) to a central server, privacy concerns, or lack of incentives. The unavailability of accurate indoor localization on a world-wide scale may be due to the calibration effort required to construct fingerprints for indoor localization systems such as WiFi-based localization system [40].

To address these two challenges, we envision a crowd-sensing system that leverages the sensor-rich always-on mobile devices to automate these two tasks [38]. Figure 1 shows a possible system architecture for realizing such system. Specifically, sensor information from mobile devices of users of a specific building are collected and sent to the cloud for processing. The user motion traces are extracted from the inertial sensors and are analyzed to estimate the building floorplan shape as well as other points of interests and higher layer semantics. Concurrently, other relevant signals for localization, such as WiFi APs or BLE beacons information, are used to

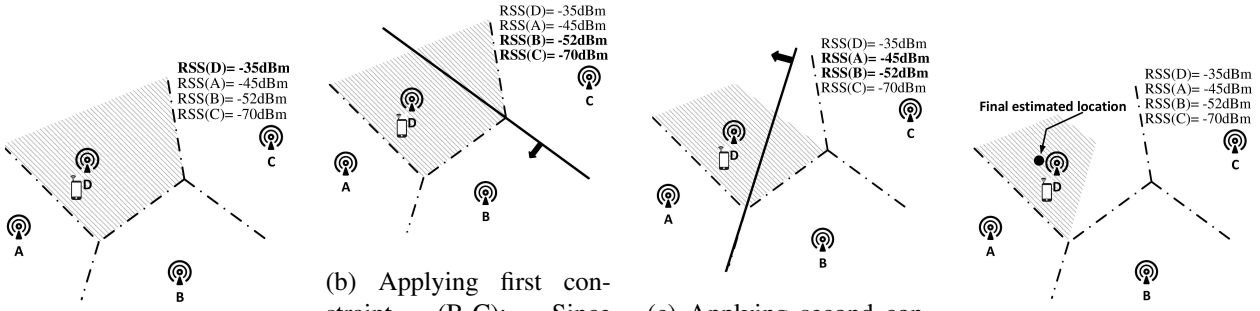
construct the required databases for localization. Other users can then query the system to obtain the floorplan of the building they are located in, in addition to estimating their location.

In the rest of this section, we give some insights about some projects that pioneered the work in automatic semantic construction of indoor floorplans and ubiquitous localization.

2.1 Automatic Construction of Semantic-rich Floorplans

The goal of this module to estimate the overall floorplan shape and the rooms and corridors layout. The CrowdInside system analyzes the collected motion traces to estimate the overall floorplan shape [9]. The idea is that areas with user traces should correspond to rooms and corridors, while blocked areas map to walls. First, it obtains a point cloud by mapping each user step to a point. Then, it applies computational geometry techniques, the alpha shapes, to estimate the floorplan layout. To separate rooms from corridors, it divides each user trace into segments at each major change in direction, e.g. a turn. After that, a classifier is used to separate the different segments based on features such as the user speed, points density, among others. Once separated, a clustering operation is applied to the points corresponding to the rooms segments, using WiFi to separate adjacent rooms. Finally, alpha shapes are applied recursively to each cluster to estimate the room shape. Many points of interest (PoIs) such as elevators, escalators, and stairs can be estimated based on their unique signature on the different mobile device sensors. Different follow-up systems have suggested using other sensors, such as the camera [20], or WiFi [22] to complement inertial sensors. In addition, other systems addressed the problem of extracting higher layer semantics, e.g. the venues names or categories in a shopping mall or landmarks in transit stations. In particular, the *CheckInside* system [15, 18] combines the phone sensors with information publicly available from location-based social networks (LBSNs) such as Foursquare, to estimate the labels of different venues inside a mall. The intuition is that the check-in operation of the LBSN provide hints about the location the user is located at. However, due to the inherent inaccuracy of indoor location determination systems and human errors in the check-in operation, whether unintentional or malicious, these labels may not match the actual user location. To address this issue and provide high accuracy in labeling the venues, the system construct a multi-modal sensor-based semantic fingerprint of the different venues. This fingerprint is used to determine the venue with fingerprint that has the closest match to the current user collected sensors. The *SemSense* system [17] further extends this approach to estimate the venue categories, e.g. restaurant or clothing shops, even for venues that are not popular. The idea is that, based on economic geography research, design of the malls divides the available space into homogeneous blocks, where venues of the same category are grouped together [34]. *SemSense* maps the venues category estimation problem into a graph theory problem, where nodes reflect venues and edges represent the physical neighborhood relation between venues. Different category scores are assigned to each node using an extended Topic Specific PageRank algorithm [21]. Only venues with a category scores above a threshold are labeled to reduce the noise effect. The resulting floorplan can be used to increase the LBSNs venues coverage ratio as well as allowing hierarchical look-up by organizing labels in a hierarchical manner (e.g. venue name (McDonaldas or Starbucks), fine-grained category (restaurant or coffee shop), or coarse-grained category (food place)).

The *TransitLabel* system [19], on the other hand, targets labeling the semantics in transit stations including detecting the locations of different ticketing and vending machines, gates, restrooms, lockers, platforms, among others. The idea is that the usage of these semantics present identifiable signatures on one or more cell-phone sensors, e.g. the beep sound of a ticket vending machine [16]. This observation is leveraged to automatically recognize these activities, which in turn are mined to infer their uniquely associated stations semantics (e.g. ticketing machine). Furthermore, the locations of the discovered semantics are automatically estimated as the center of mass of inaccurate passengers' positions when these semantics are identified given that the average of independent noisy samples should converge to the actual location based on the law of large numbers.



(a) Initial step: User initial ambiguity region is the Voronoi cell corresponding to the AP with the highest RSS.

(b) Applying first constraint (B-C): Since RSS(B)>RSS(C), the user location falls in the half plane defined by the bisector line between B and C and containing B.

(c) Applying second constraint (A-B): Similarly, the user falls in the half plane defined by the bisector line between A and B and containing A.

(d) Final location estimation: The user location is estimated as the center of mass of the last ambiguity region.

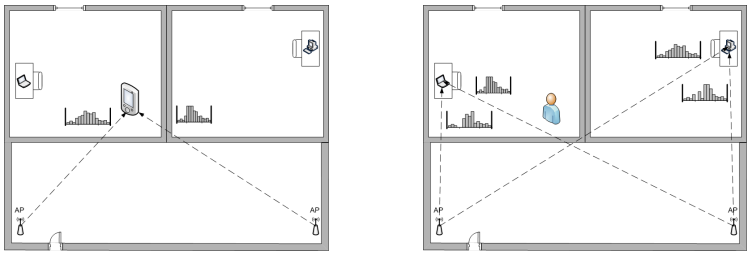
Figure 2: Example of the IncVoronoi basic approach with four APs (Figure from [12], included here by permission).

2.2 Ubiquitous Indoor Localization

RF-based localization, e.g. using WiFi or BLE [10, 12], has a potential for providing ubiquitous indoor localization as it uses the already existing wireless infrastructure. However, due to the wireless channel noise and complex propagation environment, they usually depend on constructing a radio map of the area of interest, where the signature of the APs at different locations are stored during a calibration phase. Radio-map based techniques can provide good localization accuracy. Nevertheless, constructing the radio map is a labor-intensive process that needs to be repeated if the environment changes for different devices, reducing the technology scalability to a worldwide scale.

To address this, researchers have proposed a number of approaches to reduce or eliminate the calibration overhead. To reduce the calibration overhead, propagation modeling-based approaches, such as [10], or active user crowd-sensing are used [28]. However, this comes at the cost of reduced accuracy or overhead on the user. Passive crowd-sensing, where the user is not involved has been proposed in [4], where dead reckoning based on the inertial sensors are used to estimate the user position and use it for automatic radio map construction. To reduce the error accumulation in the dead reckoning process, a Simultaneous Localization and Mapping (SLAM) framework is used that combines the motion model from the inertial sensors with an observation model of the different semantics extracted from the multi-modal phone sensors. This leads to a high-accuracy localization system with a low overhead on the user side. Moreover, since the system is crowd-sensing based, keeping the radio map up to date with environment changes is accounted for.

To completely eliminate the calibration overhead and maintain robustness over environment changes, devices, and different APs transmit power, calibration-free techniques have gained momentum recently. For example, the *IncVoronoi* system uses the relative relation between the received signal strength (RSS) of each pair of APs to determine the user location [12]. The idea is that, for a given pair of APs, the user will be located closer to the strongest AP than the other. Based on this, it starts by estimating the location of the user as the location of the Voronoi cell with the strongest RSS constructed using APs as seeds. To further reduce the ambiguity region and refine the user location, the relative relation between each pair of heard APs is used as constraints to reduce the ambiguity region recursively (Figure 2). Different modules are presented to handle practical consideration such as the noisy wireless channel, obstacles in the environment, different APs transmit power, among others.



(a) Active (b) Passive

Figure 3: Difference between active and passive radio maps construction (Figure from [33], included here by permission).

3 Device-free Sensor-less Sensing

Traditional sensor networks have been based on special hardware, i.e. sensors, that are attached to the object or installed near the phenomena being monitored. With the blooming of wireless networks at different scales ranging from personal Bluetooth, through local wireless networks such as WiFi, to city-wide cellular networks; it seems natural to try to leverage them for ubiquitous sensing on a worldwide scale. Specifically, the concept of device-free passive (DfP) localization has been introduced [42] to leverage standard RF networks to detect, track, and identify objects without them carrying any devices nor participating actively in the process. The idea is that the presence of humans or objects in an RF environment affects the received signal, which can be analyzed to sense different characteristic of the objects and the environments. For example, a wireless network used during the day for browsing the Internet or checking one's email, can be used without using any extra hardware as an intrusion detection system during the night; just by monitoring and analyzing the changes in the received signal strength at the different devices already installed in an area of interest. This opens the door for a large class of novel applications such as border protection, smart homes, gesture recognition, mobile health care, among many others. The fact that these applications can be deployed using standard RF networks, with no additional hardware, makes them a good candidate for future spatial awareness systems on a worldwide scale.

In the balance of this section, we describe the different functionalities of device-free passive detection, tracking, and identifications.

3.1 Detection

Binary detection, i.e. detecting whether there is someone inside an area of interest or not, is an important function for a number of applications such as intrusion detection and smart homes. For example, in a smart home settings, the device-free passive detection system can detect that there is no one currently in the house and turns off the light to save energy. Detection functionalities usually serve as an enabler for the other functionalities, e.g. enabling the DfP tracking module once an entity has been detected. To implement a DfP detection system, one needs to differentiate between a silence period, where no one is inside the area of interest, and a busy period, where an activity is detected. For this, simple techniques, e.g. based on estimating the variance of the signal have shown high accuracy in detection [42]. More advanced techniques, e.g. [25, 32], target robust detection in different environments and changing conditions by continuously updating the system parameters.

3.2 Tracking

Once an entity has been detected, the next step is to track its location in the area on interest. Note that this tracking is performed without the entity carrying any devices nor actively participating in the tracking process but based on its effect on the received signal strength. However, since the wireless channel is highly noisy in

indoor environments, there is no simple expression that can capture the relation between the RSS and distance. Traditionally, device-based active localization systems, e.g. [43] used a radio map to capture the different APs signatures at different locations in the area of interest. DfP tracking systems use a similar approach. However, in contrast to an active radio map, where a device is placed at the different calibration points, a passive radio map is constructed based on measuring the effect of a person standing at the different calibration points on the signal strength received at the infrastructure devices (Figure 3). Recently, there has been some effort in reducing the passive radio map construction effort by either automatically constructing the passive radio map through ray tracing [7, 13] or using models for the signal strength [14].

3.3 Identification

The most exciting and most challenging DfP functionality is the identification function. This refers to characterizing the detected object or the environment including applications such as traffic estimation, gesture recognition, mobile healthcare, emotion detection, among many others. For example, the ReVISE system [5, 23] leverages the changes in the RSS of a transmitter and receiver installed along the side of the road to differentiate between pedestrians and cars as well as estimate the car speed. The concept of device-free identification has been leveraged for a number of mobile healthcare applications. For example, [31] presents a system for fall detection based on the variance of the RSS received from RFID tags at the RFID reader. More advanced signal processing techniques have been applied to extract the breathing and heart rates [1, 26]. The idea is that the motion of chest and heart during breathing or beating modulates the RF signal, which can be processed to separate and extract the breathing and heart rates. More recently, a number of novel applications have been proposed based on the DfP concept. For example [24, 30] propose to analyze the heart and breathing signals further to estimate the user emotion. WiGest [2] analyzes the changes of the ambient WiFi signals to turn any WiFi-enabled device into a gesture recognition system. Other systems leverage the more fine-grained channel state information (CSI) to learn detailed information about the object such as the pressed key on a virtual keyboard [6], whether someone is smoking or not [44], and spy on spoken words [35].

4 Conclusion

Indoor spatial awareness on a worldwide scale is still a challenging problem that is open for new contributions. Some of the possible directions for enhancements include identifying higher layer semantic maps, such as the owner of a room. Formal modeling of the mapping process, e.g. to obtain bounds of the possible achievable accuracy is another direction for research. User incentives techniques, such as using gamification, as well as techniques to preserve the sensitive buildings privacy are open areas for contribution. Analyzing the redundancy in buildings within the same complex or floors within the same building may have interesting challenges and impact on reducing the system overhead or increasing its accuracy [38]. Another parallel direction can be in using the device-free passive identification concept for securing and controlling wearable devices. Handling the scale and limited capabilities of the Internet of Things (IoT) devices is another challenge. Developing techniques that address the heterogeneity, whether it is of using different RF technologies or devices with different capabilities, is another dimension for research [39]. Finally, looking for the next killer-app for using DfP detection, tracking, or identification and making it works robustly with high accuracy and at scale is always an open direction for contribution.

References

[1] H. Abdelnasser and et al. UbiBreathe: A Ubiquitous non-invasive WiFi-based Breathing Estimator. In *MobiHoc*. ACM, 2015.

[2] H. Abdelnasser and et al. Wigest: A ubiquitous WiFi-based gesture recognition system. In *INFOCOM*. IEEE, 2015.

[3] H. Abdelnasser and et al. Magboard: Magnetic-based ubiquitous homomorphic off-the-shelf keyboard. In *SECON*. IEEE, 2016.

[4] H. Abdelnasser and et al. SemanticSLAM: Using environment landmarks for unsupervised indoor localization. *IEEE Transactions on Mobile Computing*, 2016.

[5] A. Al-Husseiny and et al. RF-based traffic detection and identification. In *VTC Fall*. IEEE, 2012.

[6] K. Ali and et al. Keystroke recognition using wifi signals. In *MobiCom*. ACM, 2015.

[7] H. Aly and et al. New insights into WiFi-based device-free localization. In *UbiComp*. ACM, 2013.

[8] H. Aly and M. Youssef. Dejavu: an accurate energy-efficient outdoor localization system. In *ACM SIGSPATIAL GIS*. ACM, 2013.

[9] M. Alzantot and et al. CrowdInside: Automatic construction of indoor floorplans. In *SIGSPATIAL*. ACM, 2012.

[10] P. Bahl and et al. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM*. IEEE, 2000.

[11] J. Chung and et al. Indoor location sensing using geo-magnetism. In *MobiSys*. ACM, 2011.

[12] R. Elbakly and et al. A robust zero-calibration RF-based localization system for realistic environments. In *SECON*. IEEE, 2016.

[13] A. Eleryan and et al. Aroma: Automatic generation of radio maps for localization systems. In *MobiCom Workshops*. ACM, 2011.

[14] A. Eleryan and et al. Synthetic generation of radio maps for device-free passive localization. In *GLOBECOM*. IEEE, 2011.

[15] M. Elhamshary and et al. CheckInside: A fine-grained indoor location-based social network. In *UbiComp*. ACM, 2014.

[16] M. Elhamshary and et al. Activity recognition of railway passengers by fusion of low-power sensors in mobile phones. In *SIGSPATIAL*. ACM, 2015.

[17] M. Elhamshary and et al. SemSense: Automatic construction of semantic indoor floorplans. In *IPIN*. IEEE, 2015.

[18] M. Elhamshary and et al. A fine-grained indoor location-based social network. *IEEE Transactions on Mobile Computing*, 2016.

[19] M. Elhamshary and et al. Transitlabel: A crowd-sensing system for automatic labeling of transit stations semantics. In *MobiSys*. ACM, 2016.

[20] R. Gao and et al. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *MobiCom*. ACM, 2014.

[21] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 2003.

[22] Y. Jiang and et al. Hallway based automatic indoor floorplan construction using room fingerprints. In *UbiComp*. ACM, 2013.

[23] N. Kassem and et al. RF-based vehicle detection and speed estimation. In *VTC Spring*. IEEE, 2012.

[24] D. Katabi. Tracking people and monitoring their vital signs using body radio reflections. In *MobiSys workshops*. ACM, 2014.

[25] A. Kosba and et al. Rasid: A robust WLAN device-free passive motion detection system. In *PerCom*. IEEE, 2012.

[26] A. Pal and et al. A robust heart rate detection using smart-phone video. In *MobiHoc workshops*. ACM, 2013.

[27] K. Panta and et al. Indoor localisation using white LEDs. *Electronics letters*, 2012.

[28] J. Park and et al. Growing an organic indoor location system. In *MobiCom*. ACM, 2010.

[29] N. Priyantha and et al. The cricket location-support system. In *MobiCom*. ACM, 2000.

[30] M. Raja and et al. Applicability of RF-based methods for emotion recognition: A survey. In *PerCom Workshops*. IEEE, 2016.

[31] W. Ruan and et al. Tagfall: Towards unobstructive fine-grained fall detection based on UHF passive RFID tagsleds. In *MobiQuitous*. ICST, 2015.

[32] I. Sabek et al. ACE: An accurate and efficient multi-entity device-free WLAN localization system. *IEEE Transactions on Mobile Computing*, 14(2), 2015.

[33] M. Seifeldin and et al. Nuzzer: A large-scale device-free passive localization system for wireless environments. *IEEE Transactions on Mobile Computing*, 2013.

[34] G. J. Stigler. The organization of industry. *University of Chicago Press Economics Books*, 1983.

[35] G. Wang and et al. We can hear you with wi-fi! *IEEE Transactions on Mobile Computing*, 2016.

[36] R. Want and et al. The active badge location system. *ACM Transactions on Information Systems*, 1992.

[37] A. M. Youssef and M. Youssef. A taxonomy of localization schemes for wireless sensor networks. In *ICWN*, 2007.

[38] M. Youssef. Towards truly ubiquitous indoor localization on a worldwide scale. In *SIGSPATIAL*. ACM, 2015.

[39] M. Youssef. A decade later-challenges: Device-free passive localization for wireless environments. In *Proceedings of the Fifth IEEE COSDEO Workshop, IEEE PerCom*, 2016.

[40] M. Youssef and et al. The Horus WLAN location determination system. In *MobiSys*. ACM, 2005.

[41] M. Youssef et al. Multivariate analysis for probabilistic WLAN location determination systems. In *MobiQuitous 2005*. IEEE, 2005.

[42] M. Youssef and et al. Challenges: device-free passive localization for wireless environments. In *MobiCom*. ACM, 2007.

[43] M. Youssef and et al. The Horus location determination system. *Wireless Networks*, 2008.

[44] X. Zheng and et al. Smokey: Ubiquitous smoking detection with commercial wifi infrastructures. In *INFOCOM*. IEEE, 2016.

The SIGSPATIAL Special

Section 2: Event Report

ACM SIGSPATIAL
http://www.sigspatial.org

GIR'16 Workshop Report 10th ACM SIGSPATIAL Workshop on Geographic Information Retrieval San Francisco, USA 4^{th} November 2016

Christopher B. Jones[1]
[1] Cardiff University, United Kingdom
c.b.jones@cs.cf.ac.uk

Ross S. Purves[2]
[2]University of Zurich, Switzerland
ross.purves@geo.uzh.ch

(Workshop Co-chairs)

The field of Geographic Information Retrieval (GIR) is concerned with the problems of gaining access to documents and to information within documents that relate to geographical locations. The methods employed derive from those of Information Retrieval, with its emphasis upon unstructured documents and natural language processing, and of Geographical Information Systems which, while oriented to structured data, are the source of many spatial analytical and data access methods that are relevant to GIR. This workshop is the tenth of a series of workshops that have addressed research challenges for GIR, including those relating to recognising and disambiguating references to place names in text (geoparsing); determining the geographic scope of documents; developing gazetteers and ontologies to maintain knowledge of toponyms and geographic concepts; spatio-textual indexing methods that combine inverted file methods with those of spatial database indexing; managing vagueness and uncertainty in geographic terminology; extracting geo-spatial facts and events from documents; and evaluating the performance of geo-information retrieval systems.

The 10th GIR workshop was held on 31st October 2016 at the ACM SIGSPATIAL conference in San Francisco, USA. Previous workshops have been held either in combination with the SIGIR and CIKM conferences or as stand alone events in cooperation with ACM SIGSPATIAL.

At GIR'16 there were 9 presentations, of which 6 were full papers and 3 were short papers. The workshop was organized around three sessions relating to common themes of the submitted papers and a fourth session that was a discussion of ideas for shared case studies in GIR.

The first session focused on extracting information on events, the function of geographic places and the presence of people at places that they refer to in Tweets. Spitz et al presented a method for imprecise extraction of events that relate to an actor, a time and a location. They maintain a knowledge base of graphs of entities (potential actors), time and location that record relations (such as part-of and similar to) between the entities to allow searching for coarser or finer granularity instances of the elements of a query. Tardy et al presented a paper that explored the use of social media, in particular Flickr, to detect both geographic features and their function or use (as for a building). They looked at an urban context and used the Geonames gazetteer to detect geographic features along with a word sense disambiguation tool (BabelFly) to detect the sense of words. Sparks et al described a machine learning approach to detecting whether Twitter users are present at specific types of geographic facility (restaurant, airport, stadium). The intention was to exploit very widely used social media (Twitter) to supplement data on people's behaviour patterns available from less widely used social media that record location check-ins explicitly.

The second session was concerned with geo-data disambiguation and integration. Blank and Henrich used graph search and approximate string matching methods (with the Geonames gazetteer) to extract itineraries

from historic route descriptions with a depth-first branch-and-bound algorithm. Yu et al addressed the problem of place matching using multiple similarity metrics in combination with an adaption of a method referred to as Naïve Descending Extraction. Their methods improved on a baseline semantic alignment system. Golubovic et al presented a design for a system intended to alert farmers to imminent threats, related for example to weather or pests, based on automatically identifying, analysing and integrating data from multiple sources including news articles and social media.

The topic of the third session was geoparsing and evaluation. Brando et el presented the results of a comparison between several named entity recognition systems, some of which (in particular Stanford NER) use a machine learning approach and can be trained with data specific to a particular domain. Their experiments were notable for distinguishing between the standard task of recognising gazetteer names and the more challenging task of recognising various informal references to places that can include generic place types as part or all of the name. Morteza highlighted the limitations of some existing approaches to measuring precision and recall in geoparsing and proposed an approach to toponym resolution that combines a measure of confidence in the resulting geo-coded place with geographic distance between the resolved location and the gold standard location. Cai and Ye Tian studied processes employed by human annotators when resolving toponyms and noted the difference between heuristics they employed, such as hierarchical ontological relations between place references in a single article, and an assumption that places in the same article are close together. They presented a geo-referencing workbench that progressively learns a gazetteer based on manual intervention.

The final session was a lively and constructive discussion of ideas for possible shared tasks and case studies in GIR. These related to quite a variety of topics including the proposal of studies of the forms of place names and the improvement of gazetteers, the creation of shared resources for performing geo-parsing, the generation of benchmark systems and of an auto-evaluation framework for new GIR methods (analogous to some existing such frameworks in other areas of information retrieval and NLP).

join today!

SIGSPATIAL & ACM

www.sigspatial.org **www.acm.org**

The **ACM Special Interest Group on Spatial Information** (SIGSPATIAL) addresses issues related to the acquisition, management, and processing of spatially-related information with a focus on algorithmic, geometric, and visual considerations. The scope includes, but is not limited to, geographic information systems (GIS).

The **Association for Computing Machinery** (ACM) is an educational and scientific computing society which works to advance computing as a science and a profession. Benefits include subscriptions to *Communications of the ACM*, *MemberNet*, *TechNews* and *CareerNews*, full and unlimited access to online courses and books, discounts on conferences and the option to subscribe to the ACM Digital Library.

- ❑ SIGSPATIAL (ACM Member) ..... $ 15
- ❑ SIGSPATIAL (ACM Student Member & Non-ACM Student Member) ..... $ 6
- ❑ SIGSPATIAL (Non-ACM Member) ..... $ 15
- ❑ ACM Professional Membership ($99) & SIGSPATIAL ($15) $114
- ❑ ACM Professional Membership ($99) & SIGSPATIAL ($15) & ACM Digital Library ($99) $213
- ❑ ACM Student Membership ($19) & SIGSPATIAL ($6) ..... $ 25

payment information

Name _____

ACM Member # _____

Mailing Address _____

_____

City/State/Province _____

ZIP/Postal Code/Country _____

Email _____

Mobile Phone _____

Fax _____

Mailing List Restriction
ACM occasionally makes its mailing list available to computer-related organizations, educational institutions and sister societies. All email addresses remain strictly confidential. Check one of the following if you wish to restrict the use of your name:

- ❑ ACM announcements only
- ❑ ACM and other sister society announcements
- ❑ ACM subscription and renewal notices only

Credit Card Type: ❑ AMEX ❑ VISA ❑ MC

Credit Card # _____

Exp. Date _____

Signature _____

Make check or money order payable to ACM, Inc

ACM accepts U.S. dollars or equivalent in foreign currency. Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

Questions? Contact:
ACM Headquarters
2 Penn Plaza, Suite 701
New York, NY 10121-0701
voice: 212-626-0500
fax: 212-944-1318
email: acmhelp@acm.org

Remit to:
ACM
General Post Office
P.O. Box 30777
New York, NY 10087-0777

SIGAPP

www.acm.org/joinsigs

Association for Computing Machinery

Advancing Computing as a Science & Profession

The SIGSPATIAL Special

ACM SIGSPATIAL
http://www.sigspatial.org