Vita

A Versatile Toolkit for Generating Indoor Mobility Data for Real-World Buildings

Li, Huan; Lu, Hua; Chen, Xin; Chen, Gang; Chen, Ke; Shou, Lidan

Published in:
Proceedings of the VLDB Endowment

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

Vita: A Versatile Toolkit for Generating Indoor Mobility Data for Real-World Buildings

Huan Li[†], Hua Lu[‡], Xin Chen[†], Gang Chen[†], Ke Chen[†], Lidan Shou[†]
[†]Department of Computer Science, Zhejiang University, China
[‡]Department of Computer Science, Aalborg University, Denmark
[†]{lihuancs, cxzju, cg, chenk, should}@zju.edu.cn, [‡]luhua@cs.aau.dk

ABSTRACT

We demonstrate a generic, user-configurable toolkit for generating different types of indoor mobility data for real-world buildings. Our prototype generates the desired data in a three-layer pipeline. The *Infrastructure Layer* accepts industry-standard digital building information (DBI) files to generate the host indoor environment, allowing users to configure the generation of a variety of positioning devices, such as Wi-Fi, Bluetooth, RFID, etc. The *Moving Object Layer* offers the functionality of defining objects or trajectories, with configurable indoor moving patterns, distribution models, and sampling frequencies. The *Positioning Layer* generates synthetic signal strength measurements known as raw RSSI[1] measurements according to the positioning device data and trajectory data generated at relevant layers. It also generates different types of indoor positioning data through the customization of all typical indoor positioning methods on the raw RSSI data.

1. INTRODUCTION

In the past few years, indoor space has emerged as a new research frontier for data management [7]. As a Big Data paradigm, indoor mobility analytics has started to attract wide attention from various research communities and industries.

Many businesses, e.g., customer engagements, medical services, and space use analysis [8], can all benefit significantly from the insights that are very hard to obtain without indoor mobility data. In real systems, indoor mobility data is captured by positioning technologies that differ significantly from outdoor GPS. Typical indoor positioning systems employ short-range wireless technologies such as Wi-Fi, Bluetooth, RFID, etc. The localization methods used by such systems include trilateration [2], fingerprinting [5], and proximity [4], all estimating user locations on a non-continuous basis by measuring the *RSSI*[1] observed at either the mobile clients or the static stations. As a result, indoor mobility data in general has different representation formats, and their positioning accuracies are variably lower than that of GPS.

[1]Received signal strength indication, the power present in a radio signal received by a positioning device.

Proceedings of the VLDB Endowment, Vol. 9, No. 13

In order to support relevant studies, we need a versatile data generator that is able to generate various indoor mobility data according to user needs. Such a generator can be used for two purposes: (1) It can be used to generate mobility data, which is typically given as a discrete sequence of user location estimates. (2) It can provide the "ground truth" for the mobility data generated in (1), to evaluate the models/algorithms being studied. For example, a data generator can preserve the underlying *raw trajectory* while generating the corresponding synthetic *indoor positioning data*. Particularly, the former can be preserved in a fine temporal granularity, whereas the latter can be set according to a lower sampling frequency to simulate a real indoor positioning system. In this sense, such a data generator can provide "ground truth" indoor movement data, which is indispensable for effectiveness evaluations needed by mining and analytics. Such ground truth is usually missing in real indoor positioning data due to the low sampling frequency and/or low accuracy. Consequently, an object's whereabout is unknown between two consecutive reported locations in real indoor positioning data, which makes such real indoor positioning data barely useful for effectiveness evaluations.[2]

There exist few data generators [6, 9, 11] for indoor settings. MWGen [9] can generate moving object trajectories in both outdoor and indoor settings, however, with many restrictions. For a given indoor space, MWGen requires users to manually extract the building information from a floor plan. A multi-floor building is simulated by duplicating the floor plan. Given two indoor locations, MWGen generates the trajectories on a path either with minimum length or with minimum walking time. The RFID data generation tool [11] generates RFID data for testing RFID business tracking systems where objects are constrained to conveyor belts only. The tool allows for configuration on parameters such as the number of virtual RFID readers, the number of RFID tags, and the velocity of conveyor belts. IndoorSTG [6] generates semantic-based trajectories and proximity based positioning data for indoor moving objects in an artificial, simulated indoor environment. It allows for limited configuration on the virtual indoor entities (e.g., rooms, staircases, and elevators), and virtual positioning devices.

Given the variety of needs for indoor mobility analytics, the existing data generators are apparently insufficient in functionality. First, they only generate very limited types of indoor mobility data. MWGen [9] cannot generate indoor positioning data. The RFID data tool [11] only generates RFID data and produces no trajectory data. IndoorSTG [6] generates both types, but it only works for proximity based indoor positioning and ignores more popular alternatives like Wi-Fi based fingerprinting. More importantly,

[2]This is not a problem in outdoor settings where GPS data captures a moving object's raw movements to a very good degree thanks to high sampling frequency and high positioning accuracy.

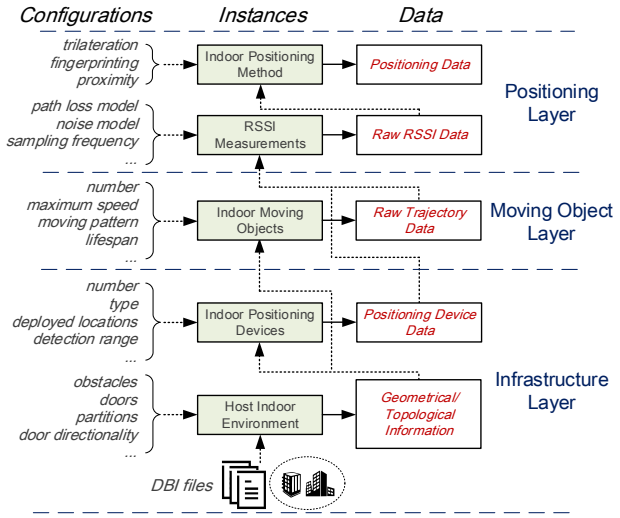


Figure 1: General Data Flow for Indoor Mobility Data Generation

both MWGen and IndoorSTG generate semantic-based indoor trajectories that lack sufficient "ground truth" details, making their output unsuitable for research that involves effectiveness studies. Second, the existing data generators do not allow real building models to be imported for data generation. Instead, they use only artificial or semi-artificial indoor settings, which are far from practical needs. Third, the existing data generators offer very limited support for indoor movements. Users have few choices on the moving patterns of indoor moving objects. All these problems make the output of these tools oversimplified and insufficient for modeling realistic indoor mobility.

This work demonstrates a toolkit named *Vita* [1] for generating indoor mobility data in real-life indoor environments that are captured in digital building information (DBI) formats. Figure 1 shows the general data flow of using the toolkit. The entire data processing pipeline is divided into three layers. In the *Infrastructure Layer*, the host indoor environment is loaded and constructed by processing the DBI file(s) for real building(s). In this layer, users are also allowed to add information about indoor positioning devices. Subsequently, in the *Moving Object Layer*, indoor moving objects and their raw trajectory data are generated based on the geometrical and topological information obtained in the infrastructure layer. Further, in the *Positioning Layer*, raw RSSI measurement data is generated from the positioning device data generated in the infrastructure layer as well as the raw trajectory data generated in the moving object layer. After that, the ultimate positioning data can be generated according to the user-specified indoor positioning method.

To the best of our knowledge, Vita is the first tool for generating multiple-typed indoor mobility data for real-world buildings. Vita is characterized by several desirable features:

- Vita accepts industry-standard DBI files and uses real-world (multi-floor) buildings with semantic attributes as the host environment for data generation.
- Vita offers a set of indoor moving patterns and distribution models for users, making its output—indoor moving objects and their trajectory data—highly configurable.
- Vita allows users to tune the sampling frequency of generating the trajectories such that the valuable "ground truth" movements are preserved to an arbitrarily detailed degree.
- Vita can generate indoor positioning data according to the most typical indoor positioning methods.

2. SYSTEM DESIGN

Vita consists of three levels of system components (see Figure 2).

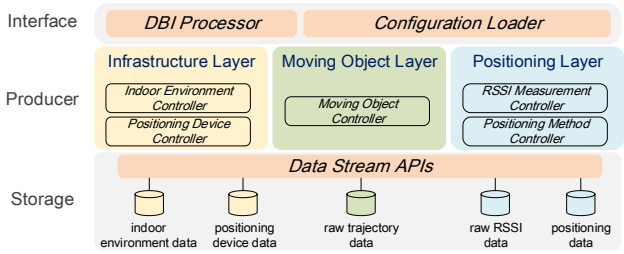


Figure 2: System Architecture

Interface reads and extracts the meta-data from user's input. It includes two modules, namely *DBI Processor* and *Configuration Loader*. The DBI Processor parses raw DBI files and automatically constructs the host indoor environment with parsed indoor entities. The Configuration Loader allows one to directly edit the parameters for data generation.

Producer consists of the three layers introduced in Figure 1, it receives meta-data from the Interface, produces specific types of data, and exchanges them with the Storage at relevant layers. In each layer, multiple controllers are designed to manage the related generation tasks:

1) The Infrastructure Layer accepts two types of input: DBI files about real buildings and configuration on indoor positioning devices. The *Indoor Environment Controller* allows a user to configure door directionality and deploy obstacles to further customize the host indoor environment. The *Positioning Device Controller* allows a user to configure the devices' number, deployed locations, type, and other type-dependent properties (e.g., the detection range of RFID readers). This layer generates two types of data: indoor environment data and positioning device data.

2) The Moving Object Layer accepts user configuration on objects' indoor movements. The *Moving Object Controller* allows a user to set object parameters including number, maximum speed, moving pattern, and lifespan. In this layer, users can also tune the sampling frequency in order to set the temporal granularity for the raw trajectory data to be generated.

3) The Positioning Layer controls the generation of raw RSSI measurement data and indoor positioning data. The *RSSI Measurement Controller* allows a user to set RSSI data generation parameters including the path loss model, the noise model, etc. The *Positioning Method Controller* (PMC) reads objects' raw RSSI data and estimates the locations according to the chosen positioning method and relevant configuration. Note that another sampling frequency can be specified in PMC for generating the positioning data. This is different from the one for generating the trajectory data. PMC offers three typical indoor positioning methods, namely trilateration, fingerprinting and proximity. Once a positioning method is chosen, a user can configure its detailed parameters. For *trilateration*, users can tune the parameters of the function that estimates distance from noisy RSSI data. For *fingerprinting*, users can select a set of reference locations for training a positioning model; they can also choose a deterministic or a probabilistic algorithm to infer objects' locations. *Proximity* does not require any extra configurations since the positioning device's detection range and frequency are already configured in the infrastructure layer.

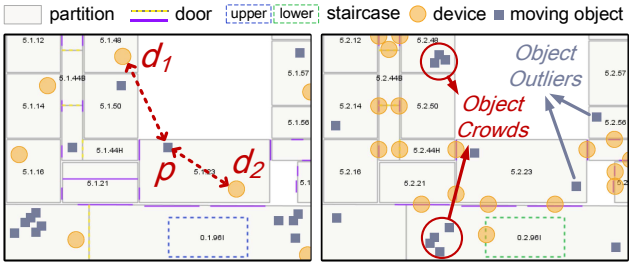
Storage serves as both the data provider and data keeper. In particular, the *Data Stream APIs* module encapsulates some commonly used functions and query processing algorithms that can be directly called by the Producer. Besides, the generated data is stored into different repositories with efficient indices.

3. PRINCIPLES OF DATA GENERATION

3.1 Modeling Indoor Object Movements

We consider three critical aspects related to indoor movements.

1) *Initial Distribution:* Vita provides two distribution models. The *uniform* model assumes that objects appear evenly in the space initially. Whereas the *crowd-outliers* model captures a more common scenario where a vast majority of objects are located around several hot areas to form crowds while others are distributed randomly as outliers. For example, customers in a mall often gather around the shops that are currently on sale. As Figure 3(b) illustrates, the object crowds are grouped in circles and the outliers are shown as small rectangles.



(a) Ground Floor (b) First Floor

Figure 3: A Data Generation Example on Real-World Floor Plans

2) *Lifespan:* Each generated object is associated with a lifespan that is randomly determined between two user-specified parameters *minimum lifespan* and *maximum lifespan*. We also support adding new objects during the generation period. Users are allowed to configure the emerging locations and starting times for new objects. For example, users can choose a Poisson distribution to set the starting times for new objects.

3) *Moving Pattern:* We considered three aspects in customizing object moving patterns, namely *intention*, *routing*, and *behavior*. For intention, *destination* model means an object moves toward its destination, and *random-way* model means it moves randomly. For routing, an object can move along a path determined by a particular routing schema, e.g., *minimum indoor walking distance* [10], *minimum walking time* [9], etc. For behavior, users can choose from pre-defined mechanisms to configure details such as the change of speed, the stop during the moving, etc. For example, in the *walk-stay* mechanism, an object will switch between the states "walking along the path to its destination" and "staying at the destination or a location on path" after a random period of time.

3.2 Modeling Positioning Devices and RSSI

Vita provides two deployment models for generating indoor positioning devices. Referring to the examples in Figure 3, the positioning devices are deployed with the *coverage* model on the ground floor. Commonly used in installing access points, this model stipulates that devices should be close to the wall to get power supply and they should be separate from each other to have maximum signal coverage. On the first floor, the *check-point* model is used to simulate the scenario where devices are deployed at entrances to rooms and/or hotspots in large rooms.

Vita uses *path loss model* [2] to generate RSSI measurements mainly based on the transmission distance between client and positioning device. Other factors such as multi-path, non-line-of-sight are also considered. Referring to Figure 3(a), although the object p's transmission distance to device d_1 and d_2 is equal, the RSSI value measured by d_2 should be greater than d_1's since the walls block the line-of-sight between p and d_1. We implement a generic, flexible path loss model as $rssi(dBm) = -10nlog10(d_t) + A + N_{ob} + N_f$. Specifically, $rssi$ is the measured value; d_t is the present transmission distance between the positioning device and the observed object. We allow users to define three variables: A is a calibration RSSI value measured at 1 meter, N_{ob} is the noise caused by influence of obstacles like walls and doors, and N_f is the noise for signal fluctuation related to temperature, humidity, etc; a default setting of these variables is provided for a quick customization.

3.3 Modeling Indoor Positioning Methods

By processing raw RSSI data in different ways, Vita currently supports three indoor positioning methods.

1) *Trilateration* [2] infers deterministic locations from the intersection of at least three circles[3]. The key is to convert an RSSI measurement to the distance between a positioning device and an object. To this end, we allow users to define their own RSSI conversion functions that derive the distances from the noisy RSSI measurements. A default function is also provided in case a user does not know how to configure the details.

2) *Fingerprinting* [5] associates RSSI fingerprints to locations. A fingerprint in a location is a vector in which each dimension corresponds to an RSSI value measured by a certain positioning device. In the *offline phase*, a site-survey is required to collect the fingerprints for a set of reference locations. The collected data is stored in *radio map* as training data. When constructing a radio map, Vita first allows users to select a set of reference locations on a given floor. After that, Vita simulates some objects to collect the fingerprints at the selected reference locations to get RSSI data for the radio map. Once the radio map is constructed, in the *online phase*, users can employ various classification algorithms such as NaiveBayes or kNN to infer locations for moving objects.

3) *Proximity* [4] estimates symbolic relative locations for moving objects. Specifically, if an object is detected by a positioning device, it is considered to be collocated with that device for the detection period. We use a thresholding method to determine the detection period for a given pair of object and device. If the RSSI measurements for the object cannot be found over the time of the device's one detection operation, we consider it has left the device's detection range, and the detection period is thus complete.

4. SYSTEM IMPLEMENTATION

Vita's main GUI is shown in Figure 4. Users can upload DBI files through the top-left panel. Each parsed DBI entity will be rendered into the map view with its detailed information shown in the text above it. The configuration and data generation elements for each layer are located in the bottom-right tabs.

Vita is designed and implemented in an extensible way for easy integration of more advanced features in the future. For example, to introduce the interference between moving objects, it can be configured to use more complicated movement generation processes like a crowd simulation model. Also, the entire prototype will be open source for free extensions and customizations.

4.1 Processing DBI files

Vita accepts the Industry Foundation Class (IFC) files for processing DBI. Generally, an IFC file is a textual file that uses data sections to describe a building's indoor entities. However, IFC files only capture indoor topology partially. We follow a previous work [3] to finish the basic processing steps. Specifically, the data errors in describing the indoor topology can be identified through

[3]A circle is centered at a positioning device's location, and its radius is the distance to an observed object.

geometry calculations or GUI-based manual checks. Rooms or hallways with irregular shapes are decomposed into balanced, smaller partitions according to their sizes and shapes, and the resultant partitions are indexed by a spatial index in order to support the indoor distance computations. Connected partitions for each door are identified through topology and geometry computations.

Vita improves the DBI processing in [3] to support multi-floor buildings. IFC models a staircase as a set of disjointed 3D points, but its connectivity to other partitions is missing. Vita performs two steps to find out a staircase's connected partitions. First, several upper (lower) vertices on the staircase's boundary are identified by geometry computation, and the floor having the maximum intersection with the upper (lower) vertices is selected as the upper (lower) connected floor. Second, within the range of the connected floors, the partition that contains the upper (lower) vertices is returned as the upper (lower) connected partition for the staircase.

Vita also supports semantic extraction by defining empirical rules. For example, a canteen will be identified if its entity name contains the word "canteen" or "dining room", a public area will be recognized in the terms of its door connectivity and floorage.

4.2 Data Storage

We use PostgreSQL+PostGIS to store the infrastructure's geometrical information. Specifically, indoor entities modeled by polygons and points are indexed by featured spatial indices. The indoor topology information is stored in a relation that supports both door directionality and multi-floor structure properties.

Raw trajectory data and RSSI measurement data are both stored in DBMS. The format for trajectory data is (o_id, loc, t), which denotes that an object identified by o_id was at location loc at time t. RSSI measurement is stored as $(o\_id, d\_id, rssi)$, where d_id is a positioning device ID and $rssi$ is the measurement value.

Vita supports multiple data formats for different positioning methods. Trilateration and deterministic fingerprinting directly produce output as (o_id, loc, t), which means that object o_id was reported at loc at time t. Probabilistic algorithms estimate one object's location with a set of samples, each containing a location loc and a probability $prob$. Thus, it is given as $(o\_id, \{(loc_i, prob_i)\}, t)$. Data generated for proximity is very different from the others. A record (o_id, d_id, t_s, t_e) indicates that object o_id was detected by a positioning device d_id from time t_s to t_e. Note that loc in all records consists of two parts, the former refers to a *buildingID* + a *floorID*, the latter can be either a *partitionID* or a coordinate point.

5. DEMONSTRATION

We will demonstrate how to generate different types of the positioning data by using a set of real DBI files for different indoor spaces. The system operations for each positioning data generation process is summarized into a common path: 1. Import a DBI file $\rightarrow$ 2. View and modify the host indoor environment $\rightarrow$ 3. Configure and generate indoor positioning devices $\rightarrow$ 4. Configure and generate indoor moving objects $\rightarrow$ 5. Configure and generate raw RSSI measurements $\rightarrow$ 6. Choose and configure a positioning method, generate the corresponding indoor positioning data.

In step 1, we will test different DBI files from clinics, malls and office buildings. In step 2, we will switch and browse through the floor plans of the real building to modify the indoor structure, e.g., decompose the irregular partitions, identify and fix parse errors, deploy obstacles, etc. In step 3, we will configure and generate the Wi-Fi, Bluetooth, RFID positioning devices respectively. Those devices are directly rendered into the floor plans for users to edit. In step 4, we will configure the properties for generating moving objects and then initialize a set of moving objects. During the data generation process, the movements of objects will be visualized in real time on the floor plan of the used real-world building. The generation can be paused at any time in order to extract a snapshot of the indoor moving objects. We will also tune the sampling frequency for generating raw trajectory data. In step 5, to generate raw noisy RSSI data, we will configure the parameters such as path loss model, sampling frequency, etc. In step 6, the interface offers supported positioning methods according to the chosen type of generated devices in step 3. In particular, all three methods can be applied to Wi-Fi devices, whereas fingerprinting currently does not apply to RFID and Bluetooth devices. When a positioning method is chosen, the system opens a generated properties file for configuring the relevant parameters. We will employ different combinations such as RFID + proximity, Bluetooth + trilateration, and Wi-Fi + fingerprinting. A short demonstration video is available on Vita's web site [1].

Figure 4: System Graphical User Interface

6. ACKNOWLEDGMENTS

This work was supported by The National Program on Key Basic Research Project of China (No.2015CB352400), and by The Overseas Collaboration Grant (No.61528207) from The National Natural Science Foundation of China.

7. REFERENCES

[1] Vita Project. `http://db.zju.edu.cn/vita/`.
[2] A. Bose and C. H. Foh. A practical path loss model for indoor WiFi positioning enhancement. In *ICICS*, pages 1–5, 2007.
[3] M. Boysen, C. de Haas, H. Lu, X. Xie, and A. Pilvinyte. Constructing indoor navigation systems from digital building information. In *ICDE*, pages 1194–1197, 2014.
[4] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.
[5] V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché. A comparative survey of WLAN location fingerprinting methods. In *WPNC*, pages 243–251, 2009.
[6] C. Huang, P. Jin, H. Wang, N. Wang, S. Wan, and L. Yue. IndoorSTG: A flexible tool to generate trajectory data for indoor moving objects. In *MDM*, pages 341–343, 2013.
[7] C. S. Jensen, H. Lu, and B. Yang. Indoor-a new data management frontier. *IEEE Data Eng. Bull.*, 33(2):12–17, 2010.
[8] H. Lu, C. Guo, B. Yang, and C. S. Jensen. Finding frequently visited indoor pois using symbolic indoor tracking data. In *EDBT*, pages 461–472, 2016.
[9] J. Xu and R. H. Güting. MWGen: A mini world generator. In *MDM*, pages 258–267, 2012.
[10] B. Yang, H. Lu, and C. S. Jensen. Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space. In *EDBT*, pages 335–346, 2010.
[11] H. Zhang, W. Ryu, B. Hong, and C. Park. A test data generation tool for testing RFID middleware. In *ICCIE*, pages 1–6, 2010.