2013 IEEE 14th International Conference on Mobile Data Management

IndoorSTG: A Flexible Tool to Generate Trajectory Data for Indoor Moving Objects

Chuanlin Huang, Peiquan Jin, Huaishuai Wang, Na Wang, Shouhong Wan, Lihua Yue

School of Computer Science and Technology, University of Science and Technology of China
No.96, Jinzhai Road, Hefei, 230027, China
jpq@ustc.edu.cn

***Abstract*—Indoor moving objects management has been a research focus in recent years. In order to get the trajectory data of indoor moving objects, people have to deploy a lot of positioning equipment, such as RFID readers and tags, which takes lots of money, time, and other costs. In addition, it is a very complex and costly process to construct different environment settings for various indoor applications. Aiming to provide experimental trajectory data for various indoor operations and mining algorithms, in this paper we present a flexible tool to generate trajectories for indoor moving objects, which is named IndoorSTG (*Indoor Spatiotemporal Trajectory Generator*). IndoorSTG can simulate different indoor environments using various elements including rooms, doors, corridors, stairs, elevators, and virtual positioning devices such as RFID or Bluetooth readers. Meanwhile, it can generate semantic-based trajectories for indoor moving objects in a specific indoor space. After an overview of the general features of IndoorSTG, we discuss the architecture and implementation of IndoorSTG. And finally, a case study of IndoorSTG's demonstration is presented.**

***Keywords*—indoor space; moving objects; semantic-based trajectories; data generator**

I. Introduction

Most people spend a considerable portion of their time in indoor environments such as office buildings, shopping centers, conferences facilities, airports, and so on. Therefore, indoor Location-Based Service (LBS) has been a new hot research topic [1]. There are many potential applications related with indoor LBS, such as hot indoor locations recommendation, indoor navigation, and indoor moving patterns analysis.

However, it is hard to perform studies towards indoor moving objects data management, due to the lack of real tracking data. Although there are some previous works focusing on indoor simulated data generation [2-6], they are designed for specific application needs and cannot suit for different indoor applications. Some researchers proposed to randomly generate simulated indoor trajectory data [2, 3], unfortunately most indoor moving objects do not have a random moving pattern. In [4], J. Xu et al. assumed that indoor moving objects may move along two kinds of shortest paths connecting two indoor locations, namely paths with minimum length and paths with minimum traveling time. They defined a constant indoor walking speed and a constant lift speed to compute objects' moving time in indoor space. However, they did not consider the correlation between moving objects and indoor locations. Furthermore, different objects may have varying moving patterns in indoor space, which has to be taken into account when generating trajectory data for indoor moving objects.

In the paper, aiming at providing simulated tracking data for indoor moving objects management, we present a flexible simulation tool, called IndoorSTG (*Indoor Spatiotemporal Trajectory Generator*). The motivation of IndoorSTG is to design an effective tool to simulate various kinds of indoor environments and generate semantic-based trajectory data for moving objects according to different moving patterns. The unique features of IndoorSTG can be summarized as follows:

(1) IndoorSTG introduces three types of semantics into the generation of indoor trajectory data, which are the types of moving objects and locations, the relations between moving objects and locations, and the moving patterns.

(2) IndoorSTG supports configuration of different indoor environments. Thus it can be used for various indoor application scenarios.

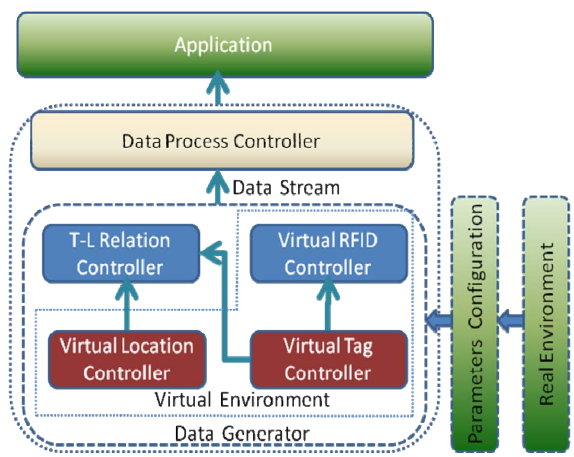

Fig.1. Architecture of IndoorSTG

II. Implementation of IndoorSTG

A. Architecture of IndoorSTG

The architecture of IndoorSTG is shown in Fig.1. We use RFID readers as the basic sensors. The RFID readers are deployed in the indoor environment, and we attach a RFID tag to each moving object so that they can be captured when passing by the RFID readers. Therefore, we use two virtual devices, namely *Virtual RFID Controller* and *Virtual Tag Controller*, to simulate RFID readers and

978-0-7695-4973-6/13 $26.00 © 2013 IEEE
DOI 10.1109/MDM.2013.51

moving objects respectively. When a moving object identified by a unique virtual tag enters into the range covered by some RIFD reader and leave out of the range, we generate a record formed (<*ID, reader_ID, tag_ID, enter_time, leave_time, move_time*>) to record the movement. Here *move_time* is used to record the time interval for the object moving from the previous RFID reader to the current one. In addition, the virtual environment also contains a *Virtual Location Controller*, which is used to maintain the information about the interested locations. The *T-L Relation Controller* in Fig.1 is designed to define the relationships between moving objects (i.e., virtual tags) and interested locations. The *Data Processing Controller* is the user interface to generate semantic-based trajectories, which also offers flexible configuration on the parameters of IndoorSTG.

B. Key Technologies in IndoorSTG

Classification of Indoor Locations. In a typical indoor environment, a moving object is usually associated with some specific locations. For instance, a student in a lab building is associated with his or her laboratory room and the supervisor's office. Hence, when a moving object enters into the indoor environment, we assume that it will stay for a long period at its associated locations and stay for less time at other locations. Meanwhile, we assume that moving objects tend to moving among their associated locations. As a result, the locations are classified into three types listed in Table I, as described as follows.

TABLE I. LOCATION CLASSIFICATION

Type	Description	Probability
0	*Primary Locations*	[0~1]
1	*Secondary Locations*	[0~1]
2	*Negligible Locations*	[0~1]

(1) *Primary Locations*. The primary locations represent the focused locations of a moving object, such as working room, where the moving object may go more frequently, and has a larger probability to firstly arrive at this location. In primary locations, he or she may stay for a long time, may not.

(2) *Secondary Locations*. The secondary locations represent the locations that are related with the moving object, where he or she may go frequently, only less than the primary locations. For example, for students, the secondary locations may include the supervisor's office, while for teachers the secondary locations may include the laboratory that his or her students stay in. Because of this relationship, the students (teacher) have a larger probability to teacher's office (students' laboratory). The types 0 and 1 can be called interested locations.

(3) *Negligible Locations*. The negligible locations are those that are not included in the above two type of locations. Particularly, those locations have little determined association with moving objects, such as corridors, stairs and elevators etc. and which have a small probability to go during the journey of a moving object.

Correlation between Objects and Locations. Indoor moving objects usually have different moving patterns; therefore it is important and useful to first make a classification on those moving objects. First, we should divide the moving objects into several types according their characters. Taking the office building of our department as an example, teachers and students are usually associated with their offices, but generally visitors have no associated locations. However, in a shopping mall, clerks are generally associated with their sites; while most customers have no associated locations. Since IndoorSTG aims at simulating the moving patterns of indoor moving objects, we simply divide all the moving objects (virtual tags) into two types, namely regular objects and random objects, each of which contains a certain number of moving objects.

Secondly, we construct the relationships among moving objects and locations. We assume that each regular object is associated with some locations among its primary and secondary locations. It means each object has a set of primary locations and another set of secondary locations. However, for random objects, there are no associated locations with them. Moreover, the set of regular and random objects as well as the associated primary and secondary locations with regular objects can be adjusted according to the specific scenario of the simulated indoor environment. For example, in the office building of our department, the number of regular objects (teachers and students) is much larger than that of random objects (visitors), whereas in a shopping mall the opposite is true. In the implementation, we construct a probability matrix between locations and objects, which indicates that a certain object has a specific probability to reach a location when moving in the indoor space. At this time, we must consider the type of moving objects, because they have different probabilities to move towards different types of associated locations. Generally, the probability for primary location is bigger than the probability for secondary locations, and much bigger than negligible locations.

Simulation of Different Moving Patterns. As different moving objects have different moving patterns in indoor space, we introduce an algorithm to generate simulated moving patterns for a given moving object. In the algorithm, we assume two situations: (1) moving on the same floor; (2) moving from one floor to another floor. When the object moves on the same floor or on the different floors, the velocity may change at the different time. When he or she moves from one floor to another floor, if taking stairs, the velocity generally differs from the velocity of moving on the same floor and may change. However, if taking elevator, the velocity of the elevator or escalator keeps constant speed, which is different from the velocity of the moving object generally. Because the object may have different moving patterns from the entrance to the destination room (location) at different time, we assign different probabilities to the interested locations of the object. Particularly, we assume that moving objects just entering into the building have different locations preferences from those that are already in the building. For instance, when a student just enters into the department building, his or her first destination location is usually the primary location (the laboratory). However, when he or she leaves the laboratory, the first destination location is surely not the laboratory itself. Hence, we differ those two cases and use different probabilities for them.

Given that one object just enters into the building from Location 1 (the current location), we first pick up the destination of the object according to the predefined probabilities. Suppose that the destination is Location 19, which is the primary location for the object. We will use the *Dijkstra* algorithm to get the minimum route from the current location to the destination, which infers that the object is very possible to reach the destination along the shortest path. Then choose the next location by the weight of each edge, i.e. probability, in the path connecting the current location with the destination until arrive at the destination. Then we adjust the probability of the other types of locations in order to get the next destination. Through this process, the probabilities of each location associated with a specific object can be adaptively updated according to the moving history of the object. When the current location and the destination are located in different floors, the generated optimal path includes stairs or elevators. Therefore we introduce additional waiting time into the trajectory to reflect this cost. Otherwise, the time cost will only include moving time. When an object arrives at the destination location, a staying time period will be randomly assigned to the object in order to simulate the real moving patterns in indoor space. For instance, in an indoor shopping mall, people may stay in a shop for a while and move on to another shop.

III. DEMONSTRATION

In our demonstration, we will first set up the indoor environment, which describes the 6-floor building of the school of computer science in our university. However, the indoor environment can be rebuilt or modified in IndoorSTG, including floors, rooms, doors, elevators, and so on. It can also load the data file of indoor environments to construct the environment. IndoorSTG is implemented in Java under Windows 7 Professional and follows the object-oriented programming standard. Our demonstration will use IndoorSTG to simulate different indoor environments and generate indoor trajectory data in this virtual environment. The user interface of the IndoorSTG is shown as Fig.2. We will first show the configuration of the parameters, such as the probability settings for the locations, the number of moving objects, the staying time of moving objects, the moving speed, and so on. Then we will show the process of generating trajectory data of all moving objects and show the data table using the format (*<ID, reader_ID, mob_ID, enter_time, leave_time, move_time>*). After that, we will show how IndoorSTG answer the two types of queries, i.e., moving object query and RFID reader query, in the user interface as well as animate their movement by visualizing the trajectories in the indoor environment.

ACKNOWLEDGEMENT

This paper is supported by the National Science Foundation of China (No. 60776801 and No. 71273010), the National Science Foundation of Anhui Province (no. 1208085MG117), and the USTC Youth Innovation Foundation.

REFERENCES

[1] C. S. Jensen, H. Lu, B. Yang, “Indoor—A New Data Management Frontier”, *IEEE Data Engineering Bulletin*, Special Issue on Spatial and Spatio-temporal Databases, 33(2): 12-17, 2010.

[2] C. S. Jensen, H. Lu, B. Yang, “Indexing the Trajectories of Moving Objects in Symbolic Indoor Space,” In: *Proc. of SSTD*, pp.208-227, 2009.

[3] C. S. Jensen, H. Lu, B. Yang, “Graph Model Based Indoor Tracking,” In: *Proc. of MDM*, 2009.

[4] J. Xu, R. H. Güting, “MWGen: A Mini World Generator”, In: *Proc. Of MDM*, pp. 258-267, 2012.

[5] C. Park, W. Ryu, B. Hong, “RFID Middleware Evaluation Toolkit Based on a Virtual Reader Emulator,” In: *Proc. of Int. Conf. on Emerging Databases*, pp. 154-157, 2009.

[6] H. P. Zhang, W. S. Ryu, B. H. Hong, “A Test Data Generation Tool for Testing RFID Middleware,” In: *Int. Conf. on Computers and Industrial Engineering*, Awaji, 2010.

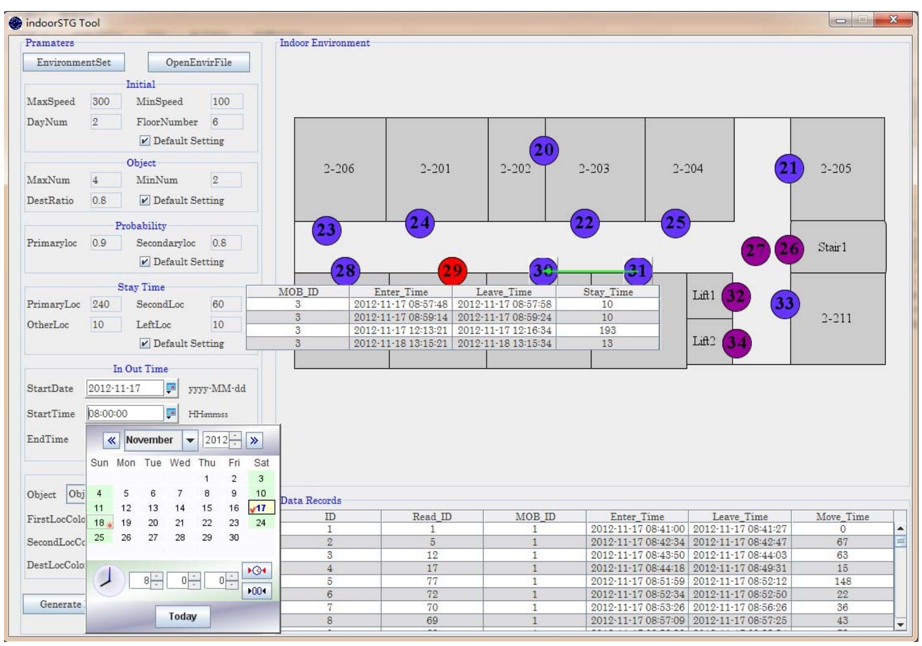

Fig.2. User Interface of IndoorSTG Demonstration