

# Infrastructures for Research on Multimodal Moving Objects

Jianqiu Xu, Ralf Hartmut Güting

Database Systems for New Applications, Mathematics and Computer Science  
FernUniversität Hagen, Germany  
{jianqiu.xu,rhg}@fernuni-hagen.de

## Abstract

Moving objects with multiple transportation modes (e.g., *Car*, *Bus*, *Walk*) can be represented by the reference method where the location is mapped to infrastructure objects, e.g., roads, buses, pavements. A scalable dataset is required to evaluate the performance of a database system managing these moving objects. To produce multimodal moving objects, infrastructure objects are essentially needed. Due to the major challenge of gaining real datasets, in this demonstration we propose a method to create these infrastructure objects for different environments including pavement areas, bus network and indoor just from a given road network and some floor plans. An indoor viewer is also implemented in a database system for visualizing 3D indoor moving objects and trajectories. The work is to be used for evaluating the performance of a database system managing generic moving objects.

## 1 Introduction

This demonstration presents a method to create infrastructure objects (e.g., pavements, bus routes), which are referenced by generic moving objects covering multiple environments for location representation, as well as a 3D visualization viewer for indoor objects and trajectories. Due to the difficulty of getting all these geographic objects, we produce all outdoor data just from a road dataset (without further information) [7] which is easy to attain and indoor data from public floor plans (e.g., [6, 5]), so that our method is also available for other researchers creating their own data. The infrastructure objects are in order to create generic moving objects for the research on moving objects with multimodal transportation modes, e.g., benchmarking a moving objects database system.

A moving objects database manages spatial objects continuously changing locations over time. Currently, the data models for representing moving objects work in specific environments, e.g., free space [1], road network [2], indoor [3]. To represent generic moving objects in all environments (e.g., *Walk*  $\rightarrow$  *Car*  $\rightarrow$  *Indoor*), the *reference* method can be used. The method is to consider the space where objects are located as a surface populated by recognizable and geographic objects that are associated with spatial attributes. Then, the location of a moving object can be represented by mapping it to those objects [9]. For a data model, one needs to do evaluation (e.g., Benchmark [8]) for the performance of DBMSs. [8] deals with one infrastructure where one mode for moving objects is considered, *Car*. But we produce the dataset of generic

moving objects covering multiple environments where all these geographic objects are necessarily required.

We partition the whole space into five infrastructures: *Free Space*, *Road Network*, *Region-based Outdoor*, *Bus Network*<sup>1</sup> and *Indoor*. Each infrastructure excluding *Free Space* consists of a set of infrastructure objects, i.e., *referenced* objects. In *Road Network*, roads and streets are infrastructure objects. The infrastructure *Region-based Outdoor* is the environment for pedestrians where pavements and zebra crossings are available areas. People can also use public transportation vehicles for their traveling, thus moving buses are infrastructure objects in *Bus Network*. The *Indoor* infrastructure comprises buildings each of which contains a set of rooms, e.g., office rooms. As *Free Space* has no infrastructure objects and there are already many available real road datasets, we focus on the last three infrastructures. *Region-based Outdoor* and *Bus network* are created based on *Road Network* because they have interaction relationships with each other. The reason is that pavements are usually located on both or one side of roads and bus routes are a subset of roads. We create polygons representing pavements and zebra crossings for *Region-based Outdoor*. *Bus Network* includes bus routes, bus stops and moving buses. For *Indoor*, the rooms of a building are created based on public floor plans.

## 2 Pavements

This section introduces the procedure to generate the environment for people walking. We create the pavements based on the real road dataset Berlin [7]. There are two kinds of pavements where one are those located on both or one side of roads and the other are zebra crossings allowing people to cross the street. Both kinds of pavements are represented by polygons.

### 2.1 Pavements besides roads

$R = \{r_1, r_2, \dots, r_n\}$  is the set of roads and each road  $r_i$  is represented by a polyline. In the real world, a road has a width value  $w_1$  and we extend the road from a polyline to a polygon using  $r_i$  and  $w_1$ . The procedure is as follows:

1.  $r_i$  is translated to two new polylines by the distance  $w_1$  where one is on the left of  $r_i$  and the other is on the right of  $r_i$ , and we denote the left part by  $r_i^l$  and the right part by  $r_i^r$ ;
2. We connect the two endpoints of  $r_i$  and  $r_i^l$ , and also the endpoints of  $r_i$  and  $r_i^r$ . Thus, two regions are formed, denoted by  $reg_i^l$  and  $reg_i^r$  where  $reg_i^l$  is created by  $r_i$  and  $r_i^l$ , and  $reg_i^r$  is created by  $r_i$  and  $r_i^r$ .
3. A third region is produced by forming the union of  $reg_i^l$  and  $reg_i^r$ , denoted by  $reg_i$ .

Figure 1 depicts the three steps for creating the polygon for  $r_i$ . Afterward, we define another width value  $w_2 = w_1 + \Delta w$  where  $\Delta w$  is the width of a pavement. Repeating the same three steps above, a larger region than  $reg_i$  can be created represented by  $reg_{-e_i}$ . The pavement area located on both sides of  $r_i$  can be obtained by  $reg_{-e_i}$  minus

---

<sup>1</sup>A rail network has a similar behavior where the vehicle moves along a pre-defined route and has a time schedule as well as a sequence of stops for passengers getting on and off.

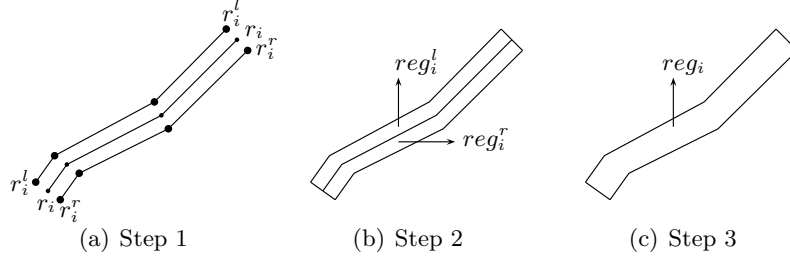


Figure 1: Create a region for a road

$reg_i$ . For each road, we follow the same procedure to create the pavements. In the end, we get a set of polygons for pavements, each of which corresponds to a road.

## 2.2 Zebra Crossings

This type of pavements are used by people to cross the street and they are usually located near the intersection position of two roads. So, at each junction intersected by  $r_i$  and  $r_j$  we create zebra crossings (represented by polygons) for each road. The zebra crossings are  $\Delta d$  (e.g., 3) meters from the junction, and they cross  $r_i$  and  $r_j$  as well as intersect the pavements located on both sides of the road.

## 2.3 Pedestrian Navigation

We perform the union on all polygons for the above two kinds of pavements and get a relatively large polygon  $P$  with many holes inside. An efficient algorithm is developed for shortest path searching inside  $P$ , which provides trip planning for a pedestrian.

## 3 Bus Network

The public bus transportation system consists of two components: *static* and *dynamic*. The *static* part is comprised of bus routes and bus stops, and the *dynamic* part includes time schedules and bus trips for each route.

### 3.1 Static

**Bus routes:** The whole bus routes can be considered as a subset of roads and we create them based on the Berlin streets data. Each bus route has a start and an end location on the road, and we select them as follows. The road network space is considered as a grid consisting of cells represented by rectangles. Each cell has a value representing the road density in that area, which is the number of roads intersecting the rectangle. All cells are classified into two groups according to the road density. Cells belonging to the high density group are the areas close to the city center or with high residential density. Cells in the low density group are the areas far away from the city center or suburbs. The start and end locations (on the roads) are randomly selected from two cells where both cells belong to the low density group, or one is from the high density group and the other is from the low density group. As cells from the high density group are close to each other, the distance between each pair is too short to create a bus route. We compute the shortest path connecting the two locations in the road network and set it as the bus route. In the end, we do refinement on the bus

routes created above, which is to merge two bus routes if the length of their common part is larger than a threshold value.

**Bus stops:** For each bus route, we create a sequence of bus stops and the distance interval between two *consequent* stops is randomly selected from a predefined distance set. Then, we merge several bus stops belonging to different bus routes if their routes have common parts and the distance between each pair of them is smaller than a threshold value, e.g., 50m.

In the real world, one bus route has two directions and for each direction there are a sequence of bus stops. To be realistic, our bus routes and bus stops are also created based on this behavior. Each bus stop is also mapped to a position in the pavement, thereby leading to connections between *Region-based Outdoor* and *Bus Network*.

## 3.2 Dynamic

The moving buses are created by their bus routes and the speed value. The bus route defines the path on which the bus should move. The speed value is set as follows. The data from [7] also has the type value for each street (e.g., main streets, side streets) which determines the maximum speed for cars moving on it. We use the maximum speed for moving buses. Each bus moves from one stop to another along its route, stops at the bus stop for 30 seconds waiting for passengers getting on and off, and then moves forward.

Each bus route has a schedule for its bus trips and different bus routes may have different time intervals for schedules. We set the schedule for each route by the traffic flow of that route, which is obtained as follows. Using BerlinMOD [8], we create a set of cars moving in the Berlin streets. And then, for each road we record its traffic flow by aggregating the number of cars passing it during a time interval. We map the bus route to its corresponding road and get the traffic flow of it. All bus routes have daytime schedules, and the routes with higher traffic flow are also treated as night bus routes.

# 4 Indoor

## 4.1 Create Buildings

For this infrastructure, we create buildings based on their floor plans. A floor plan is a diagram to show in scale the relationships between rooms, spaces and other physical features at one level of a structure. We have the floor plan of our university and there are some public resources [5, 6]. A building consists of rooms, corridors, staircases, elevators, etc. To be more general, we use the term *groom* (general room) for all of them. A *groom* is represented by a set of elements each of which has two attributes: a polygon and the height above ground level (the floor at level 0 of a building). For example, the staircase between two floors (a *groom*) contains a set of objects where each represents one footstep of the staircase. The elevator is represented by several *grooms* each of which has only one element recording the 2D area for one floor accompanied by the height above ground level. According to the floor plan, one level of the building can be created and then we translate one floor vertically to generate more floors to construct a building.

Each *groom* has one or more doors. An office room may have only one door but a corridor has several doors via which people can go from one room to another. A door

builds the connection between two *grooms*, thus we represent a door by two elements each of which has two attributes. The first attribute is the *groom* id and the second is a line value recording the position in that *groom*.

**2D areas for buildings:** Currently, we use rectangles for the bounding boxes of all building places which are obtained as follows. Let *bbox* be the bounding box of *R*, and *S* be the overall polygon of the union of all road polygons (produced by the three steps in Section 2.1).  $B = \text{bbox} \setminus (S \cup P)$  is a set of polygons which are possible areas for buildings. For each polygon, we decompose it into a set of triangles and get the maximum rectangle inside each triangle.

## 4.2 Indoor Navigation

Given all *grooms* and doors of a building, a graph can be built for indoor navigation which supports shortest path searching from one location to another. For example, “find the shortest path from my office room to conference room 1”. The start and end locations are in two *grooms* of a building. We represent the location by the *groom* id and the relative position inside that *groom*. A node of the graph denotes a door and an edge corresponds to a *groom*. The weight of an edge is the length of the shortest path connecting two doors in that *groom*. The distance value is calculated in obstructed space where Euclidean distance does not apply, because there may be obstacles (e.g., walls) in the *groom*. Before searching on the graph, one initially has to connect the start and end location to all doors in the *groom*.

## 4.3 Indoor Moving Objects

To simulate people’s movement in the building, we create indoor moving objects as follows. First, a set of indoor locations is generated. Second, with the graph in Section 4.2 the shortest path between two locations can be returned. Third, we define a speed value for a person and implement the elevator as a public transportation vehicle which has a regular and periodic movement. With the path, speed value and elevator, indoor moving objects can be generated. For each indoor moving object, we can get its trajectory in space which is represented by a 3D line.

# 5 Demonstration Overview

**Pavements and bus network:** In the user interface, one can observe all created pavements, bus routes and bus stops. We can also in the demo create online pavements for some smaller sample area. Animation of moving buses is also available. Figure 2 is the screenshot for infrastructures *Region-based Outdoor* and *Bus Network*. We will also demonstrate pedestrian navigation.

**Indoor viewer:** We can visualize floor plans of a building, indoor locations, trajectories as well as animation of indoor moving objects. Figure 3 shows the screenshot of the indoor viewer and indoor objects. In the 3D viewer, we show the building of our university where only the ground floor of a *groom* is visualized (no walls). With the 3D line, one can also visualize the indoor shortest path. In the demo, we create a set of a few hundred indoor moving objects and show their trajectories as well as animate their movement.

## 6 Conclusion

We develop a method to create infrastructure objects for representing generic moving objects in different environments. Three kinds of infrastructure objects are generated, pavement areas, bus network and buildings. Such infrastructures will be useful to evaluate multimodal moving objects databases in the future. A user-friendly interface is provided for visualizing indoor objects and the viewer can be used for research in a 3D environment. The implementation is done in an extensible database system Secondo [4]. Currently, we have the navigation for pedestrians and people inside a building. Later, we are going to provide the connection between *Region-based Outdoor* and *Indoor* as well as the navigation through all available infrastructures.

## Acknowledgment

The first author gratefully acknowledges the financial support by Chinese Scholarship Committee.

## References

- [1] R. H. Güting, and M. H. Böhlen and M. Erwig, and C.S. Jensen, and N.A. Lorentzos, and M. Schneider, and M. Vazirgiannis. A foundation for representing and querying moving objects, *ACM TODS*, 25(1), 1-42, 2000.
- [2] R.H. Güting, and V.T.de Almeida, and Z.M.Ding. Modeling and Querying Moving Objects in Networks, 15(2), 165-190, *VLDB Journal*, 2006
- [3] C.S.Jensen, and H.Lu. and B.Yang. Graph Model Based Indoor Tracking, *MDM*, 123-131, 2009.
- [4] R.H.Güting, and V.T.de Almedia, and D.Ansorge, and T.Behr, and Z.Ding, and T.Höse, and F.Hoffmann, and M.Spiekermann. SECONDO:An Extensible DBMS Platform for Research Prototyping and Teaching,1115-1116, *ICDE Demo Paper*, 2005.
- [5] <http://www.modulargenius.com/default.aspx>
- [6] [http://www.bkrfloorplans.co.uk/bkr\\_services/cinemas.html](http://www.bkrfloorplans.co.uk/bkr_services/cinemas.html)
- [7] <http://www.bbbike.de/cgi-bin/bbbike.cgi>
- [8] C.Düntgen, and T.Behr, and R.H.Güting. BerlinMOD:A Benchmark for Moving Objects Databases, 18(6), 1335-1368, *VLDB Journal*, 2009.
- [9] J.Xu, and R.H.Güting. A Generic Data Model for Moving Objects, *Manuscript in preparation*.

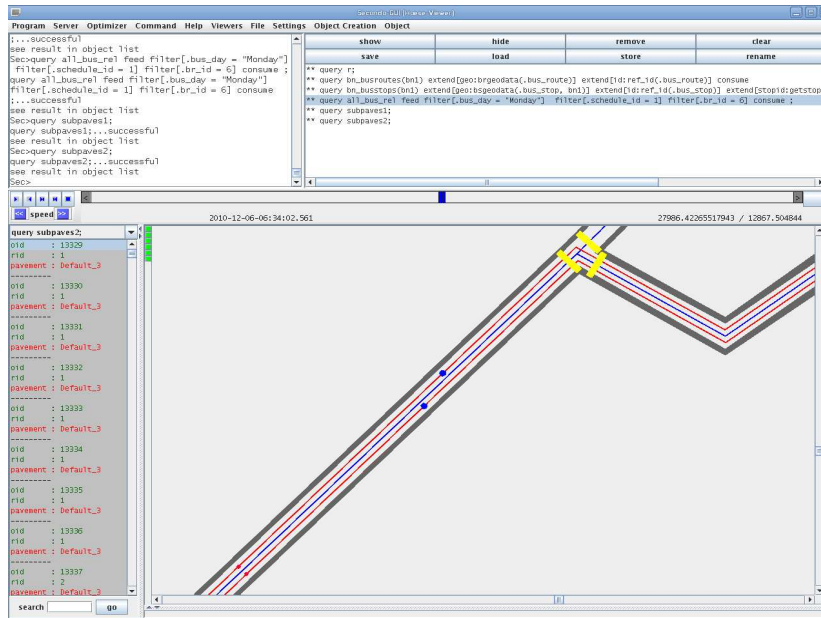


Figure 2: Pavements and Bus Network

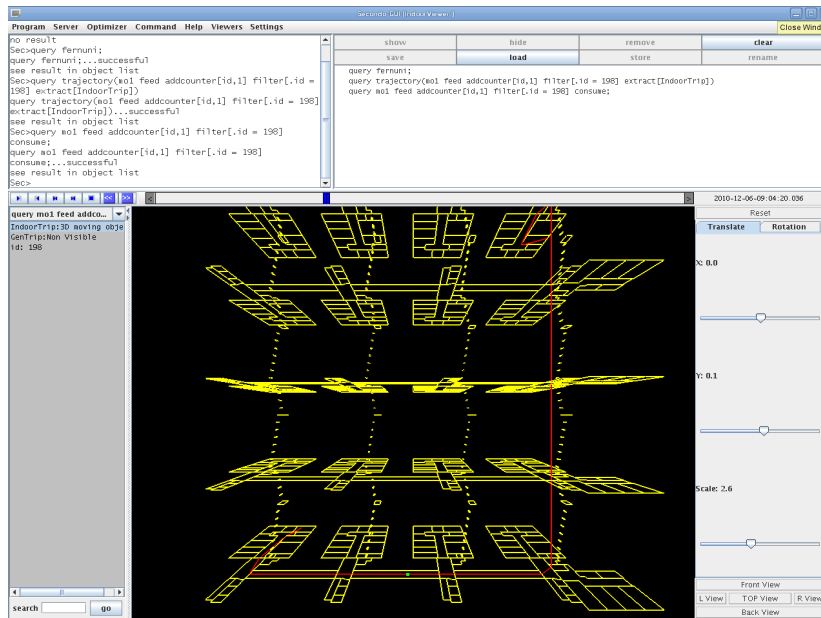


Figure 3: Indoor Visualization