

Article

# Modeling and Querying Moving Objects with Social Relationships

Hengcai Zhang <sup>1,2</sup>, Feng Lu <sup>1,2,3,\*</sup> and Jianqiu Xu <sup>4</sup>

<sup>1</sup> State Key Lab of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China; zhanghc@reis.ac.cn

<sup>2</sup> Fujian Collaborative Innovation Center for Big Data Applications in Governments, Fuzhou 350003, China

<sup>3</sup> Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China

<sup>4</sup> Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; jianqiu@nuaa.edu.cn

\* Correspondence: luf@reis.ac.cn; Tel.: +86-10-6488-8966

Academic Editors: Georg Gartner, Haosheng Huang and Wolfgang Kainz

Received: 23 April 2016; Accepted: 8 July 2016; Published: 15 July 2016

**Abstract:** Current moving-object database (MOD) systems focus on management of movement data, but pay less attention to modelling social relationships between moving objects and spatial-temporal trajectories in an integrated manner. This paper combines moving-object database and social network systems and presents a novel data model called Geo-Social-Moving (GSM) that enables the unified management of trajectories, underlying geographical space and social relationships for mass moving objects. A bulk of user-defined *data types* and corresponding *operators* are also proposed to facilitate geo-social queries on moving objects. An implementation framework for the GSM model is proposed, and a prototype system based on native *Neo4J* is then developed with two real-world data sets from the location-based social network systems. Compared with solutions based on traditional extended relational database management systems characterized by time-consuming table join operations, the proposed GSM model characterized by graph traversal is argued to be more powerful in representing mass moving objects with social relationships, and more efficient and stable for geo-social querying.

**Keywords:** moving objects; social network; data model; graph database; geo-social querying

## 1. Introduction

Moving object databases (MOD) allow modelling, manage moving entities such as people, vehicles and vessels, and have been extensively studied over the past few years, such as data modelling [1–6], indexing [7–10], and querying [11–15]. This research focuses on the management of movements and involved geographic environments, but ignores the social relationships between moving objects. Let us consider the following scenario. A user visiting a shopping mall is usually associated with geo-locations through check-ins made by mobile devices. The user has some online relationships in social networks, such as Facebook, Twitter or Foursquare. In so called location based social networks (LBSN) [16–18], there have been a bulk of query requests concerning not only movements with spatial and temporal characteristics, but also dynamic variation of social relationships:

- **Q1:** Tell me the nearest restaurants that have been checked in by my friends in this month.
- **Q2:** When, where and which colleagues did I met last week?

For the above examples, the MODs are good at answering the request "find the nearest restaurants around me", but they cannot answer the query **Q1** involving the social relationships.

The query **Q2** is also difficult for MODs because it needs to traverse both the movements and social relationships.

As we know, these geo-social data have great value for online recommender services, transportation and urban computing, human mobility research [19,20] and investigation of the connection between social ties and correlated activity [21,22]. The advancement of the wireless networks and mobile Internet further accelerate the explosive growth of the data. As of 2012, Facebook claims to have 600 million mobile users and 140 billion friend connections [23]. As of 2015, Foursquare has over 55 million users and over 7 billion check-ins worldwide [24]. The database systems should not only model the trajectories but also represent the social relationships of moving objects. More and more, real-world applications require ad hoc data management technologies, leading to a rise in new study challenges and opportunities in the field of MOD and geographical information systems (GIS) [21,25–27].

A natural approach to manage movement together with social relationships is to store them separately. For example, spatial database management systems (DBMS) or MOD could be used to manage trajectories and geographical space, and the social relationships can be managed with native graph databases. Obviously, such a simple combination will be extremely inefficient because it asks to traverse two databases constantly.

Another approach is to store together and retrieve together, where social relationships are transformed into tuple  $\langle o_1, rel, o_2 \rangle$  and stored into Spatial DBMS or MOD. However, it will generate much data redundancy, and many time-consuming table-join operations are needed to perform geo social queries. Graph database systems such as Flockdb, Neo4J, and Giraph, cannot be directly applied to store trajectories, although they are undoubtedly suitable for representing social relationships because of the lack of an abstraction level and upper logical data model [28].

There is a paucity of literature regarding integrated approaches to modelling and querying geo-social data. Although Pelekis and co-workers presented a novel graph-based model that uses a dynamic graph to represent semantic mobility timelines [17], related issues of modelling social relationships have not been addressed. This calls for the development of novel data models to store, index and query moving objects with social relationships.

In this paper, we focus on modelling the moving objects with social relationships and propose a composite graph-based data model called Geo-Social-Moving (GSM) where geographical space, trajectories and social relationships are all represented with graph structures. Geographical space is divided into a set of Voronoi regions. Each Voronoi region is modelled as a graph node and an edge connecting two Voronoi regions represents the spatial adjacency relationships between them. Trajectories of moving objects are represented with a movement graph. A trajectory is split into a set of trajectory units in different Voronoi regions. In other words, all the location points along a trajectory in a specific Voronoi region are collected and modelled as one trajectory unit, i.e., a node in the movement graph. Edges connecting trajectory units form a complete trajectory. A social graph is used to model moving objects with social relationships. It contains a set of nodes and relationships. In addition, we also create two edge types  $lk_g$  and  $lk_s$ .  $lk_g$  connects trajectory units in the movement graph and Voronoi region in the geographical graph, and  $lk_s$  connects trajectory units in the movement graph and moving objects in the social graph. Based on this model, both moving objects' trajectories and social relationships could be easily retrieved through multiple types of edges.

The contributions of this paper lie in the following aspects:

1. A comprehensive GSM model is proposed to represent geographical space, trajectories and social relationships of moving objects in an integrated manner. The type system for this model is defined and a set of operators are designed. Signatures and semantics of operators are also provided.
2. An implementation framework for a GSM model is proposed and implementation issues are addressed. A prototype based on the graph DBMS Neo4J [29] is implemented to verify the model.

- Extensive experiments with real-world geo social datasets are performed to evaluate the efficiency and performance of proposed GSM model. The results illustrated that the efficiency of queries enabled with GSM model outperforms the implementation based on the traditional relational DBMS.

The remainder of the paper is organised as follows. Section 2 summarizes related research work. Section 3 introduces the novel geo-social data model and elaborates on the basic concepts. Section 4 defines the *data types* and corresponding *operators*. Section 5 implements the presented data model and develops a prototype system. Section 6 conducts an experiment and analyses the results in detail. Finally, Section 7 concludes the paper and suggests future work.

## 2. Related Work

Early related studies on data models for moving objects focused on the representation of raw trajectories that consist of a temporal sequence of the positions of moving objects [4,17,30]. For example, the moving objects spatio-temporal (MOST) data model allowed the management of the current position of moving objects, and the prediction of future movement based on an important concept of the dynamic attribute that values change over time according to a given function of time [31–33]. Forlizzi et al. provided a discrete model for the abstract model presented by Guting and co-workers [34,35]. A discrete spatial-temporal-trajectory-based moving object database (DSTTMOD) that models trajectories as a function curve in three-dimensional space was developed [36]. It represents trajectories as a set of points, trajectory segments or a function  $f(x)$  of time. However, semantic information of trajectories such as stay points, objects' behaviour, contextual annotations or social relationships mentioned in our study was neglected [37,38]. Those data models for raw trajectories did not embrace different underlying environments such as free geographical spaces [6,37], networks [2,39], and indoor spaces [40–42]. Additionally, early related studies were more prone to simulate tracking data produced by a network-based generator [43], Oporto [44], GSTD [45], or GMOBench [46]. In addition, some research focused on the motion detection not only in dynamic scenes but also in static scenes based on the artificial neural network [47,48] and background subtraction techniques [49,50]. Chen and Huang presented a novel approach for vehicle detection based on probabilistic neural networks to detect moving vehicles in high bit-rate and low bit-rate video streams [51].

In recent years, with the development of location-based services, more and more mobile devices equipped with positioning technologies have generated massive real-world trajectories. Much additional information from the application context is considered in the modelling process. For example, the trajectories of a person can be interpreted as the sequences of points of interest such as a library, shopping mall, office building or museum. Hence, the research trends shift from raw trajectories to semantic trajectories [18,38,52–54]. A semantic trajectory can be defined as a trajectory enhanced with annotations such as activity, instant speed, transport mode and one or several complementary segmentations [37]. Among them, social relationships between moving objects are one of the important types of semantic information and need to be considered by the data management community, especially with respect to the trends of prevalent online social networks or location-based social networks [55]. This has provided the research motivation in the present paper. The first semantic model for trajectories represented a trajectory as a set of semantic places [53]. However, semantic information needs to be added to specific trajectory points. Zheni et al. introduced semantic activities into data model and presented a semantic model that depend on time geography theoretical framework [54,56]. Parent et al. incorporated semantic patterns and moving objects' behaviour to model and survey privacy issues [37]. A multi-layer framework for semantic trajectories at different levels of abstraction called "SeMiTri" introduced the concept of annotation [38,57]. In this model, spatio-temporal positions are complemented with annotations. To model more semantic information for data mining, Bogorny et al. presented a semantic trajectory conceptual data model named CONSTANT, which involves the concepts of

semantic sub trajectories, semantic points, and geographical places, events and behaviour [52,58,59]. Damiani et al. proposed a data model of symbolic trajectories representing different types of semantics [60]. In addition, data models for semantic trajectories based on ontology have been presented. For instance, Hu et al. proposed an ontology design pattern for semantic trajectories [61]. Nogueira et al. leveraged ontology to represent dynamic characteristics of trajectories [62]. To add address domain-independent and application-specific geometric and semantic facets, Yan et al. proposed an ontology framework that involves geography ontology, application domain ontology and geometric trajectory ontology [63]. A meta-model based on ontology and event approach was proposed by Boulmakoul and co-workers [64]. Pelekis and co-workers presented a graph-based model that allows management of trajectories and their semantic information [17]. It uses dynamic graph representation of the semantic mobility timelines.

### 3. Modelling Geo-Social Data

In this section, we propose the basic idea of the GSM model and introduce its core components, including the *geographical graph*, *social graph*, and *movement graph*, and related issues.

First, let us reconsider the meaning of trajectory as follows. A person with many friends in the physical world or online social networks, moves around a point of interest (POI) such as a shopping mall or museum, does something over a period of time, follows new friends in social networks, and receives their messages and the latest news. This scenario is similar to that when we surf from one web page to other web pages. Therefore, geographical space and object movements can also be represented with a graph structure similar to that used in the modelling of web pages. More specifically, POIs or regions in geographical space can be modelled as nodes in a graph. Additionally, the active records in these regions can be represented as nodes with relationships to these regions and moving objects.

#### 3.1. Geographical Graph

In the field of MOD, modelling the geographic space is beneficial to queries, e.g., finding all moving objects' trajectories in a specific region. Many subdivision methods such as grid, triangular irregular network and Voronoi diagram have been adopted. Considering that objects often move around common POIs in free space, it is helpful to connect the trajectories with POIs. In order to keep an equal number of location points in each partitioned regions, the Voronoi diagram is an effective decomposition of geographical space according to the position of discrete POIs [65]. Each POI generates a Voronoi region  $V$  that involves all points closer to this POI than to any other in Voronoi-diagram-based subdivision. Given a set of discrete POIs  $P = (p_1, \dots, p_n)$  in geographical space, a Voronoi region can be defined as:

$$V(p_i) = \{q | d(q, p_i) \leq d(q, p_j), i \neq j, p_i, p_j \in P\} \quad (1)$$

where  $V(p_i)$  is a Voronoi region,  $q$  is an arbitrary point in space, and  $d(q, p_j)$  is a distance function. This definition indicates that the distance from any point in the specific region to  $p_i$  is smaller than to other points. Figure 1a shows an example of Voronoi region. The whole geographical space is divided into eight Voronoi regions based on eight POIs.

Under the space subdivision using a Voronoi diagram, a geographical graph can be regarded as a collection of nodes and edges. A Voronoi region is represented as a single node. An edge connecting adjacent Voronoi regions represents the spatial relationships between them. The types of spatial relationships refer to the OGC (Open Geospatial Consortium) compliant nine-intersection model that presents a comprehensive definition of topological relationships between spatial objects in two-dimensional space [66,67]. Figure 1b shows the structure of the geographical graph that contains eight nodes and one kind of relationship *meet* that represents two adjacent Voronoi regions.

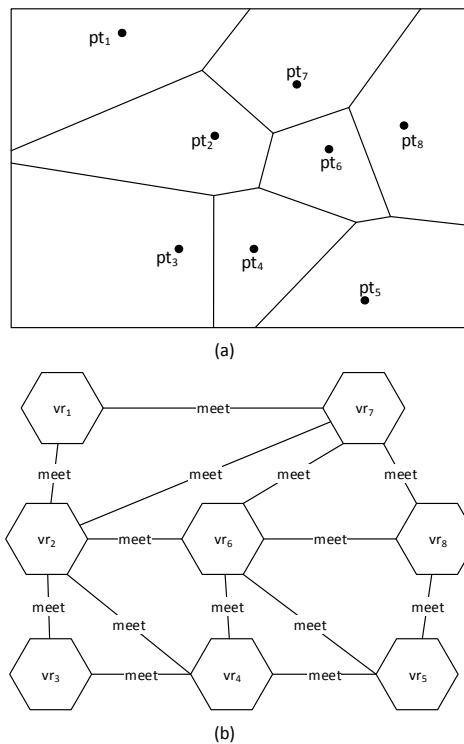


Figure 1. Geographical graph for free space.

### 3.2. Social Graph

Modelling social relationships between moving objects with graphs is taken for granted, where nodes correspond to a set of moving objects and edges correspond to a set of social relationships. The attributes of nodes could be static e.g., a name or type. The attributes of edges could be property fields representing a feature of social relationships in physical or cyber space.

In Figure 2, there are three objects ( $O_1, O_2$  and  $O_3$ ) and three types of social relationships (*friends*, *followship*, and *colleagues*).  $O_1$  works with  $O_3$ , and  $O_2$  is  $O_1$ 's friend.  $O_2$  and  $O_3$  follow each other on an online social network system.

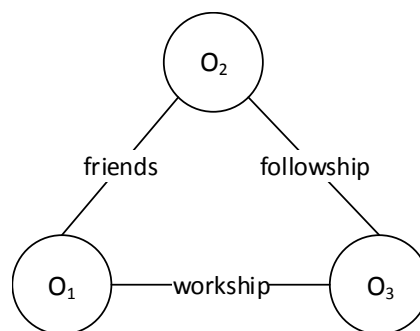
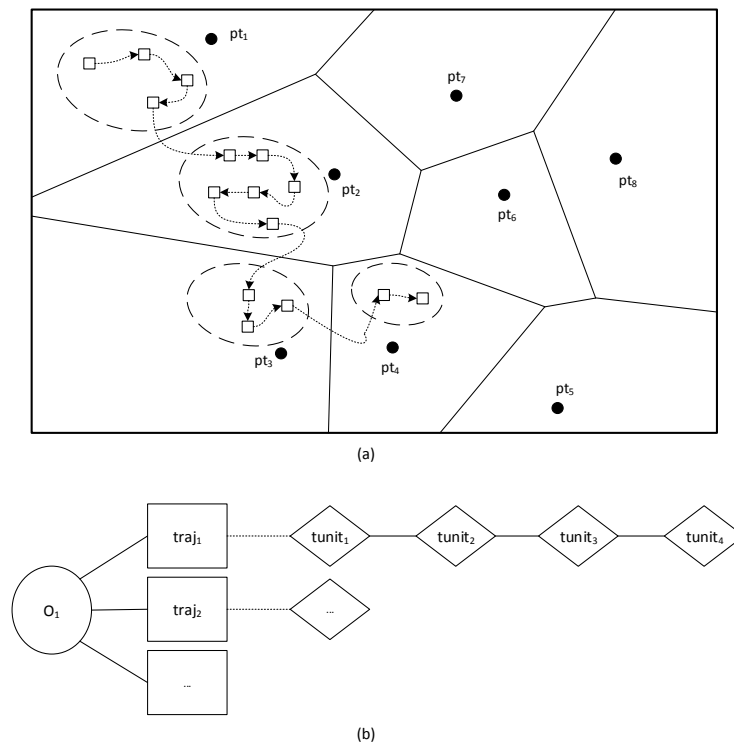


Figure 2. Social graph.

### 3.3. Movement Graph

Movement graphs are designed to model the trajectories of moving objects. In our model, an object's trajectory is split into trajectory segments using a Voronoi diagram as described by a geographical graph. Each segment is represented as a single graph node. The complete trajectory

is modelled as a set of nodes. A set of edges is then generated to associate these nodes sequentially. Figure 3 shows an example of a movement graph. In Figure 3a, there are eight Voronoi regions divided by POIs (black points). The rectangle denotes location points of moving objects. As seen from the figure, an object's location points are divided into four clusters. Hence, a definite trajectory  $traj_1$  for object  $O_1$  can be modelled into four segments, namely  $tunit_1$ ,  $tunit_2$ ,  $tunit_3$ ,  $tunit_4$ , as shown in Figure 3b.



**Figure 3.** Movement graph.

In the fields of MOD and GIS, there are two kinds of trajectory modelling approaches: *object-based* and *trip-based* approaches [68]. The *object-based* modelling approach models a moving object's continual locations as a single trajectory record. The complete trajectory of an object can thus be easily retrieved. It is also beneficial to human behaviour and mobility studies. However, this approach faces an oversize storage problem. In particular, when the time interval between two location points is too small, a trajectory may contain thousands of points.

Meanwhile, the *trip-based* modelling approach splits a continual trajectory into several trips i.e., a collection of smallest units of storage in a database system. The size of a trip is decided by segmentation methods, such as those employing regular time intervals and fixed lengths of trajectories. Although the *trip-based* modelling approach effectively avoids the oversize storage problem, it inevitably affects the efficiency of querying on whole trajectories because retrieving a complete trajectory that consists of a collection of segments requires the traversal of more records in a database. As shown in Figure 3b, all trajectories of an object can be easily retrieved with the proposed movement graph through graph traversal, which is similar to object-based modelling approaches. At the same time, the movement graph splits the whole trajectory into segments to keep the virtues of *trip-based* trajectory modelling approaches because the nodes in a movement graph naturally have links to nodes in the geographical graph (Voronoi regions).

Figure 4 illustrates an example of the proposed GSM model. A moving object's trajectory  $traj_1$  is composed of four trajectory units ( $tunit_1, \dots, tunit_4$ ). Each trajectory unit is connected to a specific

Voronoi region and also connected to a moving object, such as the edge for trajectory  $traj_1$  to a moving object  $O_1$ .

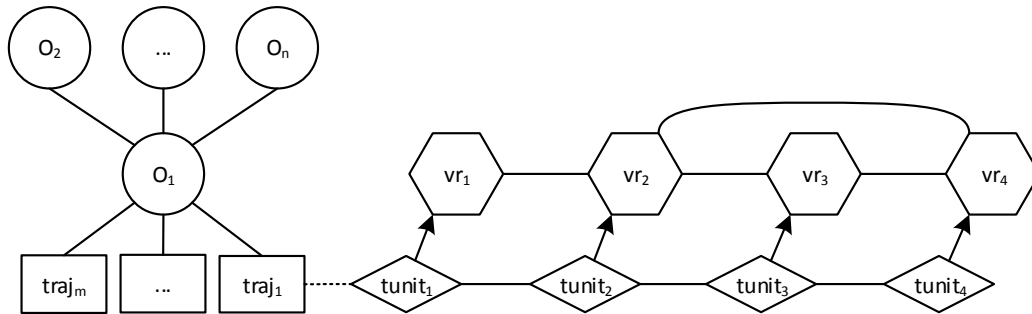


Figure 4. An example for the GSM model.

#### 4. Data Types and Operators

##### 4.1. Data Types

With the concepts defined in Section 3, this section gives a formal definition of graphs involved in the GSM model and provides *data types*  $G_g, G_s,$  and  $G_m$  to represent geographical space, social relationships and movement trajectories, respectively.

**Definition 1** (*Basic types*) Three basic types are used for the following definitions *int*, *real* and *bool*:

$$\underline{int} = Z \tag{2}$$

$$\underline{real} = R \tag{3}$$

$$\underline{bool} = \{true, false\} \tag{4}$$

**Definition 2** (*Temporal types*) Two time types are provided to represent the time *instant* and *period*:

$$\underline{instant} = R \tag{5}$$

$$\underline{period} = \{(s, e) | s, e \in \underline{instant}\} \tag{6}$$

**Definition 3** (*Geometry types*) The geometry types are employed from OGCs that release a series of specifications about geometry object models. The proposed GSM model involves three basic geometry types—*point*, *line* and *polygon*—to represent geographical space and trajectories.

$$\underline{point} = \{(lat, lon) | lat, lon \in \underline{real}\} \tag{7}$$

$$\underline{line} = \{(pt_1, pt_2, \dots, pt_n) | pt_i \in \underline{point}\} \tag{8}$$

$$\underline{polygon} = \{(l_1, l_2, \dots, l_n) | l_i \in \underline{line}\} \tag{9}$$

**Definition 4** (*Geographical graph structure*) The geographical space includes a set of Voronoi regions  $VR$  that are defined as:

$$VR = \{(rid, geom, p) | rid \in \underline{int}, geom \in \underline{polygon}, p \in \underline{point}\} \tag{10}$$

where  $rid$  is the identification of a Voronoi region,  $geom$  is represented by a geometry type *polygon*, and  $p$  denotes the point contained in the Voronoi region. Any position in this Voronoi region  $vr$  can then be defined as  $g_{pos}$ :

$$g_{pos} = \{(rid, pt) | rid \in \underline{int}, pt \in \underline{point}\}, \quad (11)$$

where  $rid$  is the identification of a Voronoi region and  $pt$  represents the latitude and longitude of the position.

A geographical graph structure  $G_g$  is represented as a pair of nodes  $V_g$  (Voronoi regions) and edges  $E_g$  (spatial relations between Voronoi regions), and is defined as follows:

$$G_g = (V_g, E_g) \quad (12)$$

$$V_g = \{(vr_1, vr_2, \dots, vr_n) | vr_i \in VR\} \quad (13)$$

In this paper, topological relationships  $R_g$  are defined as:

$$R_g = \{ \langle vr_i, vr_j, type \rangle | vr_i, vr_j \in V_g, type \in (meet, equal) \} \quad (14)$$

The relationship *meet* denotes that Voronoi regions  $vr_i$  and  $vr_j$  are adjacent to each other. In addition, *equal* means that two regions are the same.

$E_g$  represents a set of spatial relationships between Voronoi regions.

$$E_g = \{(grel_1, grel_2, \dots, grel_m) | grel_i \in R_g\} \quad (15)$$

**Definition 5** (*Social graph structure*) A moving object is defined as:

$$MO = \{(mid, name, param), mid \in \underline{int}, name \in \underline{string}\} \quad (16)$$

where  $mid$  and  $name$  are the identification and name of a moving object. In addition,  $param$  refers to the other attributes set of the object.

A social graph structure  $G_s$  is used to model moving objects and the social relations between them. It is defined as follows:

$$G_s = (V_s, E_s) \quad (17)$$

Here,  $V_s$  denotes moving objects  $MO$  and  $E_s$  denotes the social relations between two moving objects:

$$V_s = \{(mo_1, mo_2, \dots, mo_n) | mo \in MO\} \quad (18)$$

$$E_s = \{(srel_1, srel_2, \dots, srel_m) | srel_i \in R_s\} \quad (19)$$

$$R_s = \{ \langle mo_i, mo_j, type \rangle | mo_i, mo_j \in V_s, type \in \underline{string} \} \quad (20)$$

The *string* type of  $type$  represents social relations between moving objects, including *colleagues*, *friendships*, *followership*, *interest groups* or *fan relationships*, to name but a few.

**Definition 6** (*Movement graph structure*) A trajectory unit  $Traj_{unit}$  (sometimes called the sub-trajectory) of an object in a Voronoi region is defined as:

$$Traj_{unit} = \{((mo, vr), (t_i, g_{pos}^i, v_i)_{i=1}^m) | mo \in MO, vr \in VR, t_i \in \underline{instant}, v_i \in \underline{real}\} \quad (21)$$

where  $t_i$  denotes the time of location points and  $m$  denotes the count of location points. Then, a moving object's trajectory can be modelled as a collection of trajectory units.

$$trajectory = \{(mo, \sum_{i=1}^q t_{unit_i}) | mo \in MO, t_{unit_i} \in Traj_{unit}\} \quad (22)$$



A movement graph structure  $G_m$  is defined to represent the trajectories of moving objects. The novelty of this structure lies in the fact that the trajectory units that belong to a whole trajectory are represented as nodes of a graph. The geometry type *line* of segments is encoded with WKB (well-known binary) format representing vector geometry objects, and is stored as a binary attribute of a graph node. It is defined as:

$$G_m = (V_m, E_m) \quad (23)$$

$$V_m = \{(tuint_1, tuint_2, \dots, tuint_n) | tuint_i \in Traj_{unit}\} \quad (24)$$

The relationship  $E_m$  is defined as follows:

$$E_m = \{(mrel_1, mrel_2, \dots, mrel_m) | mrel_i \in R_m\} \quad (25)$$

$$R_m = \{R_m^m, R_m^s, R_m^g\} \quad (26)$$

$$R_m^m = \{\langle tunit_i, tunit_j, type \rangle | tunit_i, tunit_j \in V_m, type \in \underline{string}\} \quad (27)$$

$$R_m^s = \{\langle tunit_i, mo_j, type \rangle | tunit_i \in V_m, mo_j \in V_s, type \in \underline{string}\} \quad (28)$$

$$R_m^g = \{\langle tunit_i, vr_j, type \rangle | tunit_i \in V_m, vr_j \in V_g, type \in \underline{string}\} \quad (29)$$

where  $R_m^m$  are used to represent the ordinal relations between trajectory segments that belong to the same trajectory. They help to easily retrieve the whole trajectory of a moving object. The relationships  $R_m^s$  represent the relationships between segment *tunit* and object *mo*.  $R_m^g$  represent the relationships between segment *tunit* and Voronoi regions *vr*. They provide the capability of finding all trajectories in a specific geographical Voronoi region.

#### 4.2. Operators

In this section, we provide the bulk of operations on  $G_g$ ,  $G_s$  and  $G_m$ , as illustrated in Table 1. We use signature  $\alpha \times \beta \rightarrow \delta$  to define operation signatures and semantics. It means that the data type variables  $\alpha$  and  $\delta$  could be instantiated; for example,  $int \times int \rightarrow int$  is used to multiply two numbers together.

**Definition 7 select.** The *select* operator receives the query condition such as the field value of attributes, and the specific graph layer of the GSM model, and returns all graph nodes that meet the condition. Depending on the hierarchical graph layers, it is written as  $select(G_g, c)$ ,  $select(G_s, c)$ , or  $select(G_m, c)$  for queries on a geographical graph, social graph, and movement graph, respectively, where  $c$  is the query condition.

For example, to find graph nodes with the name 'David' over a social graph, the query expression is written as:

$$select(\underline{G_s}, name = 'David') \quad (30)$$

**Definition 8 nodevalues.** The *nodevalues* operator extracts an attribute field's value from graph node. For example, to find the email address of a moving object is written as :

$$nodevalues(\underline{mo}, email) \quad (31)$$

Note that we define the type of return value as generic type *string*. In real word applications, many data type conversion operations are needed according to actual data types. For instance, the field *age* of a moving object belongs to *int* type.

**Definition 9 getnodes.** The *getnodes* operator retrieves the adjacent nodes according to a specific relationship value. For example, retrieving all trajectory segments of moving object *mo* is written as:

$$getnodes(\underline{mo}, lk_s) \quad (32)$$

Benefitting from this graph structure, many query operations become more convenient than traditional relational database solution. For instance, the basic query ‘finding all David’s friends’ needs many table-joint operations in the traditional table approach. In contrast, the query expression in the GSM model could be written as:

$$\text{getnodes}(\underline{mo}, \text{type} = \text{friends}) \quad (33)$$

**Table 1.** Definition of operations.

Operations	Signature
<b>Select</b>	$\underline{G}_g \times \text{string} \rightarrow \underline{VR}$
	$\underline{G}_s \times \text{string} \rightarrow \underline{MO}$
	$\underline{G}_m \times \text{string} \rightarrow \underline{Traj_{unit}}$
<b>Nodevalues</b>	$\underline{VR} \times \text{string} \rightarrow \text{string}$
	$\underline{MO} \times \text{string} \rightarrow \text{string}$
	$\underline{Traj_{unit}} \times \text{string} \rightarrow \text{string}$
<b>Getnodes</b>	$\underline{MO} \times \text{string} \rightarrow \underline{MO}$
	$\underline{MO} \times \text{string} \rightarrow \underline{VR}$
	$\underline{MO} \times \text{string} \rightarrow \underline{Traj_{unit}}$
<b>Gettrajectory</b>	$\underline{MO} \times \underline{instant} \rightarrow \underline{trajectory}$
<b>Atinstants</b>	$\underline{trajectory} \times \underline{instant} \rightarrow \underline{gpos}$
<b>Atperiods</b>	$\underline{trajectory} \times \underline{period} \rightarrow \underline{Traj_{unit}}$
<b>Exinstants</b>	$\underline{trajectory} \times \underline{point} \rightarrow \underline{instant}$
<b>Distance</b>	$\underline{trajectory} \times \underline{trajectory} \rightarrow \underline{real}$

**Definition 10 gettrajectory.** The *gettrajectory* operator returns the whole trajectory *trajectory* of a moving object *MO* at specific date *instant*. For example, finding David’s trajectory at 22th July 2015 can be written as :

$$\text{gettrajectory}(\text{select}(\underline{G}_s, \text{name} = 'David'), 2015 - 07 - 22) \quad (34)$$

The implementation of this operator relies on the operator *getnodes*.

**Definition 11 atinstants.** The *atinstants* operator returns any position of a moving object at the query time *instant*. For example, a query ‘where is David at 10:00 on 22th July 2015’ can be written as:

$$\text{atinstants}(\text{gettrajectory}(\text{select}(\underline{G}_s, \text{name} = 'David'), 2015 - 07 - 22), 10 : 00) \quad (35)$$

This operator needs interpolated operation of trajectory between two adjacent points because of trajectories’ discreteness.

**Definition 12 atperiods.** The *atperiods* operator receives the query time condition *period* and the specific trajectory, and returns a trajectory unit *tunit* that meets the condition.

**Definition 13 exinstants.** The *exinstants* operator is a reverse operation of *atinstants* operator. It receives a coordinate and returns the time when an object passes through it.

**Definition 14 distance.** The *distance* operator calculates the minimum Euclidean distance between two trajectories. It finds all distance from a point of the first trajectory to any points of the second trajectory and returns the minimum value. It is written as:

$$\text{distance}(\underline{trajectory}, \underline{trajectory}) \quad (36)$$

## 5. Implementation

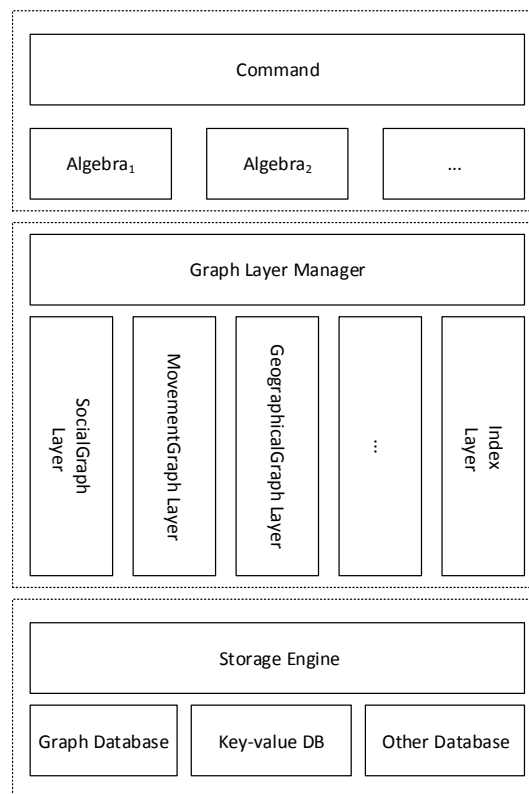
### 5.1. System Architecture

An implementation framework for the GSM model is shown in Figure 5. The framework has three major components: a storage engine, graph layer manager and algebra module.

The storage engine provides an underlying data storing solution for the GSM model's composite graph structure. It is designed to support different database systems. For a graph database, the GSM model's related structures can be stored as a basic property graph. The formats of WKB (Well-known binary) and WKT (Well-known text) can be used to represent Voronoi regions and trajectories of moving objects.

A graph layer manager provides a logic module of multiple management for the graph data, which provides more additional functions than native graph database systems that only provide methods of managing nodes and edges. In the GSM model, the graph layer manager deals with the social graph layer, movement graph layer, and geographical graph layer. In addition, the indexing structure is stored as an index graph layer.

Algebra modules group together a set of data types and operators related to the specific data model. The algebra plugin architecture provides a great deal of flexibility and control over data model implementation, similar to Secondo architecture [69,70].



**Figure 5.** Implementation framework for the GSM model.

### 5.2. Prototype Implementation

A prototype is implemented to verify the GSM model in the graph database management system *Neo4J*, as shown in Figure 6. *GSMRoot* denotes the root node of the graph database, with three child nodes (i.e., *GG*, *MG*, and *SG*) representing the root nodes for the *geographical graph*, *movement graph*, and *social graph*, respectively. The child nodes of the *GG* node are geographical Voronoi regions, with edges representing the spatial relationships between them. The *MG* node has many trajectory

segments as child nodes. Similarly, the child nodes of SG nodes denote moving objects. In addition, trajectory segments have links with geographical Voronoi regions and moving objects.

Some query examples are listed and corresponding query statements are described as follows.

**Query 1:** What is Jack's email address?

```
nodevalues(
  select( $G_s$ , name = 'jack'), email)
```

**Query 2:** Find all of Jack's friends.

```
nodevalues(
  getnodes(
    select( $G_s$ , name = 'Jack'), friendship), name)
```

**Query 3:** Find the trajectory distance between Jack and John.

```
distance(
  gettrajectory(
    select( $G_s$ , name = 'Jack'), 2015 - 07 - 22),
  gettrajectory(
    select( $G_m$ , name = 'John'), 2015 - 07 - 22)))
```

**Query 4:** Where was Jack at 10 a.m.?

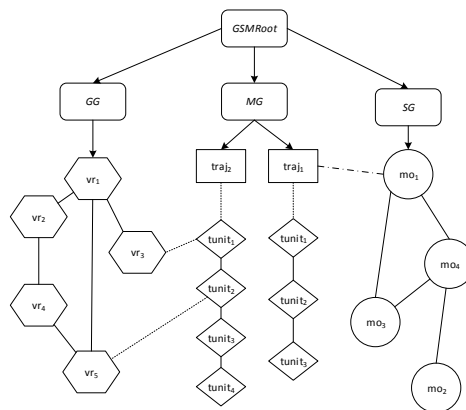
```
atinstants(
  gettrajectory(
    select( $G_s$ , name = 'Jack'), 2015 - 07 - 22), 10 : 00)
```

**Query 5:** Find Jack's trajectories between 10 a.m. and 12 a.m.

```
atperiods(
  gettrajectory(
    select( $G_s$ , name = 'Jack'), 2015 - 07 - 22), (10 : 00, 12 : 00))
```

**Query 6:** When did Jack pass Joy City shopping mall?

```
exinstants(
  gettrajectory(
    select( $G_s$ , name = 'Jack'), 2015 - 07 - 22), Joycity)
```



**Figure 6.** Implementation of the GSM model.

## 6. Experiments

### 6.1. Experimental Setting

Neo4J and Eclipse were used to implement the GSM data model on a standard personal computer with an Intel i7-4790 CPU and 8G RAM. Data collected from real-world LBSNs, namely

*Brightkite* (BK) and *Gowalla* (GO), provided by the Stanford Network Analysis Project [19], were used to conduct experiments so as to verify the effects of the presented model. Data from *Gowalla* contained 196,591 user nodes, 950,327 *friendship* edges, and 6,442,890 check-in records, while data from *Brightkite* consisted of 58,228 user nodes, 214,078 *friendship* edges, and 4,491,143 check-in records. POIs were collected from the Census Bureau's MAF/TIGER database. As shown in Figure 7, four states with more check-in records were chosen as the experimental regions; i.e., Washington (DC), California (CA), New York (NY), and Texas (TX). Table 2 describes the experimental data in detail.

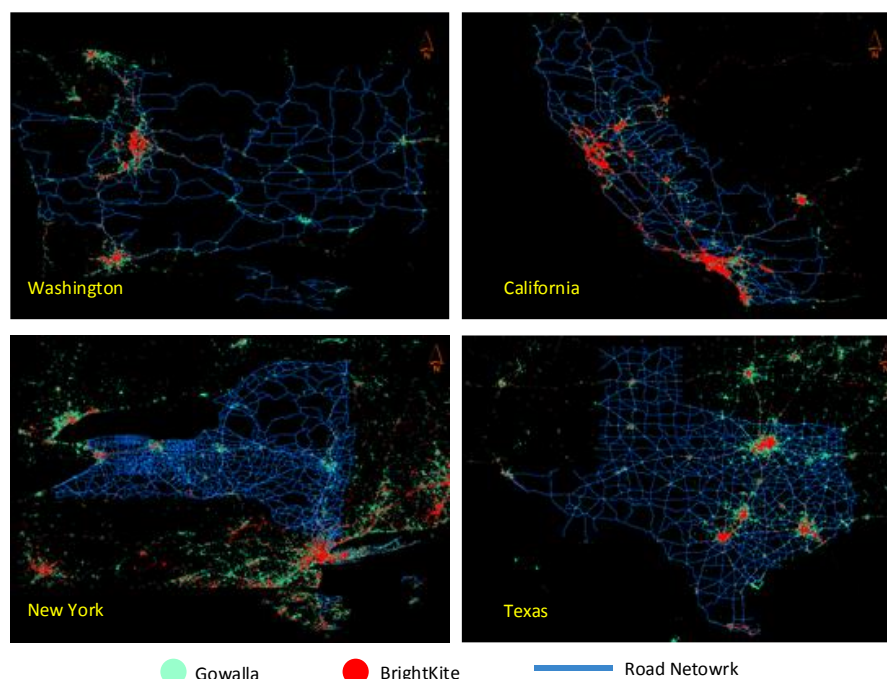


Figure 7. Spatial distribution of check-in records.

Table 2. Experimental data.

	Voronoi Regions	Moving Objects	Social Relationships	Check-in Records
DC-GO	8655	5058	163,622	199,469
DC-BK	18,655	797	59,194	93,816
CA-GO	57,251	16,684	513,406	739,687
CA-BK	57,251	2450	142,136	293,542
NY-GO	21,657	12,388	391,144	364,940
NY-BK	21,657	1789	108,678	157,020
TX-GO	40,178	17,996	524,352	984,493
TX-BK	40,178	1328	99,528	111,247

Experiments were conducted to test the performance of queries with and without GSM model support. Considering the efficiency of managing various spatial-temporal data involved in a more general environment, the widely used relational database management system PostgreSQL with a spatial extension PostGIS was used to manage these original data but without GSM model support, as a comparison, where three tables—*objects*, *socialrelation* and *trajectory*—were created to store moving objects, social relationships and trajectories, respectively, by using the following SQL expressions:

```
CREATE TABLE objects (moID int, email char, oname char);
CREATE TABLE socialrelation (fromMOID int, relation char, toMOID int);
CREATE TABLE trajectory (trajID int, traj geometry, oname char);
```

Additionally, in order to accelerate the query performance, we created three indices:  $idx_{mo}$ ,  $idx_{ids}$  and  $idx_{traj}$ , as shown in the following SQL expressions.

```
CREATE INDEX idxmo ON objects (moID);
CREATE INDEX idxids ON socialrelation (fromMOID,toMOID);
CREATE INDEX idxtraj ON trajectory USING RTREE (traj).
```

Table 3 gives four query expressions used to evaluate the performance of the proposed GSM model. The query expression *QE1* returns a moving object's location at a specified time. *QE2*, a typical geo-social query, returns the locations of a moving objects's friends at a specified time. The query expression *QE3* retrieves a moving object's trajectories in a definite region within a specified time period. Lastly, the query expression *QE4* finds all trajectories of a moving object's friends who moved around a specified region within a certain time period.

**Table 3.** Tested query expressions for the GSM model.

	Query Expression
<i>QE1</i>	<i>atinstants</i> ( <i>gettrajectory</i> ( <i>select</i> ( $G_s$ , name = 'Jack'), 2015 - 07 - 22), 10 : 00)
<i>QE2</i>	<i>atinstants</i> ( <i>gettrajectory</i> ( <i>getnodes</i> ( <i>select</i> ( $G_s$ , name = 'Jack'), friendship), 2015 - 07 - 22), 10 : 00)
<i>QE3</i>	<i>atperiods</i> ( <i>gettrajectory</i> ( <i>select</i> ( $G_s$ , name = 'Jack'), 2015 - 07 - 22), (10 : 00, 12 : 00))
<i>QE4</i>	<i>exinstants</i> ( <i>gettrajectory</i> ( <i>getnodes</i> ( <i>select</i> ( $G_s$ , name = 'Jack'), friendship), 2015 - 07 - 22)), Joycity)

## 6.2. Experimental Results

Table 4 gives the performance of different query expressions in a prototype system developed with the proposed GSM model support, and also in a naive implementation. With the increase in social relations or check-in records, the prototype system supported with the GSM model behaved more stably. This is because the operators *gettrajectory* and *getnodes* have to be translated into time-consuming relational table join operations in the naive implementation. In contrast, the implementation supported with the GSM model adopted concurrent graph traversal operations to conduct these queries.

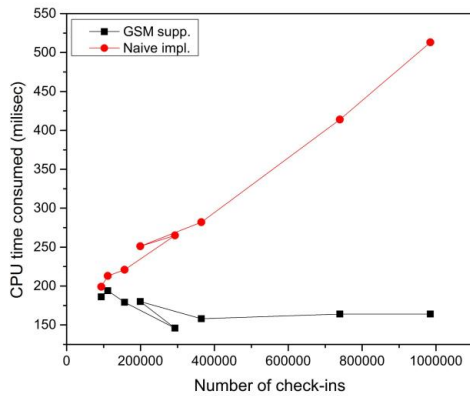
Figure 8 shows the relations between query performance and data volume. With the increase in check-in records or social relations of moving objects, the relational table join operations made the naive implementation impractical, whereas the performance of the implementation supported with the GSM model remained stable.

Figure 9 shows a schematic of a query (*QE2* and *QE4* are taken as examples) in the GSM-model-supported implementation. For *QE2*, we first used the operator *getnodes* to find Jack's friends, and grabbed their trajectory nodes through edges  $R_m^s$ . The segment nodes satisfying the time condition were then used to retrieve the nodes of the Voronoi region through edges  $R_m^g$ , and return the final results of queries. A similar schematic of *QE4* is also given in the figure.

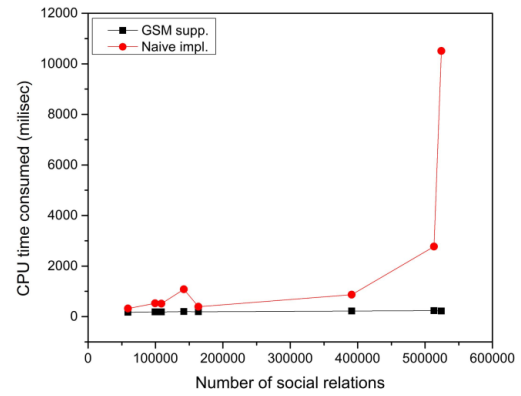
**Table 4.** Performance of queries (CPU time/ms).

	QE1		QE2		QE3		QE4	
	GSM Supp.	Naive Impl.	GSM Supp.	Naive Impl.	GSM Supp.	Naive Impl.	GSM Supp.	Naive Impl.
CA-BK	146	265	200	1082	213	256	185	2182
CA-GO	164	414	236	2771	150	385	189	3894
NY-BK	179	221	185	516	203	230	182	1474
NY-GO	158	282	219	865	142	273	171	1869
TX-BK	194	213	185	524	238	259	219	1528
TX-GO	164	513	220	10510	151	537	177	11593
DC-BK	186	199	173	318	202	204	167	1260
DC-GO	180	251	195	390	143	240	167	1367

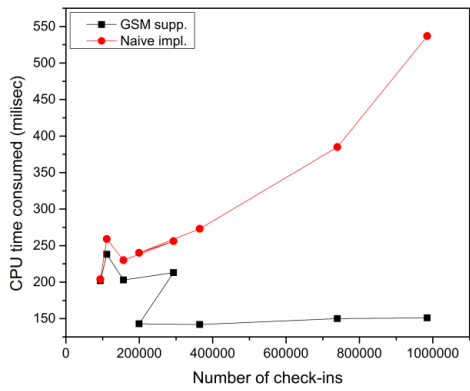
\* GSM Supp. means the implementation supported with GSM model; \* Naive Impl. means the naive implementation.



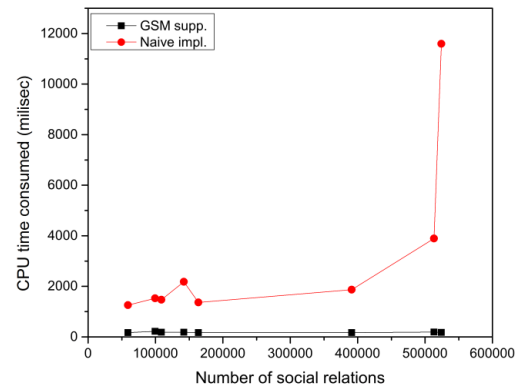
(a)QE1



(b)QE2



(c)QE3



(d)QE4

**Figure 8.** Relationships between performance and data volume in queries.

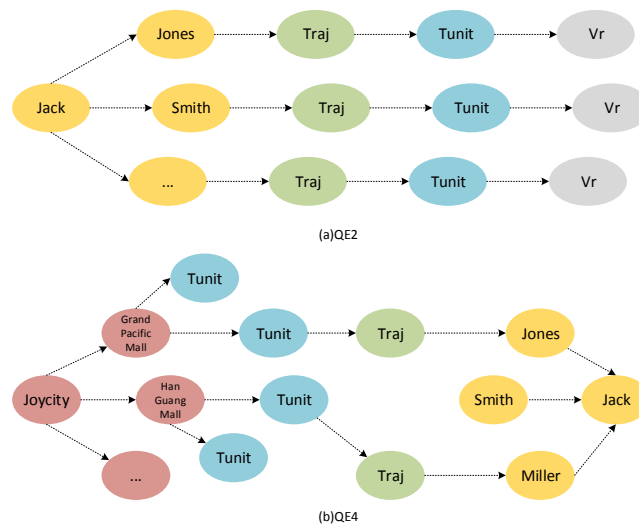


Figure 9. Schematic of queries supported with the GSM model.

## 7. Discussion

The present study proposes a GSM model for the integrated management and queries of moving objects' trajectories, social relationships and underlying geographical space. However, there are still limitations worthy of attention.

Voronoi diagrams based on POIs are used to divide free geographical space into a bulk of Voronoi regions, and these regions are then abstracted as nodes of a geographical graph. This is suitable for the uneven spatial distribution of trajectories, especially for movement traces generated by an LBSN e.g., check-in records. Nevertheless, it also raises the problem of polygonal fragmentation because of the aggregation of POIs. In this case, there will be too many small Voronoi regions and trajectory segments. This will affect the efficiency of data management and queries. A preparatory POI clustering may be helpful in mitigating the problem.

The proposed GSM model can be finely tuned to support the representation of semantic information, such as stay points or an object's behaviour, and even other activity environments such as indoor or network spaces. However, storing too much information through the attributes of nodes or edges will make the graphs overstaffed or redundant. For example, in movement graphs, some nodes may record too many location points and may hinder the querying or index performance to some extent. Compressing the trajectories and storing the attributes of graph nodes or edges with a binary system may be a useful strategy for solving the problem, with a trade-off with the efficiency of data retrieval.

The experiments conducted in this paper used check-in records from *Gowalla* and *Brightkite* systems. These records have notably different time intervals between two adjacent location points. Sometimes, objects might only have several check-in records in different locations on one day, and sometimes may have hundreds of records in a few hours. In contrast with the case for GPS-based trajectories, such as floating car data, these irregular time intervals will be a challenge to the query performance. Other kinds of trajectories will be further used to verify the proposed GSM model. In addition, the diversity of relationships for moving objects against performance will also need to be tested.

## 8. Conclusions

This paper presented a composite-graph-based data model for the integrated representation of trajectories, social relationships and geographical space for moving objects in free space in order to tackle management challenges related to the explosive growth of geo-social data. A bulk of



user-defined data types and corresponding operators were proposed to facilitate geo-social queries on the moving objects. An implementing framework was presented and a prototype developed. Experiments for the proposed data types and operators were conducted to illustrate the efficiency of the data model. It is a step forward in the data modelling of moving objects in that it considers complex social relationships and large-scale geographical spaces, and even indoor spaces.

Further work will involve the development of a corresponding index structure and various query algorithms, and the distributed implementation of a data model using a large-scale graph commutating processing frameworks such as Pregel or Bulk Synchronous Parallel.

**Acknowledgments:** This research was supported by the National Natural Science Foundation of China (Grant No. 41401460, 41421001) and State Key Research Development Program of China (Grant No. 2016YFB0502104), we also thank the anonymous referees for their helpful comments and suggestions.

**Author Contributions:** Hengcai Zhang, Feng Lu and Jianqiu Xu provided the core idea for this study. Hengcai Zhang implemented the GSM model and carried out the experimental validation. Hengcai Zhang and Feng Lu wrote the main manuscript. Jianqiu Xu made the important comments and suggestions for this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

GSM	Geo-Social-Moving
DC	Washington
CA	California
NY	New York
TX	Texas
QE1	Query expression 1
QE2	Query expression 2
QE3	Query expression 3
QE4	Query expression 4

## References

1. Guting, R.H.; Lu, J. Parallel SECONDO: Scalable query processing in the cloud for non-standard applications. *SIGSPATIAL Spec.* **2015**, *6*, 3–10.
2. Guting, R.H.; Ding, Z. Modeling and querying moving objects in networks. *VLDB J.* **2006**, *15*, 165–190.
3. Hajari, H.; Hakimpour, F. A spatial data model for moving object databases. *Int. J. Database Manag. Syst.* **2014**, *6*, 1–20.
4. Meng, X.; Ding, Z.; Xu, J. *Moving Objects Modeling, Moving Objects Management*; Springer: Berlin, Germany; Heidelberg, Germany, 2014.
5. Chen, B.Y.; Yuan, H.; Li, Q.; Shaw, S.-L.; Lam, W.H.; Chen, X. Spatiotemporal data model for network time geographic analysis in the era of big data. *Int. J. Geogr. Inf. Sci.* **2015**, *30*, 1–31.
6. Wolfson, O.; Xu, B.; Chamberlain, S.; Jiang, L. *Moving Objects Databases: Issues and Solutions*; IEEE: Chicago, IL, USA, 1998.
7. Ding, Z. UTR-tree: An index structure for the full uncertain trajectories of network-constrained moving objects. In Proceedings of the Ninth International Conference on Mobile Data Management (MDM 2008), Beijing, China, 27–30 April 2008; pp. 33–40.
8. Jung, H.; Kim, Y.S.; Chung, Y.D. QR-tree: An efficient and scalable method for evaluation of continuous range queries. *Inf. Sci.* **2014**, *274*, 156–176.
9. Pfoser, D.; Jensen, C.S.; Theodoridis, Y. *Novel Approaches to the Indexing of Moving Object Trajectories*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2000.
10. Xu, J.; Guting, R.H.; Zheng, Y. The TM-RTree: An index on generic moving objects for range queries. *Geoinformatica* **2015**, *19*, 487–524.

11. Sacharidis, D.; Skoutas, D.; Skoumas, G. Continuous monitoring of nearest trajectories. In Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas, TX, USA, 4–7 November 2014; pp. 361–370.
12. Shao, Z.; Taniar, D. Enhanced range search with objects outside query range. *World Wide Web* **2015**, *18*, 1–23.
13. Silvestri, C.; Lettich, F.; Orlando, S.; Jensen, C.S. GPU-based computing of repeated range queries over moving objects. In Proceedings of the 2014 22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Torino, Italy, 12–14 February 2014; pp. 640–647.
14. Zhang, R.; Qi, J., Lin, D.; Wang, W.; Wong, R.C.-W. A highly optimized algorithm for continuous intersection join queries over moving objects. *VLDB J.* **2012**, *21*, 561–586.
15. Trajcevski, G.; Tamassia, R.; Cruz, I.F.; Scheuermann, P.; Hartglass, D.; Zamierowski, C. Ranking continuous nearest neighbors for uncertain trajectories. *VLDB J.* **2011**, *20*, 767–791.
16. Long, J.A.; Nelson, T.A. A review of quantitative methods for movement data. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 292–318.
17. Pelekis, N.; Theodoridis, Y.; Janssens, D. On the management and analysis of our lifesteps. *ACM SIGKDD Explor. Newsl.* **2014**, *15*, 23–32.
18. Zheng, Y. Trajectory data mining: an overview. *ACM Trans. Intell. Syst. Technol. (TIST)* **2015**, *6*, 29–41.
19. Cho, E.; Myers, S.A.; Leskovec, J. Friendship and mobility: User movement in location-based social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 1082–1090.
20. Scellato, S.; Noulas, A.; Mascolo, C. Exploiting place features in link prediction on location-based social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 1046–1054.
21. Crandall, D.J.; Backstrom, L.; Cosley, D.; Suri, S.; Huttenlocher, D.; Kleinberg, J. Inferring social ties from geographic coincidences. *Proc. Natl. Acad. Sci.* **2010**, *107*, 22436–22441.
22. Tang, W.; Zhuang, H.; Tang, J. Learning to infer social ties in large networks. *Mach. Learn. Knowl. Discov. Databases* **2011**, *6913*, 381–397.
23. Fowler, G.A. Facebook: One billion and counting. *Wall Street J.* **2012**, *4*, 1–10.
24. Chorley, M.J.; Whitaker, R.M.; Allen, S.M. Personality and location-based social networks. *Comput. Human Behav.* **2015**, *46*, 45–56.
25. Ahuja, R.; Armenatzoglou, N.; Papadias, D.; Fakas, G.J. Geo-Social Keyword Search. *Adv. Spat. Tempor. Databases* **2015**, *9329*, 431–450.
26. Shi, J.; Mamoulis, N.; Wu, D.; Cheung, D.W. Density-based place clustering in geo-social networks. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, Snowbird, UT, USA, 22–27 June 2014; pp. 99–110.
27. Zhuang, H.; Tang, J.; Tang, W.; Lou, T.; Chin, A.; Wang, X. Actively learning to infer social ties. *Data Min. Knowl. Discov.* **2012**, *25*, 1–28.
28. Angles, R.; Gutierrez, C. Survey of graph database models. *ACM Computing Surv. (CSUR)* **2008**, *40*, 1–10.
29. Holzschuher, F.; Peinl, R. Performance of graph query languages: Comparison of cypher, gremlin and native access in Neo4j. In Proceedings of the Joint EDBT/ICDT 2013 Workshops, Genoa, Italy, 18–22 March 2013; pp. 195–204.
30. Schneider, M. *Moving Objects in Databases and GIS: State-of-the-art and Open Problems, Research Trends in Geographic Information Science*; Springer: Berlin, Germany, 2009.
31. Sistla, A.P.; Wolfson, O.; Chamberlain, S.; Dao, S. Modeling and querying moving objects. In Proceedings of the International Conference on Data Engineering (ICDE), Birmingham, UK, 7–11 April 1997; pp. 422–432.
32. Wolfson, O.; Chamberlain, S.; Dao, S.; Jiang, L.; Mendez, G. *Cost and Imprecision in Modeling the Position of Moving Objects*; IEEE: Chicago, IL, USA, 1998.
33. Wolfson, O.; Chamberlain, S.; Kalpakis, K.; Yesha, Y. Modeling moving objects for location based services. *Dev. Infrastruct. Mob. Wirel. Syst.* **2002**, *2538*, 46–58.
34. Forlizzi, L.; Guting, R.H.; Nardelli, E.; Schneider, M. A data model and data structures for moving objects databases. *ACM SIGMOD* **2000**, *29*, 319–330.
35. Guting, R.H.; Böhlen, M.H.; Erwig, M.; Jensen, C.S.; Lorentzos, N.A.; Schneider, M.; Vazirgiannis, M. A foundation for representing and querying moving objects. *ACM Trans. Database Syst. (TODS)* **2000**, *25*, 1–42.

36. Meng, X.; Ding, Z. DSTTMOD: A future trajectory based moving objects database. *Database Expert Syst. Appl.* **2003**, *2736*, 444–453.
37. Parent, C.; Spaccapietra, S.; RENSO, C.; Andrienko, G.; Bogorny, V.; Damiani, M.L.; Macedo, J.; Pelekis, N.; Theodoridis, Y.; Yan, Z. Semantic trajectories modeling and analysis. *ACM Computing Surv.* **2013**, *45*, 39–76.
38. Yan, Z.; Chakraborty, D.; Parent, C.; Spaccapietra, S.; Aberer, K. Semantic trajectories: Mobility data computation and annotation. *ACM Trans. Intell. Syst. Technol. (TIST)* **2013**, *4*, 49.
39. Sandu Popa, I. *Modeling, Querying and Indexing Moving Objects with Sensors on Road Networks*; University of Versailles-Saint-Quentin: Paris, France, 2010.
40. Jensen, C.S.; Lu, H.; Yang, B. Indoor—A new data management frontier. *IEEE Data Eng. Bull.* **2010**, *33*, 12–17.
41. Jin, P.; Zhang, L.; Zhao, L.; Wang, H.; Yue, L. Electronic RFID-based indoor moving objects: Modeling and applications. *Adv. Mech. Electron. Eng.* **2012**, *177*, 455–461.
42. Kim, Y.; Kang, H.; Lee, J. Development of indoor spatial data model using CityGML ADE. *J. Korea Spat. Inf. Soc.* **2013**, *21*, 11–21.
43. Brinkhoff, T. A framework for generating network-based moving objects. *GeoInformatica* **2002**, *6*, 153–180.
44. Saglio, J.-M.; Moreira, J. Oporto: A realistic scenario generator for moving objects. *GeoInformatica* **2001**, *5*, 71–93.
45. Theodoridis, Y.; Silva, J.R.; Nascimento, M.A. On the generation of spatiotemporal datasets, Advances in Spatial Databases. *Springer* 1999, *1651*, 147–164.
46. Xu, J.; Guting, R.H. GMOBench: A benchmark for generic moving objects. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 7–9 November 2012; pp. 410–413.
47. Huang, S.-C.; Chen, B.-H. Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems. *IEEE Trans. Neural Netw. Learning Syst.* **2013**, *24*, 1920–1931.
48. Huang, S.-C.; Do, B.-H. Radial basis function based neural network for motion detection in dynamic scenes. *IEEE Trans. Cyber.* **2014**, *44*, 114–125.
49. Cheng, F.-C.; Chen, B.-H.; Huang, S.-C. A background model re-initialization method based on sudden luminance change detection. *Eng. Appl. Artif. Intell.* **2015**, *38*, 138–146.
50. Cheng, F.-C.; Chen, B.-H.; Huang, S.-C. A hybrid background subtraction method with background and foreground candidates detection. *ACM Trans. Intell. Syst. Technol. (TIST)* **2015**, *7*, 21–31.
51. Chen, B.-H.; Huang, S.-C. Probabilistic neural networks based moving vehicles extraction algorithm for intelligent traffic surveillance systems. *Inf. Sci.* **2015**, *299*, 283–295.
52. Bogorny, V.; Renso, C.; Aquino, A.R.; Lucca Siqueira, F.; Alvares, L.O. CONSTAnT—a conceptual data model for semantic trajectories of moving objects. *Trans. GIS* **2014**, *18*, 66–88.
53. Spaccapietra, S.; Parent, C.; Damiani, M.L.; de Macedo, J.A.; Porto, F.; Vangenot, C. A conceptual view on trajectories. *Data Knowl. Eng.* **2008**, *65*, 126–146.
54. Zheni, D.; Frihida, A.; Ghezala, H.; Claramunt, C. A semantic approach for the modeling of trajectories in space and time. *Adv. Concept. Model. Chall. Perspect.* **2009**, *5833*, 347–356.
55. Mokbel, M.F.; Sarwat, M. Mobility and social networking: A data management perspective. *Proc. VLDB Endow.* **2013**, *6*, 1196–1197.
56. Zheni, D.; Frihida, A.; Claramunt, C.; Ghezala, H.B. A semantic-based data model for the manipulation of trajectories: Application to urban transportation. *Web Wirel. Geogr. Inf. Syst.* **2015**, *9080*, 124–142.
57. Yan, Z.; Chakraborty, D.; Parent, C.; Spaccapietra, S.; Aberer, K. SeMiTri: A framework for semantic annotation of heterogeneous trajectories. In Proceedings of the 14th International Conference on Extending Database Technology, Uppsala, Sweden, 21–24 March 2011; pp. 259–270.
58. Bogorny, V.; Avancini, H.; de Paula, B.C.; Kuplich, C.R.; Alvares, L.O. Weka-STPM: A software architecture and prototype for semantic trajectory data mining and visualization. *Trans. GIS* **2011**, *15*, 227–248.
59. Bogorny, V.; Heuser, C.; Alvares, L. A conceptual data model for trajectory data mining. *Geogr. Inf. Sci.* **2010**, *6292*, 1–15.
60. Damiani, M.L.; Valdes, F.; Issa, H. Moving objects beyond raw and semantic trajectories. In Proceedings Of the 3rd International Workshop on Information Management for Mobile Applications, Riva del Garda, Italy, 26 August 2013; Volume 4, pp. 1–10.

61. Hu, Y.; Janowicz, K.; Carral, D.; Scheider, S.; Kuhn, W.; Berg-Cross, G.; Hitzler, P.; Dean, M.; Kolas, D. A geo-ontology design pattern for semantic trajectories. *Spat. Inf. Theory* **2013**, *8116*, 438–456.
62. Paiva Nogueira, T.; Bezerra Braga, R.; Martin, H. An ontology-based approach to represent trajectory characteristics. In Proceedings of the 2014 Fifth International Conference on Computing for Geospatial Research and Application, Washington, DC, USA, 4–6 August 2014; pp. 102–107.
63. Yan, Z.; Macedo, J.; Parent, C.; Spaccapietra, S. Trajectory ontologies and queries. *Trans. GIS* **2008**, *12*, 75–91.
64. Boulmakoul, A.; Karim, L.; Lbath, A. Moving object trajectories meta-model and spatio-temporal queries. *Int. J. Database Manag. Syst.* **2012**, *4*, 35–54.
65. Kolahdouzan, M.; Shahabi, C. Voronoi-based k nearest neighbor search for spatial network databases. In Proceedings of the Thirtieth International Conference on Very Large Data Bases-Volume 30, Toronto, Canada, 29 August–3 September 2004; pp. 840–851.
66. Egenhofer, M.J.; Franzosa, R.D. Point-set topological spatial relations. *Int. J. Geogr. Inf. Syst.* **1991**, *5*, 161–174.
67. Egenhofer, M.J.; Herring, J. A mathematical framework for the definition of topological relationships. In Proceedings of the Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland, 23–27 July 1990; pp. 803–813.
68. Düntgen, C.; Behr, T.; Guting, R.H. BerlinMOD: A benchmark for moving object databases. *VLDB J.* **2009**, *18*, 1335–1368.
69. Guting, R.H.; Behr, T.; Düntgen, C. Secondo: A platform for moving objects database research and for publishing and integrating research implementations. *IEEE Data Eng. Bull.* **2010**, *33*, 56–63.
70. Guting, R.H.; Almeida, V.; Ansoorge, D.; Behr, T.; Ding, Z.; Höse, T.; Hoffmann, F. Spiekermann, M. SECONDO: An extensible DBMS platform for research prototyping and teaching. In Proceedings of the 21st International Conference on Data Engineering (ICDE'05), Tokyo, Japan, 5–8 April 2005; pp. 1115–1116.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).