Design and Implementation of Spatial-temporal Data Model in Vehicle Monitor System

Jingnong Weng, Weijing Wang, ke Fan, Jian Huang
College of Software, Beihang University
No. 37, Xueyuan Road, Haidian District, Beijing China, 100083.
Telephone: 86-10-82314373
FAX: 86-10-82316078
Email: wengjingnong@hotmail.com

Abstract

In recent years, Global Positioning System (GPS) has been greatly utilized in collecting real-time geographic data for a mobile unit. With the latest developments in the technologies of Geographic Information System (GIS) and wireless communications, the location data collected by a GPS unit could be easily displayed in a real-time fashion. However, for large scale applications such as a taxi fleet, the task to store and query the database with continuous GPS location data collected over time still remains a challenge. In this paper, a spatial-temporal model that is designed specifically for managing extensive amounts of GPS position data is described. The model consists of two major components, i.e., cube cell and trajectory. A cube cell is a three-dimensional space determined by a pre-defined time interval as well as the x and y axles for spatial coordinates. A trajectory, on the other hand, has a time attribute and may be stored as a line object in software applications such as Oracle Spatial. In the model, the position data that form a moving trajectory is stored in each cube cell to record a given object's locations over the specified time interval and a trajectory is defined as the basic visiting unit. The data structure allows extensive GPS data to be efficiently stored and queried. The model is also capable of eliminating redundant data for objects remaining at the same locations, which may further reduce the storage requirement. For evaluation purpose, the model was implemented in a Microsoft Windows environment using the GPS data collected from the taxi monitoring system in Guangzhou, China. Oracle 9i and Oracle Spatial were used in the model to create and query the database. The model was shown to significantly reduce storage requirement and greatly enhance the query speed.

Key words: space-time cube; GPS; spatial-temporal data model; spatial-temporal query; trajectory; cube cell

1. Introduction

Now, more and more vehicles are equipped with Global Positioning System (GPS), especially in taxis, ambulances, fire engines, etc. As a result, tracking the changing positions of continuous moving objects is becoming increasingly feasible and possible. A monitor system is set up to manage its monitored vehicle. GPS in every vehicle will update its position data every several seconds or minutes depending on its demand, so we can easily know where the monitored vehicle is and can also accept a signal from vehicle such as an alarm, accident information, etc. This will be benefit for accident handling and vehicle schedules, and also it will make driving more effective and safety. The historical trajectories data of a vehicle can be used for operational decision support and improving the quality of service of and transportation planning. The scale of the monitor centre depends on its number of monitored vehicles. For example, the newly built taxi monitor centre in Guangzhou, China, now it has 8,000 taxis and accepts GPS position data every several seconds or minutes, differently with varied taxis. If we keep records for 30 days, it will be more than 800 million records in all. Usually, it all historical records need to be kept for three months. The problem is that how we can handle such huge amounts of data, in addition, its storage and its trajectory display and varied queries from these data, etc. In this paper, a spatial-temporal model for managing these data is presented, which are used in the Guangzhou taxi monitor system. First, we will analyze three storage methods for GPS position data: direct position data storage, motion function description method and spatial-temporal model method. After analysis and comparison, we will point out that the spatial-temporal model method is an effective way to solve this problem. In the discussion of model designing, we present an space-time model and introduce two concepts: cube cell and trajectory. The main ideas of space-time cube and software module architecture are discussed. The project is implemented in Microsoft Windows environment with Oracle 9i as the database management system and Oracle Spatial for implementing the model. This model reduces storage volume and improves query efficiency. Finally, model validation and comparison between the space-time cube model and the improved model under different conditions are presented.

2. Three Storage Methods for GPS Position data

The vehicle attributes can be divided into two categories: static attributes and dynamic attributes. So called static attributes refer to some features unrelated to its location, such as vehicle ID, vehicle type, capacity, load, manufacture's date, etc. These data can be managed by an ordinary relational database. As for dynamic attributes, it mainly refers to its real time position information. Because the position of vehicle is dynamic continuous changing over time when moving, it will produce a lot of data need to store and query. We usually want retrieve vehicle's historical trajectory in some time interval or time point.

To satisfy the demand of varied queries to vehicle GPS position data, three storage methods are usually adopted. The methods are discussed as follows:

2.1 Direct Position Data Storage

This method uses relational database to store every position data in turn when vehicle is moving, which includes x, y coordinate, moving direction, speed, time, etc. Its feature is that all GPS positions data will be stored in a database in original form, so its advantage is flexible and can be satisfied varied retrieve demand. However, it will make storage volume very big and query efficient lower. This method can mainly be used in historical trajectory display.

2.2 Motion Functions Description Method

This method represents a vehicle's trajectory by constructing a motion function, which can be used to calculate vehicle's location and its moving direction in some time interval. This greatly reduces the volume of data and can satisfy query demand by using some movement parameters. In addition, it also can help us to forecast vehicle's next motion location. However, this method has its condition restriction, especially in urban areas, it is hard to get a properly motion function description.

2.3 Spatial-Temporal Database Method

This method has been widely used in the field of spatial-temporal GIS. It uses spatial database to store GPS position data and regard historical position data as a spatial object with the time stamp. The main difference with method 2.1 is that this method use spatial database. Space-time cube is presented and used as the spatial-temporal model in the research. Spatial database can manage spatial data effectively. It can also satisfy spatial query by means of spatial index. An effective way needed to be found to manage data and reduce its volume and improve its query performance. The detail is discussed in the sections below.

3. Spatial-Temporal Model Design

We use the space-time cube as the spatial-temporal model. A three-dimensional cube represents one time and two space dimensions, as shown in Figure 1. The cube has three axes: x, y, t and the curve in the cube represents a movement trajectory of a vehicle. Given a certain time value, the corresponding time slice may be obtained, which represents a vehicle trajectory status. If the moving object stops for a long time, much data redundancy will result. Some redundant dots in the trajectory are omitted when it is stored. In addition, a trajectory is used to represent dynamic attributes of moving object and stored as a line object implemented by Oracle Spatial. Trajectory is the minimum visiting unit in the application software system.

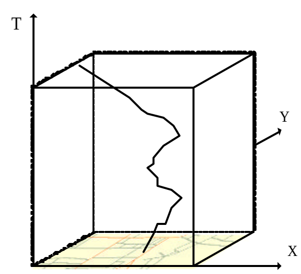

Figure 1: Space-time Cube

The model is shown in Figure 2. The essence of the model is as follows:

1) The space–time cube is divided into many regular parts along the time-axis in certain time interval and different parts in the x-y plan. It will produce many sub space-time cubes, called cube cells. The time interval is represented by the height of a cube cell and the x-y area is measured by its length and width. The volume of the spatial-temporal model is determined by the time interval, the area range of monitoring, and the distribution of roads.

2) In every cube cell, position data will form a moving trajectory.

3) A time interval may be adjusted to improve query performance.

4) The size of a cube cell is different according to its location. For example, roads in an

urban area may have a relatively high density. Therefore, more position data will be needed to describe a moving trajectory. On the other hand, roads outside an urban area may have a relatively lower density, thus less position data are needed to describe a moving trajectory. According to different time and area, different size of a cube cell is presented.

This data model greatly reduces the search scope and makes query operation simple and efficient. A trajectory, with its time attribute, may be stored as a line object in software applications such as Oracle Spatial. In the model, he position data that form a moving trajectory is stored in each cube cell to record a given object's locations over the specified time interval and a trajectory is defined as the basic visiting unit. Because a trajectory stored as a line object, not dot object, information redundancy is reduced and storage efficiency is increased.

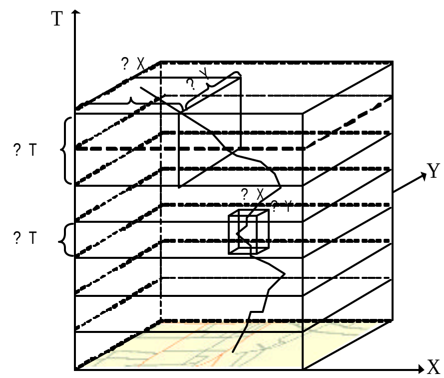

Figure 2 Time-Space Cube Model

4. Data Model Implementation

The model is implemented in Microsoft Windows environment. The database is built using oracle 9i. Spatial information and non-spatial information are stored separately. Oracle object type is used to represent the trajectory and Oracle self-defined type to represent the corresponding position data of the Trajectory. At the same time, a time interval is set to produce a specific trajectory. The time definition of a trajectory is in year-month-day format. Self-defined data types are presented below.

4.1 Vehicle Location Object

This object describes a vehicle's location, speed, direction, and its status.

```
CREATE TYPE “CAR” “VEH_OBJ” AS OBJECT
(“GPS_TIME”   DATE,
“GPS_SPEED” CHAR(3),
“GPS_DIRECTION” CHAR (3),
“GPS_STATE1” CHAR (1))
```

4.2 Vehicle Location Set Object

This object describes multi-location's information.

```
CREATE TYPE “CAR”.“VEH_ARR” AS VARRAY (1048576) OF “CAR”.“VEH_OBJ”
```

4.3 Vehicle GPS Trajectory Object

This object describes the status information of GPS trajectory.

```
CREATETYPE “CAR”.“VEH_TRACK” AS OBJECT
(“GPS_TRACK” “CAR”. “VEH_ARR”)
```

4.4 Historical trajectory Table

GEOLOC store trajectory as a line object and TRACK stores its corresponding status in the following table:

```
CREATE TABLE “CAR”. “GPS_TRACK”
(“MI_PRINX” NUMBER (11) NOT NULL,
“CARNO” VARCHAR2 (20) NOT NULL,
“TVERSION” NUMBER (10) NOT NULL,
“GEOLOC” “MDSYS”. “SDO_GEOMETRY”,
“TRACK”   “CAR”. “VEH_TRACK”,
CONSTRAINT “GPS_TRACK_INX”
PRIMARY KEY(“CARNO”, “INDATE”) )
```

4.5 Spatial Index

One of the features of the spatial database is its support to spatial query. Oracle Spatial supports two kinds of spatial indexing methods: R-tree and Quartic-tree. The R-tree, one of the most popular access methods for rectangles, is based in the heuristic optimization of the area of the enclosing rectangle in each inner node. The tree needs adjustment when a new node is inserted. The Quartic-tree is a grid based indexing mechanism and is easy to implement. Insertion of new data will not affect its performance. As GPS position data updates frequently in the monitor system, a quartic-tree index to GEOLOC is used in the model to reduce burden to system.

5. Model Validation

The spatial query module architecture of the vehicle monitor system is shown in Figure 3. The software module may be divided into four parts: spatial database, Spatial-Temporal Data Engine (STDE), Data Server, and User Interface (UI). STDE realizes uniform visit to spatial data. We validate the module and its performance in practical operation.

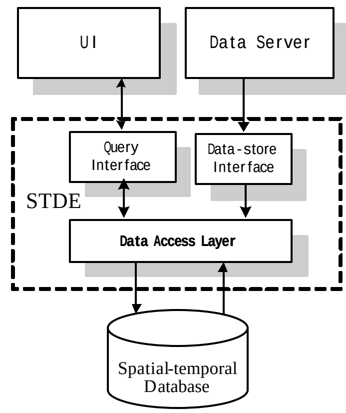

Figure 3: Spatial query module Architecture

The application server has a P4 2.66GHz CPU, one Gigabits of memory, and running database Oracle 9i version 9.2. The number of taxis is 4,131. Tables 1 through 3 present volume comparison, historical query comparison in different time, and historical query comparison between Space-time Model and Direct Position Date Storage method in different area, respectively. Table 1 gives record and volume reduced 90% and 50% respectively compared between space-time model and improved model. Table 2 provides historical query comparison in 2, 8, 12, 16, 20, 24 hours intervals, it appears the performance improved obviously. Historical Query Comparison in 0.2*0.2, 0.5*0.5, 0.8*0.8, 1.0*1.0, 1.2*1.2, 1.5*1.5 km^2 is given in Table 3.

Table 1 Volume Comparison

	Direct Position Date Storage	Space-time Model	Performance improved
Records	6,347,763	552,400	Reduced 90%
Volume	582.5M	293.6M	Reduced 50%

Table 2 Historical Query Comparison in different time

Time(hours)	Direct Position Date Storage(seconds)	Space-time Model (Seconds)
4	0.219	0.047
8	0.328	0.049
12	0.407	0.051
16	0.457	0.054
20	0.547	0.058
24	0.786	0.062

Table 3 Historical Query Comparison in different area

Area(km^2)	Direct Position Date Storage (minutes)	Space-time Model(seconds)
0.2*0.2	More than 5 minutes	10.72
0.5*0.5		11.61
0.8*0.8		17.921
1.0*1.0		18.001
1.2*1.2		14.687
1.5*1.5		23.609

6. Conclusions

Currently, spatial-temporal database applications are being developed in an ad-hoc fashion. We presented a space-time cube model and introduced two concepts, cube cell and trajectory, for monitoring the GPS equipped taxis. This model reduces storage volume and improves query efficiency. It needs to be much more examined in practice with the number of increased taxies and so on.

7. Acknowledgements

We would like to thank the Guangzhou Transportation Committee for their financial support from and the Zhitong Science and Technical Limited Company for their cooperation. Dr. Zhongwei Huang and Zhuxing Wei provided many suggestions during the course of this research. Thanks also go to Dr. Fang Zhao and Dr. Mintang Li, for their comments on this paper.

References

Beckmann, N., Kriegel, H., Schneider, R., Seeger,B. The R* -tree: an Efficient and Robust Access Method for Points and Rectangles [A] ACM SIGMOD, 1990.

Bonan Li, Guoray Cai, A General Object-Oriented Spatial Temporal Data Model, Symposium on Geospatial Theory, Processing and Applications, Ottawa 2002

Dirter Pfoser, Christian S. Jenson, Trajectory Indexing Using Movement Constrains, GeoInformatica 9:2,93-115,2005

Hoda Mokhtar, Jianwen Su, Oscar Ibarra, On Moving Object Queries. Symposium on Principles of Database Systems , Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems ,Madison, Wisconsin ,Pages: 188 - 198 , 2002

O.Wolfson, B.XU, S.Chamberlain, L.Jiang, Moving Object Database: Issue and Solutions[A] Proc. 10th Int. Conf. on Science and Statistical Database Management[C],Capri, Italy, July 1998 111-122.

Peuquet, D.J. & Duan, N. (1995). An Event-based Spatio-Temporal Data Model (ESTDM) for temporal analysis of geographical data. Int. J. Geographical Information Systems, vol. 9, pp.7 – 24.

Theodoridis Y., et al: Specifcaitons for Efficient Index in Spatiotemporal Databases[A]. Proc. 10th Int. Conf. On Scientific and Statistical Database Management[C], Capri, It aly, 1998.

Victor Teixeira de almeida and Ralf Hartmut Guting, Indexing the Trajectories of Moving Objects in Networks, GeoInformatica 9:1,33-60,2005-5-29

Xiaofeng Meng, Zhiming Ding , DSTTMOD: A Future Trajectory Based Moving Objects Database. DEXA 2003: 444-453