Location Prediction and Queries for Tracking Moving Objects

Ouri Wolfson Bo Xu
Department of Electrical Engineering and Computer Science
University of Illinois at Chicago
{wolfson,bxu}@eecs.uic.edu

Sam Chamberlain
Army Research Laboratories
wildman@arl.mil

1 Background

Consider a database that represents information about moving objects and their location. For example, for a database representing the location of taxi-cabs a typical query may be: retrieve the free cabs that are currently within 1 mile of 33 N. Michigan Ave., Chicago (to pick-up a customer); or for a trucking company database a typical query may be: retrieve the trucks that are currently within 1 mile of truck ABT312 (which needs assistance); or for a database representing the current location of objects in a battlefield a typical query may be: retrieve the friendly helicopters that are in a given region, or, retrieve the friendly helicopters that are expected to enter the region within the next 10 minutes. The queries may originate from the moving objects, or from stationary users. We will refer to applications with the above characteristics as moving-objects-database (MOD) applications, and to queries as the ones mentioned above as MOD queries.

In the military MOD applications arise in the context of the digital battlefield (see [1]), and in the civilian industry they arise in transportation systems. For example, Omnitracs developed by Qualcomm (see http://www.qualcomm.com/ProdTech/Omni/prodtech/omnisys.html) is a commercial system used by the transportation industry, which enables MOD functionality. It provides location management by connecting vehicles (e.g. trucks), via satellites, to company databases. The vehicles are equipped with a Global Positioning System (GPS), and they automatically and periodically report their location.

Tracking using a moving-objects-database also enables futuristic applications such as augmented reality. In such applications various images, charts, and other voluminous data (that cannot be stored in a portable/wearable computer for a large geographic area) is delivered "just-in-time" to the mobile computer; the delivered information pertains only to the geographic location in the immediate vicinity of the mobile computer, which continuously changes as the mobile computer moves. Finally, in electronic commerce tracking enables to deliver location-dependent dynamic travel information (e.g. local traffic conditions, local motel availability/price, local sales of interest) to a mobile subscriber.

2 The Demonstration

Our Mobitrack prototype is intended to serve as a platform, or a toolkit for developing MOD type of applications. The system is the third in a three-layer architecture. The first layer is an Object Relational DBMS. The database stores the information about each moving object, including its plan of motion. The second layer is a GIS that adds capabilities and user interface primitives for storing, querying, and manipulating geographic information. The third layer, Mobitrack, adds temporal capabilities, capabilities of managing the uncertainty that is inherent in future motion plans, capabilities for location prediction, and a simulation testbed. Currently, Mobitrack runs on both Unix and MS/Windows. On both platforms Mobitrack uses the Arc-View GIS. It uses the Informix DBMS on Unix, and DBAccess on MS/Windows.

We will demonstrate the following features of Mobitrack.

2.1 Location Modeling

The database may have various levels of information about the location of a moving object. It may know the current exact point-location, or it may know a general area in which the object is located but not the exact location, or it may know an approximate motion plan (e.g. traveling north on I95, at 60 miles per hour), or it may know the complete motion plan. The motion plan of a moving object is a sequence of way time points, (p_1,t_1), (p_2,t_2),... (p_n,t_n), indicating that the unit will be at geographic point p_1 at time t_1, at geographic point p_2 (closer to the destination than p_1) at time t_2 (later than t_1), etc. Mobitrack supports all these levels of location information. In order to do so efficiently, it employs the concept of a dynamic attribute, i.e. an attribute whose value changes continuously as time progresses, without being explicitly updated. So, the location of a moving object is given by its dynamic attribute, which is is instantiated by the motion plan of the object.

In Mobitrack a motion plan is specified interactively by the user on a GIS on a map. Mobitrack is currently using maps from GDT Corp. ; the map contains the length of each city block, the coordinates of its endpoints, and the average traffic speed along each city block. The speed information in the GDT maps is static, but we update it using real-time traffic information collected periodically from a web site (http://www.ai.eecs.uic.edu/GCM/CongestionMap.html). Based on this information the current location of the object is computed at any point in time. This motion plan may be automatically updated by GPS information transmitted from the moving object.

In Mobitrack a moving object equipped with a GPS can use one of several policies to update the locations database. One of the policies uses a cost based approach approach to quantify the tradeoff between uncertainty and communication cost. Specifically, it updates the database whenever the deviation from the database location exceeds a prespecified bound b given in terms of distance or time [2]. The update includes a revised plan and possibly a new bound on the deviation. The bound b is computed using on a cost based approach that takes into consideration the cost of update messages (in terms of wireless bandwidth consumption and processing power) and the cost of the deviation.

2.2 Spatio-temporal Capabilities

Maintaining motion plan information enables the system to answer queries pertaining to the current, future or past locations of the moving object, for example: Q1 = Retrieve the moving objects that are expected to be within a certain distance or travel time from a region R sometime during a given time interval I. (I may be a time interval that lies entirely in the future, i.e. after the time when Q1 is entered).

The above query pertains to the stored trajectory of each object. In many situations, particularly emergencies, a user may need to retrieve the objects that *can* get to a particular location within a certain time interval, even if they have to change their trajectory. For example, a police-vehicle dispatcher may be interested to retrieve the vehicles that can get to the scene of a crime within 10 minutes, even if they have to change their existing trajectories. Thus, Mobitrack also supports queries of the type: Q2 = Retrieve the moving objects that can arrive within a certain distance or travel time (e.g. 0) from a region R (which can be a point) sometime during a given time interval I.

We will demonstrate the spatial and temporal primitives of the query language and its answer-display screen. The primitives are given in graphical format, and they can be combined with textual SQL in a natural and intuitive way. For example, in the query Q1 above the region R may be drawn with a mouse on a real GIS map, and the time interval I may be specified on a graphical time line. Clearly, the query can also include regular literals, e.g., WEIGHT > 5000. Information about the moving objects that satisfy the query is displayed in textual form, and the location of each such moving object is displayed as a point on the map. The spatio-temporal functions constitute a data-blade in an ORDBMS.

2.3 Simulation Testbed

We will demonstrate a simulation testbed in which the performance of a MOD application can be evaluated. The input to the simulation system is a set of moving objects, their motion plans, their speed variations over time, the costs of deviation, the cost of uncertainty, the cost of communication, the wireless bandwidth distribution over the geographic area, and the location update policy used by each moving object. The objective is to determine the performance of MOD queries, as well as to answer questions such as: How many objects can be supported for an average imprecision that is bounded by x, and a wireless bandwidth allocated to location updates that is bounded by y? Or, given n moving objects and a bound of 10% on the imprecision, what amount of bandwidth that is necessary for location updates?

2.4 Location Prediction

In some cases, the motion plan giving the future locations of an object is unavailable. For example, a consumer may want to be reminded by her car computer to buy bread at a local supermarket ; she will probably want to be reminded ahead of time (say 10 minutes in advance) when the supermarket is on her future trajectory. Although the car computer does not know the future trajectory, it may be able to infer it based on the past trajectories traveled by the consumer, and stored in the computer. We will demonstrate the capability of Mobitrack to predict the future trajectory of a moving object.

References

[1] Chamberlain S., *Model-Based Battle Command: A Paradigm Whose Time Has Come*, 1995 Symp. on C2 Research and Technology, June 1995.

[2] Wolfson O., Sistla P., Chamberlain S., Yesha Y., *Updating and Querying Databases that Track Mobile Units*, invited paper, in a special issue of the Distributed and Parallel Databases Journal on Mobile Data Management and Applications, 7(3), 1999, Kluwer Academic Publishers, pp. 257-288.