Category: Concepts, Trends, and Challenges

Indexing Mobile Objects

Panayiotis Bozanis
University of Thessaly, Greece

INTRODUCTION

The past few years have shown a significant increase in the volume and diversity of data stored in database management systems. Among these are spatiotemporal data, one of the faster developing categories of data. This phenomenon can be attributed to the flurry of application development concerning continuously evolving spatial objects in several areas: mobile communication systems, military equipment in battlefields, air traffic, truck fleets, and others.

In standard database applications, data remain unchanged unless an update is explicitly stated. Applying this mode of operation to constantly moving objects would require frequent updates to be performed; otherwise, the database would be inaccurate and unreliable. In order to capture continuous movement and to avoid unnecessary updates, object positions are stored as time-dependent functions, requiring updates only when a function parameter changes. The moving objects are considered responsible for updating the database about alterations in their movement. In the following article is a short review on basic indexing schemes for accommodating moving objects in database systems so that complex queries about their location in the past, present, and future can be served.

BACKGROUND

The indexes developed for moving objects can be classified according to criteria such as the kind of supported queries, the type of the accommodated objects, and their applicability to various-use cases. This article focuses on the last criterion. Thus, an index is characterized as *practical* when it has been implemented and experimentally investigated. Otherwise, it is *theoretical*, and its main utility is indicating inherent complexities. The following addresses practical indexes.

Database Representation

Most applications usually assume objects of fixed shape and volume and are interested only in their location as a geometric point at various instances in time. The optimal way to register such spatiotemporal information depends on its intended use, which is either the postprocessing of the recorded data, or the exploration of current and future locations.

In the first case, object trajectories must be indexed. The trajectories are usually calculated by the linear interpolation of sampled locations, both to accommodate storage limitations but also—more important—because of the very nature of the data generators (e.g., GPS equipment generates discrete location data). As a result, each trajectory consists of a sequence of connected line segments in the three-dimensional space, for which semantically related line segments must be indexed. This indexing operation requires cautious extensions to spatial indexes (Gaede & Günther, 1998).

In the second case, the object position is considered as a time function $x(t)$. Usually $x(t)$ is modeled as a linear *parametric* function of time:

$$x(t) = x(t_{ref}) + v(t - t_{ref}),$$

specified by two parameters: the *reference position* $x(t_{ref})$ at a specific time t_{ref}, and the *velocity vector* **v**, which defines a space *dual* to the *time-location* space.

Query Types

Location-Based Queries

Location-based queries are further categorized into *range* and *nearest-neighbor (NN)* or *proximity* queries. A range query is characterized as (i) *time-slice* or *snapshot* when, given a hyper-rectangle r located at time t, it asks for all moving objects contained in r at that time; (ii) *window* when it requests reporting all objects crossing a hyper-rectangle r during a time interval $\mathbf{t}=[t_s,t_e]$; and (iii) *moving* when it specifies a trapezoid τ by connecting a hyper-rectangle r_1 at time t_s and a hyper-rectangle r_2 at time t_e and inquires all objects crossing τ.

On the other hand, nearest-neighbor queries request the nearest moving object to a given location at time t or during a time interval $\mathbf{t} = [t_s,t_e]$. The generalization this type of queries demands the *k nearest-neighbors* (kNN). Sometimes, *reversed nearest neighbor* (RNN) searching is required for reporting all objects having a given object as their nearest neighbor.

Trajectory-Based Queries

Trajectory-based queries concern (i) information about the semantics of the movement (e.g., objects entering, leaving, crossing, and bypassing a region during a given time instance or interval). For example, "*find all mobile phones entering a particular cell between 2 p.m. and 5 p.m. today*"; and (ii) Derived or navigational information, such as travelled distance, velocity, and so forth; for example, "*report all objects whose travelled distance between 2 p.m. and 5 p.m. 3 days ago was smaller than 40 mph.*"

Continuous Queries

Continuous queries considers that the range or point query is also moving: "*find all my nearest restaurants as I drive towards the current direction for the next 5 minutes*." Despite its conceptual simplicity, these queries are inherently complex, because they are equivalent to constantly posing location-based queries.

Soundness-Enriched Queries

Soundness-enriched queries demands answers enriched with validity (temporal or spatial) information; namely, the future time t at which the result expires, and the change that will occur at time t. For instance when one asks "*report the pharmacy nearest to my position as I am now moving,*" the database will return the nearest pharmacy ID i, the future time t that i ceases to be the closest pharmacy and a new pharmacy $i2$ that would be the next nearest one at time t. Alternatively, the database returns a validity region r around the query position, within which the answer remains valid. Returning to the example, i will be accompanied by the region for which it is the nearest.

This class aims at reducing the number of generated queries: Because the query and the server responses are delivered via a wireless network, the extra information could spare network bandwidth, because the user will release new inquiries only when it is absolutely necessary.

MOBILE OBJECT INDEXING

Practical indexes can be classified into two subcategories according to the time dimension: The structures capable of answering queries about the present and the future belong to the first group, and the second one comprises indexes serving inquires about the past. The presentation of the indexes will focus on (i) the type of the queries they support; and (ii) the main performance and implementation characteristics they exhibit. Table 1 summarizes the presentation.

Querying About the Present and Future

The members of this grouping are further classified based on three types of supporting inquiries:

Structures Supporting Range Queries

Kollios, Gunopoulos and Tsotras (1999b) suggested solutions for one- and two-dimensional range querying that

Table 1. Categorization of presented indexes

Querying about the Present and Future	
Query	**Index**
Range	Kollios et al. (1999b)
	Šaltenis et al. (2000)
	Benetis et al. (2002)
	Tao et al. (2003)
Simple Proximity	Kollios et al. (1999a)
	Aggarwal and Agrawal (2003)
Continuous Proximity	Song and Roussopoulos (2001)
	Tao et al. (2002)
	Iwerks et al. (2003)
Time-Parameterized	Tao and Papadias (2002)
Validity	Zheng and Lee (2001)
	Zhang et al. (2003)
Querying the Past	
Query	**Index**
Reporting	Pfoser et al. (2000)
	Pfoser and Jensen (2001)
	Porkaew et al. (2001)
	Hadjieleftheriou et al. (2002)
	Lazaridis et al. (2002)
Aggregating	Papadias et al. (2002)
	Tao et al. (2004)
	Sun et al. (2004)