Spatio–Temporal Indexing Techniques

Michael Vassilakopoulos  
TEI of Thessaloniki, Greece

Antonio Corral  
University of Almeria, Spain

INTRODUCTION

Time and space are ubiquitous aspects of reality. Temporal and spatial information appear together in many everyday activities, and many information systems of modern life should be able to handle such information. For example, information systems for traffic control, fleet management, environmental management, military applications, local and public administration, and academic institutions need to manage information with spatial characteristics that change over time, or in other words, spatio-temporal information. The need for spatio-temporal applications has been strengthened by recent developments in mobile telephony technology, mobile computing, positioning technology, and the evolution of the World Wide Web.

Research and technology that aim at the development of Database Management Systems (DBMSs) that can handle spatial, temporal, and spatio-temporal information have been developed over the last few decades. The embedding of spatio-temporal capabilities in DBMSs and GISs is a hot research area that will continue to attract researchers and the informatics industry in the years to come.

In spatio-temporal applications, many sorts of spatio-temporal information appear. For example, an area covered by an evolving storm, the changing population of the suburbs of a city, the changing coast lines caused by ebb and tide. However, one sort of spatio-temporal information is quite common (and in some respects easier to study) and has attracted the most research efforts: moving objects or points, for example, a moving vehicle, an aircraft, a wandering animal.

One key issue for the development of an efficient spatio-temporal DBMS (STDBMS) is the use of spatio-temporal access methods at the physical level of the DBMS. The efficient storage, retrieval, and querying of spatio-temporal information demands the use of specialized indexing techniques that minimize the cost during management of such information.

In this article, we report on the research efforts that have addressed the indexing of moving points and other spatio-temporal information. Moreover, we discuss the possible research trends within this area of rising importance.

BACKGROUND

The term spatial data refers to multidimensional data, like points, line segments, regions, polygons, volumes, or other kinds of geometric entities, while the term temporal data refers to data varying in the course of time. Since in database applications the amount of data that should be maintained is too large for main memory, external memory (hard disk) is considered as a storage means. Specialized access methods are used to index disk pages and, in most cases, have the form of a tree. Numerous indexing techniques have been proposed for the maintenance of spatial and temporal data. Two good sources of related information are the survey by Gaede and Günther (1998) and the survey by Saltzberg and Tsotras (1999) for spatial and temporal access methods, respectively.

During last years, several researchers have focused on spatio-temporal data (spatial data that vary in the course of time) and the related indexing methods for answering spatio-temporal queries. A spatio-temporal query is a query that retrieves data according to a set of spatial and temporal relationships. For example, “find the vehicles that will be in a distance of less than 5km from a specified point within the next 5 minutes”. A number of recent short reviews that summarize such indexing techniques (especially, indexing of moving points) have already appeared in the literature. There are several ways for categorizing (several viewpoints, or classifications of) spatio-temporal access methods. In the rest of this section, we report on the approach followed by each of these reviews and on the material that the interested reader would find there.

In the book Spatiotemporal Databases: The ChoroChronos Approach that was authored within the ChoroChronos project and edited by Sellis et al. (2003), Chapter 6 is entitled “Access Methods and Query Processing Techniques” and reviews spatio-temporal access methods that have appeared up to 2001. The main classification followed in this chapter is between methods
Spatio-Temporal Indexing Techniques

belonging in the R-tree family and methods belonging in the Quadtree family. The principle guiding the hierarchical decomposition of data distinguishes between these two indexing approaches. The two fundamental principles, or hierarchies are:

- **the data space hierarchy**: a region containing data is split (when, for example, a maximum capacity is exceeded) to sub-regions in a way that depends on these data (for example, each of two sub-regions contains half of the data), and
- **the embedding space hierarchy**: a region containing data is split (when a certain criterion holds) to sub-regions in a predefined way (for example, a square region is always split in four quadrant sub-regions).

R-trees are data-driven, while Quadtrees are space-driven access methods.

The June 2002 (Vol. 25, No. 2) issue of the IEEE Data Engineering Bulletin (http://sites.computer.org/debull/A02june/issue1.htm) is devoted to “Indexing of Moving Objects” and is an excellent source of updated information. Pfoser (2002) reviews techniques that index the trajectories of moving objects that follow unconstrained movement (e.g., vessels at sea), constrained movement (e.g., pedestrians), and movement in transportation networks (e.g., trains or cars). These techniques allow the answering of queries concerning the past of the objects.

On the other hand, Papadopoulos et al. (2002) review techniques that index mobile objects and allow answering of queries about their future positions. The basis of these techniques is the duality transform, that is, mapping of data from one data space to another data space, where answering of queries is easier, or more efficient (mapping of a line segment representing a trajectory to a point in space of equal dimensionality).

Agarwal and Procopiuc (2002) classify indexing techniques according to the consideration of time as another dimension (called time oblivious approach that can be used for answering queries about the past), the use of kinetic data structures (that can be used for answering present queries or even queries that arrive in chronological order), and the combined use of the two techniques (that can be used for efficiently answering queries about the near past or future).

Jensen and Saltenis (2002) discuss a number of techniques that may lead to improved update performance of moving-objects indices. Their paper is a good source of possible future research trends.

Chon, Agrawal, and El Abbadi (2002) report on managing object trajectories by following a partitioning approach that is best suited to answering time-dependent shortest path queries (where the cost of edges varies with time).

Mokbel, Ghanem, and Aref (2003) review numerous spatio-temporal access methods classifying them according to their ability to index only the past, only the present, and the present together with the future status of data. A very descriptive figure that displays the evolution of spatio-temporal access methods with the underlying spatial and temporal structures is included.

Tzouramanis, Vassilakopoulos, and Manolopoulos (2004) review and compare four temporal extensions of the Linear Region Quadtree and can store and manipulate consecutive raster images and answer spatio-temporal queries referring to the past.

**MAIN TRUST OF THE ARTICLE**

In this section, we briefly review the most fundamental spatio-temporal indexing techniques.

**Quadtree-Based Methods**

The Quadtree is a four-way tree where each node corresponds to a subquadrant of the quadrant of each father node (the root corresponds to the whole space). These trees subdivide space in a hierarchical and regular fashion. They are mainly designed for main memory; however, several alternatives for secondary memory have been proposed. The most widely used Quadtree is the Region Quadtree that stores regional data in the form of raster images. More details appear in Gaede and Günther (1998).

Tayeb, Ulusoy, and Wolfson (1998) used the PMR-quadtree for indexing future trajectories of moving objects. The PMR tree is a tree based on quadtrees, capable of indexing line segments. The internal part of the tree consists of an ordinary region quadtree residing in main memory. The leaf nodes of this quadtree point to the bucket pages that hold the actual line segments and reside on disk. Each line segment is stored in every bucket whose quadrant (region) it crosses. This causes the problem that data replication is introduced (every trajectory, a semi-infinite line, is stored in all the quadrants that it crosses).

Raptopoulou, Vassilakopoulos, and Manolopoulos (2004) used a new Quadtree-based structure, called XBR tree, for indexing past trajectories of moving objects. XBR trees (External Balanced Regular trees) are secondary memory balanced structures that subdivide space in a quadtree manner into disjoint regions. In their paper, XBR trees are shown to excel over PMR trees when used for indexing past trajectories.

Tzouramanis et al. (2003, 2004) present and compare four different extensions of the Linear Region Quadtree (the Time-Split Linear Quadtree, the Multiversion Linear Quadtree, the Multiversion Access Structure for Evolv-
Related Content

Roles of Resource and Data Contention on the Performance of Replicated Distributed Database Systems
[www.igi-global.com/article/roles-resource-data-contention-performance/51115?camid=4v1a]

Cooperation with Geographic Databases
[www.igi-global.com/chapter/cooperation-geographic-databases/26975?camid=4v1a]

An Efficient Index Structure for Spatial Databases
[www.igi-global.com/article/efficient-index-structure-spatial-databases/51164?camid=4v1a]

RORIB: An Economic and Efficient Solution for Real-Time Online Remote Information Backup
Scott J. Lloyd, Joan Peckham, Qing Yang and Jian Li (2003). *Journal of Database Management* (pp. 56-73).
[www.igi-global.com/article/rorib-economic-efficient-solution-real/3299?camid=4v1a]