IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB10783

Chapter VI

# Approximate Computation of Distance-Based Queries[1]

Antonio Corral, University of Almeria, Spain

Michael Vassilakopoulos
Technological Educational Institute of Thessaloniki, Greece

## Abstract

*In spatial database applications the similarity or dissimilarity of complex objects is examined by performing distance-based queries (DBQs) on data of high dimensionality (a generalization of spatial data). The R-tree and its variations are commonly cited as multidimensional access methods that can be used for answering such queries. Although the related algorithms work well for low-dimensional data spaces, their performance degrades as the number of dimensions increases* (dimensionality curse). *To obtain acceptable response time in high-dimensional data spaces, algorithms that obtain approximate solutions can be used. In this chapter, we review the most important approximation techniques for reporting sufficiently good results quickly. We focus on the design choices of efficient approximate DBQ algorithms that minimize the response time and the number of I/O operations over tree-like structures. The chapter concludes with possible future research trends in the approximate computation of DBQs.*

# Introduction

The term "Spatial Database" refers to a database that stores data for phenomena on, above or below the earth's surface, or in general, various kinds of multidimensional entities of modern life, for example, the layout of a Very Large Scale Integration (VLSI) design. In a computer system, the spatial data are represented by points, line segments, regions, polygons, volumes and other kinds of geometric entities and are usually referred to as spatial objects. For example, a spatial database may contain polygons that represent building footprints from a satellite image, or points that represent the positions of cities, or line segments that represent roads.

The key characteristic that makes a spatial database a powerful tool is its ability to manipulate spatial data, rather than simply storing and representing them. One of the main targets of such a database is answering queries related to the spatial properties of data. Some typical spatial queries are the following.

- A "Point Location Query" seeks for the objects that fall on a given point (for example, the country where a specific city belongs).
- A "Range Query" seeks for the objects that are contained within a given region, usually expressed as a rectangle or a sphere (for example, the taxis that are inside the historic center of a city).
- A "Join Query" may take many forms. It involves two or more spatial datasets and discovers pairs (or tuples in the case of more than two datasets) of objects that satisfy a given spatial predicate, such as *overlap* (for example, the pairs of boats and stormy areas for boats sailing across a storm).
- Finally, a "Nearest Neighbor Query" seeks for the objects residing more closely to a given object. In its simplest form, it discovers one such object (the Nearest Neighbor). Its generalization discovers $K$ such objects ($K$ Nearest Neighbors), for a given $K$ (for example, the $K$ ambulances closer to a spot where an accident with $K$ injured persons occurred).

As extensions of the above "fundamental" spatial queries, new query forms have emerged. For example, to examine the similarity or dissimilarity of large sets of complex objects, high-dimensional feature vectors are extracted from them and organized in multidimensional indexes. The most important property of this feature transformation is that the feature vectors correspond to points in the multidimensional Euclidean space (a kind of generalized spatial data). Then, DBQs are applied on the multidimensional points.

The multidimensional access methods belonging to the R-tree family, the R*-tree (Beckmann, Kriegel, Schneider, & Seeger, 1990) and particularly the X-tree

## Related Content