Chapter XXX
Query Processing in Spatial Databases

Antonio Corral
University of Almeria, Spain

Michael Vassilakopoulos
University of Central Greece, Greece

INTRODUCTION

Spatial data management has been an active area of intensive research for more than two decades. In order to support spatial objects in a database system several important issues must be taken into account such as: spatial data models, indexing mechanisms and efficient query processing. A spatial database system (SDBS) is a database system that offers spatial data types in its data model and query language and supports spatial data types in its implementation, providing at least spatial indexing and efficient spatial query processing (Güting, 1994).

The main reason that has caused the active study of spatial database management systems (SDBMS) comes from the needs of the existing applications such as geographical information systems (GIS), computer-aided design (CAD), very large scale integration design (VLSI), multimedia information systems (MIS), data warehousing, multi-criteria decision making, location-based services, etc.

Some of the most important companies in the commercial database industry (Oracle, Informix, Autodesk, etc.) have products specifically designed to manage spatial data. Moreover, re-
search prototypes as Postgres and Paradise offer
the possibility to handle spatial data. The main
functionality provided by these products includes
a set of spatial data types such as the point, line,
polygon and region; and a set of spatial operations,
including intersection, enclosure and distance.
The performance enhancement provided by these
operations includes spatial access methods and
query algorithms over such indexes (e.g. spatial
range queries, nearest neighbor search, spatial
joins, etc). We must also cite the Open Geo-
graphic Information Systems (OGIS) consortium
(http://www.opengis.org/), which has developed a
standard set of spatial data types and operations
and SQL3/SQL99, which is an object-relational
query language that provides the use of spatial
types and operations.

In a spatial database system, the queries are
usually expressed in a high-level declarative lan-
guage such as SQL; therefore specialized database
software has to map the query in a sequence of
spatial operations supported by spatial access
methods (Shekhar & Chawla, 2003). Spatial query
processing refers to the sequence of steps that a
SDBMS will initiate to execute a given spatial
query. The main target of query processing in the
database field is to process the query accurately
and quickly (consuming the minimum amount
of time and resources on the computer), by using
both efficient representations and efficient search
algorithms. Query processing in a spatial environ-
ment focuses on the design of efficient algorithms
for spatial operators (e.g. selection operations,
spatial joins, distance-based queries, etc.). These
algorithms are both CPU and I/O intensive, despite
common assumptions of traditional databases
that the I/O cost will dominate CPU cost (except
expensive distance-based queries) and therefore
an efficient algorithm is one that minimizes the
number of disk accesses. In this article we focus
on spatial query processing and not spatio-tem-
poral query processing, where the queries refer
to both spatial and temporal characteristics of the
data (Manolopoulos et al., 2005).

**BACKGROUND IN SPATIAL QUERIES AND PROCESSING**

From the query processing point of view, the
following three properties characterize the dif-
ferences between spatial and relational databases
(Brinkhoff et al., 1993): (1) unlike relational
databases, spatial databases do not have a fixed
set of operators that serve as building blocks for
query evaluation; (2) spatial databases deal with
extremely large volumes of complex objects,
which have spatial extensions and cannot be sorted
in one dimension; (3) computationally expensive
algorithms are required to test the spatial opera-
tors, and the assumption that I/O costs dominate
CPU costs is no longer valid.

We generally assume that the given spatial
objects are embedded in d-dimensional Euclidean
space ($E^d$). An object $obj$ in a spatial database is
usually defined by several non-spatial attributes
and one attribute of some spatial data type (point,
line, polygon, region, etc.). This spatial attribute
describes the geometry of the object $obj.G \subseteq E^d$,
i.e. the location, shape, orientation and size of the
object. The most representative spatial operations,
which are the basis for the query processing in
spatial databases (Gaede & Günther, 1998; Shek-
har & Chawla, 2003), are: (1) update operations;
(2) selection operations (point and range queries);
(3) spatial join; and (4) spatial aggregate queries,
and (5) feature-based spatial queries.

- **Update operations**: Standard database op-
erations such as modify, create, etc.
- **Point Query (PQ)**: Given a query point $p \in E^d$, find all spatial objects $O$ that contain
  it.
- **Range Query (RQ)**: Given a query polygon $P$, find all spatial objects $O$ that intersect $P$.
  When the query polygon is a rectangle, this
  is called a window query.
- **Spatial Join Query (SJQ)**: Given two col-
  lections $R$ and $S$ of spatial objects and a
  spatial predicate $\theta$, find all pairs of objects...
Related Content

Enterprise Application System Reengineering: A Business Component Approach
www.igi-global.com/article/enterprise-application-system-reengineering/3358?camid=4v1a

Internet Map Services and Weather Data
www.igi-global.com/chapter/internet-map-services-weather-data/20714?camid=4v1a

CORBA-Based Run-Time Architectures for Workflow Management Systems
www.igi-global.com/article/corba-based-run-time-architectures/51159?camid=4v1a

Federated Process Framework in a Virtual Enterprise Using an Object-Oriented Database and Extensible Markup Language
www.igi-global.com/article/federated-process-framework-virtual-enterprise/3289?camid=4v1a