Introduction to Spatial Database Systems

by Cyrus Shahabi

Spatial Databases: A Tour, Shashi Shekhar and Sanjay Chawla

* Hart Hartmut Guting’s VLDB Journal v3, n4, October 1994
Value of SDBMS

• Traditional (non-spatial) database management systems provide:
  – Persistence across failures
  – Allows concurrent access to data
  – Scalability to search queries on very large datasets which do not fit inside main memories of computers
  – Efficient for non-spatial queries, but not for spatial queries

• Non-spatial queries:
  – List the names of all bookstore with more than ten thousand titles.
  – List the names of ten customers, in terms of sales, in the year 2001

• Spatial Queries:
  – List the names of all bookstores with ten miles of Minneapolis
  – List all customers who live in Tennessee and its adjoining states
Value of SDBMS – Spatial Data Examples

• Examples of non-spatial data
  – Names, phone numbers, email addresses of people

• Examples of Spatial data
  – Census Data
  – NASA satellites imagery - terabytes of data per day
  – Weather and Climate Data
  – Rivers, Farms, ecological impact
  – Medical Imaging
Value of SDBMS – Users, Application Domains

• Many important application domains have spatial data and queries. Some Examples follow:
  – **Army Field Commander**: Has there been any significant enemy troop movement since last night?
  – **Insurance Risk Manager**: Which homes are most likely to be affected in the next great flood on the Mississippi?
  – **Medical Doctor**: Based on this patient's MRI, have we treated somebody with a similar condition?
  – **Molecular Biologist**: Is the topology of the amino acid biosynthesis gene in the genome found in any other sequence feature map in the database?
  – **Astronomer**: Find all blue galaxies within 2 arcmin of quasars.
Applications+

- Various fields/applications require management of geometric, geographic or spatial data:
  - A geographic space: surface of the earth
  - Man-made space: layout of VLSI design
  - Model of rat brain
What is a SDBMS?

• A SDBMS is a software module that
  – can work with an underlying DBMS
  – supports spatial data models, spatial abstract data types (ADTs) and a query language from which these ADTs are callable
  – supports spatial indexing, efficient algorithms for processing spatial operations, and domain specific rules for query optimization

• Example: Oracle Spatial Extension
  – can work with Oracle 10g DBMS
  – Has spatial data types (e.g. polygon), operations (e.g. overlap) callable from SQL3 query language
  – Has spatial indices, e.g. R-trees
What is an SDBMS?*

• Common challenge: dealing with large collections of relatively simple geometric objects

• Different from *image* and *pictorial* database systems:
  – Containing sets of objects in space rather than images or pictures of a space
SDBMS Example

- Consider a spatial dataset with:
  - County boundary (dashed white line)
  - Census block - name, area, population, boundary (dark line)
  - Water bodies (dark polygons)
  - Satellite Imagery (gray scale pixels)

- Storage in a SDBMS table:

```sql
create table census_blocks ( 
    name string, 
    area float, 
    population number, 
    boundary polygon );
```
Modeling Spatial Data in Traditional DBMS

- A row in the table `census_blocks`
- Question: Is Polyline datatype supported in DBMS?
Spatial Data Types and Traditional Databases

• Traditional relational DBMS
  – Support simple data types, e.g. number, strings, date
  – Modeling Spatial data types is tedious

• Example: next slide shows modeling of polygon using numbers
  – Three new tables: polygon, edge, points
    • Note: Polygon is a polyline where last point and first point are same
  – A simple unit square represented as 16 rows across 3 tables
  – Simple spatial operators, e.g. area(), require joining tables
  – Tedious and computationally inefficient
Mapping “census_table” into a Relational Database

<table>
<thead>
<tr>
<th>Census_blocks</th>
<th>Polygon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Area</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>340</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boundary-ID</th>
<th>edge-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1050</td>
<td>A</td>
</tr>
<tr>
<td>1050</td>
<td>B</td>
</tr>
<tr>
<td>1050</td>
<td>C</td>
</tr>
<tr>
<td>1050</td>
<td>D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Edge</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>edge-name</td>
<td>endpoint</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
</tr>
</tbody>
</table>
Evolution of DBMS technology

File Systems

Network DBMS  Hierarchical DBMS

Relational DBMS

Object-Oriented Systems (OODBMS)

Object-Relational ORDBMS
Spatial Data Types and Post-relational Databases

• Post-relational DBMS
  – Support user defined abstract data types
  – Spatial data types (e.g. polygon) can be added

• Choice of post-relational DBMS
  – Object oriented (OO) DBMS
  – Object relational (OR) DBMS

• A spatial database is a collection of spatial data types, operators, indices, processing strategies, etc. and can work with many post-relational DBMS as well as programming languages like Java, Visual Basic etc.
How is a SDBMS different from a GIS?

• GIS is a software to visualize and analyze spatial data using spatial analysis functions such as
  – **Search** Thematic search, search by region, (re-)classification
  – **Location analysis** Buffer, corridor, overlay
  – **Terrain analysis** Slope/aspect, catchment, drainage network
  – **Flow analysis** Connectivity, shortest path
  – **Distribution** Change detection, proximity, nearest neighbor
  – **Spatial analysis/Statistics** Pattern, centrality, autocorrelation, indices of similarity, topology: hole description
  – **Measurements** Distance, perimeter, shape, adjacency, direction

• GIS uses SDBMS
  – to store, search, query, share large spatial data sets
How is a SDBMS different from a GIS?

• SDBMS focuses on
  – Efficient storage, querying, sharing of large spatial datasets
  – Provides simpler set based query operations
  – Example operations: search by region, overlay, nearest neighbor, distance, adjacency, perimeter etc.
  – Uses spatial indices and query optimization to speedup queries over large spatial datasets.

• SDBMS may be used by applications other than GIS
  – Astronomy, Genomics, Multimedia information systems, ...

• Will one use a GIS or a SDBM to answer the following:
  – How many neighboring countries does USA have?
  – Which country has highest number of neighbors?
Three meanings of the acronym GIS

• Geographic Information Services
  – Web-sites and service centers for casual users, e.g. travelers
  – Example: Service (e.g. AAA, mapquest) for route planning

• Geographic Information Systems
  – Software for professional users, e.g. cartographers
  – Example: ESRI Arc/View software

• Geographic Information Science
  – Concepts, frameworks, theories to formalize use and development of geographic information systems and services
  – Example: design spatial data types and operations for querying
Components of a SDBMS

• Recall: a SDBMS is a software module that
  – can work with an underlying DBMS
  – supports spatial data models, spatial ADTs and a query language from which these ADTs are callable
  – supports spatial indexing, algorithms for processing spatial operations, and domain specific rules for query optimization

• Components include
  – spatial data model, query language, query processing, file organization and indices, query optimization, etc.
Spatial Taxonomy, Data Models

• Spatial Taxonomy:
  – multitude of descriptions available to organize space.
  – Topology models homeomorphic relationships, e.g. overlap
  – Euclidean space models distance and direction in a plane
  – Graphs models connectivity, Shortest-Path

• Spatial data models
  – rules to identify identifiable objects and properties of space
  – Object model help manage identifiable things, e.g. mountains, cities, land-parcels etc.
  – Field model help manage continuous and amorphous phenomenon, e.g. wetlands, satellite imagery, snowfall etc.
Modeling*

- WLOG assume 2-D and GIS application, two basic things need to be represented:
  - Objects in space: cities, forests, or rivers
  - \( \rightarrow \) modeling single objects
  - Space: say something about every point in space (e.g., partition of a country into districts)
  - \( \rightarrow \) modeling spatially related collections of objects
Modeling* ...

• Fundamental abstractions for modeling single objects:
  
  – Point: object represented only by its location in space, e.g., center of a state
  
  – Line (actually a curve or ployline): representation of moving through or connections in space, e.g., road, river
  
  – Region: representation of an extent in 2d-space, e.g., lake, city
Modeling* ...

• Instances of spatially related collections of objects:
  
  – Partition: set of region objects that are required to be disjoint (adjacency or region objects with common boundaries), e.g., thematic maps
  
  – Networks: embedded graph in plane consisting of set of points (vertices) and lines (edges) objects, e.g. highways, power supply lines, rivers
Modeling* ...

- Spatial relationships:
  - *Topological* relationships: e.g., adjacent, inside, disjoint. Are invariant under topological transformations like translation, scaling, rotation
  - *Direction* relationships: e.g., above, below, or north_of, southwest_of, ...
  - *Metric* relationships: e.g., distance

- Enumeration of all possible topological relationships between two simple regions (no holes, connected):
  - Based on comparing two objects boundaries ($\delta A$) and interiors ($A^0$), there are 4 sets each of which be empty or not $= 2^4=16$.
    - 8 of these are not valid and 2 symmetric so:

- 6 valid topological relationships:
  - disjoint, touch, overlap, cover, in, equal
Modeling* ...

• DBMS data model must be extended by SDTs at the level of atomic data types (such as integer, string), or better be open for user-defined types (OR-DBMS approach):

  **relation** states (sname: STRING; area: REGION; spop: INTEGER)

  **relation** cities (cname: STRING; center: POINT; ext: REGION; cpop: INTEGER);

  **relation** rivers (rname: STRING; route: LINE)
Spatial Query Language

• Spatial query language
  • Spatial data types
    • e.g. point, linestring, polygon, ...

• Spatial operations
  • e.g. overlap, distance, nearest neighbor, ...

• Callable from a query language (e.g. SQL3) of underlying DBMS

```
SELECT     S.name
FROM Senator S
WHERE S.district.Area() > 300
```
Query Processing

- Efficient algorithms to answer spatial queries
- Common Strategy - filter and refine
  - Filter Step: Query Region overlaps with MBRs of B, C and D
  - Refine Step: Query Region overlaps with B and C
Fundamental spatial algebra operations:

• *Spatial selection*: returning those objects satisfying a spatial predicate with the query object
  
  – “All cities in Bavaria”
    
    SELECT sname FROM cities c WHERE c.center inside Bavaria.area
  
  – “All rivers intersecting a query window”
    
    SELECT * FROM rivers r WHERE r.route intersects Window
  
  – “All big cities no more than 100 Kms from Hagen”
    
    SELECT cname FROM cities c WHERE dist(c.center, Hagen.center) < 100 and c.pop > 500k

(conjunction with other predicates and query optimization)
Querying* ...

- **Spatial join**: A join which compares any two joined objects based on a predicate on their spatial attribute values.
  - “For each river pass through Bavaria, find all cities within less than 50 Kms.”

```sql
SELECT r.rname, c.cname, length(intersection(r.route, c.area))
FROM rivers r, cities c
WHERE r.route intersects Bavaria.area and
  dist(r.route, c.area) < 50 Km
```
File Organization and Indices

• A difference between GIS and SDBMS assumptions
  • GIS algorithms: dataset is loaded in main memory (a)
  • SDBMS: dataset is on secondary storage e.g disk (b)
  • SDBMS uses space filling curves and spatial indices
    • to efficiently search disk resident large spatial datasets
Organizing spatial data with space filling curves

• Issue:
  • Sorting is not naturally defined on spatial data
  • Many efficient search methods are based on sorting datasets

• Space filling curves
  • Impose an ordering on the locations in a multi-dimensional space
  • Examples: row-order (a), z-order (b)
  • Allow use of traditional efficient search methods on spatial data
  • More details on next sessions

![Example](image)
Spatial Indexing

• To expedite spatial selection (as well as other operations such as spatial joins, …)

• It organizes space and the objects in it in some way so that only parts of the space and a subset of the objects need to be considered to answer a query.

• Two main approaches:
  • Dedicated spatial data structures (e.g., R-tree)
  • Spatial objects mapped to a 1-D space to utilize standard indexing techniques (e.g., B-tree)
Spatial Data Mining

• Analysis of spatial data is of many types
  • Deductive Querying, e.g. searching, sorting, overlays
  • Inductive Mining, e.g. statistics, correlation, clustering, classification, …

• Data mining is a systematic and semi-automated search for interesting non-trivial patterns in large spatial databases

• Example applications include
  • Infer land-use classification from satellite imagery
  • Identify cancer clusters and geographic factors with high correlation
  • Identify crime hotspots to assign police patrols and social workers
Summary

• SDBMS is valuable to many important applications

• SDBMS is a software module
  – works with an underlying DBMS
  – provides spatial ADTs callable from a query language
  – provides methods for efficient processing of spatial queries

• Components of SDBMS include
  – spatial data model, spatial data types and operators,
  – spatial query language, processing and optimization
  – spatial data mining

• SDBMS is used to store, query and share spatial data for GIS as well as other applications