Introduction to Spatial Database Systems

by Cyrus Shahabi

from

Ralf Hart Hartmut Guting's

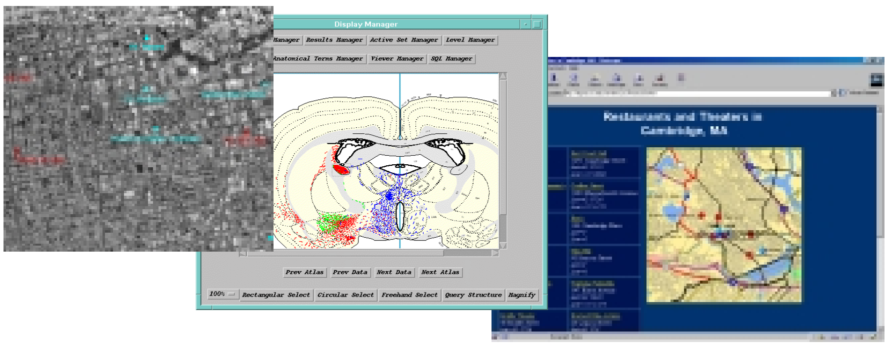
VLDB Journal v3, n4, October 1994

Outline

- Introduction & definition
- Modeling
- Querying
- Data structures and algorithms
- System architecture
- Conclusion and summary

Introduction

- 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



Introduction …

- 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

Definition

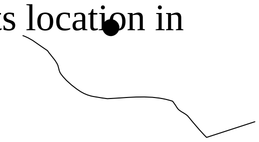
- A spatial database system:
  - Is a database system
    - A DBMS with additional capabilities for handling spatial data
  - Offers spatial data types (SDTs) in its data model and query language
    - Structure in space: e.g., POINT, LINE, REGION
    - Relationships among them: (*l intersects r*)
  - Supports SDT in its implementation
    - Providing at least <u>spatial indexing</u> (retrieving objects in particular area without scanning the whole space)
    - Efficient algorithm for <u>spatial joins</u> (not simply filtering the cartesian product)

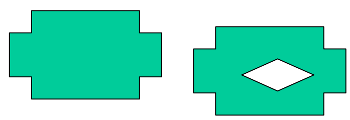
Modeling

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

Modeling …

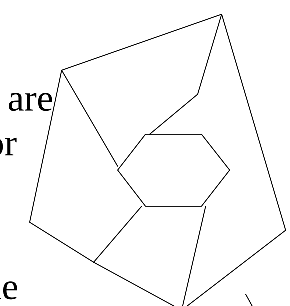
- Fundamental abstractions for modeling single objects:
  - Point: object represented only by its location in space, e.g., center of a state
  - Line (actually a curve or ployline): representation of moving through or connections in space, e.g., road, river
  - Region: representation of an extent in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

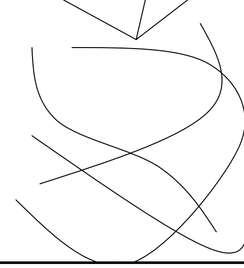




Modeling …

- 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 …

A sample (ROSE) spatial type system

EXT={lines, regions}, GEO={points, lines, regions}

- Spatial predicates for topological relationships:
  - **inside:** *geo* x *regions* → *bool*
  - **intersect, meets:** *ext1* x *ext2* → *bool*
  - **adjacent, encloses:** *regions* x *regions* → *bool*
- Operations returning atomic spatial data types:
  - **intersection:** *lines x lines* → *points*
  - **intersection:** *regions x regions* → *regions*
  - **plus, minus:** *geo x geo* → *geo*
  - **contour:** *regions* → *lines*

Modeling …

- Spatial operators returning numbers
  - **dist:** *geo1 x geo2* → *real*
  - **perimeter, area:** *regions* → *real*
- Spatial operations on set of objects
  - **sum:** *set(obj) x (obj→geo)* → *geo*
  - A spatial aggregate function, geometric union of all attribute values, e.g., union of set of provinces determine the area of the country
  - **closest:** *set(obj) x (obj→geo1) x geo2* → *set(obj)*
  - Determines within a set of objects those whose spatial attribute value has minimal distance from geometric query object

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, sothwest_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 (δA) and interiors (A^o), 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, in, touch, equal, cover, overlap

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)

Querying

- Two main issues:
  1. Connecting the operations of a spatial algebra (including predicates to express spatial relationships) to the facilities of a DBMS query language.
  2. Providing graphical presentation of spatial data (i.e., results of queries), and graphical input of SDT values used in queries.

Querying …

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."

    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

Querying …

- Graphical I/O issue: how to determine "Window" or "Bavaria" in previous examples (input); or how to show "intersection(route, Bavaria.area)" or "r.route" (output) (results are usually a combination of several queries).
- Requirements for spatial querying [Egenhofer]:
  - Spatial data types
  - Graphical display of query results
  - Graphical combination (overlay) of several query results

  (start a new picture, add/remove layers, change order of layers)
  - Display of context (e.g., show background such as a raster image (satellite image) or boundary of states)
  - Facility to check the content of a display (which query contributed to the content)

Querying …

- Extended dialog: use pointing device to select objects within a subarea, zooming, …
- Varying graphical representations: different colors, patterns, intensity, symbols to different objects classes or even objects within a class
- Legend: clarify the assignment of graphical representations to object classes
- Label placement: selecting object attributes (e.g., population) as labels
- Scale selection: determines not only size of the graphical representations but also what kind of symbol be used and whether an object be shown at all
- Subarea for queries: focus attention for follow-up queries