Database Challenges of Spatiotemporal Data

Jan Chomicki
Dept. CSE
University at Buffalo
State University of New York
`http://www.cse.buffalo.edu/~chomicki`

Workshop on Spatiotemporal Data Models for Biogeophysical Fields, April 9, 2002, San Diego, California.

Relational data model

De-facto standard for business data [Codd, 1970].

Basic notions:

- relation **schema**: a finite set of **attributes**
- relation **instance**: a finite set of (flat) **tuples**

SSN	*Name*	*Salary*
123456789	John Smith	80K
333333333	Mary White	95K

Limitations of the relational model

First Normal Form:

- values are atomic
- complex values need to be unnested

“Bare-bones” type system:

- only atomic types
- no subtyping/inheritance
- no encapsulation of operations with data

No **object identity**.

Structural rigidity:

- no support for unstructured/heterogenous data

Spatial data in the relational model

Boundary representation (vector):

Name	x	y
Birnam Wood	1 03'	50 49'
Birnam Wood	1 10'	50 45'
Birnam Wood	1 02'	50 36'

One-dimensional encoding (raster).

Problems (some):

- low level
- no notion of spatial object/type
- mismatch with the query language

Beyond relational I

Object-relational:

- abstract data types (blackbox/whitebox)
- row types and references
- inheritance

Example ADT `Polygon`:

- constructors
- methods: containment, overlap,...
- `Rectangle` **isa** `Polygon`

Query language: **SQL:1999**.

Beyond relational II

Constraint databases [Kanellakis et al., 1990]:

- **constraint tuple**: a finite set (conjunction) of atomic constraints
- **constraint relation**: a finite set (disjunction) of generalized tuples
- semantics: **infinite point-sets**
- usually linear arithmetic constraints (may be more general)

Example:

$$0 \leq x \leq 2 \wedge y \leq y \wedge y \geq 0.$$

Query languages: relational calculus, relational algebra.

Theoretically appealing but few implemented systems.

Spatiotemporal phenomena

What is changing **where** and **how**.

What:

- $0D$ points
- $1D$ lines
- $2D$ regions
- $3D$ volumes.

Where:

- in $1D$ space (line)
- in $2D$ space (plane)
- in 3D space.

How:

- continuous **movement**
- continuous **evolution**
- discrete **evolution**
- birth, death, split, merge....

Examples

Transportation: truck or ship movement, airplane flights.

Natural disasters: oil spills, forest fires.

Ecology: species migration, habitat or land cover changes.

Climate: season or vegetation changes.

Society and economy: urban growth, land use changes, epidemics.

Ownership or administrative changes.

Spatiotemporal objects

Ading the **time dimension** to spatial objects.

Object-relational:

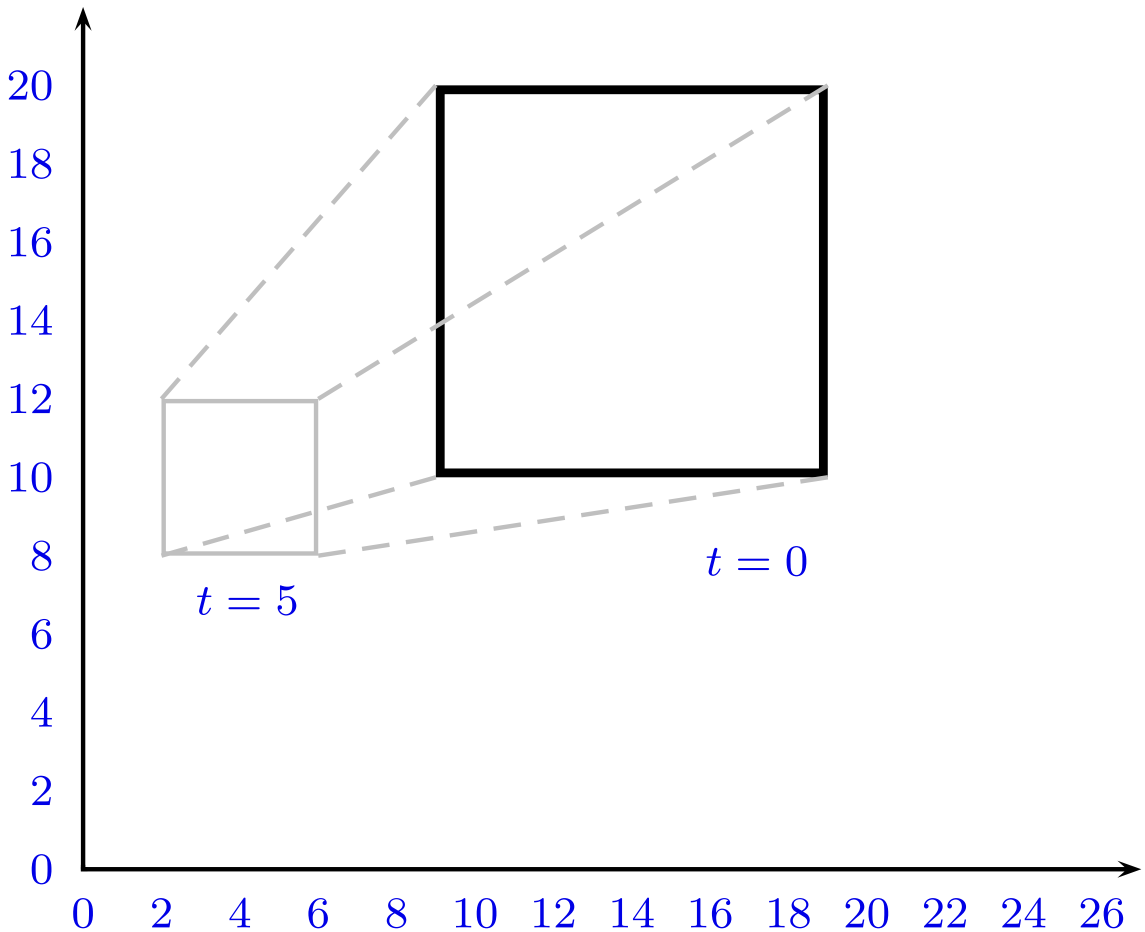
- temporal **lifting**
- **concrete** representation: polyhedron in 3D
- **closure** problematic

Constraint databases:

- one extra **time variable**:

$$t \geq 0 \wedge t \leq 5 \wedge x + y \leq t \wedge x \geq 0 \wedge y \geq 3$$

- **closure** guaranteed



20
18
16
14
12
10
8
6
4
2
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26
$t = 5$
$t = 0$

Representation problems

Real spatiotemporal data:

(A) data comes as **discrete observations**:

– within a snapshot (TINs)

– in different snapshots

(B) data lacks clearly **identifiable** and **delineated** objects

(C) modelled movement/evolution **irregular**

(D) data does not have regular (**polyhedral**) 3D structure.

Solution to (A), (C), and (D):

1. convert each snapshot to a set of polygons
   - intrasnapshot interpolation can be also expressed using constraints
2. interpolate/approximate between the snapshots.

$\Rightarrow$ the ADT or constraint approach more suitable to construct **approximations**.

Object definition:

- large **collections** of points
- complex **conditions**: $temperature > 32F$
- results of scientific analysis **programs**.

Will relational databases and relational query languages be still useful in that context?

Arrival of spring query:

Find the regions where the spring arrived earlier than a year before.

$$\exists t, t'.[t < t' < t + 365 \wedge S(t, x, y) \wedge \neg S(t-1, x, y) \wedge S(t', x, y) \wedge \neg S(t'-1, x, y)].$$

Other challenges

Data models:

- *type* systems
- representing *uncertainty*: *speed between 40 and 60 mph*
- *integrating* different representations
- resolving *inconsistencies*

Query languages and interfaces:

- multidimensional aggregation (*spatiotemporal OLAP*)
- *visualization*
- *animation*:
  - *explicit* representations (ADTs) better than *implicit* ones (constraints)

Databases with moving objects

[Wolfson et al., 1997-; Su et al., 2001-].

Moving object:

- point movement in 2D
- satisfies *motion continuity*
- component functions (*motion vector*) infinitely differentiable

Moving object database (MOD):

- finite set of moving objects
- the instant NOW

Querying MOD

Operations:

- location
- direction, distance, length,...
- spatial/spatiotemporal predicates
- speed, acceleration,...

Temporal dimension:

- queries about the *past*: *"where was truck #123 at 5pm yesterday?"*
- queries about the *present*
- queries about the *future*

Query languages:

- SQL3 [Forlizzi et al., SIGMOD 2000]
- relational calculus with built-in functions [Su et al., SSTD 2001]
- temporal logic [Wolfson et al., ICDE 1997]

Location issues

Uncertainty:

- object information may be out of date
- *certain/possible* query answers
- probability distributions

Updates:

- cost vs. imprecision tradeoff
- not practical to report every change to the motion vector: *a winding road*

Commercial technology: Qualcomm Omnitracs, Mobitrac.

Some outstanding issues in MOD

Modeling movement:

- **1.5D**: movement on fixed road networks
- what instead of precise location of an object it is enough to know whether it will arrive to some location by a certain **deadline**?

Query processing:

- queries with **uncertainty** factors
- instantaneous vs. **continuous** queries
- location **sampling**
- **conflict** resolution

A final look at MOD

MOD is now a separate research area:

- important practical applications
- specific technical issues: precision/uncertainty/probability
- specialized query languages
- specialized indexing techniques

Can the success of MOD be replicated?

Bottom-up approach:

- adding spatiotemporal constructs to existing GIS, in response to applications' demands
- problems with generality, interoperability etc.

Top-down approach:

- design a general model with clean semantics based, for example, on constraint databases
- will anyone use it in practice?

Adapt **general** query languages to a **broad spectrum** of spatiotemporal data.