Mobility Data Management and Exploration: Theory and Practice

Chapter 4 - Mobility data management at the physical level

Nikos Pelekis & Yannis Theodoridis
InfoLab, University of Piraeus, Greece
infolab.cs.unipi.gr

A guided tour on Mobility Data Mgmt

I. Mobility data storage and querying
   Efficient trajectory indexing and storage in MODs
Mobility data storage and querying

“πάντα ρει - Everything changes and nothing remains still”
Heraclitus (500 BC)

Key questions that arise

- How to store and query trajectories in a DBMS?
  - Is a trajectory simply a sequence of (x, y, t) tuples?

- What kind of analysis is suitable for mobility data?
  - In particular, trajectories of moving objects?
  - How does infrastructure (e.g. road network) affect this analysis?
  - Which patterns / models can be extracted out of them?
    - Clusters, frequent patterns, anomalies / outliers, etc.
  - How to compute such patterns / models efficiently?
  - How to protect privacy / anonymity?
  - trade-off between privacy protection and quality of analysis
Efficiently trajectory indexing and storage in MODs

Indexing techniques
MOD engines

R-trees for spatial data

- For $d$-dimensional point or region data
- Is it portable to mobility data?
  - In other words, is space + time simply a 3D space?
What ?? for mobility data

- Challenges:
  - Both space and time are equally important!
  - But! time is not simply a 3rd dimension
    - think of its monotonicity
    - Also, density in the ‘spatio-temporal’ space could be quite different from place to place

- Diversity in space, in time, in space-time ...

Diversity in space

Time: 05/06/2008 05:03 pm

The number of objects varies in different regions
Diversity in time

Time: 05/06/2008 05:03 pm  Time: 05/06/2008 08:05 am

The number of objects changes with time.

Diversity in space & time

Time: 05/06/2008 05:03 pm  Time: 05/06/2008 08:05 am

The distribution of objects also changes with time.
Assume we have stored trajectories (i.e., the recorded locations of a moving object over time)

Main Question: how can we approximate a trajectory?
- Like, e.g. MBRs for (static) spatial objects?
- Then, we could build indices upon the approximations

Two approaches: native vs. parametric space

Typically approximate using MBRs; then index these MBRs
- we can use R-trees etc. 😊
- trajectories are lines, thus MBRs add extensive empty space 😞

How many MBRs per trajectory?
- One MBR per trajectory (too much empty space...) or one MBR per segment (too many MBRs...)

Can we do anything better?
- Smart “partitioning” for MBRs [Hadjieleftheriou et al. 2002]
MOD indexing techniques

- Indexing the past (= trajectories)
  - unconstrained movement: the trajectory-bundle tree (TB-tree) [Pfoser et al. 2000]
  - network-constrained movement: the fixed-network-restricted tree (FNR-tree) [Frentzos, 2003]
- Indexing the present (and anticipated future)
  - Data partitioning: TPR-tree [Saltenis et al. 2000], TPR*-tree [Tao et al. 2003]
  - Space partitioning: B*-tree [Jensen et al. 2004], ST²B-tree [Chen et al. 2008]
- (Hybrid solution for) Indexing the past & present
  - RPPF-tree [Pelanis et al. 2006]

We focus on historical tracks of moving objects (trajectories) concentrate on the 1st group

TB-tree

- [Pfoser et al. 2000] Maintains the ‘trajectory’ concept
  - Each node consists of segments of a single trajectory
  - nodes corresponding to the same trajectory are linked together in a chain
  - Effective for trajectory-oriented queries
FNR-tree

- (Frentzos, 2003) a forest of 1D (temporal) R-trees on top of a 2D (spatial) R-tree
  - There is an additional “Parent” 1D R-tree which indexes the temporal intervals of the 1D R-trees leaf nodes

Moving Objects Database Systems

- From traditional DBMS to Moving Object Database (MOD) engines
  - Data types, indices, query processing & optimization strategies for trajectories
  - Spatial and temporal dimensions are considered as first-class citizens.
- Several prototype MOD engines
  - **SECONDO** (Güting et. al.) IDEAS’00, ICDE’05, MDM’06
  - **PLACE** (Aref et al.) SSDBM’04, VLDB’04
  - **HERMES** (Pelekis et. al.) EDBT’06, SIGMOD’08, IJKBO’11
A generic DBMS framework that can be filled with implementation of various data models (R, OR, XML) and data types (spatial data, moving objects)

Built on top of Berkeley DB.

A MOD is a set of SECONDO objects of the form \((name, type, value)\), where \(type\) is one of the implemented algebras

About 20 implemented algebras
- standard algebra, relational algebra, R-Tree algebra, spatial algebra, etc.

Query optimizer includes optimization of conjunctive queries, selectivity estimation, and implementation of an SQL-like query language

---

**PLACE**

Continuous evaluation of queries over spatio-temporal data streams

Shared execution among concurrent continuous queries

Built on top of PREDATOR database system

Incremental evaluation of continuous queries

Spatio-temporal query operators
A palette of Abstract Data Types on top of an extensible DBMS

- Moving point, moving line, moving polygon, etc.
- Linear vs. Arc movement
- R-tree and TB-tree indexing support

Summary
Summary on Mobility Data Management

- Open issues:
  - Trajectory indexing
    - *Hybrid* (past- and present- locations) indexing techniques
  - Advanced **MOD architectures**
    - centralized vs. distributed vs. stream-oriented

Questions
Reading list

- Mobility data modeling & MOD engines
  - de Almeida, V.T. et al. (2006) Querying Moving Objects in SECONDO. Proceedings of MDM.

- Mobility data modeling & MOD engines (cont.)
  - Mokbel, M.F. et al. (2004a) Continuous Query Processing of Spatio-temporal Data Streams in PLACE. Proceedings of SSDBM.
Reading list

Mobility data modeling & MOD engines (cont.)

MOD Indexing
Reading list

MOD Indexing (cont.)
