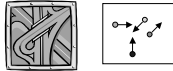


## Topic 2: Spatiotemporal Data



- Large collections of spatio-temporal data are becoming available.
- Example: widespread use on-line use of mobile personal information appliances.
- Applications that need to manage efficiently such data include:
  - Fleet management systems
  - Location-based services
  - Intelligent transportation systems
  - Traffic supervision systems
  - ...

## Three Types of Spatiotemporal Data Management

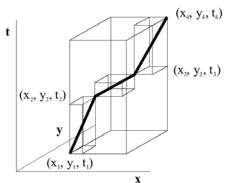
- Indexing and querying the movements of objects in the past (trajectories).
  - Applications: data analysis for traffic supervision, managing recordings for security systems, etc.
- Indexing and querying the current movements of objects in a centralized server.
  - Applications: querying GPS data for traffic supervision systems, military systems, etc.
- Querying the current movements of objects in a distributed mobile environment.
  - Applications: Real-time navigation systems, highly dynamic wireless network applications using GPS, autonomous sensors, etc.

## 1. Querying Past Movements

- Various query types are applied on past data movements:
  - *Timestamp / interval* queries find objects which intersect a spatial window at a specific timestamp / *time-window* :  
*find all cars which were in Central at 2:00 p.m yesterday*  
*find all cars which were in Central between 2:00 p.m and 3:30 p.m. yesterday*
  - *Topological* queries find the space/time intersections of objects with a spatial window:  
*when and where did car X enter and exit Central yesterday?*
  - *Navigational* queries retrieve information such as speed and heading of objects:  
*what was the speed (travel distance) of car X in Central, yesterday?*
- We may also retrieve *aggregate* information for many objects:  
*find areas in Central with the highest traffic congestion yesterday*

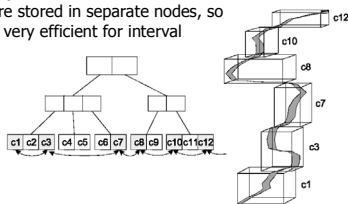
## Indexing Object Trajectories

- Good for timestamp/interval queries. Also useful for topological queries.
- A simple method is to use a 3-dimensional R-tree to index the object trajectories.
- However, due to the nature of data a lot of *dead-space* is covered by the node MBRs.
- Some specialized indexes consider the special nature of object trajectories and queries to improve query performance.



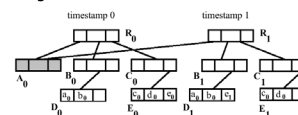
## The Trajectory-bundle Tree (TB-tree)

- The TB tree is a 3D R-tree where each leaf node stores information for *only one* trajectory.
- Thus retrieving a single trajectory is very efficient.
- As a trade-off trajectories close to each other in the 3D space are stored in separate nodes, so the TB-tree is not very efficient for interval queries.



## The Historical R-tree (HR-tree)

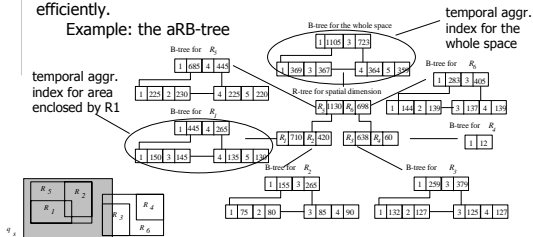
- A possible way to index spatiotemporal data is to build one R-tree for each timestamp. However, this is space-inefficient.
- The HR-tree "compresses" multiple R-trees, so that R-tree nodes which are common in two timestamps are shared by the corresponding R-trees:



- A data structure that combines the 3D R-tree with the HR-tree is the MV3R-tree (it will be described in presentation 3)

## Indexing Historical Aggregate Information

- In many cases we do not want to retrieve information about individual objects, but *aggregate* information about a region
- Specialized indexes can help retrieving this information efficiently.



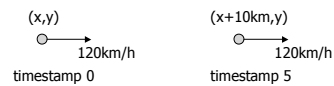
Dr. N. Mamouli

Advanced Database Technologies

7

## 2. Querying Current and Future Movements

- The location of moving objects is changing constantly over time and updating it in a database is expensive and impractical in many cases.
- Thus it is preferred to store the "current movement" of objects as a function over time rather than as a series of individual positions.
- For example if I know that an object was at location  $(x,y)$  5 minutes ago, and it was moving east with speed 120km/hour I can *speculate* that it is currently located at  $(x+10km,y)$ .



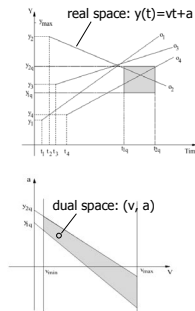
Dr. N. Mamouli

Advanced Database Technologies

8

## Using Geometric Duality to Index Mobile Objects

- In a simple approach we can represent the object movements by lines in 3 dimensional space and index these lines. Queries can be modeled as 3D rectangles and objects which intersect the rectangles are retrieved.
- However, indexing long lines even on the plane is a hard problem. In the dual geometric space, we can represent lines by points and index them efficiently. However, the queries are also transformed to shapes that do not facilitate fast search.



- More in presentation 1...

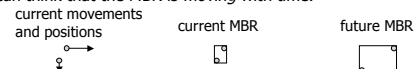
Dr. N. Mamouli

Advanced Database Technologies

9

## The Time-parameterized R-tree

- The TPR-tree stores the current object positions and their movement functions.
- The MBR of a directory node is parameterized to be the MBR of the indexed objects an *any* time instance in the future. You can think that the MBR is *moving* with time.



- The indexed objects are clustered with respect to their relative positions in the future, and not their current positions.
- The tree is valid (useful) only for a limited time interval (after that the structure degenerates). This *validity* period is an input of the construction algorithm.

- More in presentation 2...

Dr. N. Mamouli

Advanced Database Technologies

10

## Time-parameterized Queries

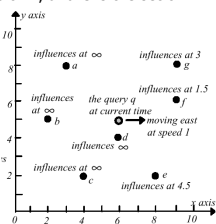
- In TP queries the query object or data objects are moving, so the result changes over time.
- A TP query is defined by a triple  $\langle R, T, C \rangle$ , where R is the current result, T is the *expiry* time of R, and C is the set of objects that will affect R at T.

- Example: a NN TP query

$\langle \{d\}, 1.5, \{f\} \rangle$

- Each object has an influence time  $T_{INF}$  indicating when it will influence the query

- The problem can be transformed to finding the object with the min.  $T_{INF}$



Dr. N. Mamouli

Advanced Database Technologies

11

## 3. Mobile Distributed Environments

- In the future, the number of large centralized databases will shrink, as they are not flexible in updating highly dynamic information.
- We are already experiencing the de-centralization of information. Peer-to-peer network architectures are becoming more and more popular, mobile devices are becoming more intelligent and with storage potential.
- Thus, spatiotemporal data could be managed autonomously without needing to update centralized servers. This model can be implemented easily, it is flexible and more natural.
- However, query processing over networks can be expensive and tricky; message exchanging overloads the network and slows communications. So we still need time to realize the future *dataspace*.

Dr. N. Mamouli

Advanced Database Technologies

12

## The Future DATASPACE

- Objects (e.g., people cars, etc.) will be employed by GPS units; therefore they will be capable of querying mobile information.
- This information will be updated by intelligent sensors, which do not only produce data, but are able to store and analyze it.
- Thus all information is stored where it is produced and accessed "on demand" by other devices. A moving device may also "post" information to a near-by static device.
- Management of spatiotemporal queries is becoming very challenging in this new dataspace. The result of the query depends on the availability of data in the nearby static or dynamic units.
- Message-route scheduling becomes a highly dynamic problem involving both network and spatiotemporal query management.

## Summary



- Several new applications need to manage efficiently spatiotemporal data.
- Queries may refer to the past object positions, or may speculate their future location and movement.
- Indexing methods for both cases have been proposed for centralized database systems.
- The future challenge is to manage spatiotemporal queries autonomously by message exchange between autonomous intelligent mobile appliances.
- In the presentations that follow we will see in detail some indexing methods for spatio-temporal data management:
  1. Indexing current and future object locations
  2. Indexing past locations