Location-dependent Query Processing

- Location-dependent applications
- Location-dependent queries
- Mobile positioning techniques
- Location-dependent databases
  - Data representation
  - Indexing and retrieval
  - Query processing

Location-dependent applications

Traffic Monitoring

- How many cars are in the downtown area?
- Send an alert if a non-friendly vehicle enters a restricted region
- Report any congestion in the road network
- Once an accident is discovered, immediately send alarm to the nearest police and ambulance cars
- Make sure that there are no two aircrafts with nearby paths

Location-based Object Finder / Advertisement

- Where is my nearest Gas station?
- What are the fast food restaurants within 3 miles from my location?
- Am I near to a restaurant while any of my friends are there
- Send E-coupons to all customers within 3 miles of my stores
- Where is Bus 973 now? How long do I have to wait for its arrival?

Location-dependent applications

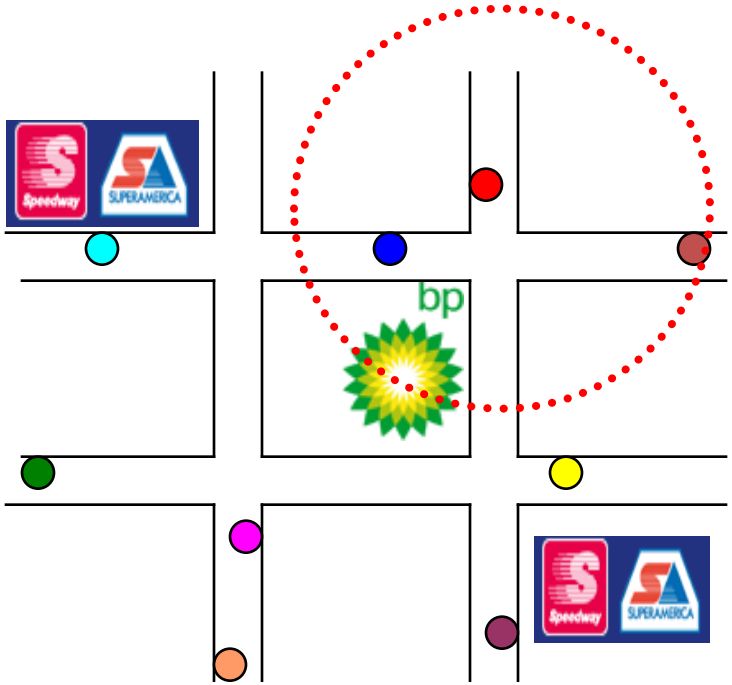
- **Location-dependent applications have three key elements**
  - *Location-dependent queries*:
    - Results depend on the location of the queries or the objects being queried for.
  - *Locations*:
    - Obtained by using mobile positioning technologies.
    - Maintained by spatial databases.
  - *Location-dependent database*:
    - Representation, storage, and retrieval of location-dependent data
    - Processing of location-dependent queries

Location dependant query

- **Query whose result depends on the geographical location of the origin of the query**
  - Moving query
  - Example: What is the distance of Hung Hom station from here?
- **Query whose result depends on the geographical location of the object being queried for**
  - Moving object
  - Example: Where is the Taxi with plate no. 12345678?
  - **User continuously asks the same question - every time the answer is different but correct.**
- **How about query whose context is related to locations?**
  - Find out the top 5 roads in KLW that have the most traffic between 7:00-8:00pm

Location-dependent queries

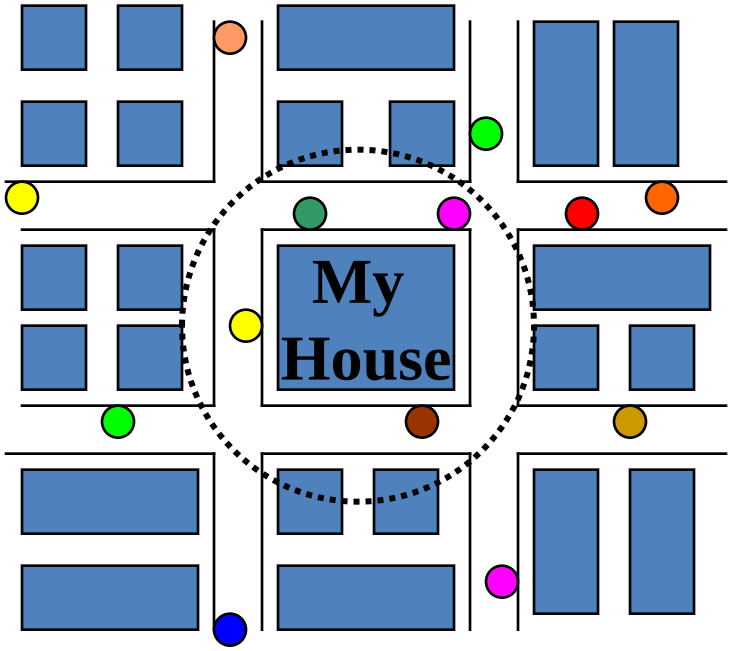
Moving query on stationary objects



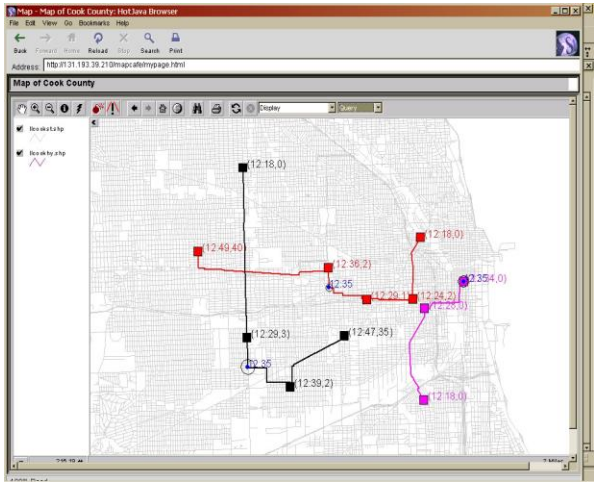
Find the nearest gas station(s) within 1 miles of moving red car

Location-dependent queries

Stationary query on moving objects

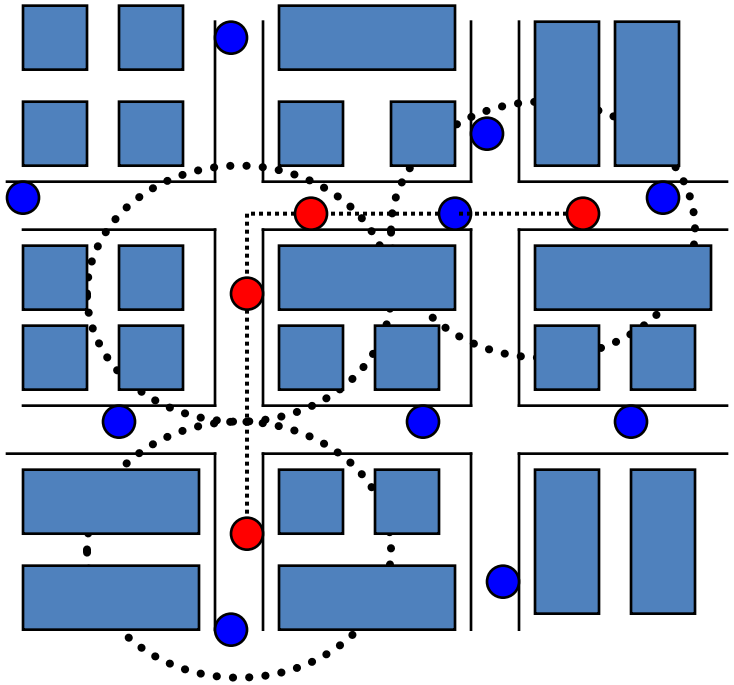


Continuously find all vehicles within 1 miles of my house



Location-dependent queries

Moving query on moving objects

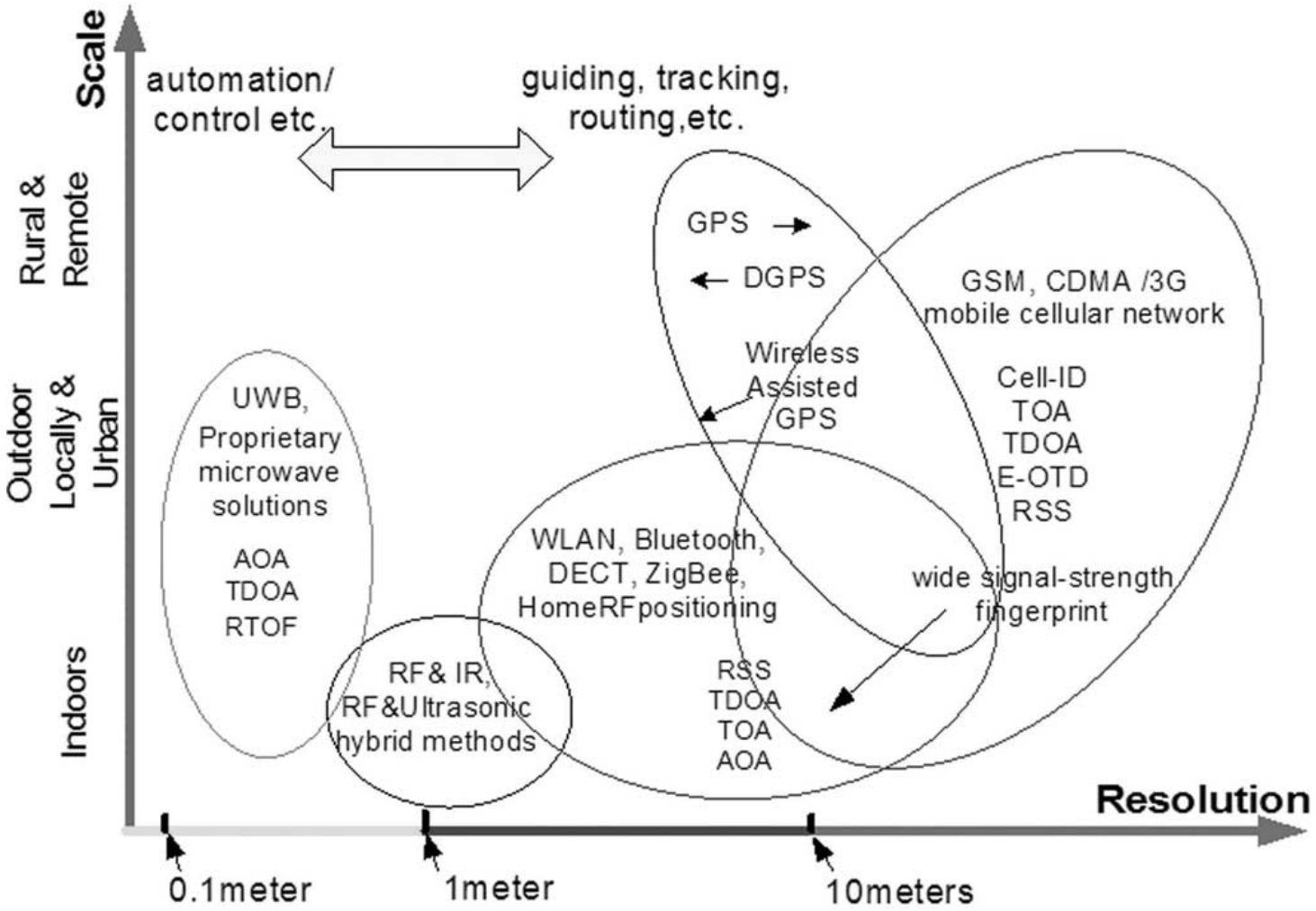


Continuously find all police cars within 3 miles of the moving red car

Mobile positioning techniques

- **Requirement of location-dependent queries:**
  - Continuous monitoring of location of the origin of the query, or of the object being queried for – need mobile positioning technologies.
- **Objective of all positioning technologies is to capture the location of a mobile device and convert it into a meaningful X, Y, Z coordinate**
  - Many technologies are available, deciding which one to use depends on the combination of *accuracy* and *cost*.
  - When the accuracy increases, so does the cost
  - Cost is usually shared between the mobile user and wireless carrier

Mobile positioning techniques



Location representation

- **2 models**
  - Geometric model

    specify a location as an n-dimension coordinate
  - Symbolic model

    logical, real-world entities describing the location space

    e.g., Restaurant A

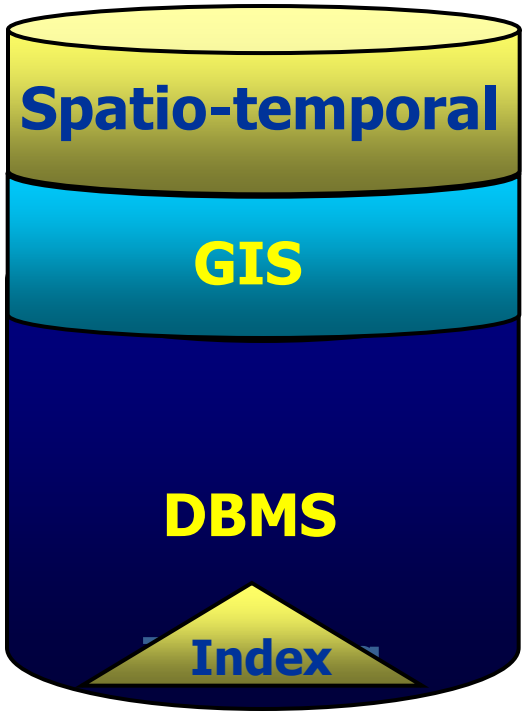
    (250,3000)

    { Mall }
- **Spatial data: data representing / associated with locations**
  - Space: defined by a set of locations

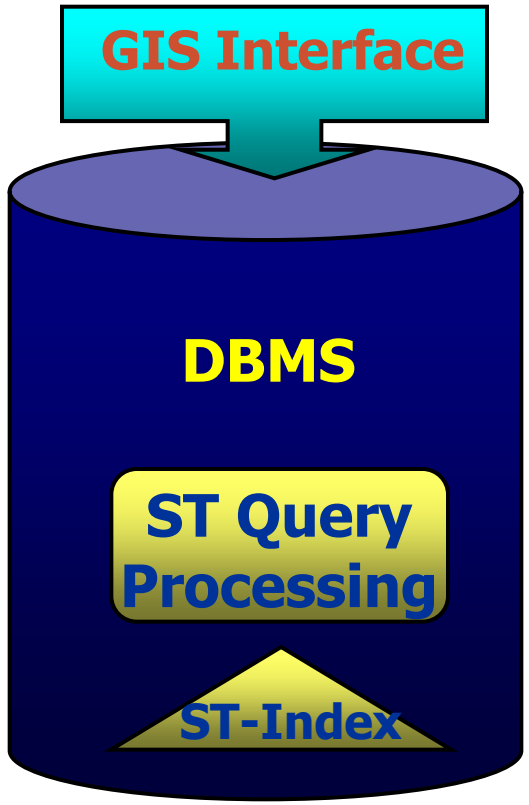
Location-dependent databases

- **Processing of location-dependent queries**
  - Representation, Indexing and retrieval of location-dependent, spatial, and mobile data

Layered Approach



Built-in Approach



Location-dependent data

- **Data whose value is functionally dependent on location.**
  - e.g., city tax, regional weather, hotel rent
- **In database, multiple correct values exist - needs *location binding* or *location mapping function***

Report the trucks in my area

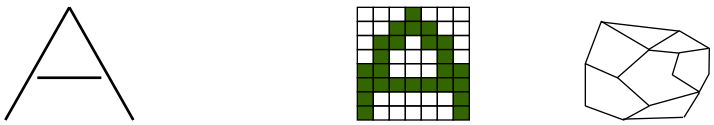
```
SELECT O.ID
FROM Object O
WHERE O.type = "truck"
INSIDE Area My-area
```

Spatial data

- **Data about space or entities occupying space.**
  - Target at representing the *geometry* of spatial features of the entity, e.g., the location of a road, its orientation.
  - Attribute data represents other information about the spatial features, which are merely regular database data, e.g., the name of a road, its speed limit.
- **Spatial objects can be of**
  - zero dimension (a *point*),
  - one dimension (a *line*),
  - two dimensions (an *area*)
  - and so on

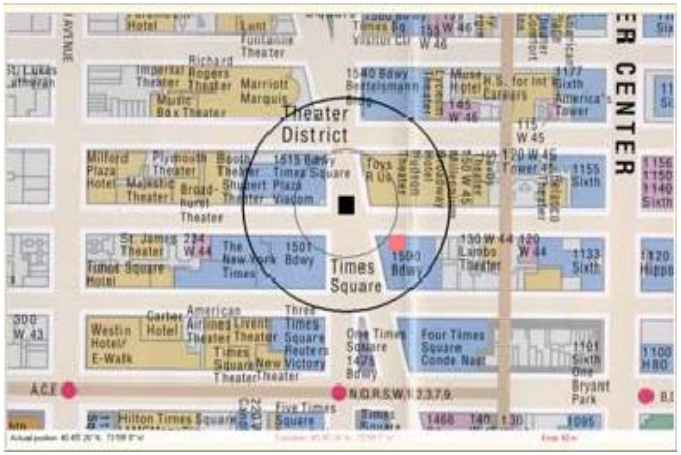
Spatial data representation

- **Spatial features can be *discrete* or *continuous*.**
  - Discrete features do not exist between observations but are individually distinguishable, e.g., road and land use.
  - Continuous features exist between observations, e.g., elevation, rainfall on land.
- **Spatial features can be modeled using *vector data model* or *raster data model*.**
  - Vector data model: using points (and their *x*- and *y*-coordinates) as basis to model lines, areas and so on.
  - Raster data model: using grid to represent spatial feature. Each cell contains a value about the feature at that location.
    - Generalized raster model is called *tessellation model*, in which the elements can be of arbitrary shapes.



Spatial data representation

- **Geographical Information System (GIS) is the computer-based system to manipulate spatial data**
  - A mapping software that translates the low-level position information from the locating infrastructure into meaningful location.
  - building addresses, street layouts, population densities, plethora of other information.



Spatial queries

- **In standard databases, the comparison operator is based on logical and relational operators only.**
- ***Spatial queries* include new operators:**
  - Containment: whether a point or an area is contained in another area.
  - Overlap: whether two areas overlap in some part.
  - Neighbor: whether two entities are close to each other.
  - General spatial relationship: e.g., whether a point is to left of a box.
  - Spatial join: joining two relations containing spatial objects, e.g.

```
select  L.name
from  Legco L, Company C
where  L.district.area() > 20 and within(C.location, L.district);
```

- **General spatial queries include range query for objects within a certain area, nearest neighbor (object that is nearest to another object), *k*-nearest neighbors.**

Location-dependent vs Spatial queries

- **Location-dependent queries are supported by spatial databases and subsume spatial queries.**
- **Similarity to spatial queries:**
  - Operate on spatial data, i.e. data associated with space occupied by object.
  - Expressed with spatial query operators, e.g., contains, contained in, intersects, neighboring, north of.
  - Relying on spatial select and spatial join.
  - Based on spatial data structures for storage and indexing.
- **More than spatial queries:**
  - Client may move in location-dependent queries - location of client may be part of the query itself, and result may depend on direction of movement.
  - Data may not directly contain location information, but may have multiple values corresponding to different locations.
  - Requires dynamic spatial data processing.
  - Sometimes temporal features are also implied (e.g. gas stations in the next 20 minutes).

Variety of location-dependent queries

Continuously report the number of cars in the freeway

- *Type:* Range query
- *Time:* Present
- *Duration:* Continuous
- ***Query:*** **Stationary**
- ***Object:*** **Moving**

What are my nearest McDonalds for the next hour?

- *Type:* Nearest-Neighbor query
- *Time:* Future
- *Duration:* Continuous
- ***Query:*** **Moving**
- ***Object:*** **Stationary**

Send E-coupons to all cars that I am their nearest gas station

- *Type:* Reverse NN query
- *Time:* Present
- *Duration:* Snapshot
- ***Query:*** **Stationary**
- ***Object:*** **Moving**

What was the closest dist. between Taxi A & me yesterday?

- *Type:* Closest-point query
- *Time:* Past
- *Duration:* Snapshot
- ***Query:*** **Moving**
- ***Object:*** **Moving**

Indexing

- **Data are stored within a relational database as tuples and tables.**
- **To answer queries, indexes are built to allow efficient retrieval of data satisfying querying conditions (especially for relational joins).**
  - Allow direct access to tuples satisfying the condition.
  - Most indexes are built based on hashed table (for point query) and B-tree or B^+-tree (for range query).
  - In practice, B^+-trees are used more often, since they allow sequential access to range data by chaining together leaf nodes.

Indexing spatial data

- **In a spatial database, spatial objects can be stored in their primitive forms**
  - e.g., a polygon is represented by the points defining the lines enclosing it or by the polyline.
- **Again, indexes must be built to allow effective processing of additional spatial query types.**
  - Partition the set of objects into subsets according to their locations
  - Examples include R-tree, R*-tree, R^+-tree.
    - B-tree is based on ordering on a one-dimension key attribute.
    - Spatial data (e.g. an area) is of two dimensions - R-tree etc. generalize the B-tree and B^+-tree into their two dimensional equivalence.

R-Tree

- **R-tree is the 2D extension of B^+-tree.**
- **In range query, how do you get a set of points or a set of areas contained within a certain bounding box?**
  - A *minimum bounding box* (mbb) is the smallest rectangular region that just completely covers the area.
  - It is easily computed by taking x_1 and x_2 to be the minimum and maximum value of x-coordinate of all points in the area, and y_1 and y_2 to be the minimum and maximum value of y-coordinate.
- **It is not difficult to get a set of points within an area, by generating a range query on the x- and y-coordinate of the corner points of the bounding box (point query).**

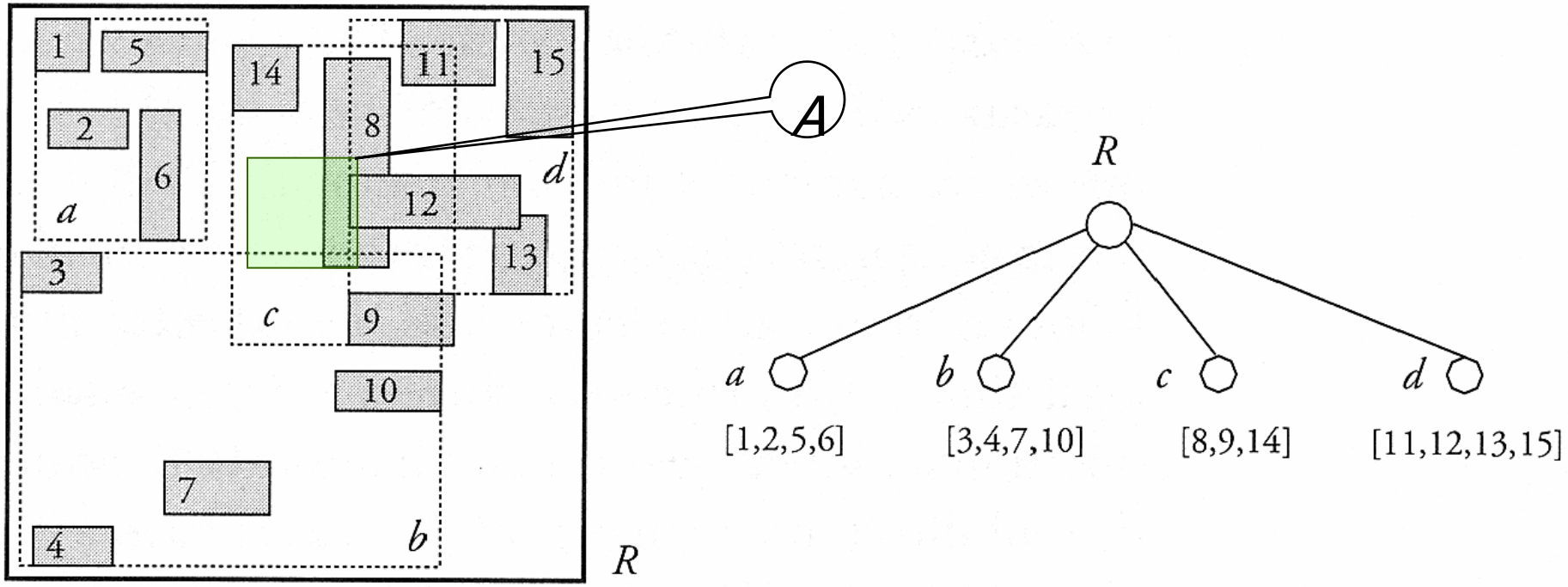
```
select  p.x, p.y from allPoints p
where  x1 <= p.x and p.x <= x2 and y1 <= p.y and p.y <= y2;
```

R-Tree

- **In R-Tree, objects (points or rectangles) are clustered into a group, represented by the mbb of the group and is considered a node.**
- **Nodes are considered again as objects for clustering into next level, until the root is reached.**
  - Note that now the mbb of interior nodes could overlap.
- **For point search, check the mbb of each branch.**
  - If the mbb of a branch contains the point, search recursively down the sub-branches of the branch.
- **For range search, check the mbb of each branch.**
  - If the mbb of a branch intersects with the query rectangle, search recursively down the sub-branches of the branch.
- **Insertion with R-tree is complicated.**
- **R*-tree performs some optimizations to R-tree for insertion.**

R-Tree

- **An example of an R-tree.**
  - How to search for *A*?

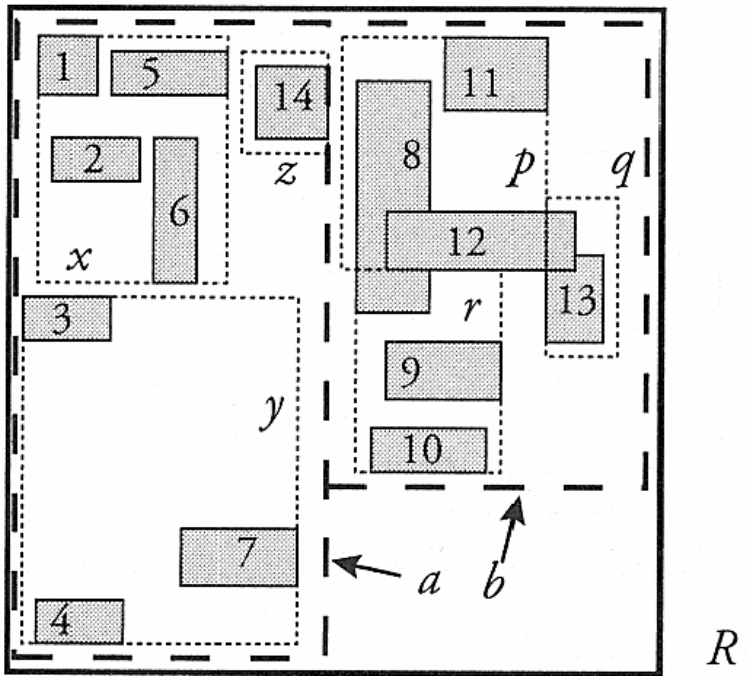


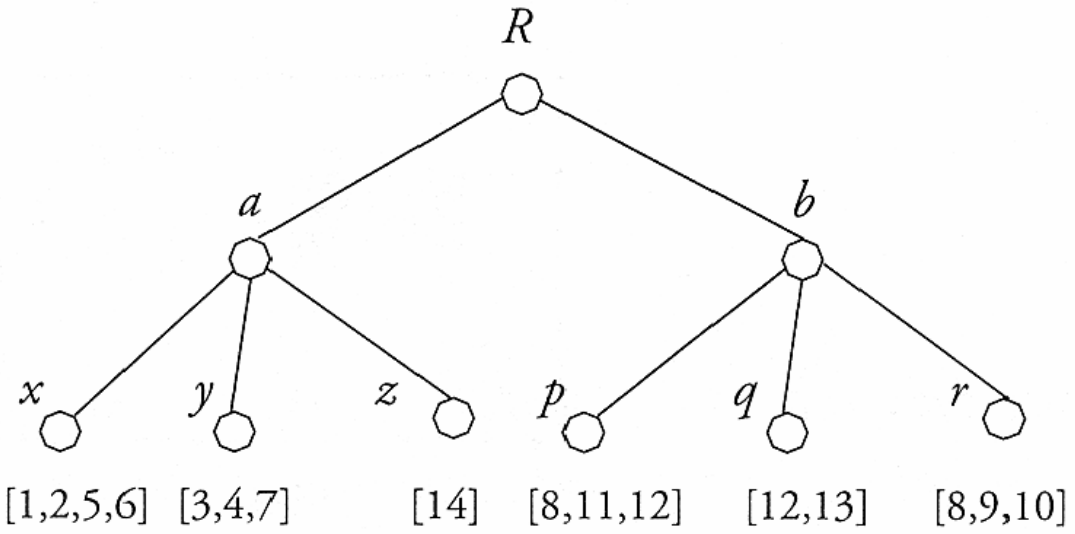
R^+-Tree

- Searching in R-tree could be expensive, since multiple branches need to be explored if the mbb of a branch overlaps with the query.
- R^+-tree is designed to ensure that the mbb for branches at the same level **do not overlap**.
  - The tradeoff is that the object may have to be duplicated.
- Searching in R^+-tree is similar to searching in quadtree.
- Compared with R-tree, the storage overhead for R^+-tree is higher, insertion is more complicated, but searching is much faster.

R+-Tree

- **An example of an R+-tree.**
  - Here objects 8 and 12 are duplicated.



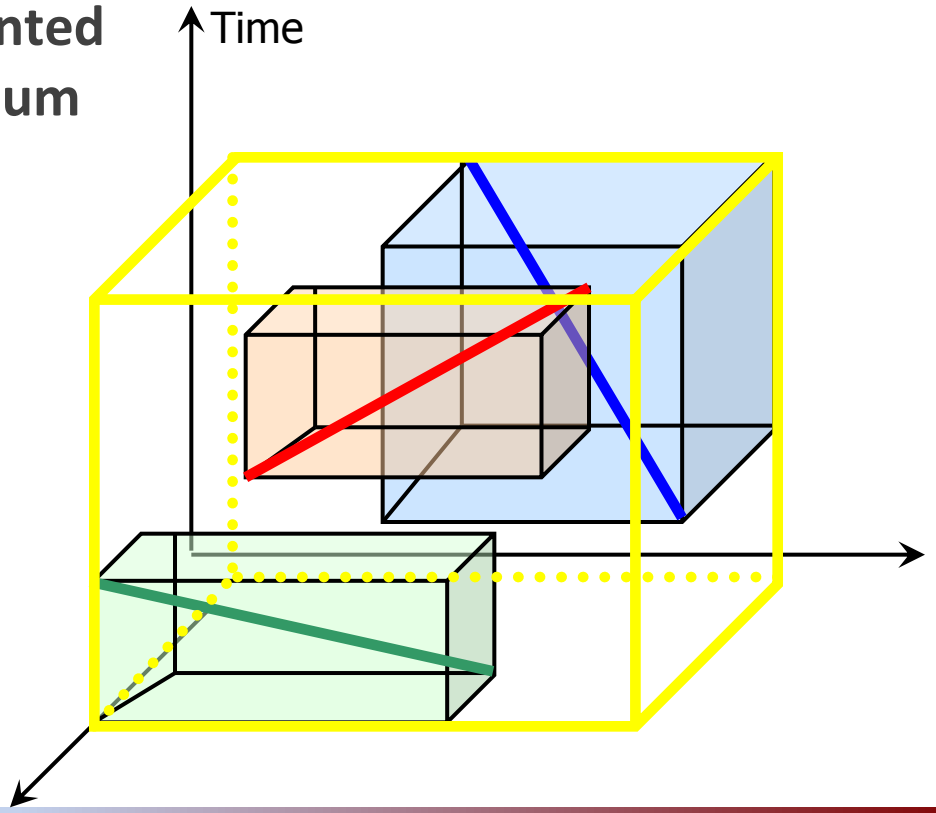


Indexing moving object

- ***Moving object database*** **- spatial database storing locations of moving objects.**
  - E.g. You may want to get the nearest taxi (or the nearest green taxi if you are in Northern N.T.). Database must store the current location of all taxis, since a similar query can be issued at different location.
- **Cost of insertion of an object in spatial database is very high. Worse yet, this needs to be done often for a moving object.**
- **Research is done to search for fast methods to insert and query moving objects, with effective indexing.**
- **We can trade off precision with update efficiency: update important objects more often but others less often.**

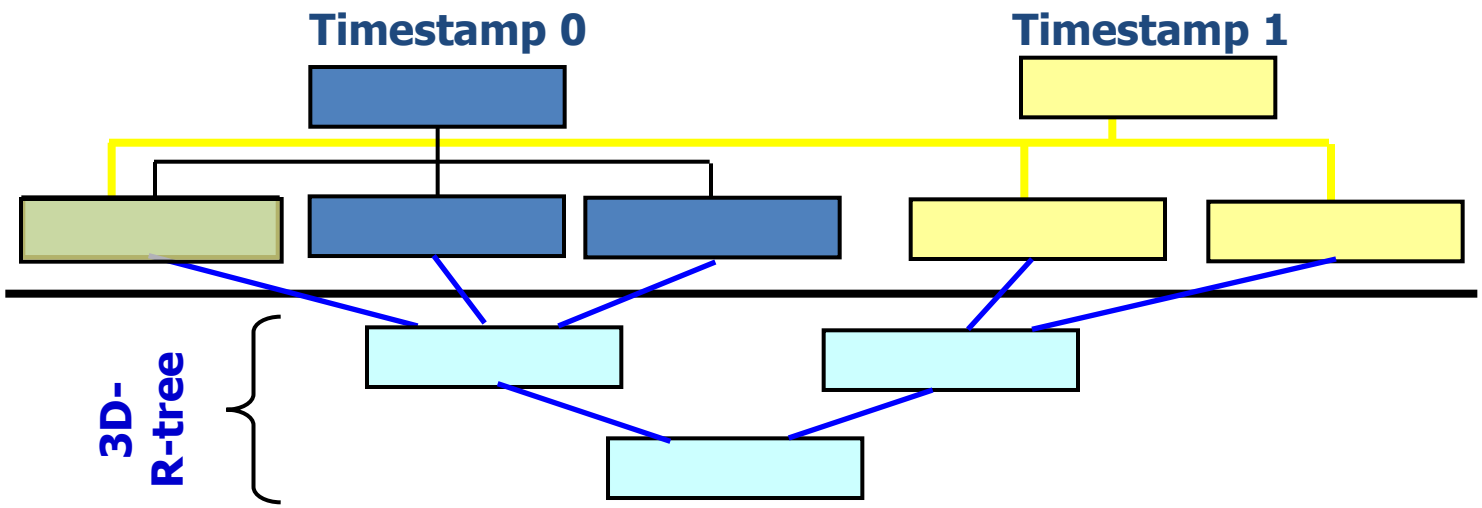
Indexing time dimension

- **Querying Along the Spatial-Temporal Dimension**
  - *What was the location of a certain object from 7:00 AM to 10:00 AM yesterday?*
  - *Find all objects that were in a certain area at 7:00 AM yesterday*
  - *Find all objects that were close to each other from 7:00 AM to 8:00 AM yesterday*
- **Historical trajectories are represented by their three-dimensional Minimum Bounding Rectangle (MBR)**
  - 3D-R-tree is used to index the MBRs
  - Technique simple and easy to implement
  - Does not scale well
  - Does not provide efficient query support



Indexing time dimension

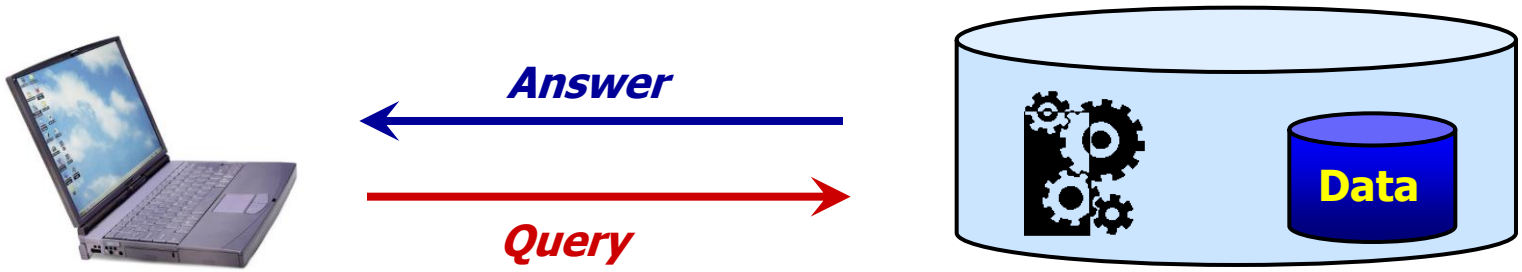
- Multi-version indexing structures: maintain an R-tree for each time instance
- R-tree nodes that are not changed across consecutive time instances are linked together



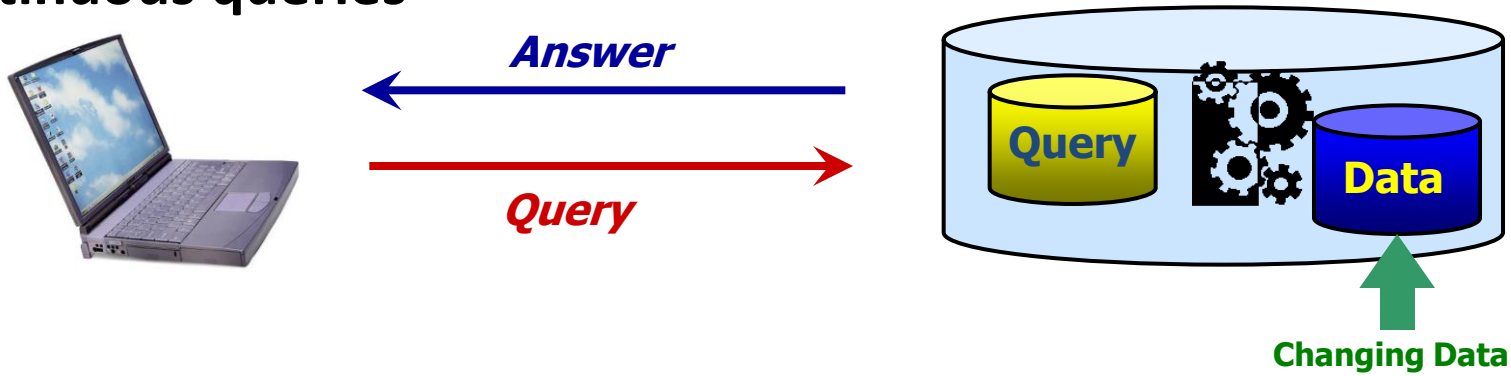
- A multi-version R-tree can be combined with a 3D-R-tree to support interval queries

Continuous query

- **Query the present – time is always NOW**
  - *Find the number of objects in a certain area*
  - *What is the current location of a certain object?*
- **Traditional (Snapshot) queries**



- **Continuous queries**



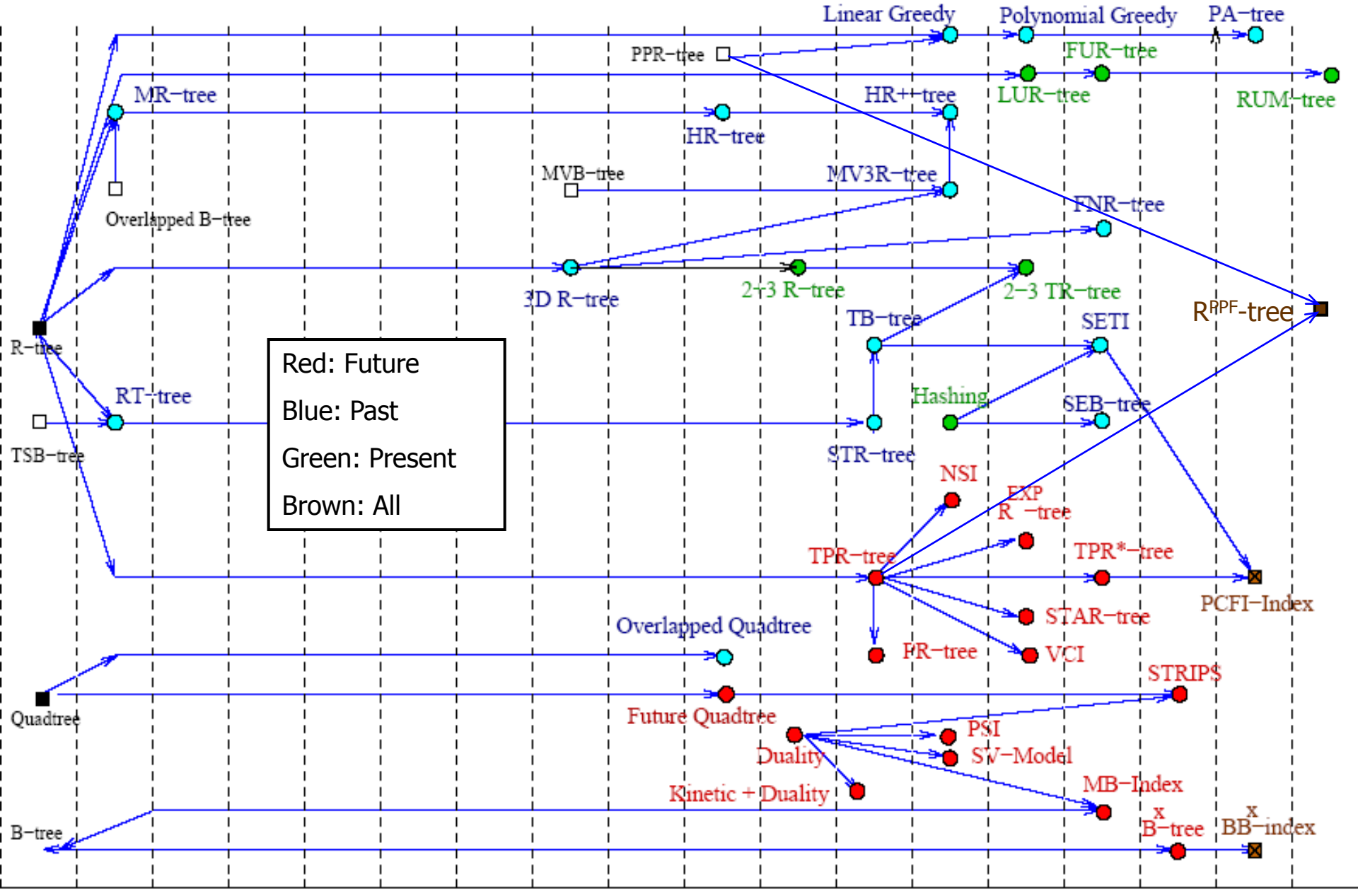
Continuous query

- ***Straightforward Approach***
  - Abstract the continuous queries to a series of snapshot queries evaluated periodically
- ***Result Validation***
  - Valid time (t) and Valid region (r)
- ***Result Caching***
  - Consecutive evaluations of a continuous query yield similar results
  - Upon evaluation, retrieve more data that can be used later
- ***Result Prediction***
  - Based on future trajectory movement, answer can be pre-computed in advance
- ***Incremental Evaluation***
  - Query is evaluated only once, then only updates of the ansqer are evaluated – add or delete objects

Current status of research

- 50+ spatial index structure
- 30+ spatio-temporal indexing structure
- Wide variety of spatio-temporal query processing techniques

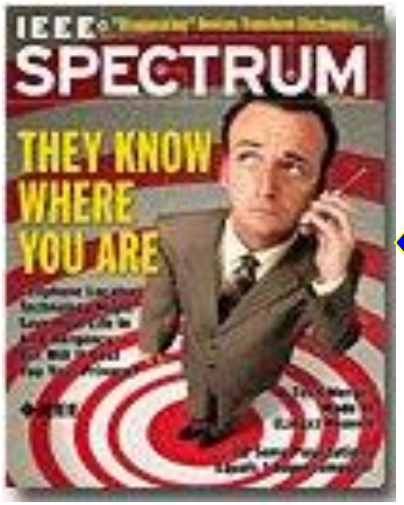
Spatio-temporal access methods



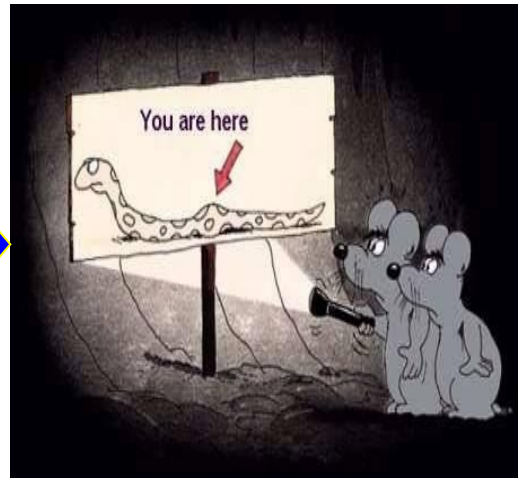
Open Research Issues

Location Privacy



YOU ARE TRACKED… !!!!



"New technologies can pinpoint your location at any time and place. They promise safety and convenience but threaten privacy and security"

Open Research Issues

Spatio-temporal data mining

- Mining the history ➔ Predicting the future
- Online outlier detection for moving objects
- Suspicious movement in video surveillance
- Analysis of tsunami, hurricanes, or earthquakes
- Phenomena detection and tracking