Spatial trees

And the problems they solve
Overview

- What are spatial trees?
- Partitioning on data or space?
- R-tree
- KD-tree
- Alternatives
- Wrap-up
Trees that allow efficient indexing of spatial data (points, lines, polygons) in 2, 3 or higher dimensions.
What are spatial trees?

How would you find your data?
Range queries? \( x_0 < R.x < x_1 \) AND \( y_0 < R.y < y_1 \)
Normal range queries don't work for geometry (only for point data)
How to implement e.g. INTERSECT?
SELECT * FROM R WHERE INTERSECTS(R.polygon, BBOX(x₀, y₀, x₁, y₁));

What are spatial trees?
Partitioning

- Three typical approaches:
  - Use a spatial tree that partitions on data
  - Use a spatial tree that partitions on space
  - Use B-tree with 2D to 1D project (space-filling curve)

- Trees w. data-partitioning:
  - R-tree, Segment tree etc

- Trees w. space-partitioning:
  - K-D tree, Quad-tree etc
R-tree (1984)

http://en.wikipedia.org/wiki/R-tree

- **Core idea**: Index objects by Minimum Bounding Rectangle (MBR)
- **Objects**: Coordinates, Rectangles, Polygons
- **Internal nodes**: Minimum Bounding Rectangle (MBR) of objects in subtree
- **Leafs**: Single spatial object (with MBR)
R-tree (1984)
R-tree is balanced (like B-tree):
- All leaf nodes are at the same height
- Organizes the data in pages
- Designed for storage on disk
- Each page can contain a maximum number ($M$) of entries
R-tree (1984)

Queries: Intersection, containment, nearest-neighbor

Key idea: Use bounding boxes to decide whether or not to search inside a subtree

Big data: Nodes paged to memory when needed

Challenges: Balanced tree, not cover too much empty space, not overlap too much
R-tree (1984)

Challenges:

- Build a balanced tree
- Rectangles should not cover too much empty space
- Rectangles should not overlap too much

Inserting elements (original idea):

- Always insert into the subtree that requires least enlargement of its bounding box
R-tree (1984)

Improve the way the tree is built:
- Building an efficient tree from scratch (bulk-loading)
- Performing changes on an existing tree (insertion and deletion)

Variants:
- $R^+$ tree (1987)
- $R^*$ tree (1990)
- Prioritized R-tree (2004)
**KD-tree (1975)**

- Binary tree over k-dimensional points
- Every node corresponds to a point
- Left child contains points to the left
- Query: Nearest neighbour search
Alternatives to spatial trees

- xD to 1D (and use a B-tree)
  - Space-filling curves
  - Geohash

Z-order curve

A.C. Meyers Vænge 15

http://geohash.org/u3but6uvpt
Wrap up

Many more spatial trees:
- http://en.wikipedia.org/wiki/M-tree

We should be covered :-}
Wrap up

Take away:
- R-tree is the workhorse of spatial databases

Databases that support spatial queries:
- **RDBMS:** PostGIS\%, MySQL\%, Oracle Spatial\%+, SQL Server\$
- **NoSQL:** CouchDB/GeoCouch\%, SpaceBase\%, MongoDB\#

\%: R-tree index
\+: Quad-tree
\$: B-tree + space-filling curve
\#: B-tree + geohash
Final thoughts:

What the gaming industry considers state-of-the-art

What the map industry considers state-of-the-art

Thank you :-) kostas@diku.dk