

Ranking in Spatial Databases

Gisli R. Hjaltason and Hanan Samet

Abhishek Srivastava

Student ID: 861307778

February 21, 2017

CS 236, Winter 2017

The problem:

The paper notes that indexing provides an ordering for the attributes and in general we only do indexing of a particular attribute, So if there is an another attribute present we have to do additional indexing on that as well which is mostly the case of spatial databases where multiple attribute is present and doing additional indexing takes more time and space. It also poses an issue when dimensional unit differs between attributes.

The contribution:

The authors propose an algorithm to find the neighbors with increasing distance from the query point in spatial database environment and it makes the use of hierarchical structure indexes are stored in and it should also decompose the space into blocks, so this method can be used in anyone of the data structure used for indexing like R-tree variants, quadrees and kdb trees.

The method:

The Authors present that there are two ways of storing indexes which also preserve locality between the records: Explicit and Implicit. In explicit method hashing used for mapping similar to space filling curve to store the indexes. Implicit methods use bucketing to store the indexes. Author uses *container blocks* for his algorithm which is similar to minimum bounding region.

Author uses implicit ordering method and proposes two ways to implement:

- **Bottom-Up Approach:** In this approach we first locate the block which contains the query object after that we try to find nearest objects by traversing in clockwise order among adjacent neighboring blocks. In this approach we do not need to start at the root to find the neighboring objects. some drawbacks are it may have to traverse among all the neighboring blocks. It uses pruning in some way to minimize the blocks it has to traverse in recursive fashion.
- **Top-Down Approach:** In this approach we find the leaf node which contains the query point. Then we use recursion to keep track of the blocks and the objects which are already been traversed or encountered. We use priority queue to maintain this. Blocks and objects are also inserted in the priority queue while searching for the leaf node of query point but can be identified if there are visited or not during recursion.

Comments:

The paper presents a simple algorithm which uses hierarchical properties of index storing data structures. This indexing can be used in multiple variety of indexing methods and can be with any-dimensional spatial data. Worst performance is $O(N \log N)$.

However, there are some shortcomings:

- Works on static indexing and it will need to rebuilt if new points are introduced.
- Only works for a query point and does not support bounding regions to find neighbors.