

# Efficient Processing of Spatial Joins Using R-trees

Thomas Brinkhoff, Hans-Peter Kriegel, Bernhard Seeger

Abhishek Srivastava

Student ID: 861307778

February 9, 2017

CS 236, Winter 2017

---

## The problem:

The paper notes that Join operations or multi-scan queries in spatial data are of huge importance but objects have to be accessed multiple times and give quadratic cost. Using conventional join operations such as sort-merge or hash-based on the spatial data are also not efficient because of high disc access and they loose locality information which is very important in spatial data.

## The contribution:

The author propose that we can extend the properties of R-tree class to support spatial join queries and do it in an efficient way by taking advantage of spatial indexing of R-trees. Space filling curves are not used because of high redundancy of references for join operations. The author choose R\*-tree among R-tree variants.

## The method:

The Author present spatial join which used technique similar to pruning on the subtrees of R\*-tree. If rectangle of two directory entries do not overlap which implies that there will be no intersection and can be pruned. So only those rectangle entries are consider who overlap with other entries. This reduces the search space and thus increasing the efficiency of the join operation. Problem with these approaches were: High CPU time consumption and Not consideration of I/O cost.

Authors presents following mechanisms to handle these problems:

- CPU Time Tuning: Search space can be even more confined by sorting the entries according to their spatial location and consider only those entries which intersect the intersection of the rectangle entries. These changes will not affect storage and query performance of R\*-tree. This algo is robust and easy to implement as well. Spatial sorting and Plane sweep improves the tuning much more better and reduces CPU-timing even more.
- I/O Tuning: Spatial partitioning and clustering of R\*-tree is exploited to determine read schedule so that local optimization can be used to reduce the I/O cost by reducing number of disc access. Local plane sweep order, Local 2 order is used to improve I/O tuning. This is a heuristic approach to solve issue to minimize disc access, finding an optimal solution is NP-Hard problem.

One of the main challenges in this paper is the handling R\*-tree with different heights.

## Comments:

The paper presents novel method of using R\*-tree to perform spatial joins in efficient way. Factors considered for evaluation were: CPU and I/O cost, number of disc access. However, it has some sort coming:

- Authors also concur with that the spatial joins are suitable for large problem sizes because for small problem overhead will be much more than the improvement so other methods can be used.
- I/O tuning is a heuristic and may not always give optimal results.