# ALGORITHM FOR SPATIAL JOIN OPERATION

Mr.Jay Bharat Vora
*M.Tech, Computer Science and Engineering*
*National Institute of Technology, Calicut*

***Abstract*—A spatial database system is a database system that supports geographic data types in its implementation and offers spatial data types in its data model and query language. Spatial data are today more common than ever before. A few examples of spatial data are meteorological maps, biological and scientific data, socioeconomic and agricultural data, and geotagged social media. The variety of applications and areas where such data are gathered and processed is growing at a startling and unprecedented rate, as is the volume of geographical data generated every single day. In spatial database systems, the spatial join is a common operation, and evaluating it is a well-researched subject.**

## I. INTRODUCTION

We can distinguish between two different sorts of queries in a database system (DBS). One particular sort of query, known as a single-scan query, only needs to visit an item once, hence the execution time is only ever linear in the quantity of objects kept in the relevant relation. In a spatial database, window and point queries are excellent instances of single scan queries. The other form of query is a "many scan query," which requires accessing an item several times. As a result, execution time is typically not linear but rather super linear in the quantity of objects. The spatial join and the map overlay are the two crucial multiple-scan queries in a spatial DBS.Consider the spatial relations Forests and Cities as an illustration, where an attribute in each relation denotes the boundaries of the two entities, woods and cities, respectively. An illustration of a spatial join is the query that returns all forests that are located within a city.

## II. VARIOUS TYPES OF SPATIAL JOIN TECHNIQUES

| | |
|---|---|
| Internal Memory Methods | 1. Nested Loop Join<br>2. Plane Sweep<br>3. Z-Order |
| External Memory Methods Both Sets Indexed | 1. Hierarchical Traversal<br>2. Non-Hierarchical Method<br>3. Multi-dimensional point methods |
| External Memory Methods One Set Not Indexed | 1. Construct a Second Index<br>2. Index as Partitioned Data<br>3. Index as Sorted Data |
| External Memory Methods Neither Sets Indexed | 1. External Plane Sweep<br>2. Partitioning Algorithms |

## III. ALGORITHMS

### A. Spatial Join Algorithm

The breadth-first search is used by BFRJ (breadth-first R-tree join) to traverse the R-tree for the spatial join. The overlap pair nodes of two R-trees are recorded at each level of the descending tree using a number of IJI (intercede join index) tables. Beginning at the R-roots, tree's BFRJ repeatedly performs a step known as a node-pair join, working its way down the tree level by level until it reaches the leaf nodes.The node-pair join process compares the offspring of a node in one R-tree to the offspring of a different node in the other R-tree in a pair [1]. If a node overlaps another node, the two nodes are recorded in an IJI for the subsequent overlap check. The offspring of leaf nodes, which make up a pair of nodes, are objects rather than nodes. Instead of being saved to IJI, a pair of overlap objects are returned to the users. To facilitate the breadth-first search, a queue, called SJQ, is used to save elements [1].

### B. kNN Query Algorithm

kNN Algorithm is divided into 3 parts

*1) Object Classification:* Object classifiers are objects that indicate their classification for a particular classifier set, and are added to the R-tree by augmenting the node structure with extra data [1].

*2) Distance Classification:* Distance classifiers should generalise very large regions in relation to the amount of the data when used to map a Euclidean distance/increment generator to a specific region. Every sub-regions should utilise the same distance classification as their nearest classified ancestor, and there should be a global distance function [1].

*3) Concentric Circle Query Region:* An initial search circle is generated using a distance classification based on regional characters. When the R-tree is traversed, the algorithm checks whether the current MBB intersects the query circle and that it contains the classifications of the query. If all MBBs have been checked and there are still not k objects found, a function is applied to the current circle to incrementally increase it. If the MBB is contained in the current query circle and not completely contained in the previous query circle, its sub tree is searched and the algorithm can significantly reduce the nodes that it has already searched. The extended search algorithm for kNN on an R-tree takes as parameters the QPOINT, CLASSIFICATIONS, CIRCLE, PCIRCLE, and RESULT. The QPOINT is the center of the search circles, the CLASSIFICATIONS are an array of classifications that represent the query, and the CIRCLE is the current query circle. The RESULT is a functionally global array that holds the sorted array matches found by the algorithm [1].

### *C. Plane Sweep Algorithm*

The two-dimensional plane-sweep of a set of iso-oriented rectangles has two passes: the first sorts the rectangles in ascending order, and the second sweeps a vertical scan line from left to right. The key to the algorithm is its ability to keep track of the active rectangles and perform the intersection test.[2] The plane-sweep algorithm uses three operations to track active rectangles: INSERT, REMOVE INACTIVE, and SEARCH. A spatial join determines all pairs of intersecting rectangles in A and B, while a plane-sweep algorithm applies a sweep structure for both A and B. The sweep structures' implementation data structure may significantly affect how well they perform as measured by their performance studies. For smaller data sets, where the overhead of more complex structures is unnecessary, it is suitable to choose a block list structure (many objects in each list entry). More complex structures are ideal for larger data sets or data sets that are heavily skewed[2].

```
procedure PLANE_SWEEP(setA, setB);
begin
  listA←SORT_BY_LEFT_SIDE( setA );
  listB←SORT_BY_LEFT_SIDE( setB );
  sweepStructureA←CREATE_SWEEP_STRUCTURE();
  sweepStructureB←CREATE_SWEEP_STRUCTURE();
  while NOT listA.END() OR NOT listB.END() do
    /* get left most rectangle from the two lists */
    if listA.FIRST() < listB.FIRST() then
      sweepStructureA.INSERT(listA.FIRST());
      sweepStructureB.REMOVE_INACTIVE(listA.FIRST());
      sweepStructureB.SEARCH(listA.FIRST());
      listA.NEXT();
    else
      sweepStructureB.INSERT(listB.FIRST());
      sweepStructureA.REMOVE_INACTIVE(listB.FIRST());
      sweepStructureA.SEARCH(listB.FIRST());
      listB.NEXT();
    end if;
  end loop;
end;
```

Fig. 1: Plane Sweep Algorithm

### *D. Partition Based Spatial Merge (PBSM)*

PBSM algorithm operates in the following two steps.

1) **Filter Step** : The PBSM algorithm uses an approximation of the spatial feature to get a "rough estimate" of the characteristics of the spatial attribute, partitioning large inputs into smaller chunks, and using a computational geometry plane-sweeping technique to join the chunks.
2) **Refinement Step** :As the filter step "joins" the inputs based on the MBR of the joining attributes, it is possible for two non-overlapping spatial features to have overlapping MBRs. As a result, the filter step typically produces a superset of the joint outcome. The R and S tuples that are represented by the OID pair created in the previous step are retrieved from disc in the refinement stage, and their join attributes are checked to see if the join predicate is indeed satisfied.

## IV. FUTURE SCOPE

There are many researches going on how to reduce the join time. One proposed solution is using Machine Learning to reduce the join time. One such algorithm is Spatial Join Machine Learning (SJML). Model for distributed spatial join using machine learning that estimates selectivity and costs. Because it allows spatial join, it varies from other ML-based query optimizers. This is applicable to any input data that wasn't used during training. Additionally, because it supports skewed data, including real datasets, and operates on clearly stated data statistics that can be calculated with a straightforward data scan, it solves the shortcomings of previous theoretical spatial join models.[3].

## REFERENCES

[1] K. Bhima, T. A. Sri, K. D. Ramaiah, and A. Jagan, "Exerting spatial join and knn queries on spatial database," in *2012 International Conference on Recent Advances in Computing and Software Systems*. IEEE, 2012, pp. 260–266.

[2] E. H. Jacox and H. Samet, "Spatial join techniques," *ACM Transactions on Database Systems (TODS)*, vol. 32, no. 1, pp. 7–es, 2007.

[3] T. Vu, A. Belussi, S. Migliorini, and A. Eldawy, "A learned query optimizer for spatial join," in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 2021, pp. 458–467.