

# Size Separation Spatial Join

Nick Koudas and Kenneth C. Sevcik

Abhishek Srivastava

Student ID: 861307778

February 16, 2017

CS 236, Winter 2017

---

## **The problem:**

The paper notes that previous methods for doing spatial join operations often needed to replicate the input datasets or intermediate results. Replicating data often leads to storing more data on the disc or in the main memory, therefore it leads to more I/O operations and more disc block reads. Paper also points out that existing join methods were not able to support few predicates if involved in join operations.

## **The contribution:**

The authors propose a new algorithm to perform spatial join of multiple datasets with no indexes. The algorithm is extension of Filter Tree and includes dynamic hierarchical decomposition of the space to store the spatial data with no repetitions. The algorithm proposed by the authors in some way extends relation join as well by generalizing relational *Sort Merge Join* for multidimensional data which makes its implementation easier.

## **The method:**

The Authors present size separation spatial join which constructs *Filter Tree* dynamically and does not build whole indices. It uses this property to capture large entities at higher levels and small entities as lower level.

Filter tree captures two property for each entity:

- Hilbert Value: Hilbert value of the center of Minimum Bounding Region.
- Level of Filter Tree:  $\text{Level}(x_l, y_l, x_h, y_h)$  represents at which level of the filter tree entity resides.

Level for each entity in all datasets is computed and partitioned by keeping data of same level with each other.  $L+1$  pages needed to store this information, where  $L$  is the max number of levels in the filter tree. Then sorting is performed for each level file of each dataset so that Hilbert values of entities are monotonically increasing and after that join operation is performed by scanning over all pages at all level, reading one page only once. This algorithm can take advantage by reading each page only once and can operate join operation on that data. The partition of data into hierarchy mainly depends on the size of the data.

Performance of this algorithm can be increased by applying filtering like other existing algorithms which reduces the number of unnecessary operations performed by the join operations. This algorithm uses Dynamic Spatial Bitmaps to perform filtering which is similar to relational bitmap indices join. Size of bitmap depends on the size of level chosen, but there is a trade-off between the size of bitmap and its effectiveness.

## **Comments:**

The paper presents an extension of Filtering tree and generalizing it for the multidimensional data and can be performed on data with no indexes. This methods compare its results with Partition based spatial merge joins and spatial hash joins over different distribution of data with different coverage and it outperforms others.

However, it has some short comings:

- If the data is very large with many small partition calculation of hilbert center and level can be overhead .
- Does not perform well in case of skewed spatial data.