Spatial Joins Using Seeded Trees

Ming-Ling Lo and Chinya V. Ravishankar

Abhishek Srivastava
Student ID: 861307778

February 14, 2017
CS 236, Winter 2017

The problem:
The paper notes that Join operations for tree like index based algorithms were optimized for spatial selection rather than spatial join operation thus performing badly due to large number of random disc access for such operations. Also tree based indexing required pre-computed indexes of whole data to operate and holding the indexes for large trees can be substantially big and is not optimal. Storing indexes also does not help in non-spatial queries and multiple join operations.

The contribution:
The author propose to utilize the information present during spatial join such as size of input data sets and their spatial attributes to dynamically construct index trees called seeded trees. This supports queries to dynamically indexes for the derived data sets as necessary to support spatial join.

The method:
The Author present seeded tree which is optimized for join operation and low construct costs. It has few working assumptions as well: R-tree exists for at least one dataset & seeded tree is generated at run time using R-tree from the other dataset. Seeded tree is divided into two parts: Seed levels and Grown levels.

Authors presents following phases to construct a seeded tree:

- Seeding Phase: Top k levels are copied from R-tree. Seed nodes can either store tree minimum boundary or center points of minimal bounding boxes. These nodes values may be changes later during data insertion but the structure of seed levels never changes.
- Growing Phase: Data is inserted at slot level which results in R-tree subtrees, it follows all the properties of R-tree like node splitting etc. Criterion for selecting child node depends whether it has stored an area or central points of minimal bounding box. Multiple update policies are provided by authors.
- Cleanup Phase: This process is done after all data is inserted in seeded tree. It adjusts the bounding boxes if required to make relevant data consistent and deleting empty slots.

Tree matching is performed after these phases to produce join results. Intermediate linked lists can be used during growing phase to increase the construction cost and reduce the size of seeded trees.

Comments:
The paper presents novel method of using seeded tree to perform spatial joins in efficient way. Factors considered for evaluation were: Total Cost, Construction Cost and Matching Cost against various data set sizes and different degree of spatial clustering. It gave significant performance win over other methods(R-tree Join, Brute Force Join).

However, it has some short comings:

- Tree matching algorithm to find join result can be improved .
- How to handle cases when both data sets to join do not have an R-tree.