Spatial Hash-Joins

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 Hash based Join operations always outperform Tree based Join operations because of the few number of disc access and works well for the operations where relations are quite large compared to available memory. Existing hash-join methods can be extended to integrate changes for spatial join and also its costs depends on input data sizes which can be easily estimated. Paper also discusses the difficulties for applying hash-join to spatial joins and need some modifications.

The contribution:
The authors propose a new framework for spatial hash join operations. The methods proposed are extended version of relation hash joins. This method does not need pre-computed indexes and are applicable to different kind of spatial join operations. The authors provide reasons for applying difficulties with directly applying hash join operation such as it does not preserve spatial locality and thus need some modification.

The method:
The Authors present spatial hash join and provide a new framework to implement spatial hash joins and it has two components: set of buckets extents and assignment function.

Authors provide a framework which is used to do the assignment to the buckets and it has two Phases:

- Partition Phase: Using spatial partition functions it places inner and outer dataset objects into buckets.
- Join Phase: To obtain results it joins corresponding inner and outer buckets.

The partion of input datasets in bucket extents must be done approximately into equal-sized buckets and is very crucial for performance. Assignment criterion is selected to map bucket extents with the data object and number of buckets it can be assigned to. If proper assignment criterion is not used it can result in imbalanced buckets, hence poor performance. Authors present different assignment criterion's for inner and outer datasets considering bounding box as data objects.

Inner Bucket extents are choosed using bootstrap seeding technique and based upon these dataset is partitioned. This reduces the total area and overlap area as well. Outer bucket extents are immutable and partition is done by matching overlap with inner buckets, if no overlap is found it is discarded. The remaining buckets are then joined to produce result.

Comments:
The paper presents a novel Spatial Hash Join method which outperforms tree based spatial join operations for both pre-computed and dynamically generated indices for input datasets. This method can be used for the data where there is no pre-computed indexing is present.Experiments performed against Seeded tree join and R-tree join for different parameters such as Dataset Size, Spatial Clustering, Buffer size.

However, it has some short comings:

- Performs badly when inner dataset is larger than outer relations.
- Does not perform good when Minimum bounding region has to be calculated in run time from raw data.