Models and Optimization Methods of Spatial Query on Multi-Platforms

Uses of spatial data sets have been increased exponentially. Due to huge amount and complex structure of spatial data, the optimization of spatial query becomes a difficult and important from research perspectives. Authors have focused on the queries based on spatial characteristics. The spatial query models of the following platforms are analyzed.

- **Query Model of Oracle Spatial**: It provides a SQL model and functions to store, retrieve, update and query the spatial data sets in the Oracle Database. Oracle Spatial using two-tier query model and it follows primary filtering and secondary refinement operation.
- **Query Model of ArcSDE**: It is a tool that allows users to store and manage spatial data in their chosen relational database management system.
- **Query Model of OGR/GEOS**: These are the libraries which implement spatial predication functions. These are the functions can help us implement the spatial query based on two spatial tables. There is as such no query model and developers are given freedom to implement the spatial queries.

Authors have analyzed the indexed spatial join method based on and have proposed the 2FSJM optimization approach for improving the performance of multi-platforms. The first level filtration is the Minimum Boundary Rectangle of spatial table, or layer and the second level filtration is the MBRs of geometries. Authors have also conducted experiments which show that their method improves the spatial query performance for these multi platforms.

Efficient Query Processing on Spatial Networks

Jagan Sankaranarayanan, Houman Alborzi, and Hanan Samet

Center for Automation Research, Institute for Advanced Computer Studies Department of Computer Science, University of Maryland.

The authors have addressed special queries from special networks perspectives. They have proposed a new framework (SILC) for efficient processing of spatial queries on spatial networks. Another strong statement made by authors is that it is possible to pre compute the shortest path and distance between all pairs of vertices in a spatial network, which is otherwise, quite an expensive method. The third contribution of the paper is the concept of progressive refinement of the inter-object distances between objects in a spatial network.

The authors have successfully tested their approach by performing three experiments by taking random samples from a large road-network dataset.

Top-k Spatial Joins

Manli Zhu, Dimitris Papadias, Jun Zhang, and Dik Lun Lee

Joins are involved in several types of spatial database queries. Even after their heavy usage, the cost involved in join queries is quite high and needs effective performance optimization. Generally the algorithms proposed for this optimization give a large output. In several applications, however, the user may be interested in only a small subset of the output. This paper focuses on such query types, called the top-k spatial join. Authors not only propose a TOP-k SPATIAL JOIN ALGORITHM (TS) which is based on the branch and bound paradigm but have applied two optimization techniques to improve the efficiency of this algorithm.
Authors have also successfully evaluated the proposed algorithms in terms of node/page accesses and CPU cost.

**Threshold Query Optimization for Uncertain Data**

In this paper, authors address the optimization problem of general SQL-based probabilistic threshold queries that involve selections, projections and joins. PTQ are the queries where all results satisfying the query with probabilities that meet the threshold requirement are returned. Examples of such queries are nearest neighbor and range queries etc. Although threshold queries have been studied in various settings for probabilistic data, this paper is focused on optimizing threshold queries for basic database operations, such as selections, projections and joins.

Experimental validation is used for proving the efficiency of these queries. They have used two datasets in this. One is a real data set for attributes with continuous uncertainty, and the other is a synthetic data set for attributes with discrete uncertainty. Authors have also suggested some optimization rules to reduce the query processing time through experiments on both of these data sets.

**Spatial Join Techniques**

This is a survey paper in which authors not only give the literature survey but analyze in detail the methodology of different joining techniques. They examined different techniques using nested-loop joins, indexed nested-loop joins, or variants of the plane-sweep technique. Authors have taken different cases and have made recorded as to which technique should be used in which case. For e.g "If both datasets are indexed, but with incompatible types of indices, such as an R-tree [Guttman 1984] and a point quadtree [Finkel and Bentley 1974], Corral et al. [1999] suggest ignoring one index and performing an indexed nested-loop join". One of the points which author throws light upon is that many other spatial join techniques are also there which could not be implemented because of space limitations. Therefore it is always a tradeoff. Authors have also given a brief idea of top – k joins.

The intention of this survey is not to judge which technique is superior for which type of scenario but to provide an idea as to what methods works best in which situation.

**Chapter Organizations**

**Chapter 5: Query Optimization and Processing**

- **Extension of book section 5.1**: We plan to add new algorithms for spatial join from the papers “Spatial Join Techniques” and “Models and Optimization Methods of Spatial Query on Multi-Platforms”. We can also add the concept of top k spatial queries and joins.
- **Addition of new section 5.6**: We plan to add the Query models of various spatial platforms like Oracle Spatial, ArcSDE etc. We feel that by going through a few examples from functional spatial systems will help in understanding behind the scene query execution framework. We can move the summary section to 5.7.
5. Query Processing and Optimization

5.1 Evaluation of Spatial Operations

5.1.1 Overview
5.1.2 Spatial Operations
5.1.3 Two-Step Query Processing of Object Operations
5.1.4 Techniques for Spatial Selection
5.1.5 General Spatial Selection
5.1.6 Algorithms for Spatial Join-Operations
5.1.7 Strategies for Spatial Aggregate Operation: Nearest Neighbour

5.2 Query Optimization

5.2.1 Logical Transformation
5.2.2 Cost-Based Optimization: Dynamic Programming

5.3 Analysis of Spatial Index Structures

5.4 Analysis of Spatial Index Structures

5.5 Parallel Spatial Database Systems

5.5.1 Hardware Architectures
5.5.2 Parallel Query Evaluation
5.5.3 Application: Real-Time Terrain Visualization

5.6 Summary
5.3.1 Enumeration of Alternate Plans
5.3.2 Decomposition and Merge in Hybrid Architecture

5.4 Distributed Spatial Database Systems
5.4.1 Distributed DBMS Architecture
5.4.2 The Semijoin Operation
5.4.3 Web-Based Spatial Database Systems

5.5 Parallel Spatial Database Systems
5.5.1 Hardware Architectures
5.5.2 Parallel Query Evaluation
5.5.3 Application: Real-Time Terrain Visualization

5.6 Query Models of Spatial Platforms
5.6.1 Oracle Spatial
5.6.2 ArcSDE
5.6.3 OGR/GEOS

5.7 Summary