Volume 2, Issue 4, April 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Comprehensive Study on Spatial Query

Mr. Shaik Salam[1] Dr. T Venkateswar Rao[2]
M.V.R Maneesha[3], Mohammed Sajjad Shaik[4]
[1]Associate Professor, Dept of CSE, Sree Vidyanikethan Engg. College, A.Rangempet
[2]Professor, Dept of CSE, K.L University, Vijayawada
[3,4]M.Tech Student, Dept of CSE, Sree Vidyanikethan Engg. College, A.Rangempet,
email:maneesha.mrc4@gmail.com

ABSTRACT: Spatial query is the core function of GIS analysis, the query data is mostly geometric point, line, polygon which has the location information and formatted query condition is a variety of spatial relationships. In spatial query it has been changing the query process to give efficient results to solve real world applications. This survey focuses on categorizing and reviewing the current progress on spatial query and their metrics and method.

KEYWORDS: GIS, Spatial query, ANN, GNN, RNN.

1. INTRODUCTION:

In the last decade we have different types of query processing are taken place in spatial data.Spatial Queryinvolves selecting features based on location or spatial relationships, which requires processing of spatial information. For instance a question may be raised about parcels within one mile of the freeway and each parcel.

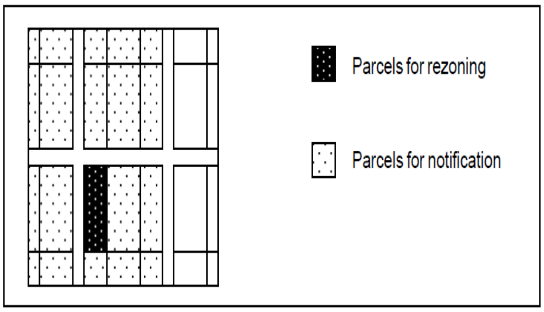

Fig 1: Land owners within a specified distance from the parcel to be rezoned identified through spatial query

In this case, the answer can be obtained either from a hardcopy mapor by using a GIS with the required geographic information as shown in fig1.

Let us take one spatial query example where a request is submitted for rezoning, all owners whose land is within a certain distance of all parcels that may be rezoned must be notified for public hearing. A spatial query is required to identify all parcels within the specified distance. This process cannot be accomplished without spatial information. In other words, the attribute table of the database alone does not provide sufficient information for solving problems that involve location.

Spatial query processing has been put forth on Nearest Neighbour (NN) queries which play an important role in decision making. For instance, a tourist information system supporting nearest neighbour queries may assist tourists to find the nearest attractive points of their interest. In the same way all-nearest-neighbours queries, range nearest-neighbour query, group nearest neighbour queries, nearest surrounded queries.

2. NEAREST NEIGHBOUR QUERIES:

An efficient branch-and –bound R-Tree Traversal algorithms to find the nearest neighbour object to a point and then generalize it to find the k-nearest neighbours.In this using a R-trees data structure as shown in figure 2(b) for a spatial data collection as shown figure 2(a)[2].R-trees is a extension of B-Tree in higher than one dimension. They combine most of the nice features of both B-Tree and quad trees.Metrics are 1) MINDIST is a minimum Distance of the object O from P and 2) MINMAXDIST is a minimum of the maximum possible distances from P to a face of the MBR (minimal bounding rectangle) containing O.This algorithm is an ordered Depth first traversal. It begins with the R-tree root node and proceeds down the tree.

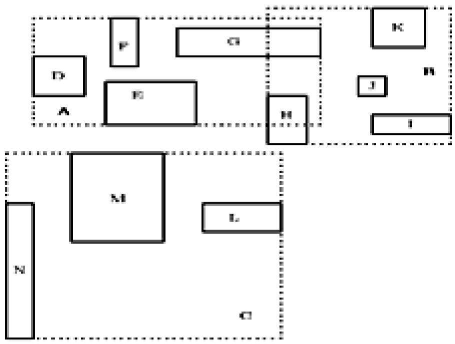

Fig 2(a): Collection Build

During the descending Phase, at each newly visited non-leaf node. The algorithm computes the ordering metric bound for all Its MBR's and sorts them into active Branch list. We than apply pruning strategies to ABL to remove unnecessary branches. The algorithm iterates on this ABL, until the ABL is empty. The first metric (MINDIST) Produces the most optimistic ordering possible;whereas the-

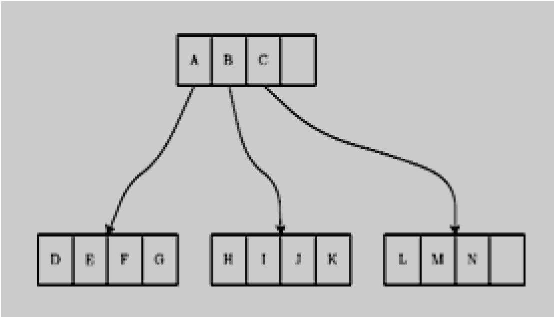

Fig 2(b): The R-Tree

second metric (MINMAXDIST) produces the most pessimistic ordering that ever need be considered. The both metrics were to be valuable tools in effectively directing and pruning the nearest neighbour search.

3. ALL-NEAREST-NEIGHBOUR QUERIES:

ANN queries constitute a hybrid of nearest neighbour search and spatial joins.We already given about NN and Spatial join is the spatial join between two datasets A and B finds the object pairs in the Cartesian product $A \times B$ which satisfy a spatial predicate, most commonly *intersect* (assuming the datasets contain objects with spatial extent)[3]. Depending on the existence of indexes, different spatial join algorithms can be applied. The *R-tree algorithm*, proposed in [2], computes the spatial join of two inputs indexed by R-trees. RJ synchronously traverses both trees, starting from the roots and following entry pairs which intersect.

ANN is simply defined as follow,Given two sets A and B of multidimensional objects, the all-nearest-neighbours (ANN) query retrieves for each object in A its nearest neighbour in B.

Index based ANN method the main aim to reduce the processing cost .It having several methods,they are a) the*multiple nearest neighbour* (MNN) algorithm can be considered as the counterpart of index-nested-loops in relational databases. In particular, MNN applies NNqueries (one for each point in A) on R-tree. The order of the NN queries is very important since if two consecutive query points (of A) are close to each other, a large percentage of pages from R-Treeneeded during the second query will be in the LRU memory buffer due to the first one.

b) In order to reduce the high computational cost of MNN, we overcome by a*batched* NN (BNN) method, which retrieves the nearest neighbours for multiple points at a time. BNN splits the points from A into a number of n groups G_{A1}, $G_{A2}\ldots G_{An}$, such that $UG_{Ai}= A$ and $\forall_{i,j}$, $1 \le i < j \le n$; $G_{Ai} \cap G_{Aj} = \emptyset$. Then for each group, rootis traversed only once, reducing considerably the number of distance computations.

ANN on non- Indexed method that is a hash-based ANN algorithm.It having two phase (i) hashes the points from A and B in spatial partitions, (ii) loads pairs $<H_A, H_B>$, $H_A \in A$, $H_B \in B$ of buckets covering the same region and searches for each $a_i \in H_A$its NN in H_B. In order to distribute the points evenly to the hash buckets and using spatial hashing method. A fine regular grid that contains more cells than the number of hash buckets is used to partition the points. Each bucket contains multiple tiles (i.e., grid cells) at different areas of the space. Since $H_A$and $H_B$cover the same area, the NN of each point $a_i \in H_A$is likely to be in H_B.

4. RANGE NEAREST NEIGHBOUR QUERIES:

A range nearest-neighbour (RNN) query retrieves the nearest neighbour (NN) for every point in a range[4]. It is a natural generalization of point and continuous nearest-neighbour queries and has many applications. In this we consider the ranges as rectangles and efficient in-memory processing and secondary memory pruning techniques for RNN queries in both 2Dand high-dimensional spaces. These techniques are generalized for kRNN queries, which return the k nearest neighbours for everypoint in the range.

The algorithm taking a input as a query range and output as a candidate setof RNN. The procedure involved in this RNN is using BFS paradigm in that a priority queue. The process run until queue is empty, in that we are checking a minimum distance between candidate and range is greater than or equal to maximum distance between a nodes to range. If condition is true then we performing the face-based pruning and call to the same function with a parameter as new candidate set.

5. GROUP NEAREST NEIGHBOUR QUERIES:

Given two sets of points P and Q, a group nearest neighbour (GNN) query retrieves the point(s) of P with the smallest sum of distances to all points in Q[5]. Consider, for instance, three users at locations q_1, $q_2$and $q_3$that want to find a meeting point (e.g., a restaurant); the corresponding query returns the data point p that minimizes the sum of Euclidean distances $|pq_i|$ for $1 \le i \le 3$.

GNN queries are processing by three algorithms. They are

(1) **Multiple query method**:The *multiple query method* (MQM) utilizes the main idea of the *threshold algorithm*, i.e., it performs incremental NN queries for each point in Q and combines their results. For instance, Consider Q having {$q1$, $q2$}), MQM retrieves the first NN of $q1$ and compute the distance between p to $q2$. Similarly, it finds the first NN of $q2$. The point with the minimum sum of distances to all query points becomes the current GNN of Q.

(2)**Single point method**:MQM may incur multiple accesses to the same node through different queries. To avoid this problem, the *single point method* (SPM) processes GNN queries by a single traversal. First, SPM computes the *centroid* q of Q, which is a point in space with a small value of *dist*.(q,Q) (ideally, q is the point with the minimum *dist*.(q,Q)). The intuition behind this approach is that the nearest neighbour is a point of P "near" q. It remains to derive (i) the computation of q, and (ii) the range around q in which we should look for points of P, before we conclude that no better NN can be found.

(3)**Minimum Bounding Method**: The minimum *bounding method* (MBM) performs a single query, but uses the minimum bounding rectangle *M* of *Q* (instead of the centroid *q*) to prune the search space. Specifically, starting from the root of the Rtree for dataset *P*, MBM visits only nodes that may contain candidate points.

6. NEAREST SURROUNDER QUERIES:

Nearest Surrounder (NS), which searches the nearest surrounding spatial objects around a query point [6]. NS query can be more useful than conventional nearest neighbour (NN) query as NS query takes the object orientation into consideration. To address this new type of query, we are using angle-base bounding properties and distance-bound properties of Rtree index. The former has not been explored for conventional spatial queries. With these identified properties, we are using two algorithms, namely, Sweep and Ripple. Sweep searches surrounders according to their orientation, while Ripple searches surrounders ordered by their distances to the query point. Both algorithms can deliver result incrementally with a single dataset lookup.

A new class of spatial queries is the Nearest Surrounder (NS) queries and its variant, the multi-tier NS queries (*m*NS). These queries, searching the nearest spatial objects surrounding a query point, are different from the conventional NN queries which have been extensively studied in the past decade. It having angle-based bounding properties and different search strategies. Based on these we are using *Sweep* and *Ripple* algorithms.

An NS query can be executed as a collection of finer NS queries that search surrounds in disjointed angular ranges. This allows us to employ appropriate algorithms in different angular ranges based on different object distribution.

7. CONCULSION:

This paper surveys the various spatial query,explains how it will processed on a given query to reduce the cost.In this survey we have different type of spatial query techniques to solve the different real world applications.From this study, a basic method NN query is involved in many query types and it is simple to develop.Further it can be extended in perspective of various spatialqueries that can be applied on database.

REFERNCES:

[1] N. Roussopoulos, S. Kelly, and F. Vincent. “Nearest NeighbourQueries” In *Proceeding of the 1995 ACM SIGMOD Conf.,San Jose, CA, May 22-25*, pages 71–79, 1995.

[2] N. Backmann, H.-P. Kriegel, R. Schneider, and B. Seegar. ,“The R*-Tree: An Efficient and Robust Access Method forPoints and Rectangles”, In *Proceedings of the ACM SIGMOD Conf., Atlantic City, NJ, May 23-25*, pages 322–331, 1990.

[3] Jun Zhang, Dimitris Papadias, Nikos Mamoulis, Yufei Tao, “All-Nearest-Neighbours Queries in Spatial Databases”, Procd. 16th intl. conf. on database mange, June, 2004.

[4]D.Papadias, Qiongmao Shen,Yufei Tao, Kyriakos Mouratidis, “Group Nearest Neighbour Queries”,procd. Of the 20th Intl. Conf. On Data engg.,2004.

[5]Haibo Hu and Dik Lun Lee, “Range Nearest-Neighbour Query” IEEE Trans. onknowledge and Data Engg., vol. 18, no. 1, Jan,2006.

[6] Ken C.K. Lee, Wang-Chien Lee, Hong Va Leong, “Nearest surrounder Queries”, IEEE Trans. on knowledge and Data Eng., vol. 22, no.10, Octo, 2010.

