International Journal of Engineering Research and Development
***e*-ISSN: 2278-067X, *p*-ISSN: 2278-800X, www.ijerd.com**
Volume 3, Issue 5 (August 2012), PP. 35-40

Spatial Data Ranking For Selected Location Using Quality of Features

R.Krishnaraj[1], M. Sathiamoorthy[2], A.Akila[3], G.Karthik[4]
[1,3,4]*Final Yr M.Tech, Department of Computer Science and Engineering, Christ College of Engineering and Technology, Pondicherry, India.*
[2]*Department of Computer Science, Pondicherry University, Puducherry-14, India.*

Abstract- Spatial database management system (SDBMS) contains spatial data in space and provides special storage for handling the spatial data. With the perception of users attraction towards few best objects rather than a large list of best objects as a result, in this paper, we propose an approach to generate top k ranking of spatial data for selected location based on quality of features. Search Algorithm and Enhanced Branch & Bound Algorithm have been used to achieve this. Search Algorithm is used for searching specific set of objects. Enhanced Brand & Bound Algorithm takes input as search result of search algorithm to produce top most objects according to the quality of features (hotel, school, hospital, and market) for searched objects. For example, consider a real estate database, in that customers want to search particular location with top k flats in accordance to features. The top most flat for selected location with good quality is obtained by using the above algorithms. The experimental results would prove the efficiency of this proposed work.

Keywords— spatial databases, search algorithm, branch and bound algorithm, location, rank, real estate.

I. INTRODUCTION

A spatial database is a database that is optimized to store and query data that is related to objects in space, including points, lines and polygons. While typical databases can understand various numeric and character types of data, additional functionality needs to be added for databases to process spatial data types [1]. In addition, spatial databases provide structure for storage and analysis of spatial data. The spatial data is comprised of objects in multi-dimensional space. Storing spatial data in a standard database would require excessive amounts of space. Queries to retrieve and analyze spatial data from a standard database would be long and cumbersome task. Spatial databases provide much more efficient storage, retrieval, and analysis of spatial data [2].

The term "Spatial Database" refers to a database that stores data for phenomena on, above or below the earth's surface or in general, various kinds of multidimensional data of modern life. In a computer system, these data are represented by points, line segments, regions, polygons, volumes and other kinds of 2D/3D geometric entities and are usually referred to as spatial objects. Spatial databases include specialized systems like "Geo graphical Information Systems", "CAD databases", "Multimedia databases", "Image databases", etc. The role of spatial databases is continuously increasing in many modern applications during last years. Mapping, urban planning, transportation planning, resource management, geo marketing, archeology and environmental modeling are just some of these applications.

The key characteristic that makes a spatial database a powerful tool is its ability to manipulate spatial data, rather than simply to store and represent them [Güt94]. The most basic form of such a manipulation is answering queries related to the spatial properties of data. Some typical spatial queries are the following. A "Point Location Query" seeks for the spatial objects that fall on a given point. A "Range Query" seeks for the spatial objects that are contained within a given region (usually expressed as a rectangle or a sphere). A "Join Query" may take many forms. It involves two or more spatial datasets and discovers pairs (or tuples, in case of more than two datasets) of spatial objects that satisfy a given spatial predicate. The spatial distance join was recently introduced in spatial databases to compute a subset of the Cartesian product of two spatial datasets, specifying an order on the result based on distance.

The spatial data is divided into two types. They are Point Data and Region data. The point data are the Points in a multidimensional space, E.g., *Raster data* such as satellite imagery, where each pixel stores a measured value. The Region Data are the Objects have spatial extent with location and boundary, E.g., Road network. Spatial database system can be used in variety of applications such as Computer aided Design and geo-application such as agriculture, banking, real estate, road networks, etc.

In this project, the spatial database is the real estate database system application and the spatial data are the spatial attributes (features such as school, hospital, market, hotel, etc.). As the user looking more interested in few best objects rather than a huge list of best objects, we propose top k ranking of spatial data for selected location query to retrieve few best objects. In paper [4], the Top-k spatial preference queries provide top most best data objects based upon feature score of objects in the spatial neighborhood. In which the feature score values can be obtained by a rating provider.

The rest of the paper is organized as follows. In Section II, the literature survey of this work has given. Section III presents the SDBMS architecture followed by formal definition of this problem in Section. IV. Section V presents the algorithm used in this paper. Section VI contains an extensive experimental evaluation and section VII concludes the paper with directions for the future work.

II. LITERATURE SURVEY

The top-k queries produce ordered result by using some calculated score. Generally, users are interested in top-k join result. For this, the top-k queries require joins to produce top-k result. The relational processors should not process the ranking queries with join efficiently. To solve this type of queries, many of the researchers proposed several methods.

In 2004, F. Ilyas et.al.,[5] introduced a new rank-join algorithm that made use of the individual orders of its inputs to produce join results ordered on a user-specified scoring function. They had experimentally evaluated their proposed rank join operators and analyze its performance. In 2006, Rank-aware query optimization framework [6] by Ihab F. Ilyas et.al. fully integrated the rank-join operators into relational query engines and shown the performance of the proposed framework. In 2007 Yiu et.al., [4] proposed the Branch and bound (BB) and Feature join algorithm (FJ) that rank objects based on the qualities of features. They proved that their proposed work is better than simple and Group probing algorithms with real and synthetic data.

In 2011, Joao et.al.,[7] proposed a mapping of pairs of data and feature objects to a distance-score space, which made us to identify and materialize the minimal subset of pairs that is sufficient to answer any spatial preference query. They had done an extensive experimental evaluation over their algorithm which outperforms the state-of-the-art algorithms in terms of I/Os and execution time. In 2011, Yiu et.al.,[8] proposed the Object Influence score algorithm and Upper Bound Computation algorithm that rank objects based on the combination quality of features using influence score. They had evaluated the performance of the proposed algorithms with the previous algorithms such as simple, Group probing, Branch & bound and feature join. Related to existence, we propose an enhanced Branch and Bound algorithm for selecting top k query results from the selected spatial data base. In order to get the result, the spatial data is indexed with r-tree index. This indexing method is used in search and enhanced branch & bound algorithms to solve the proposed query.

In this paper, to achieve efficient search, we propose a new technique in which the output of search algorithm is given as input to the enhanced branch and bound algorithm. These algorithms produce top k objects for selected location with highest score and feature points are calculated by user rating obtained from rating provider.

III. SPATIAL DATABASE MANAGEMENT SYSTEM ARCHITECTURE

Spatial database management system **(**SDBMS) is a Three Layer Architecture as shown in Fig.1. SDBMS works with a spatial application at the front end and a DBMS at the back end [2].

- **SDBMS has three layers:**
  - Interface to spatial application
  - Core spatial functionality
  - Interface to DBMS
- ***Interface to spatial application:*** This layer provides interface to the spatial application.
- ***Core spatial functionality*** **:** This layer supports spatial data representation, logical models & query languages, spatial access methods, query processing and spatial algorithms.
- ***Interface to DBMS***: This layer provides interface for the database.

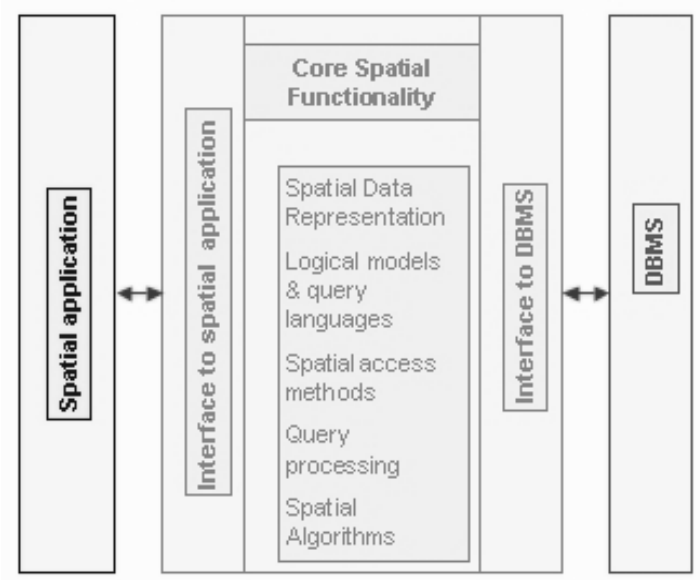

Fig.1. SDBMS Three Layer Architecture

IV. PROBLEM DEFINITION

Traditionally, the spatial preference query ranks the top k objects with the highest score, such that it ranks the entire objects. For example, in real estate database user want to find the flats with good quality, the spatial preference query rank the top most flats with highest score for all flats available in the database.

The problem found here is, the spatial preference query does not rank the top k objects for specific location. For example: If a user wants to find the best flat in specific location, the spatial preference query does not rank top flat for selected location. To solve this problem, we generate a top k feature rank for selected location query. This query helps to rank the objects based on quality of features for user selected location. For example, user wants to buy best flat in Pondicherry location, our proposed query find and display the top most flats only in Pondicherry location based on good quality of features. Two algorithms such as search algorithm and Enhanced Branch and bound algorithm have been used to answer this query and produce efficient result. Motivated by this, in the next section we present the steps of algorithms to solve the query.

V. PROPOSED ALGORITHM

In this section, the indexed R-Tree data structure used for representing and implementing the spatial data is given followed by the algorithm used for retrieving top k best featured query result for selected location query.

A. R-Tree

R-Tree is a spatial access method .The R-tree data structure was the first index specifically designed to handle multidimensional extended objects. It essentially modifies the ideas of the B-tree to accommodate extended spatial objects. The key idea of R-tree is to group nearby objects and represent them with their minimum bounding rectangle (MBR) in the next higher level of the tree. Time complexity of search r-tree

- If MBBs do not overlap on q, the complexity is O (log mN).
- If MBBs overlap on q, it may not be logarithmic, in the worst case when all MBBs overlap on q, it is O (N).

B. Top k feature rank for selected location query

The top k feature rank for selected location query is used to retrieve few best objects according to quality of features. To answer this query two algorithms used. They are Search algorithm [9] and Enhanced Branch & bound algorithm. The block diagram of the proposed query functioning with search and proposed enhanced BB algorithm is given in Fig.2.The top k feature rank query will be inputted to enhanced Branch and bound algorithm. It gets the data for getting processed from the data base by search algorithm and produces top k rank featured query results for the selected location.

1) Search algorithm

The spatial objects can be searched by use of search algorithm [9]. In this algorithm, the data partitioning method such as R-Tree index is used. The basic search algorithm on R-trees, similar to search operations on B-trees, traverses the tree from the root to its leaf nodes.

Algorithm SEARCH

//Let E be the entry, T be the root of a given R-tree and S be the search rectangle, the algorithm is intended to identify all index records whose rectangles overlap S . Denoting an index entry E's rectangle by E.I.

(1) Search **sub trees**: If T is not a leaf, check each entry E to determine whether E.I overlaps S or not. If it overlaps then we start search for the consecutive non leaf node.

(2) Search **leaf node**: If T is a leaf, check all entries E to determine whether E.I overlaps S and result E is found.

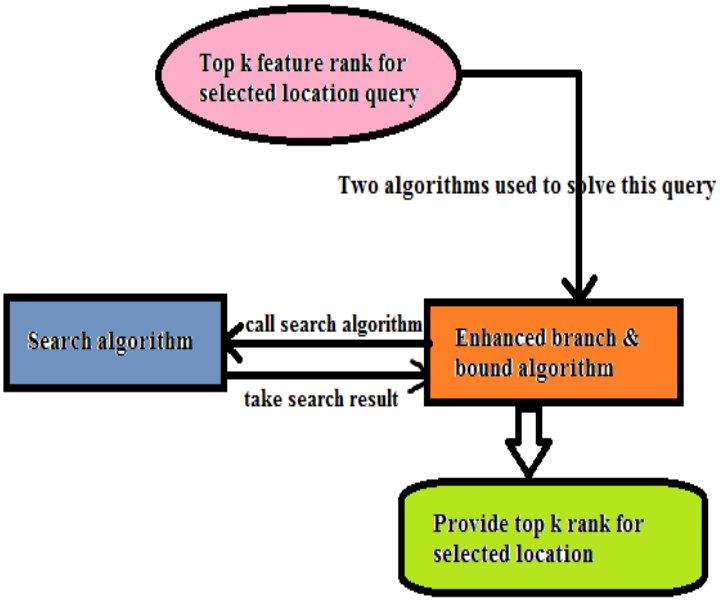

Fig.2. Block diagram of the proposed query

1) Branch and Bound algorithm

The top most spatial objects can be getting by use of branch and bound algorithm [8],[10]&[11].

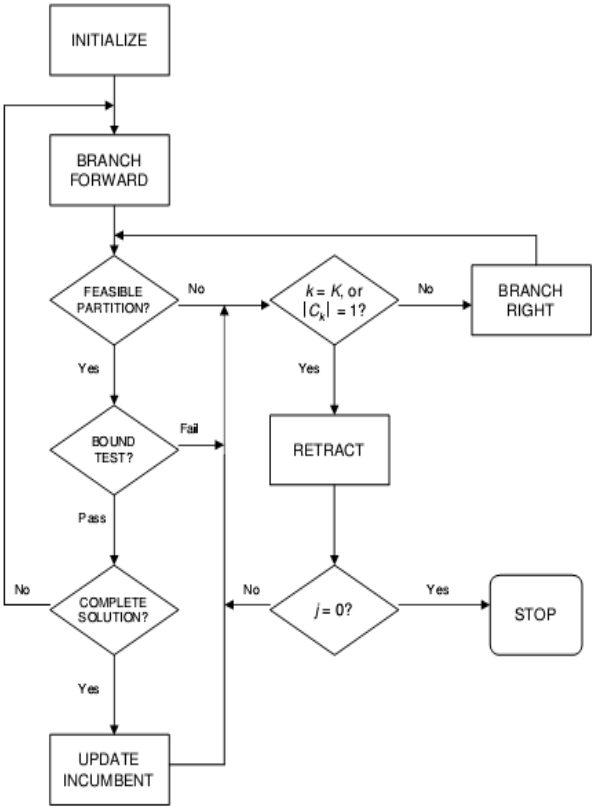

Fig.3.A Flowchart steps of the branch & bound process

Algorithm Description

The key idea is to compute component score, for non-leaf entries E in the object tree D, an upper bound T(E) of the score T(p) for any point p in the sub tree of E.

The algorithm uses two global variables: Wk is a min-heap for managing the top-k results and γ represents the top-k score so far (i.e., lowest score in Wk). The pseudo-code of branch and bound algorithm (BB) is called with N being the root node of D. If N is a non-leaf node, in this algorithm, the scores T (E) for non-leaf entries E can be computed concurrently.

Recall that T (E) is an upper bound score for any point in the subtree of E. The techniques for computing T (E) will be discussed shortly. The component score Tc (E) is the range score, take maximum quality of points. With the component scores Tc (E) known so far, we can derive T+ (E), an upper bound of T (E). If T+ (E) ≤ γ, then the subtree of E cannot contain better results than those in Wk and it is removed from set V.

In order to obtain points with high scores early, we sort the entries in descending order of T (E) before invoking the above procedure recursively on the child nodes pointed by the entries in V. If N is a leaf node, we compute the scores for all points of N concurrently and then update the set Wk of the top-k results. Since both Wk and γ are global variables, the value of γ is updated during recursive call of BB. To improve the performance of Branch and bound algorithm, we develop the enhanced branch and bound algorithm as follows.

Enhanced Branch and Bound algorithm

Algorithm:Enhanced Branch and Bound
Wk: = new min-heap of size k (initially empty);
γ: =0;
// k-th score in Wk

1: ***Call*** search algorithm
// Take input as search result E from search algorithm
2: V: {E| E ε N}; //V denotes set in which points are to be stored
3: ***If*** N is non-leaf ***then***
4: ***for*** c: =1 to m ***do***
5: compute T (E) for all E ε V concurrently;
6: remove entries E in V such that T+ (E) <= γ;
7: ***for each*** entry E ε v such that T (E) > γ ***do***

8: read the child node N pointed by E;
9: continue step 2;
10: **else**
11: ***for*** c: =1 to m ***do***
12: compute T (E) for all E ε V concurrently;
13: remove entries e in V such that T+ (E)<=V;
14: Sort entries E ε V in descending order of T (E);
15: Update Wk (and γ) by entries in v;

In branch and bound algorithm, changes have been made in getting input values and also about sorting the entries, resulted with enhanced branch and bound algorithm. The input values of enhanced BB are the output of searching algorithm. Instead of performing sorting individually on each node among its child nodes, entire tree node have been sorted after this process is over. This will reduce the time effectively and improve the performance.

VI. EXPERIMENTAL RESULT

In this section, we evaluate our proposed work, by implementing the algorithms executed on a Visual studio. In our experiment we take real data as New York from a real estate website http://century21grand.com/dbtypes.htm to test our proposed query. This can be achieved by giving input for searching top most flats in New York country for Newburgh city.

Our proposed query find the set of entries that match with Newburgh entries by using search algorithm and the enhanced branch & bound algorithm based on good quality of features and obtain the result.

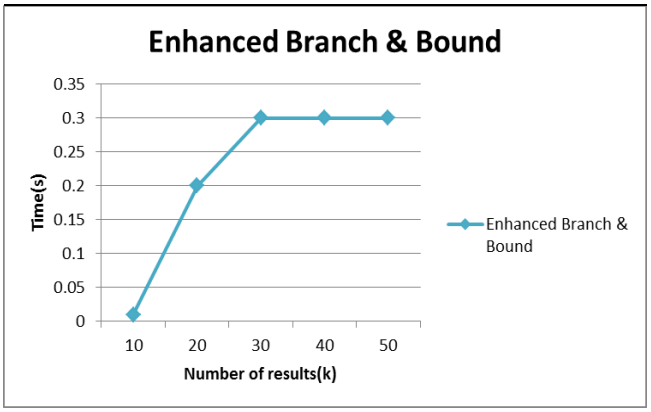

Fig.4. Enhanced branch and bound performance

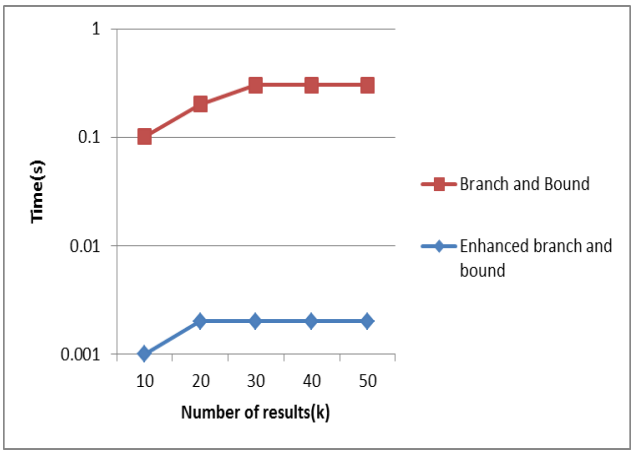

Fig.5. Effect of real data with number of result and time

The Fig.5 compares the cost of the algorithm for New York dataset. Branch & Bound (BB) algorithm requires a higher number of inputs and performs sorting individually on each node, so it takes much time to produce top-k result set. However, the Enhanced Branch & Bound algorithm requires fewer inputs and performs sorting on all nodes than BB. So, it takes less time to produce top-k result. Therefore Enhanced Branch & Bound algorithm performs slightly better than BB.

VII. CONCLUSION

In this paper, we use two algorithms like search algorithm and enhanced branch and bound algorithm to solve this problem. Specifically, search algorithm search objects and the enhanced branch and bound algorithm require input as search result obtains from search algorithm and efficiently produce top-*k* result for selected location. As confirmed with extensive experiments, the generated query produces efficient result.

REFERENCES

[1]. http://en.wikipedia.org/wiki/Spatial_database.

[2]. Elizabeth Sayed , Elizabeth Stoltzfus. "Spatial Databases GIS Case Studies". Industrial Engineering and Operations Research (IEOR) , Dec, 2002.

[3]. Ramakrishnan .R, Gehrke.J. "Database management Systems". McGraw-Hill, Third edition, 2003.

[4]. Yiu.M.L, Dai. X, Mamoulis. N, Vaitis.M. "Top-k Spatial Preference Queries". In Proc. of IEEE 23rd Int.Conf. on Data Engineering (ICDE), April 2007.

[5]. Ilyas. F, Aref .W. G, Elmagarmid.A. "Supporting Top-k Join Queries in Relational Databases". International Journal on Very Large Data Bases (VLDB),Vol.13,3, 2004.

[6]. Ihab. F, Walid.G, Ahmed.K, Hicham.G, Rahul shah, Jeffrey Scott Vitter. "Adaptive Rank-Aware Query Optimization in Relational Databases". ACM Transactions on Database Systems,Vol. 31,4, pp.1257–1304,2006.

[7]. Joao.B, Akrivi vlachou, Christos Doulkeridis, Kjetil Norvag. "Efficient Processing of Top-k Spatial Preference Queries". Very Large Data Bases (VLDB) Endowment,Vol.4,2,2011.

[8]. Yiu. M. L, Dai.X,Mamoulis.N, Vaitis.M. "Ranking Spatial data by quality preferences".IEEE Transaction on Knowledge and Data Engineering,Vol.23, 3, pp.433 – 446, 2011.

[9]. Yong Zhang, Lizhu Zhou, Jun Chen. "Layered r-tree: an index structure for three dimensional points and lines". International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII,B4.

[10]. Antonio Leopoldo, Corral Liria. Algorithms for the Processing of Spatial Queries using R-trees. The Closest Pairs Query and its Application on Spatial Databases Almeria, January 2002

[11]. Yunjunga gao,"Optimal-location-selection query processing in spatial database"knowledge and data engineering, august 2009.