OPTIMIZATION OF LOCATION BASED QUERIES USING SPATIAL INDEXING

S. Geetha[1] and S. Anu Velavan[2]
Department of Computer Science and Engineering, Anna University – Regional Office, Tiruchirapalli, India
E-mail: [1]sgeetha2501@gmail.com, [2]avelvan26@gmail.com

Abstract
The recent development in the technology leads to the introduction of various mobile terminals and there is a demand that the client requires effective location based services. The valid regions expand and also query retrieval time increases which lead to poor performance of query processing. The spatial indexing techniques are one of the most effective optimization methods to improve the quality of services. In existing system NN queries and window queries are used. In that R-tree and grid indexing has been used for increasing the query efficiency. But the Grid-index technique support low memory and thus large databases cannot be handled effectively. In the proposed system we are using Ordered grid index and EVR-tree to minimize the query retrieval time and to decrease the depth of the search index. The Ordered grid index and EVR-tree to speed up the spatial query processing.

Keywords:
Spatial Indexing, Location Based Queries, LBS, Spatial Data Mining, Spatial Queries

1. INTRODUCTION

Spatial data mining is the process of discovering of knowledge, spatial relations or other patterns explicitly stored in spatial databases such mining demands an integration of data mining with spatial database technologies.

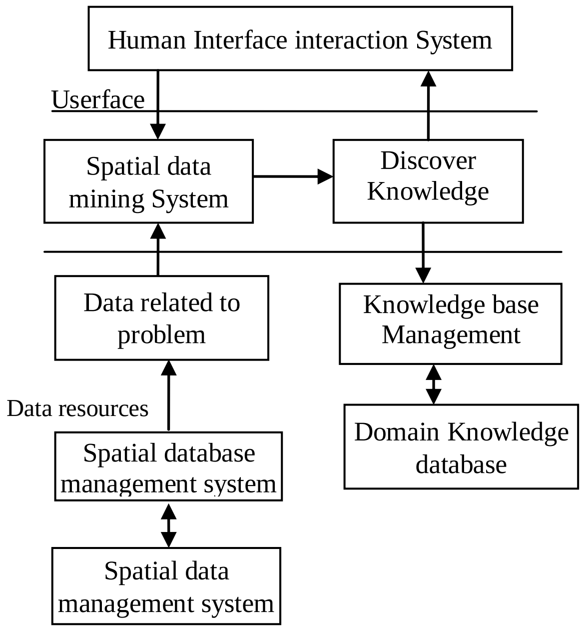

Fig.1. Architecture of spatial data mining

It can be used for understanding spatial data, discovering spatial relationship and relationships between spatial and non-spatial data, constructing spatial knowledge bases, reorganizing spatial database and optimizing spatial queries. Spatial database stores and maintain huge volume of spatial related data such as maps, pre-processed remote sensing or medical imaging data. The spatial data contains both spatial and non-spatial attributes.

The spatial data contains both spatial and non-spatial attributes. The spatial attributes are used to describe the location of spatial objects such as longitude and latitude, elevation of spatial objects. The non-spatial attributes of spatial objects to describe non-spatial features of spatial objects such as name, unemployment rate and population for a city. The spatial related data are known as spatial data. For example consider the road map which contains two-dimensional objects such as points, line, polygon to represent cities, roads, and boundaries of states respectively.

1.1 UNIQUE FEATURES OF SPATIAL DATA MINING FROM CLASSICAL DATA MINING

The following some of the features differentiate the spatial data mining from classical data mining.

Input to spatial data mining process

The input of spatial data mining are more complex than classical data mining process because it is in the form of lines, points, polygons. This spatial data contain two types of attributes respectively spatial and non-spatial data. The spatial objects have implicit relationship among spatial objects such as overlap, intersect. The non spatial have the explicit relations among spatial objects member of, subclass of, arithmetic relation, ordering of data.

Output pattern of spatial data

The spatial data mining contain four important form of output form such as spatial clustering, spatial outlier, predictive model, spatial collocation patterns.

Computational process

To improve the computational efficiency of spatial data mining require spatial auto correlation, low dimensionality in space and spatial indexing approach retrieve the result of spatial query quickly. The generic algorithms are generalized to apply to spatial data mining.

1.1.1 Types of Relationship Between Spatial Objects:

The spatial relationship is used to find relation between objects which satisfy certain relation

- Topological
- Direction
- Distance

The spatial relations are classified into topological, distance and direction relations which may be combined by logical operators to show the neighborhood relation between spatial objects. The spatial objects are points, lines, polygons which are represented by a set of points. The topological relations between two related objects are based on the boundaries, interiors and complements of those objects. some of the relation operators are disjoint, meets, overlaps, covers, covered by, contains, inside. Distance relations between two objects are calculated by compare the distance of two objects with a given constant using one of the arithmetic comparison operators [2]. The direction relation between two object O1 and O2, consider one representative point of the object O1 as the origin of a virtual coordinate system whose quadrants and half-planes define the directions. To fulfill the direction predicate, all points of O2 have to be located in the respective area of the plane

1.2 SPATIAL QUERY

The spatial query is the type of query to search the available data in the spatial database to satisfy the user conditions. It is the special form of database query supported by spatial and geo databases. The advantage of spatial query over SQL query is which make use of spatial relationship between geometrics and allow geometric data such as lines, polygons, points for spatial query processing.

1.2.1 Types of Spatial Queries:

There are several types of spatial queries that are explained below.

Range query

The queries are associated with bounded regions. The data retrieved for this query may be overlapped or contain some regions. In the range query spatial objects are related to each other within particular region or distance. For example find all the hospital within 10 miles of the trichy city.

Nearest neighbor query

Here the data is retrieved which is close to the location at which the query is generated. For example a user issues a NN query to find the nearest hospital with respect to my current location.

Window query

It is a special kind of range query which is used to find all the specified data items within the specified range of window frame. This query is very useful when user want to view particular portion of data or image. For example user issues a query to find all hospital in the particular town.

Spatial join query

Join query is the combination of more than one query for that spatial join operation is used. For example user issues a query to find all the hospital in the particular city.

1.3 SPATIAL INDEXING

The spatial database are using spatial index to optimize spatial query, which is one of the optimization methods to improve the quality of location based services.

1.3.1 Need of Spatial Indexing:

Now a day most of the application GIS, location based services operate on spatial data that is in the form of points, lines, polygons, circles. But traditional DBMS can't store huge spatial data and unable to process spatial data effectively due to the following reasons:

- Spatial data are very huge, very complex in structures and relationships
- To retrieve data from spatial database complex spatial operators are needed such as intersection, adjacency.
- In spatial database ordering of spatial data is very difficult. For that conventional sort-ordering methods can't be used.

To avoid those entire problems spatial indexing is needed to improve and optimize the performance of query processing.

The conventional indexing method does not handle the spatial query processing while the distance between two objects are very long, the objects are bounded with some boundaries. The spatial indexing classified into R-tree and its variants, Grid index structure, Quad tree, Oct tree etc., which are all used to improve efficiency of query processing.

1.4 LOCATION BASED SERVICES

Location based services is also known as location dependent information services. Spatial queries are one of the important Location based server. Based on the spatial constraints, spatial queries can be divided into several categories. They are NN queries and window queries. Both queries are used for finding the query results based on the category of spatial queries.

2. RELATED WORK

In previous year so many researches were conducted for these query processing and spatial access methods. Here we are going to give some concepts which are related to our proposed work. The geographic information system contains the data like point, lines, polygons which contain the spatial relationship between those data. When the spatial data contain more complexity and large data the computational and storage complexity is very low.

The uniform grid index structure has the problem of very large data base can't be handled effectively when the memory is resident. The adaptive grid index filter algorithm is proposed to distribute spatial data throughout all grid cells. This algorithm construct boundary grid, internal grid when research mode of spatial data are in the form of polygon [5]. Due to the recent development in the technology lot of mobile devices have been developed. The mobile phones, PDAs even on the development has lot of disadvantages such as the limited bandwidth and also limited battery life and restrained memory space. The method proposed was mainly aimed to provide fast query processing with the minimal number of nodes. This proposes a new technique where the wireless broadcast environments for which the location based maintenance strategies. These strategies predict the next data that is about to be queried by the mobile client and thus perfectness the data closes to the client's current location. The cache maintenance strategies proposed can be categorized into two steps they are nearest pre-fetch first and hierarchical farthest away replace [7].

The object-based and solution-based indexing are used to NN query and other related queries. The object based indexing suffers from the backtracking problem and main issue in object based indexing is the low storage overhead. The solution based indexing overcomes the backtracking problem but occupies a larger index space than the object based indexing. To overcome these drawbacks this proposes the concept of Grid-partition index. This technique of grid partition index splits the solution space of the NN query in to grid cells. Thus the query point and the data that are located near can be easily mapped into the grid cell. Due to the implementation of this technique the search space is reduced to a greater extent. The data that falls within the grid cell that are related to the query are stored as the objects in the grid index. Thus this approach is a combination of the object based and the solution based indexing [8]. An indexing and semantic cache method for location dependent queries based on voronoi diagrams and location based queries. It pre-process the data very quickly and for validate the data item semantic caching is used. They also introduced cache replacement algorithm based on the size of the area and the distance between centres of the cached data item [6].

The R*-tree is the extension and modification of B*-tree which use only linear single attribute. The proposed R*-tree is based on minimum bounding rectangle used to identify the overlapping region of query and it is mainly concentrate to reduce area, margin and overlap of the directory rectangles [11]. The location based queries is an approach to estimate the validity of the queries that are processed previously based on the current location of the mobile users. To make this possible an additional property called the validity region is added by the server. This mainly focuses on the two popular types of queries nearest neighbor and window queries. This technique is aimed to reduce the number of queries to the server. The Voronoi diagram is used to compute the nearest neighbor when a nearest neighbor query reaches the server [13].

3. PROPOSED WORK

The client issues a query to server following operations are performed. The proposed system contains the two main steps such as query processing and indexing of data. This proposed system use the proxy based with spatial indexing approach to process spatial query in efficient manner and fast retrieval of data with updating user location in frequently. In the dataset processing selected dataset records has inserted into the database and used to remove the unnecessary, missing or redundant data in the data base. This is used to improve the quality of query processing.

The distance between the user current location and the location of object he needs. The distance has been estimated based on the longitude and latitude of object from the given dataset using the Euclidian Distance formula as below.

Based on the distance only user can determine whether the client requested location is within the estimated valid region or the location is out of region from server. Next query is processed and stored in the database. The indexing is mainly used for searching the location of user data based on issued spatial queries. For indexing operation proxy maintains temporary cache of query results for the frequent data item retrieval. Basically, this approach has been classified into NN-queries and window queries. For nearest neighbor query data is retrieved which is close to the location at which the query is generated.

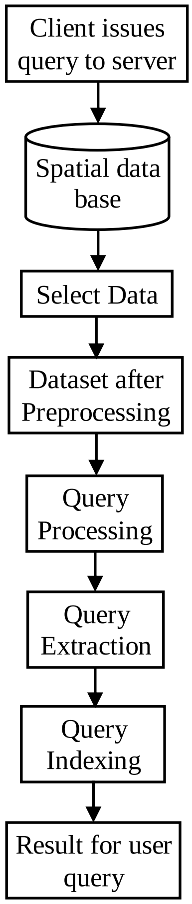

Fig.2. Architecture for processing of data

NN Query Algorithm

```
Input: Spatial Dataset
Output: Query Results based on ordering
Input Parameters: longitude, latitude, distance from dataset
begin
While (dataset value.next())
{
  For each Object in Dataset
  Input belong to N number of objects
  For (int i = 0; i <= n; i++)
  {
        NN query result = select current location, preferred
        location, distance based on choosing category;
        NN query Indexing Result: select (category,
        distance) order by distance;
     }
}
End
```

These query results are stored and maintained in the estimated valid region tree for quick retrieval of data. The window query is used to find all the specified data items within the specified range of window frame which updates the frequent data objects. These frequent updated data objects are stored in the ordered group cell grid index structure.

Window Query Algorithm

```
Input: Spatial Dataset
Output: Query Results based on ordering
Input Parameters: longitude, latitude, distance from dataset
begin
While (dataset value.next())
{
  For each Object in Dataset
  Input belong to N number of objects
  For (int i = 0; i <= n; i++)
 {
    W query result = select current location, preferred
    location, distance based on choosing category;
    Window query Indexing Result: select (category,
    distance) order by distance based on the current region;
  }
}
End
```

This approach use the efficient query processing algorithm that can effectively utilize Ordered cells of grid index to speed up the processing of spatial queries. The Ordered group cell index groups the cells of grid into groups which contain equal or similar the number of objects based on distance. This group is called as ordered cell group and each group contain unique identifier to identify the cell object. This means that which map the ordered cell groups to data nodes of index instead of cells of grid. The index structure is maintained by splitting the ordered group into two ordered groups when overflow occur in the specified group cells.

The each divided sub group contains equal number of objects in each group. This splitting of group is used to effect utilization of storage space. For this divided sub group key identifier is identified by median object position in the ordering sequence of cells. Then ordered grid index search efficiency is improved by using identifiers of ordered cells to increase the out of index node and decrease the depth of index.

Ordered Grid Indexing Algorithm

```
Input: Spatial Dataset
Output: Query Results based on ordering
Input Parameters: longitude, latitude, distance from dataset
begin
While (dataset value.next())
{
  For each Object in Dataset
  Input belong to N number of objects
   For (int i = 0; i <= n; i++)
   {
        Indexing result = select (category, distance) order by
        distance;
     }
}
End;
```

4. PERFORMANCE EVALUATION

The performance evaluation is to compare the performance of existing indexing system with proposed indexing system. The ordered grid indexing retrieval time is compared with R and grid indexing for number queries issued by client. There are two types queries such as NN query and window query. To calculate query retrieval time client issues a query to server. The comparison is as follows:

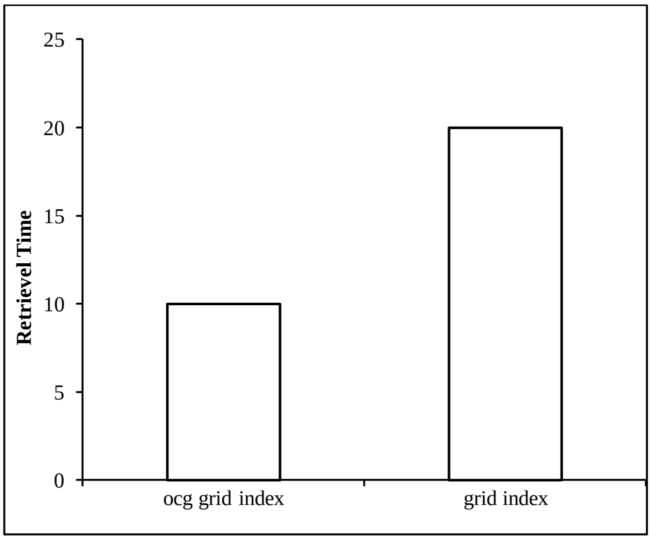

Fig.3. Query retrieval time for window query

The Nearest neighbour query is used to find location and the indexing is used to retrieve data in the database quickly.

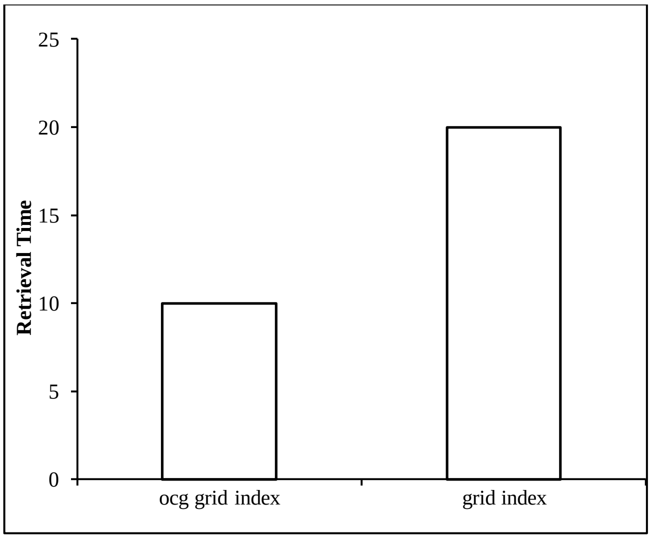

Fig.4. Query retrieval time for NN query

5. CONCLUSION

The index structure for LBSs should be efficient for spatial query processing in the presence of frequent location updates. The client sends a query to server to retrieve location based on his current location. Based on type of window query or nearest neighbor query which are processed and results are stored. R-tree and ordered Grid index is applied for the processed results to quick retrieval of results.

REFERENCES

[1] Petr Kuba, "Data structures for spatial data mining", Faculty of Informatics Masaryk University Report series, 2004.

[2] Peter van Oosterom, "*Spatial access methods*", Wiley pages, Vol. 1, pp. 385-400, 1999.

[3] Beng Chin Ooi, Ron Sacks-Davis and Jiawei Han, "Indexing in spatial data bases", *Natural Sciences and Engineering Research Council of Canada*.

[4] Martin Ester, Hans-Peter Kriegel and Jörg Sander, "Algorithms and Applications for Spatial Data Mining", *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS, Taylor and Francis*, 2001.

[5] Hao Lu, Ershun Zhong, Xieliu Shu, Tianbao Wang and Shaohua Wang, "An Effective Algorithm for Spatial Query using Spatial Adaptive Grid Index Filter", *19th International Conference on Geoinformatics*, pp. 1-5, 2011.

[6] Kwangjin Park and Young-Sik Jeong, "A Caching Strategy for Spatial Queries in Mobile Networks", *Journal of Information Science and Engineering*.

[7] B. Zheng, J. Xu, W.-C. Lee and D.L. Lee, "Grid-Partition Index: A Hybrid Method for Nearest-Neighbor Queries in Wireless Location-Based Services", *The VLDB Journal*, Vol. 15, No. 1, pp. 21-39, 2006.

[8] Baihua Zheng, and Dik Lun Lee, "Semantic caching in location dependent query processing", *Advances in Spatial and Temporal Databases*, *Lecture Notes in Computer Science*, Vol. 2121, pp. 97-113, 2001.

[9] Shiow-yang Wu and Kun-Ta Wu, "Dynamic Data Management for Location Based Services in Mobile Environments", *Proceedings of the 7th International Database Engineering and Applications Symposium*, pp. 16-18, 2003.

[10] Maruto Masserie Sardadi, Daut Daman, Mohd Shafry and Zahabidin Jupri, "QuadR-Tree Indexing Selection Engine for Tuning Spatial Database System using Mobile Geographical Information System Technology", *New Advanced Technologies*, 2010.

[11] N. Beckmann, H.-P. Kriegel, R. Schneider and B. Seeger, "The R-*Tree: An Efficient and Robust Access Method for Points and Rectangles", *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Vol. 19, No. 2, pp. 322-331, 1990.

[12] Sandi Winn Aye, "A Caching Scheme in Location-Dependent Query Processing", *International Journal of Scientific and Technology Research*, Vol. 2, No. 3, pp. 223-227, 2013.

[13] J. Zhang, M. Zhu, D. Papadias, Y. Tao and D.L. Lee, "Location-Based Spatial Queries", *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 443-454, 2003.

[14] Muhamed Ilyas and Vijayakumar, "A Proxy based Framework for Efficient Range Query Processing in a Cellular Network", *International Journal of Information Technology and Computer Science*, Vol. 2, No. 2, pp. 1-8, 2010.