ISBN 978-93-84468-20-0
Proceedings of 2015 International Conference on Future Computational Technologies (ICFCT'2015)
Singapore, March 29-30, 2015, pp. 136-142

Location-Based R-Tree and Grid-Index for Nearest Neighbor Query Processing

Myat Hsu Mon[1], and Myo Min Than[2]
Department of Information Technology Engineering, Yangon Technological University, Yangon, Myanmar
[1]myattsumonn@gmail.com, [2]dr.myominthan79@gmail.com

***Abstract*:** *Location Based Services (LBS) is a service that integrates mobile device's location and other information relating with current location to the user. Geospatial data identifies the geographic location of features and boundaries on Earth. Spatial index can be used for indexing geographic data. Spatial indexes can improve spatial query efficiency. Using spatial index method is suitable for huge volume of data. There are many spatial index methods such as R-tree, B-tree, kd-tree, Quad-tree, Grid index. In this study, nearest neighbors (NN) that can be used by various levels of people (e.g. stores, clinics, mini-marts, schools, convenience stores, bazaars, etc.,) are showed with detailed information. This study uses R-tree method to get detailed NN results efficiently. But R-tree will generate much overlapping and coverage between MBR. So R-tree by combining with Grid-partition index is used because grid-index can reduce the overlap and coverage between MBR. The query performance will be efficient by using these methods.*

Keywords: *LBS, NN, R-tree, Grid-Partition Index.*

1. Introduction

As mobile devices are very popular, location based services are important in people's daily lives. Location of an object or a person is its geographical position on the earth with respect to a reference point. This information can be characterized by using a number of different representations including latitude/longitude/altitude or street addresses, etc. Mobile devices and mobile databases that aim at providing data access anywhere and at any time are widely used in Location Based Query [2].

There are many types of queries. In this study, nearest neighbors search queries based on user's current location are focused on. Nearest Neighbors queries need Location Dependent Data (LDD) for computing the result. For instance, "find the nearest convenience stores within 5km of my device", is a location dependent query because it depends on the geographical location of the mobile unit which initiated the query [1]. Spatial database is to retrieve data items quickly and efficiently according to their spatial locations. In order to handle spatial data efficiently, as required in computer aided design and geo-data applications, a database system needs an index mechanism that will help it retrieve data items quickly according to their spatial locations[7]. However, most indexing techniques are not for multidimensional data. For example, K-D-B tree works only point of data. In B-tree, relational system cannot also handle the new kinds of data. So, novel access methods were proposed. One of these structures, the R-tree, was proposed by Guttman in 1984, aimed at handling geometrical data, such as points, line segments, surfaces, volumes, and hyper volumes in high-dimensional spaces [7]. But R-tree method may overlap between MBR. Moreover, the existing algorithms in spatial database consider that the data are indexed by a spatial access method (e.g. R-Tree) and utilize some branch-and-bound approach to restrict the search space [6]. Main memory grid index is an effective structure for processing continuous range queries over moving objects this method can reduce the overlap between MBR.

In this system, the user asks for specific stores information using a search query then the system uses R-tree index for retrieving the results. A grid-partition method is used to locate MBR and support mobile nearest-neighbor search. The remainder of this paper is organized as follows. Section II describes the background of the theory. Related Work is presented in Section III. Section IV expresses the proposed system of the design. The brief experimental results are given in Section V. Finally, conclude the paper in Section VI. .

2. Background Theory

LBSs are information services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the mobile device. There are classifications of queries in location based services. They are Range Query (RQ), Nearest Neighbor Query (NNQ), Navigation Query (NQ) and Geo-Fence Query (GFQ). In this paper, search nearest neighbor queries based on user's current location. Nearest neighbor (NN) queries are responsible for getting the objects closest to a specific location [3]. The application areas of LBS are Emergency Services, Navigation Services, Information Services, Tracking and Management Services and Billing Services. This study proposes Information Services to find the nearest places.

The data or information that identifies the geographic location is called spatial data. Spatial indexing is a special access method used to retrieve data from data-store. It is also used to decrease the time it takes to locate features that match a spatial query. Spatial data are large in quantity and are complex in structures and relationships. So, an appropriate spatial index is important, especially when users are working with large amount of data. Efficient processing of queries manipulating spatial relationships relies upon auxiliary indexing structures. In order to find spatial objects efficiently based on proximity, it is essential to have an index over spatial locations. The underlying data structure must support efficient spatial operations, such as locating the neighbors of an object and identifying objects in a defined query region [4]. There are many spatial index methods that use in nearest neighbor search.

2.1. R-Tree

R- tree, was proposed by Guttman in 1984, aimed at handling geometrical data, such as points, line segments, surfaces, volumes, and hyper volumes in high-dimensional spaces. R-tree is a height-balanced tree similar to B-tree. It seems that due to modern demanding applications and after academia has paved the way, the industry recently recognized the use and necessity of R-trees. Thus, R-trees are adopted as an additional access method to handle multi-dimensional data. Nowadays, spatial databases and geographical information systems have been established as a mature field, spatiotemporal databases and manipulation of moving points and trajectories are being studied extensively, and finally image and multimedia databases able to handle new kinds of data, such as images, voice, music, or video, are being designed and developed. An application in all these cases should rely on R-trees as a necessary tool for data storage and retrieval. R-tree applications cover a wide spectrum, from spatial and temporal to image and video (multimedia) databases [8].

R-tree is used in spatial database as a spatial access method. Spatial database is the core of GIS. The aim of spatial access method is improving query performance by incorporating an additive data structure since the high query performance is one of the key features of successful GIS systems. R-tree has root node, intermediate node and leaf node. Every leaf record is a smallest bounding box. Root has at least two children. Leaf node does not store the actual spatial objects, but it stores the minimum bounding rectangle of the actual spatial objects. The tree is constructed hierarchically by grouping the leaf boxes into larger, higher level boxes which may themselves be grouped into even larger boxes at the next higher level. The operations of R-tree are inserts, deletes, updates and queries/searches (e.g. Names of all the roads in 1 sq km area?). Fig. 2 shows some rectangles, organized to form a corresponding R-Tree [7].

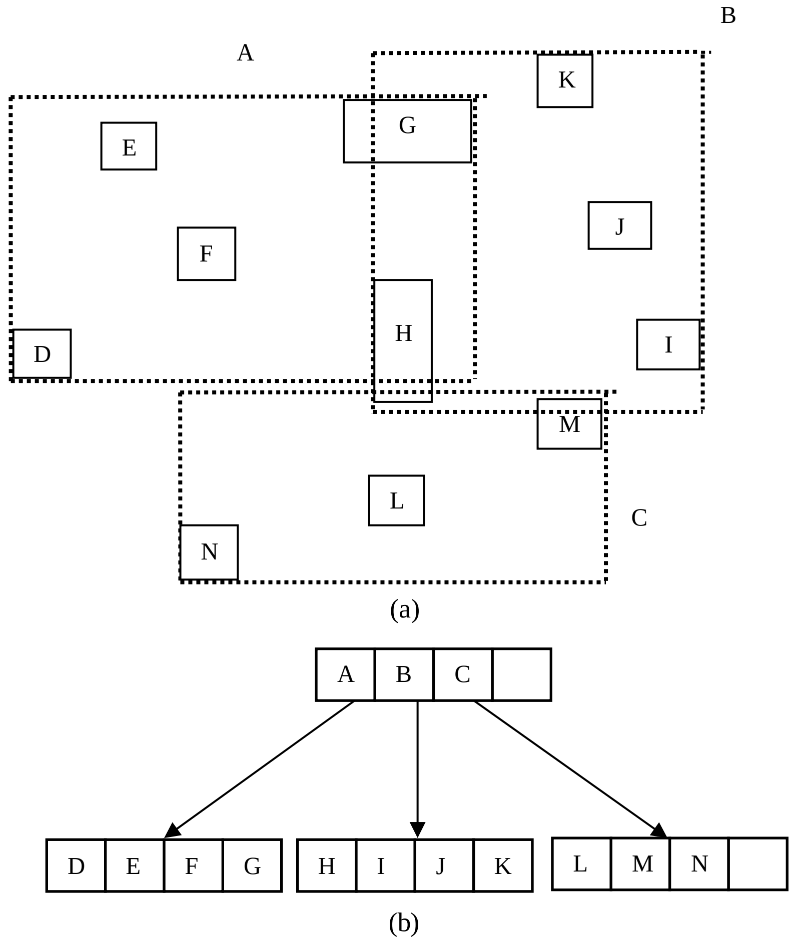


Fig. 1: Example of R-tree (a) data MBRs and their MBRs (b) The corresponding R-tree

2.2. Grid-Index

Grid indexing algorithm is a simple and efficient way of spatial indexing. A grid-based spatial index has the advantage that the structure of the index can be created first, and data added on an on-going basis without requiring any change to the index structure; indeed, if a common grid is used by disparate data collecting and indexing activities, such indices can easily be merged from a variety of sources. Many commercial software apply this type of indexing despite its shortcomings, especially when the spatial distribution of objects is not equal-distributed. The optimistic scenario would be that every grid should contain closely the same number of related objects. Spatial grid indexes define imaginary x/y grids for a feature class. There may be one, two, or three of these grid levels defined per feature class. Fig. 2 shows the example of grid file.

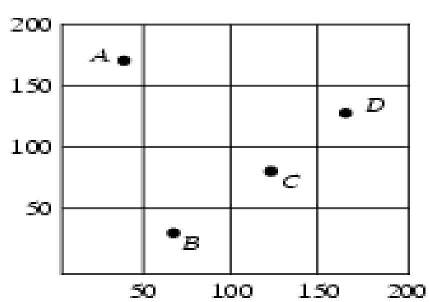


Fig. 2: Grid file

3. Related Works

Mate et al. proposed continuous query in location based services [1]. This system retrieves the results only from the server. H. Dunham et.al described the formalization of location related in queries [2]. Several in-memory algorithms of continuous range queries over collections of moving objects are evaluated in [5]. R-tree index is proposed for spatial searching [7] and [8]. Combining R-tree and B-tree were presented in [9]. This system uses R-tree to retrieve the results and plots the results on the map and B-tree is used to show more detail. Kwangjin Park proposed a spatial index and query processing algorithm that takes the broadcast environment into consideration. The proposed index partitions the local area into grid-cells and assigns the unique ID number to each grid-cell [10]. Zhang et al. proposed a grid cell based continuous k-NN query processing method (CkNN) [6].

4. Proposed System

This section presents the design of the proposed system. The main objective of the proposed system is the design of searching nearest neighbors with detailed information based on user's current location. Nearest Neighbor search is an important class of LBS. LBS is a useful application that provides mobile data services. Users with mobile devices can query about their surroundings based on current locations at any place, anytime. Nearest Neighbor search is the query of Location-Dependent Query that provides data based on location data specified in a query.

The user can check his/her current location on Google Map located by marker. The user can find services like stores, schools, restaurants, clinics, hospitals, mini-marts that locate within a specific range. To get point of interests, user needs to send information of current location to the server. Current location is converted into x, y coordinate. Firstly, latitude and longitude value has been converted into radians using following formula:

$$LAT= (latitude *pi) /180 \tag{1}$$

$$LON= (longitude*pi) /180 \tag{2}$$

The radian coordinates have been reconverted into x and y coordinates using following formula:

$$x= R*Sin (LAT) *Sin (LON) \tag{3}$$

$$y=R*Cos (LAT) \tag{4}$$

The spherical coordinate system has earth's radius (R) approximately 6371 km. Then sort objects in R-tree using MINDIST ordering metric. MINDIST (q, R) is the minimum distance between a point q and rectangle R. If the point is inside R, then MINDIST=0. If q is outside of R, MINDIST is the distance of q to the closet point of R. The following equation is to calculate MINDIST.

$$\text{MINDIST (P, R)} = \sum_{i=1}^{d} |qi - ri|^2 \tag{5}$$

where:

$$r_i = s_i \ if \ q_i < s_i \tag{6}$$

$$r_i = t_i \ if \ q_i > t_i \tag{7}$$

$$r_i = q_i \ otherwise \tag{8}$$

The followings are nearest neighbor search algorithm using R-tree and grid-partition index.

1. Initialize the nearest distance as finite distance (e.g. 5km,7km)
2. Traverse down the tree from the root. During this situation, at each newly visited non-leaf node, the algorithm computes the ordering metric bounds (MINDIST) for all its MBRs and sorts them.
3. Apply pruning strategy 1 and 2 to the ABL to remove unnecessary branches.
4. Iterate on this ABL until it is empty. For each iteration step, the algorithm selects the next branch in the list and applies the step 3 recursively to its children node corresponding to the MBR of the branch.

5. At a leaf node, computes each object's distance from the query point, compares with the distance of the nearest distance so far, updates it if necessary.
6. Then MBRs containing the point object are divided in grid coordinates to get the smallest bounding rectangle.
7. In grid implementation, divide space into M-by-M grid of squares, create linked list for each square, use 2D array to directly access relevant square. Insert: insert MBRs that contain objects'(x, y) into corresponding grid square and examine only those grids squares that could have points in the rectangle.
8. Finally, nearest neighbors are returned when ABL is empty.

In server, current location and nearest objects are executed by using proposed methods. Then server sends the corresponding reply to the user. The user will get the point of interests with details.

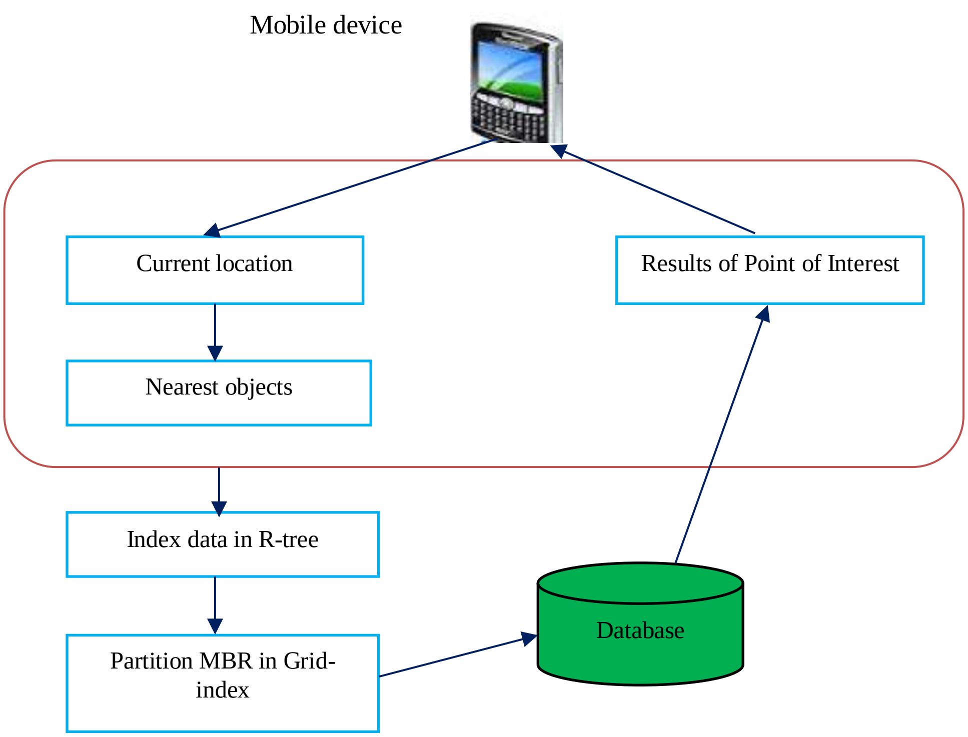
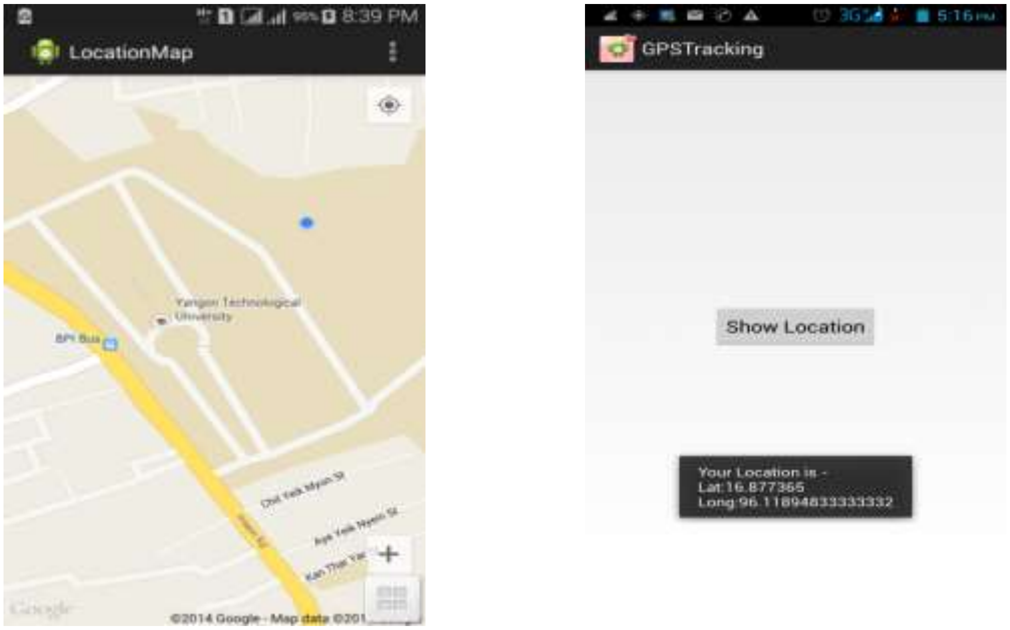


Fig. 3: System Architecture Design

5. Experimental Results

This section presents the experimental results. The applications used in the study are the followings: Google Map (API), Android Developer Tool, MySQL server. This system considers nearest neighbors of user's location. The system will give detailed information of objects which the user wants to know to the user. User can get his/her location from GPS of mobile device. Fig. 4 (a) shows user's location and the address of user's location (latitude and longitude) is shown in Fig. 4 (b).



(a) (b)

Fig. 4: Mobile device display (a) user's current location (b) the address (latitude and longitude) of current location

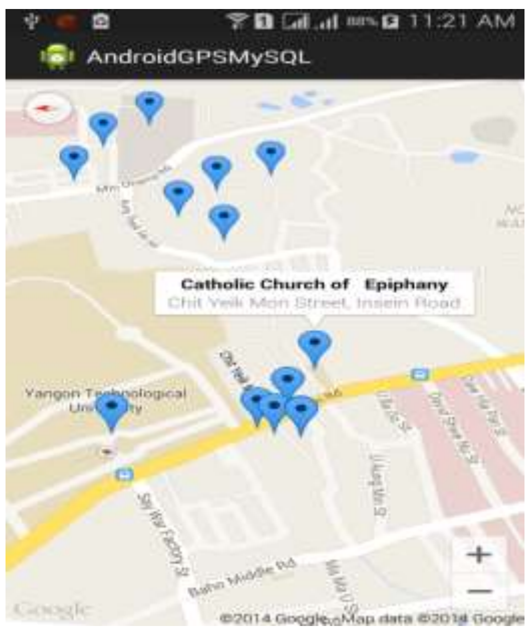


Fig. 5: Nearest places of user on map

Fig. 5 shows the nearest places of user's location. User can find places whatever he/she wants. For example, user wants to find a clinic within 2 km.

6. Conclusion

Location Based Services can help user to find stores, clinics, hospitals, gas stations or any other facility of interest requested by user within a specified range. GPS in mobile device can update user's location as soon as his/ her location changes. Spatial indexing is efficient for query processing in LBS. Indexing techniques such as

R-tree, kd-tree, B-tree, Quad-tree and grid are experimented for different goals. The important point is to retrieve any data quickly and effectively based on the current location. This paper proposes a hybrid index structure that combines R-tree and grid partition index. The query performance is not efficient by using R-tree method because of overlapping in MBR of R-tree. So grid-partition index is used to solve the overlap and coverage between MBR. This study is intended to get detailed information by the user. Moreover, this study gives data objects such as stores, schools, clinics, bazaars, mini-marts that can be used by various levels of people. The system can help people that search nearest places by reducing time and distribution cost.

7. Acknowledgement

I specially thank to my teachers for providing guidance. I would also like to thank everyone who supports me with respect to this study.

8. References

[1] Pramila Mate, Hariram Chavan, Vishwaji Gaikwad, "Android Based Continuous Query Processing in Location Based Services", *IJARCSSE*, vol.4, Issue 5, May 2014.

[2] Ayse Y.Seydim, Margaret H. Dunjam, Vijay Kumar, "Location Dependent Query Processing", *MobiDE 2001 ACM 1-58113-412-6/01/05*, USA.

http://dx.doi.org/10.1145/376868.376895

[3] Raghunath Rajachandrasekar, Zoya Ali, Srinivas Hegde, Vilobh Meshram, Nishanth Dandapantula, "Location-Based Query Processing: Sensing Our Surroundings", Department of Computer Science and Engineering, Ohio State University.

[4] Raghunath Beng Chin Ooi, Ron Sacks-Davis, Jiawei Han, "Indexing in Spatial Databases".

[5] D.V. Kalashnikov, S. Prabhakar, W. G. Aref, S. E. Hambrusch, "Efficient Evaluation of Continuous Range Queries on Moving Objects", *13th International Conference on Database and Expert Systems Applications- DEXA 2002*.

[6] Wei Zhang, Jianzhong Li and Haiwei Pan, "Processing Continuous k- Nearest Neighbor Queries in Location Dependent Application", *IJCSNS International Journal of Computer Science and Network Security,* vol.6 No. 3A, March 2006.

[7] Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching", *in Proc ACM SIGMOD International Conference on Management of Data*, Boston, Ma, pp: 45-57, 1984.

[8] Yannis Manolopoulos, Alexandros Nanopoulos, Apostolos N. Papadopoulos, Yannis Theodoridis, *R-Trees: Theory and Application*.

[9] Marwa A. M. Abd Elwahab, Khaled M. Mahar, Hatem Abdelkader, Hatem Awad Khater, "Combining R-Tree and B-Tree To Enhance Query Processing".

[10] Kwangjin Park, "Location-Based Grid Index for Spatial Query Processing", Expert Systems with Applications 41 (2014) 1294–1300.

http://dx.doi.org/10.1016/j.eswa.2013.08.027