

Grid-based Spatial Index Method for Location-based Nearest Neighbour Search

Aung Zaw Myint ¹⁺, Khin Mo Mo Tun ²

¹University of Computer Studies, Yangon

²University of Information Technology, Yangon

Abstract. Geo-spatial data becomes more and more large amount of data on the Web. On the other hand, managing massive spatial data is one of challenges for supporting spatial queries and high performance computational is also needed to support spatial queries. Thus there is needed to solve this criteria is to create a better spatial indexing method. The proposed method is to create Grid-based R-tree index structure for k nearest neighbour query and range query. R-tree is constructed with Minimum Bounding Rectangle (MBR) that contains a group of objects. The proposed system is combined R-tree with grid index that is reduced overlapping and covering area. The proposed system is to support spatial queries efficiently and also supports speed up computational performance.

Keywords: k Nearest Neighbor Search, R- tree, Grid-index, LBS

1. Introduction

The increasing of inter-network technologies and Internet of Things (IoTs) devices has become commonly useful for consumers on their mobile smart aware devices. Thus the quality of spatial data is needed to support on their real-time location via location-based services (LBS). Google Maps Services and Social networking services (SNS), such as Twitter and Facebook are provided online users to reveal their location during emergencies, retrieve information about nearby restaurants and hotel, and acquire local traffic. The current growth of LBS services, companies is boosting their efforts to automatically locate consumers for advertising and marketing purposes. LBS software services are also considered for mobile applications to represent a new business sector and E-commerce. Spatial index is used by spatial databases to optimize spatial queries.

Spatial databases are fully-fledged databases that, in addition, enable the storage, retrieval, manipulation, querying, and analysis of geometries like points, lines, and regions representing, for example, the geometries of cities, roads, and states respectively. In order to handle spatial data efficiently, database management system needs an index mechanism that will output the data quickly according to their current location. There are many index structures such as K-D-B tree works only in points data, B- tree, R-tree, Hilbert-Curve tree, BSP-tree and Quad-tree. Most spatial database application uses R-tree indexing method because it is the most widely accessed method [1].

This study proposes a new algorithm to speed up the performance of k-Nearest Neighbour (k-NN) retrieval on spatial database. The proposed method named as R-tree based grid index is a hybrid index structure of R-tree and grid indexing technique [2].

Grid index is used for locating objects. As grid index is easy in implementations such as updating and creating index, it is simple and efficient way of spatial indexing. The grid index extracts only locations from nearest indices and sends these indices to R-tree. R-tree is used to retrieve k-nearest neighbour objects [3].

⁺ Corresponding author. Tel.: +959425003327.
E-mail : aungzawmyint@ucsy.edu.mm

Additionally, the remainder of this paper is organized as follows. In section II, we describe related works of this paper. In section III, we describe background theory. In section IV, we discuss the overview of proposed system. In section V, we explain computing grid index. In section VI, we describe construction of R-tree. In section VII and VIII, we discuss our expected experimental and then conclude the paper.

2. Related Works

Processing k-Nearest Neighbour (KNN) query based on location has been well studied in spatial database. R-tree method was proposed by Guttman in 1984 in order to handle spatial data efficiently, as required in computer aided design and geo-data applications and to extract the objects by current location [4, 5]. The proposed index used Distributed grid index from server transmission and clients examine the received index with the unique ID number to each grid-cell. Zhang et al. proposed a grid cell based continuous k-NN query processing method (CkNN) [6].

Inverted file-R* tree and R* tree – inverted file are geo-textual indices that loosely combine the R* tree and inverted file [7]. Hariharan, B. Hore, C. Li, S. Mehrotra [8] proposed the KR*-tree structure that captures the joint distribution of keywords in space and significantly improves performance. Z. Li, K. C. K. Lee, B. Zheng, W. C. Lee, D. L. Lee, X. [2] proposed IR-tree. This paper proposed to support efficient geographic document and IR-tree enables the pruning of textually and spatially irrelevant subsets.

As to k-NN query algorithm [3], its system has studied well in traditional databases. The idea of this system is to establish a static R-tree-like structure which is developed from Rtree. It cannot handle continuous queries and update queries. DJ Oneil et al. X. Cao, G. Cong, Christian S. Jesen, Jun J. Ng, Beng C. Ooi, N. T. Phan, D. [7] proposed Spatial Web Object Retrieval System (SWORS) that is capable of efficiently retrieving spatial web objects that satisfy spatial keyword queries. This system use IR tree and inverted file for index. SWORS supports location-aware top-k text retrieval (LkT) query and the spatial keyword group (SKG) query that retrieves a group of objects that cover the query keywords.

3. Background Theory

The rapidly expanding technology of mobile communication will give mobile users capability of accessing information from anywhere and anytime. It is also a distributed system with a network to communicate between different machines. Wireless communication is needed to enable mobility of communicating devices. In recent years, Location-Based Service (LBS) is growing rapidly among mobile users. As the mobility is the most distinguishing feature of the mobile computing environment, location becomes an important piece of information for LBS. Therefore, spatial database for spatial locations has become an important area of people's interest and research. A fundamental issue in this area is how to store and operate spatial data efficiently. Quickly executing k-Nearest-Neighbour (k-NN) and range queries in spatial database applications requires an informative and efficient index structure that can effectively reduce the search space.

A. R-tree Index Structure

R-trees index method was presented by Guttman [1]. R-tree is an index tree-structure that uses multi-dimensional indexes. R-tree derived from B-tree. An R-tree is a depth-balanced tree in which each node corresponds to a disk page (i.e., the number of entries in each node is limited). R-tree structure constructs tree structure that has root node with at least two children (unless it is a leaf). All leaves must be at same level. R-trees are based on Bounding Box. Objects are entirely contained inside the bounding box. The actual objects are recorded in the leaves point that are enclosed minimum bounding rectangles (MBRs). This is also the reason for the benefits R-trees have in the matter of dynamic indexing. But the bounding boxes used in R-tree nodes can overlap.

The following figure is to illustrate this situation for a simple two-dimensional structure. Figure 1 shows R-tree structure.

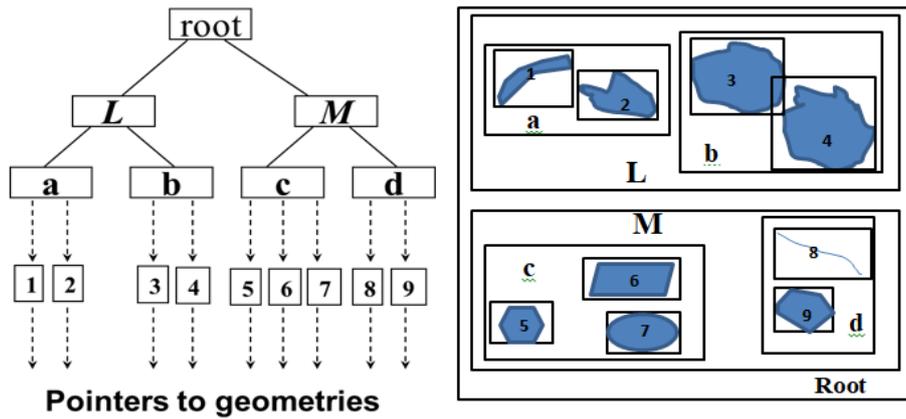


Figure 1: R-tree structure

B. Grid Index

Grid is a pre-defined partitioning index in which the region is partitioned into rectangular cells because it indicates the pre-defined spatial region. Grid covers a part of the space that object position is inside the boundaries of a grid cell and this object belongs to this cell. Šidlauskas et al [9] proposed the grid index which is one of the principal framework index structures for moving objects database. The performance of grid index commonly uses in multi-dimensional queries. The grid index file can be stored entirely in memory. The storage structure of a grid index file stores the size parameters m and n that is block pointer of the grid. Then index files stores the buckets of the grid. Grid is a 2-d array that every cell in the array matches to a square cell with a length of grid cell size. In Every grid cell in the array has a link to a list of buckets which contains the data object [10]. Each bucket has bucket size and metadata fields where metadata contains the next bucket, the number of entries and pointer to the next bucket in Figure 2.

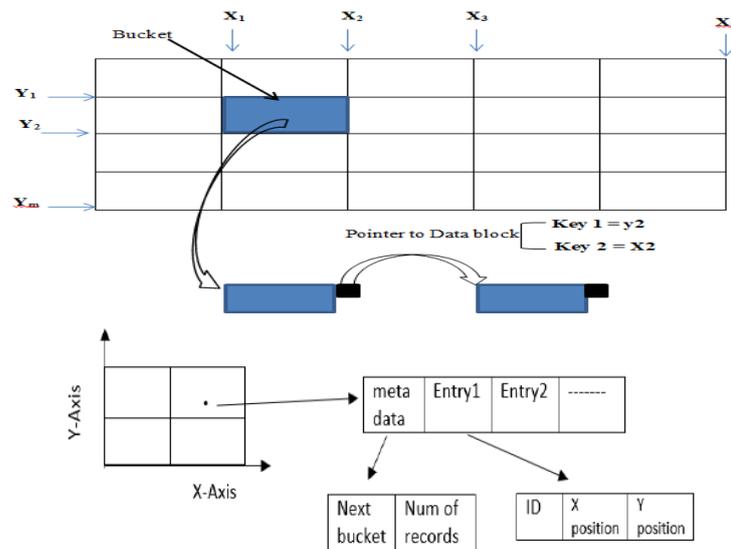


Figure 2: Grid Index Structure

4. Proposed System

The proposed system is explained in detail. This system offers k nearest neighbour results to the user based on their current location quickly by using R-tree based grid indexing technique.

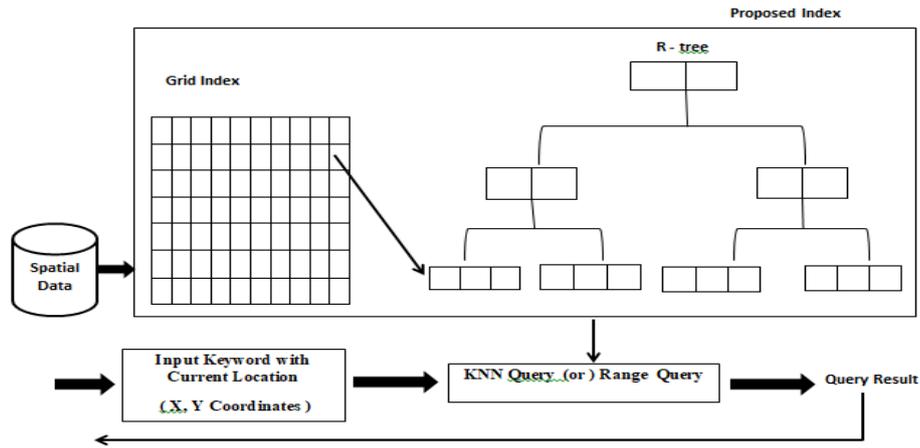


Figure 3: System Overview

First of all, locations of objects (latitude and longitude values) can be acquired via GPS in user device. The objects are located in two dimensional grid cells in Figure 3. Then, it identifies coordinates of the top left corner and right bottom corner within specified area. We collect clinics, mini-marts, restaurants, shops, and stores within above coordinates. Then index for each collected location and current location are computed. Thus nearest indices around index of current location can be acquired. The grid index extracts only locations from nearest indices and sends these indices to R-tree. Therefore, the usage of large memory space in R-tree can be reduced. In R-tree, nearest indices that are sent by grid index are constructed as a tree. R-tree is composed of root node, intermediate node and leaf node. R-tree sort active branch list (ABL) by ordering MINDIST. ABL is a list that calculates the distance of objects.

5. Computing Grid Index

The grid index is composed by grid cells. Each cell represents a region of space generated by partitioning the domain using a uniform grid, which can then be assigned unique identifiers and used for spatial indexing purposes. It uses coordinates of objects and sorts them into grids, where grids have their own identifier index for faster querying [10]. It is simple and efficient way of spatial indexing. A grid index file is organized into a two dimensional structure that is shown in Figure 4. The geographically related data (i.e.: points that are close to each other) are stored in the same data block. The grid index cannot represent objects, it can only present points. In fact, the only kind of index that can handle where queries: is given by a location (i.e., coordinate), and then find the objects that contains the location. Therefore, performance of grid index for range queries is to find objects that are located within a certain range and such performance of grid index for nearest neighbor queries is to find the nearest neighbor of a data point.

```

Input latitude, longitude
Output grididx
Begin
Latlng ygentl = (17.136160,96.009705);
LatLng ygnbr = (16.7851054,96.1791551000);
latdiff = (ygentl.latitude- latitude);
londdiff = (ygentl.longitude - longitude);
w = (latdiff / factor);
h = (londdiff / factor);
grididx = (h * maxwidth) + w;
End

```

Figure 4: Grid Index Algorithm

6. Construction R-tree

R-tree can be used (nearest neighbor) search for some places. In R-tree, nearest indices that are sent by grid index are constructed as a tree. R-tree is composed of root node, intermediate node and leaf node. The processing step of the R-tree construction for nearest neighbour search is shown in Figure 5.

In this algorithm, p is query point and points in a node are objects. MBR means Minimum Bounding Rectangle of each leaf node. The two ordering metrics in R-tree are $MinDist$ and $MaxDist$. $MinDist$ is the distance of object O from query point P . $MinMaxDist$ is the minimum of the maximum possible distances to a face or vertex of the MBR containing O . $MinDist$ and $MinMaxDist$ offer a lower and an upper bound on the actual distance of object O from the query point P respectively.

```

begin
  if node is a leaf node
    Compute distance from p to all points contained in current node.

    Update the current best NNs and Current best distance if the distances
    between p and all points are less than the maximum distance of nearest
    NNs list.
  else
    Sort the MBRs contained in current node using MINDIST in ascending
    order.
    This sorted list is denoted the Active Branch List (ABL).
    Compute the minimum of the MINMAXDIST of all MBRs.
    Apply pruning strategy to the ABL to remove unnecessary branches.

    Iterate on this ABL until it is empty. For each iteration step, apply pruning
    strategy recursively to its children node.
    When the ABL is empty,
    Return nearest neighbor.

```

Figure 5: Nearest Neighbor Search on R-tree

7. Experimental Result

The performances evaluations of the proposed system are shown in Figure 7. The processing time and response time are covered. Moreover, the comparisons on evaluation results for the proposed indexing scheme, no indexing scheme, and traditional R-tree indexing scheme. In order to implement an efficient k-NN search application needs a structured framework shown in Figure 6.

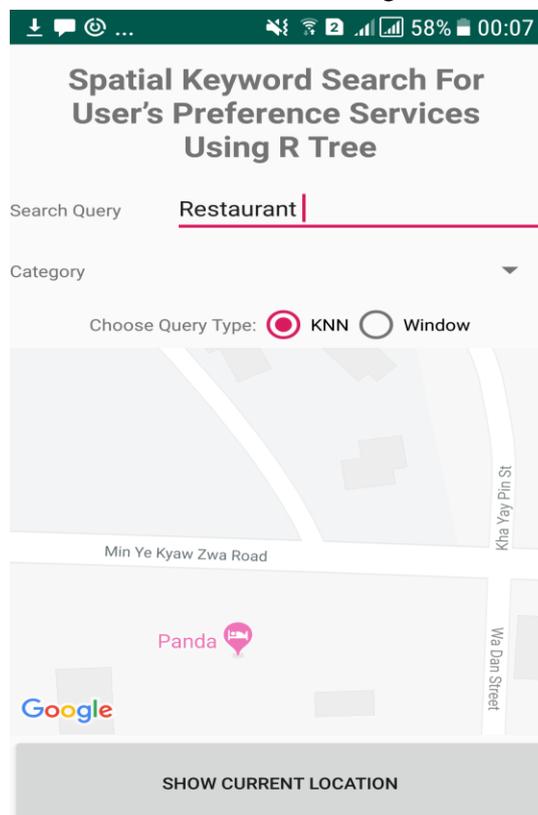


Figure 6 : User Interface of Proposed System

Table1. Excepted Results for Proposed System

Service Types	R-tree Index	Proposed Index
Restaurants	40	15
ATM Machines	30	12
Hospitals	55	20
Hotels	70	40
Cinemas	68	43
Gift Shops	77	34

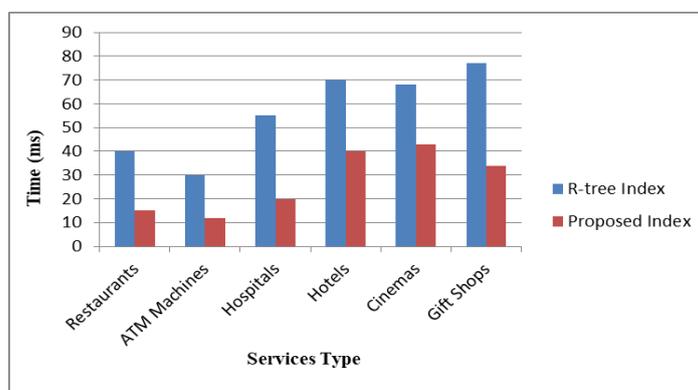


Figure 7: Processing Time of Different Indexing Schemes

8. Conclusion

Nearest neighbor search such as k-nearest neighbor query and range query based on user's current location is very important to get useful geo-information efficiently. A k-nearest neighbour query retrieves k objects in a given objects set which are closet to the query point. Processing k-nearest neighbour query efficiently requires spatial indexing methods. In this paper, we proposed grid based R-tree indexing method system. The system can be used effectively the retrieval of user relevant geo-spatial information. The proposed system can reduce searching time and reduce unnecessary node visiting cost. The memory space can be reduced by using that proposed indexing techniques. The speed performance of this system outperforms the traditional R-tree system. As further work, we will consider on moving objects retrieval using the proposed index structure.

9. References

- [1] A. Guttman, "R-Tree: A dynamic index structure for spatial searching", ACM SIGMOD, pp.47-57, 1984
- [2] Z.Li, K.C., K.Lee, Zheng, W.C.Lee "Ir-tree: An efficient index for geographic document search", IEE TKDE, 23(4):585-599, 2011.
- [3] G.Cong, C.S.Jensen, and D.Wu : Efficient retrieval of th top-k most relevant spatial web objects:, PVLDB, 2(1):337-248, 2009.
- [4] I.D Felipe, V. Hristidis, and N. Rische , " Keyword search on spatial databases", in ICDE, pages 656-665, 2008.
- [5] NRSKF. Vincent, " Nearest Neighbor Queries" , ACM SIGMOD , 1995.
- [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] X.Cao, G.Cong, Christian S. Jesen, Jun.j, N.T.Phan, D. Wu "SWORS:A System for the Efficient Retrieval of Relevant Spatial Web Objects".
- [8] R. Hariharan, B. Hore, C.Li, S.Mehrotra, " Processing spatial-keyword (sk) queries in geographic information retrieval(gir) systems", pages 655-665, 2008.
- [9] D. Šidlauskas, S. Šaltenis, C. W. Christiansen, J. M. Johansen, and D. Šaulys, "Trees or grids?: indexing moving objects in main memory," in Proceedings of thnternational conference on Advances in Geographic Information Systems, 2009, 236-245.
- [10] Kwangjin Park, "Location-Based Grid Index for Spatial Query Processing", Expert Systems with Applications 41(2014) 1294–1300.