

Radix Priority Search Tree in Circular Range Searching for GIS

Linn Linn Phway
University of Computer Studies, Yangon

Myint Myint Sein
University of Computer Studies, Yangon

ABSTRACT

This paper presents a method for circular range searching in 2D geographical data for GIS. The proposed is based on the priority search tree (PST) developed by Edvard T. McCreight in the mid eighties. The Priority Search Tree is a data structure used for performing semi-infinite range queries. In the solution presented in this paper, the two operation of PST, Insert operation and EnumerateRectangle are combined for getting all data-points in a circular range. A query in this new data structure returns all points in circular range. The proposed circular range searching is the querying of all datapoints P which are the subset of dataset R and also locating in the circular range with the radius and the center is query point.

Keywords:

Circular Range Search, GIS, Priority Search Tree.

1. INTRODUCTION

Circular range searching is used in extracting surrounding services for user's define distance which is allows user to search for businesses and other requirement services that satisfy the user define geographical constraint. This system offer many advantages to the users to retrieve the information about their current locaion and retrieve data to get more useful informaton near from their location. This system allows users to access geographic data from portable devices. There is necessary to defined specific location range so the user can decide to choose their destination more easily and quickly. The system provice for searching desired servics on desired distance range around user according to user preferences.

The areas of computer science where the data is geometric computer graphics such as geographic information systems (GIS) and spatial data structures are important. The data structures store a set R of geometric objects such that various queries can be answered efficiently. The most widely used queries are range queries, which request for all objects in R in a query range Q . Another important usage queries is nearest-neighbor queries, where one asks for the object from R closest to a query point [4].

In the case of range searching consists of pre-processing a set R of objects, in order to determine which objects from R intersect with a query object, called. For example, R may be a set of points corresponding to the services of the area of city, and we want to find the services within a certain latitude and longitude range. Range searching problems and data structures are the basic important study area of computational geometry and the range searching problem finds applications not only in areas of geometrical data processing such as geographical information systems (GIS), and computerized aided drafting (CAD), but also in databases [8].

A range search is a query where the query key is a range. Searching for a range leads to more than one value is being reported. In one dimension the data set can be seen as points on a line, and the query key will be an interval within which

all points shall be reported. Such a query can be efficiently solved using a regular binary tree, or even a sorted array [8].

2. RELATED WORK

Haibo Hu and Dik Lun Lee (2006) consider the (hyper)rectangles range and propose the method for efficient in-memory processing and secondary memory pruning techniques for Range Nearest Neighbor queries in 2D and high-dimensional spaces. Their techniques are generalized for kRNN queries for getting the k-nearest neighbors for every point in the defining range. Their idea is to speed up any type of NN query by supporting with auxiliary solution-based index EXO-tree [2].

Jon Louis Bentley and Jerome H. Friedman listed some search methods such as Se Sequential Scan, Projection, Cells, Kd Trees, Range Trees, k-ranges and Quad Tree for range searching. They provide not only the methods for attacking multikey searching problems and also display the data structures available for solving the particular problem of range searching [3]

The min-max priority search tree is the combination of a binary search tree and a min-max heap. The min-max priority search tree algorithm to enumerate all maximal empty axis-parallel rectangles amongst points in a rectangular region R in R^2 in $O(m \log n)$ time with $O(1)$ extra-space, where m is the total number of maximal empty rectangles introduced in Minati De, Anil Maheshwari, Subhas C. Nandy, Michiel Smid(2011) [5].

Robin Y. Flatland, Charles V. Stewart (2000) present algorithms that obtain efficient overall query times by performing novel searches of multiple range trees and extending k - nn trees. When queries $\Theta(N)$ points or require $E = \Omega(N)$ extensions with other algorithms in two dimensions, the overall retrieval time of their algorithms is $O(E+N)$. Their research give efficient algorithms for two instances of extending neighborhood problems stated as extending orthogonal range reporting and extending L_∞ k-nearest neighbors [6].

Sunil Arya, David M. Mount (2006) show that given a bounded range Q of diameter ω and $\epsilon < 0$, an approximate range query. They show that in any fixed dimension d , a set of n points in R^d can be preprocessed in $O(n \log n)$ time and $O(n)$ space, such that approximate queries can be answered in $O(\log n + (1/\epsilon)^d)$ time. They also present a lower bound for approximate range searching based on partition trees of $\Omega(\log n + (1/\epsilon)^{d-1})$, which implies optimality for convex ranges [7].

This system is possible for users to use GIS and for location searching. In this system, GPS device is used to define and collect geographic coordinates for all places on earth. Download these points onto computer and integrate them with mapping programs. The own Data-Set is created for Yangon region which containing the latitude, Longitude. Calculating a Ring Range around a point with user preference distance and generating all latitudes and longitudes at a given circle range

to extract accurate services according to their geographic coordinates. The specific information according to user is extracted by the preference ring range.

3. PRIORITY SEARCH TREE

Priority search tree algorithms can also be used to enumerating all intersecting pairs of rectangles. Priority search tree, for representing a dynamic set D of ordered pairs $[x,y]$ over the set $0,1,\dots,k-1$ of integers and a set of algorithms that operate on the priority search tree to implement the following operations [1]:

InsertPair (x,y) : Insert a pair $[x,y]$ into D .

DeletePair (x,y) : Delete a air $[x,y]$ from D .

MinXInRectangle (x_0,x_1,y_1) : Given test integers x_0, x_1 and y_1 among all pairs $[x,y]$ in D such that $x_0 \leq x \leq x_1$ and $y \leq y_1$, find a pair whose x is minimal.

MaxXInRectangle (x_0,x_1,y_1) : Given test integers x_0, x_1 and y_1 among all pairs $[x,y]$ in D such that $x_0 \leq x \leq x_1$ and $y \leq y_1$, find a pair whose x is maximal.

MinYInXRange (x_0,x_1) : Given test integers x_0 and x_1 , among all pairs $[x,y]$ in D such that $x_0 \leq x \leq x_1$, find a pair whose y is minimal.

EnumerateRectangle (x_0,x_1,y_1) : Given test integers x_0, x_1 and y_1 , enumerate those pairs $[x,y]$ in D such that $x_0 \leq x \leq x_1$ and $y \leq y_1$ [1].

Priority search tree have basically two types;(i) Radix Priority Search Tree and (ii)Balanced Priority Search Tree. In our propose system radix priority search tree is emphasized. A radix priority search tree representing a set of n pairs occupies $O(n)$ words of storage, where each word is $O(\log k)$ bits long. In conventional algorithm-analytic terms, a radix priority search tee is a linear-space data structure [1].

The first invariant is a priority queue invariant on y -values. It asserts that for any node t in a radix priority search tree, if $t.left$ is not NIL then $t.p.y \leq t.left.p.y$, and if $t.right$ is not NIL then $t.p.y \leq t.right.p.y$. This first invariant constrains only direct ancestor-descendant relations; it does not constrain sibling relations at all [1].

The second invariant is a radix search tree invariant on x -values. It asserts that associated with each node t in a radix priority search tree is an x -interval $[lower, upper]$ within which $t.p.x$ lies. The x -interval associated with the root of the radix search tree is the interval $KeyBound$. For any node t such that $t.left$ is not NIL, the x -interval associated with the node $t.left$ is $[lower .. floor((lower+upper)/2))$. For any node t such that $t.right$ is not NIL, the x -interval associated with the node $t.right$ is $[floor((lower + upper)/2)..upper]$ [1].

The y -value of new pair smaller than the root, the new pair “beats” the already root pair in $InsertPair$ procedure. If not, then the new pair is inserted recursively into either the left or right subtree, determined by its x -value.

```

PROCEDURE InsertPair(VAR t : RPSTPtr, newPr : Pair,
    lowerX : KeyRange, upperX : KeyBound);
VAR
    p : Pair,
    middleX : KeyRange
BEGIN
    IF t = NIL THEN
        BEGIN
            NEW(t);(*add a newleaf node*)
            t.p := newPr;
            t.left := NIL;
            t.right := NIL;
        END
    ELSE IF t.p.x <> newPr.x(*assumes unique values*)
    THEN
        BEGIN
            IF newPr.y < t.p.y THEN(*new pair beats existing one*)
                BEGIN p := t.p; t.p := newPr END
                (*exchange new/ existing*)
            ELSE p := newPr;
            middleX := (lowerX + upperX) DIV 2;
            IF p.x < middleX
                THEN InsertPair(t.left, p, lowerX, middleX)
                ELSE InsertPair(t.right, p, middleX, upperX);
        END;
    (*ELSE this pair already present, so don't insert it*)
    END;(*of InsertPair*)

```

EnumerateRectangle is a depth-first enumeration that calls the functionReport whenever a pair is found within the constraint rectangle. Report returns TRUE if the enumeration should continue, and FALSE if it should terminate [1].

EnumerateRectangle return all elements in the rectangle defined by the lines, $x=x_L, x=x_R, y=y_0, y=y_T, x_L \leq x \leq x_R$ and $0 \leq y \leq y_T$.

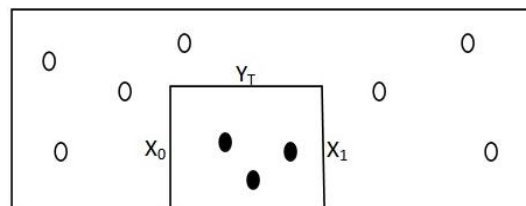


Fig 1: Example of EnumerateRectangle Operation

```

Function EnumerateRectangle
(t : RPSTP, x0, x1, y1, KeyRange);
Function Report (Pair) : BOOLEAN;
lowerX : KeyRange; upperX : KeyBound) : BOOLEAN;
VAR
    continue : BOOLEAN;
    middleX : KeyRange;
BEGIN
    IF t <> NIL THEN
        IF t.p.y <= y1 THEN
            BEGIN(*node passes y test*)
                IF (x0 <= t.p.x) AND (t.p.x <= x1) THEN
                    continue := Report(t.p)
                ELSE continue := TRUE;
                    middleX := (lowerX + upperX) / 2;
                IF continue AND (x0 < middleX) THEN
                    continue := EnumerateRectangle
                    (t.left, x0, x1, y1, Report, lowerX, middleX);
                IF continue AND (middleX <= x1) THEN
                    continue := EnumerateRectangle
                    (t.right, x0, x1, y1, Report, middleX, upperX)
                EnumerateRectangle := continue;
            END
        ELSE EnumerateRectangle := TRUE
            (*node fails y test*)
        ELSE EnumerateRectangle := TRUE;
            (*empty subtree*)
        END;(*of EnumerateRectangle*)
    
```

4. PROPOSE SYSTEM

In this system, the circular range searching is developed by using priority search tree and Vincenty's method. Vincenty's method is used to define the rectangular range which covers the circular with user requirement range. By using rectangular range is defined in preprocessing step, the execution time can be reduced. Vincenty's formulae are two related iterative methods used in geodesy to calculate the distance between two points on the surface of a spheroid, developed by Thaddeus Vincenty (1975) [9].

They are based on the assumption that the figure of the Earth is an oblate spheroid, and hence are more accurate than methods such as great-circle distance which assume a spherical Earth. The first (direct) method computes the location of a point which is a given distance and azimuth (direction) from another point. The second (inverse) method computes the geographical distance and azimuth between two given points. They have been widely used in geodesy because they are accurate to within 0.5 mm (0.020") on the Earth ellipsoid [9].

Firstly, we use Vincenty's direct method to determine the maximum and minimum bounded that cover for the user requirement range. When defining the ring range for user preference distance, Vincenty Direct method computes the location of a point which is a given distance and azimuth (direction) from another point.

Table1: Sample of Storage Data-type

Place Name	Service Type	Latitude(N)	Longitude(E)
Shwe Dagon Pagoda	Pagoda	16 47 54.37	96 8 58.07
Sule Pagoda	Pagoda	16 46 27.84	96 9 31.68
Parami Children Hospital	Hospital	16 50 59.93	96 9 22.60

Dagon Centre-1	Shopping Centre	16 48 14.49	96 8 15.37
Hel Pin Bus stop	Bus stop	16 47 28.07	96 8 27.23
Link Lan Bus stop	Bus stop	16 48 22.34	96 8 50.35

Latitude and Longitude of user current location as *lat*₁, *lon*₁, user's define range is used as *s* and *α* have four variable such as 0°, 90°, 180° and 270°. In geographic coordinate system, true north is 0° and it is used for maximum latitude, 90° is maximum longitude, 180° is minimum latitude and 270° is minimum longitude. After processing this step, we get the two pairs of latitude and longitude which are minimum latitude, minimum longitude and maximum latitude, maximum longitude. These two pair are cover the user define range [8].

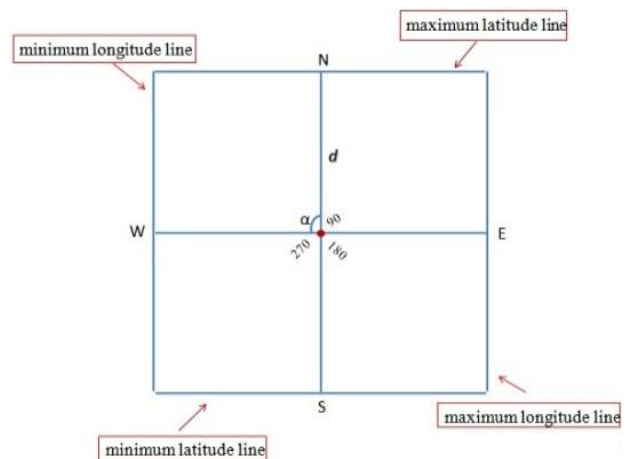


Fig 2: Defining Max/Min Bound

When we get the min/max latitude and longitude, these are use as input to the proposed algorithm which is modifying the EnumeratedRectangle to eliminate the data point which are not include in user define rectangular range.

Finally, we are using the binary search tree to exactly extract the data point in circular range. In this part Vincenty's inverse formula is used to determine the distance between two points, i.e., from the current location to service location in a rectangular range.

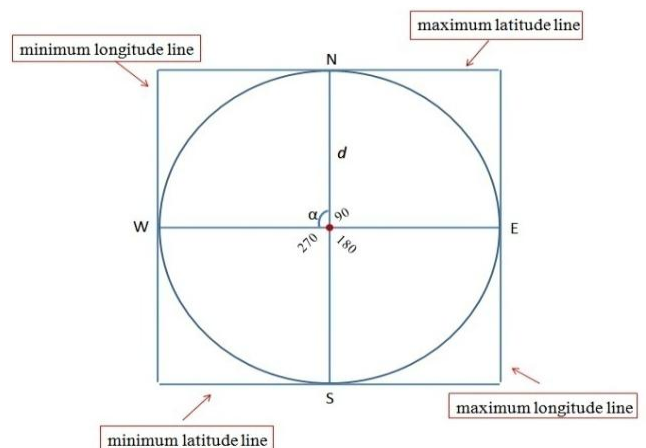


Fig 3: Defining Circular Range

Binary search trees provide an excellent structure for searching a list and at the same time for inserting and deleting data into the list. The common properties of binary search trees are as follows:

- ~ The left sub-tree of a node contains only nodes with keys less than the node's key.
- ~ The right sub-tree of a node contains only nodes with keys greater than the node's key.
- ~ The left and right sub-tree each must also be a binary search tree.
- ~ Each node can have up to two successor nodes.
- ~ There must be no duplicate nodes.
- ~ A unique path exists from the root to every other node [10].

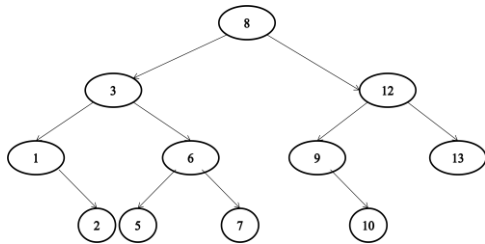


Fig 5: Example of Binary Search Tree

Proposed Algorithm ($R; x_0; y_0; x_{mi}; x_{mx}; y_{mi}; y_{mx}$)

1. Input $R = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$;

2. Define: $t = Ptr$;

$newPr = pair \in R$;

$x_{mi} = \minLat; x_{mx} = \maxLat$;

$y_{mi} = \minLon; y_{mx} = \maxLon$;

$x_0 = user's\ current\ x; y_0 = user's\ current\ y$;

$t.p = (x_0, y_0)$;

$D = t.p$

3. for $i = 1; n$;

if ($x_{mi} \leq newPr.x$ and $x_{mx} \geq newPr.x$) then;

if ($y_{mi} \leq newPr.y$ and $y_{mx} \geq newPr.y$) then;

$t_i.left.p = newPr$;

$D = t_i.left.p$;

else

$t_i.right.p = newPr$;

end

else

$t_i.right.p = newPr$;

end

end

4. Return $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_r, y_r)\} \subset R$;

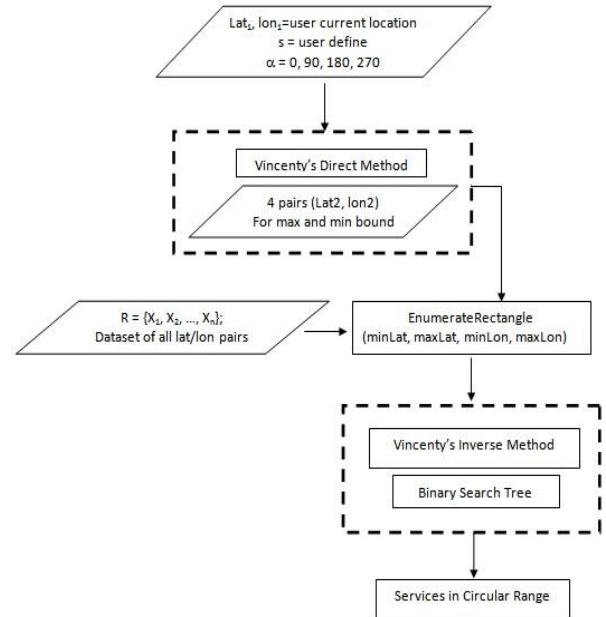


Fig 4: System flow diagram

5. RESULT AND DISCUSSION

Fig 6: Starting window

Service In Rectangular Range !

	Services	Distance From User (Meters)
1	3th Street Bus Stop	542.69
2	Hel Pin Bus Stop	1219.7
3	Link Lan Bus Stop	889.69
4	Shwe Gone Taing Bus Stop	935.04
5	Shwedagon Pagoda (East Gate) Bus Stop	246.88
6	Shwedagon Pagoda (North Gate) Bus Stop	312.91
7	Shwedagon Pagoda (South Gate) Bus Stop	546.20

Fig 7: Showing Services in Rectangular Range

Service are ordered by their distances from user's current location.

	Service	Distance From User (Meters)
1	Shwedagon Pagoda (East Gate) Bus Stop	246.88
2	Shwedagon Pagoda (North Gate) Bus Stop	312.91
3	3th Street Bus Stop	542.69
4	Shwedagon Pagoda (South Gate) Bus Stop	546.20
5	Link Lan Bus Stop	889.69
6	Shwe Gone Taing Bus Stop	935.04

Fig 8: Showing Services in Circular Range

In this paper, we demonstrate about the modifying EnumerateRectangle operation of radix priority search tree for circular range searching. The analysis of EnumerateRectangle is complicated because each Report calls itself recursively on both sons of a tree node. EnumerateRectangle can visit a node only if the pair in the node's father was Reported. It follows that if EnumerateRectangle in fact enumerates S pairs, it runs in a time bound of $O(\log k+S)$. This is true whether or not the enumeration is terminated by the Report function [1]. In our system, we are only searching all nodes in defining range in dataset not identically to the EnumerateRectangle operation. In EnumerateRectangle operation, the root node is not NUL. Therefore our system is similar to the combination of Insert operation and EnumerateRectangle operation. This implies the time of $O(\log k)$ on depth of recursion and the number of nodes visited.

6. CONCLUSION

The circular range searching system is useful for foreigner to easily search bus stop, restaurant, department station, etc., and easily to estimate how far from there. Most current location based services are providing on online and need wireless connection. The system is based on geographic coordinates system and also useful in offline. The system provide for searching desired services on desired distance range according to user preference. This system is planned to extend by enumerating the nearest neighbour for all services in the circular range. When the user is on the move, it is necessary to get the updated information in time, so we idealize to calculate the moving rang query and defying shortest part for end users.

7. REFERENCES

- [1] Edward M. McCreight "Priority Search Tree" (Society for Industrial and Applied Mathematics Journal, Vol. 14, No.2, May 1985.)
- [2] Haibo Hu and Dik Lun Lee "Range Nearest-Neighbor Query" (IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 1, January 2006.) 24)
- [3] Jon Louis Bentley, Jerome H. Friedman, "Data Structures for Range Searching" (Computing Surveys, Voi. 11, No. 4, December 1979)
- [4] M. de Berg, J. Gudmundsson, M. Hammar, M. Overmars, "On R-trees with low query complexity", (Computational Geometry, Volume 24, Issue 3, April 2003, Pages 179–195)
- [5] Minati De, Anil Maheshwari, Subhas C. Nandy, Michiel Smid "An In-Place Min-Max Priority Search Tree" (23rd Canadian Conference on Computational Geometry, 2011)
- [6] Robin Y. Flatland, Charles V. Stewart "Extending range queries and nearest neighbors" (Computational Geometry 17 (2000).
- [7] Arya, David M. Mount "Approximate Range Searching" (Computational Geometry: Theory and Applications, 17 (2000), 135-163.)
- [8] http://en.wikipedia.org/wiki/Range_searching
- [9] <http://www.movable-type.co.uk/scripts/latlong-vincenty-direct.html>
- [10] http://en.wikipedia.org/wiki/Binary_search_tree