Based on the Half-space Pruning to Continuously Monitoring Reverse KNN

Du Xuedong, Xie Tingting
College of Information Science and Engineering
Shandong University of science and technology.
Qingdao 266510,China

***Abstract* -Reverse k Nearest Neighbor (RKNN) query is proposed based on the k Nearest Neighbor (KNN) query, which can be used to evaluate the influence of the query objects. At present RKNN query algorithms are mostly based on static or continuous objects, however, with the widespread of mobile device, algorithms for efficiently answering queries about large populations of moving objects are gaining interest. Given a group of nearby space objects as the query input, the authors propose the continuously monitoring RKNN algorithms based on the half-space pruning, compute the minimal enclosing circle containing the query objects and consider the objects in the circle as a whole. Then, in order to get the query's final RKNN results, we design a filter method based on R-tree and a refinement process to realize continuous RKNN monitoring.**

Keywords-Half-space Pruning, RKNN, Continuously Monitoring, minimal enclosing circle

I. INTRODUCTION

RNN (Reverse nearest neighbor) query is the most important problem in the spatial database and is a new query type based on nearest neighbor query. The situations of traditional RNN query are the static query point or non-continuous query point set, however, with the development of database technology, dynamic continuous monitoring query becomes research hotspot.

RNN query problem is proposed by F. Kom and S. Muthukrishnan in 2000[1] and is received everyone's attention in subsequent years. The basic methods include: static and dynamic reverse nearest neighbor query issues, pretreatment methods and space pruning reverse nearest neighbor query problems, and so on. The data pretreatment method is just efficient in dealing with static and less updated data, but the query performance will drop dramatically in frequent updated data.

R*K*NN (reverse *K* nearest neighbor) of half-space pruning method is proposed by Y. Tao and D. Papadias, taking advantage of the geometric properties of RNN. This method is flexible because it does not require pretreatment of the nearest neighbor information.

Continuous RNN query will recalculate query results when the location of objects is changed. This paper calculates tne minimal enclosing circle of a group of inquire objects and considers the inquire objects as a whole. As long as inquire goal is still in the minimal enclosing circle, we do not recalculate.The application and definition of the minimal enclosing circle reduces the computational cost, meanwhile, also reduces the communication costs between client and server.

R-tree index and filter-refinement framework model are adopted in this paper. In filtering stage, we calculate the minimal enclosing circle containing all inquire points, then using tne minimal enclosing circle and a group of subset of data points to calculate R*K*NN search area, whose algorithm is CINCH. All objects in this query area are candidate results; In refining phase, improved Range-*k* validation methods is applied to query and removed non-results of candidates, thus finally we get search results. Experimental results show that the proposed algorithm's efficiency is optimal.

II. RELATED RESEARCH

A. the defination of RKNN[1]

Given a set of any multidimensional points S and a query point $q \notin S$, a RKNN query retrieves points $q \in S$, $\mathrm{D(p,q)} \leq D(p,p')$ where D is a distance metric that is assumed to be Euclidean distance in this paper, and p' is the *k*th nearest point to p.

B. the defination of Range-k(p,k,r)[2]

Judged whether *k* is larger than the number of point, the point is that the distance between point p and around p is less than r, whether if {o | dist (p, o) < r} < *k* is tenable, then returning true, otherwise false.

C. the half-space pruning

Considering perpendicular bisector between inquire point *q* and any data point *p*, where a bisector is shown that divides the space into two half-spaces [3]: *q-half-space* (the half-space containing *q*) and *p-half-space* (*p* located in this half-space). Any point that lies in *p-half-space* is always closer to *p* than *q* and cannot be the RNN for this reason. Normally, this requirea multiple half-spaces to prune a query point region. As shown in Figure 1, for example, half-spaces between *q* and *c* (*Hq* and *Hc*), any points located in *Hc* region are closer to *c* and cannot be the RNN of *q* for this reason. This figure is given the half-spaces between *q* and *a*, *q* and *b*, *q* and *c* respectively, if *q* is their nearest points then we verify candidate *a*、*b*、*c*.

Published by Atlantis Press, Paris, France.

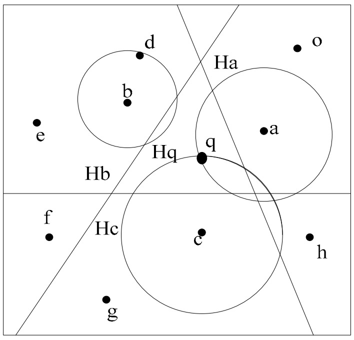


Figure 1. half-space pruning

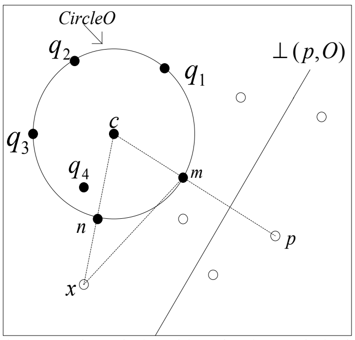


Figure 2. the relationship of point and circle

D. *INCH(Intersections' Convex Hull) spital shrinking method*

Tan[4] also employed filter-refinement framework model and proposed INCH algorithm which is based on half-space pruning, its basic ideas: to compute the search scope of a RKNN query which constraints the search space, and every time when a new candidate is found, the search scope will be narrowed, ultimately we get the candidate set. INCH algorithm is better tan half-space pruning in the spital pruning ability, but the method only aims at single inquire point.

E. *the minimal enclosing circle*

The minimal enclosing circle of fixed-point set that contains all fixed-points and radius of which is minimal, Figure2 shows, circle O is the minimal enclosing circle of point set {q1,q2,q3,q4}. The calculated method of the minimal enclosing circle refer to reference[5].

F. *vertical bisector between point and circle*

The vertical bisector between point p and circle O by point c for center and r for radius, that is betweenr poing p and the intersection point m with linear *l*(p,c) and circle O, using ⊥(p,O)said. *l*(p,c) is formed by point p and c. Figure 2 shows that the space is divided into two parts, including point p called p-half-space (also general designation with ~O-half-space), containing circle O called O-half-space.

G. *distance between point and circle*

The distance between point p and circle O by point c for center r for radius is donated by d(p,O), when dist(p,c)>r, d(p,O) = dist (p, c) - r; and when dist(p,c)≤r, d(p,O) = 0. As shown in Figure 2 that here is point x of external circle O, d(x,O)=dist(x,n), n is intersection point with linear l(x,c) and circle O.

III. R*K*NN QUERY ALGORITHM

In this paper, we adopt index structure of R-tree by pruning method, under the enlightenment of half-space pruning and INCH algorithm, use filter-refinement framework model. In filtering stage, determine R*K*NN search area of the minimal enclosing circle and get candidate set. And in refining stage, apply Range-*k* validation method to eliminate errors, and get a final result set.

A. *filtering procedure*

The filter-refinement framework model is adopted for R*K*NN query. Filtering stage gets a group of candidate set through quickly triming some ibjects. Considering separatly all inquire points will complecate matter, so taking into account the minimal enclosing circle of all inquire points as a whole then we prune.

First, calculate the minimal enclosing circle O of R*k*NN inquire point set Q, think over circle O as a whole and use the given spatial data subset S⊆P to make vertical bisector betweenr circle O and every point of subset S, so space is divided into several polygon. As shown in Figure 3 (a) that S = {p1, p2, p3, p4, }. Then calculate each polygon's level value which is the ~O-half-space's number of this polygon. Figure 3 (b) shows, if polygon's level value is less than *k*, explaining that the polygon is within *k*-1 ~O-half-space at most, such polygons together form R*k*NN candidate results of all points in circle O, Figure 3 (c) shows thick border area. Many polygons formed candidate area is not fit for store, so shaped convex polygon of thick border area as Figure 3(d). The convex polygon is R*K*NN search area. All data objects in the search area and set S are candidate.

B. *CINCH algorithm*

Based on the filter-refinement framework model to realize R*K*NN query. The Search area determined in filtering phase, is actually R*K*NN search area of all points in the minimal enclosing circle. Because R*K*NN inquire points are in the circle point set, data points within the search area are also R*K*NN inquire candidate, and the search area is also the search area of R*K*NN query. Proposed CINCH algorithm views the minimal enclosing circle as a whole to calculate the search area. The basic idea of CINCH algorithm: given an object set S ∈ P, P is the whole dataset, for all objects, respectively make vertical bisector with circle O, thus space will be divided into many a polygons, calculate level value of every polygon, formed convex polygons whose level value is less than *k* is namely the search area. Union set with objects of the search area and set S is R*K*NN candidate set.

Algorithm 1.CINCH（O,*k*,S）

Input: O: tne mimimal enclosing circle including inquire points; *k*: *k* value of R*K*NN; S: subset of the whole data set

Output: search area formed by convex polygon

Description：

STEP1: calculate vertical bisectrix between each object of set S and circle O;

STEP2: calculate intersection point;

STEP3: calculate level value of each intersection point

STEP4: put the intersection, its level value is less than *k,* into set I

STEP5: return convex polygon area formed by all points of set I.

C. *Filtering algorithm*

According to given a set of data object $S \in P$, first, calculate the minimal enclosing circle O of inquire set Q, then calculate search area SR by making use of CINCH algorithm. Based on set S and SR, the candidate set is $S \cup S'$, $S' = SR \cap (P - S)$.

Set S can affect the size of the candidate set and can also influence the efficiency of refining stage to exclude incorrect candidates. And the search area has the following properties: (1) with the increase of objects in S, the search area will increasingly curtailed. (2) input object closer circle O into set S,the search area will be more reduced.

We adopt iteration method according to above two properties. Let S=φ, successively add data object to S, thus diminish search space. We adopt R-tree to realize filtering algorithm, the method of extended S is that point closer circle O has higher privilege joined S. Use R-tree as data index structure, sort data in ascending order, the data object outside O join S in order.

Algorithm 2. FCINCH-Filter(O,*k*)

Input: O: tne mimimal enclosing circle including inquire points; *k*: *k* value of R*K*NN

Output: candidate set Scnd including inquire results

Description：

```
S=φ；Scnd=φ;SR=the whole data space
Initialize a min-heap H with (root entry of R-Tree,0)
while H is not empty do
    de-heap an entry (p,key)
    if(p.MBR and SR jion)then
        if(p is a index node)then
            for( every node m of p)do
                insert(m,min-dist(m,c))into H
        else if(p is child node)then
            for( every data object n of p)do
                insert(n,dist (n,c))into H
        else if Scnd=Scnd∪{p}
            if(key>0)then
                S=S∪{p}
                SR=CINCH(O,k,S)
return Scnd,S
```

FCINCH-Filter algorithm sets up a minimum pile of H formed (p, key) to traverse - tree by using breadth first search, and p cloud be an index node of R-tree, also may be child node, or a data object; key is minimum distance between p and circle O; S is initialized by empty and set aside for data set formed search area SR; Scnd is initialized by empty and used to store candidate set. H is initialized by inserting root node of R-tree. In the process of traversing R-tree, we choose object which is nearest to the center of circle c to insert into Scnd, if p's corresponding key value is greater than 0, we insert p into S, because all data objects in circle O is a whole, it is meaningful to find the data object outside circle O to join S (key = 0 says that p is in circle or on the circle), meanwhile, CINCH algorithm will cut search area. Because the search space is iterative shrinking with data object continuously join S, when CINCH algorithm has a new data object, need to update and mainten search area and related information (for example, perpendicular bisector, intersection point and their level value).

D. *Refining stage*

In refining stage, exculde incorrect objects of RKNN candidate set which obtain in filtering stage.

Algorithm 3：FCINCH-Refine(Q,O,*k*,S,Scnd)

Input: Q: inquire set; O: tne mimimal enclosing circle including inquire points; *k*: *k* value of R*K*NN; S: subset of the whole data set; Scnd: candidate set

Output: R*K*NN result set

Description：

```
SR = CINCH(O,k+1,S)
for(every data object p in Scnd) do
    if(p is not in SR)then
        Scnd = Scnd-{P}
for(every data object p in Scnd) do
    if(p∈Q)then r = min{dist(p,{Q-p})}
    else r = min{dist(p,{Q })}
if(Range-k(p,k,r})return false)then
    Scnd = Scnd-{P}
return Scnd
```

Refinement can be divided into two stages: the first phase, algorithm CINCH (O, k + 1, S) calculate new region to exclude error items; the second phase, use Range-*k* validation method to validate each candidate for excluding incorrect items, the candidate set Scnd may contain one or more inquire points. Because a inquire point may be RKNN of another inquire, when candidate is inquire point, parameter of function Range-*k* r=min {dist (p, {Q - p}}; however,when candidate is not inquire point, parameter of function Range-*k* r= min{dist(p,Q)}.

IV. ANALYSIS OF ALGORITHM

This section will test the algorithm FCINCH performance of proposed RKNN query. In the context of the same size of inquire set, the enclosing circle size and *K* change will be effect on algorithm efficiency, and then compared with FINCH algorithm. Algorithm is simulated completely in microcomputer. Tests use North America Nap (data quantity 24254) data sets, buliding up R-tree index.

Performance indicator is based on the actual running time including CPU time and I/O time. FINCH's running time is the sum of every inquire points' running time, candidate result is also the sum of each inquire points'.

As shown in Figure 4, setting inquire amount is 5, as the enclosing circle is growing, algorithm FCINCH also has its usage scope. When the radius is greater than 0.8, algorithm FINCH is superior to FCINCH, because the performance of FINCH algorithm calculated every inquire point is the similar, the running time increases manyfold and is irrelevant to inquire point position. So when the number of inquire pointis is the same, the running time is basically similar. But algorithm FCINCH is different, it is associated with inquire point position. In this state of determinate inquire number, the minimal enclosing circle formed from inquire point set is greater, algorithm performance will be lower. As shown in Figure 5, along with circle are bigger, candidate number is greater for algorithm FCINCH, but algorithm FINCH has little change.

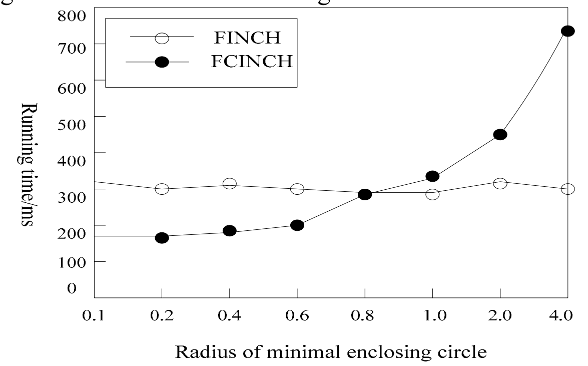


Figure 4. the mimimal enclosing circle size on the influence of algorithm performance

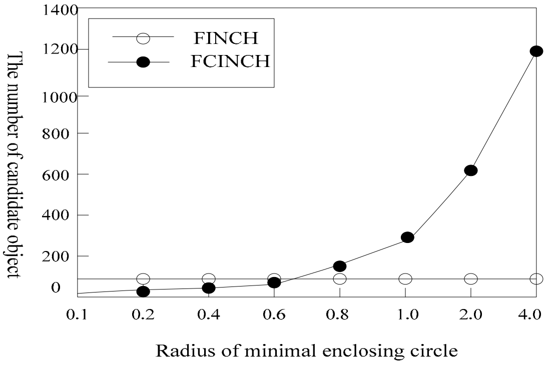


Figure 5. the minimal enclosing circle size on the influence of candidate number

But when the distance among inquire is near, the performance of algorithm FCINCH will better than algorithm FINCH. As shown in Figure 6, given 5 inquire points, along with *k* value from 1 to 10, the execution performance compares in Nap data set. Figure 7 shows the comparion of the same number of candidates, algorithm FINCH get more candidate set than algorithm FCINCH along with the increase of *k*. The two Figures show that FCINCH is more effectively than FINCH in processing R*K*NN query problem with closer inquire points. As inquire points are more nearer, algorithm FINCH calculate search area one by one to get their candidate set, their search area cloud overlap and candidate set of each inquire pointwill also have a lot of repetition, thus the algorithm performance will reduce, but algorithm FCINCH avoids the situation, it delimate inquire points in the minimal enclosing circle, considering the circle as a whole will avoid the repetition. So in this case, FCINCH is more effective than FINCH.

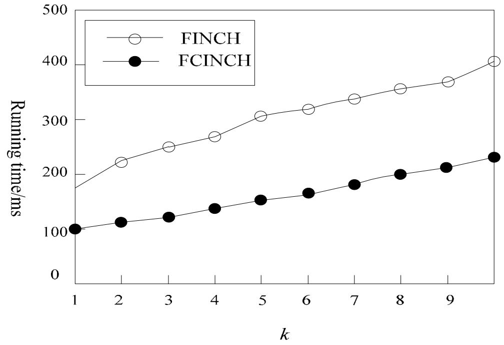


Figure 6. *k* value on the influence of algorithm performance

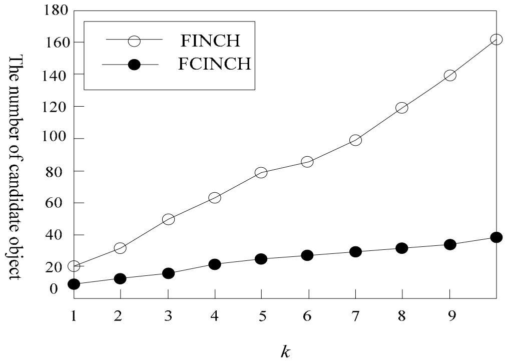


Figure 7. *k* value on the influence of candidate number

V. CONCLUSION

With the development of spatial databases, RKNN query has broad prospects. This paper adopts R-tree index structure and uses filteg-refinement model structure to realize R*K*NN query. In filtering stage, considering tne minimal enclosing circle as a whole to prune and using CINCH algorithm to determine the search area, and finnally getting candidate result set. In refining stage, using Range-*K* to verify and eliminate data points of the error output, and ultimately obtains the result set. And performance of algorithm is tested by employing real data set. With a group of neighboring space objects as inquire point set, the experiment result shows that the proposed algorithm's efficiency is better than direct algorithm. This algorithm also has practical application value, could be used to evaluate the set object influence. Future work will focus on improving algorithm, researching R*K*NN and high-dimensional R*K*NN.

VI. ACKNOWLEDGMENT

This work is supported by the National High-Tech Research and Development Program of China (863 Program) under Grant No. 2009AA062703, and the State Key Laboratory of Rail Traffic Control and Safety of Beijing Jiaotong University under Contract No. RCS2008K006.

REFERENCES

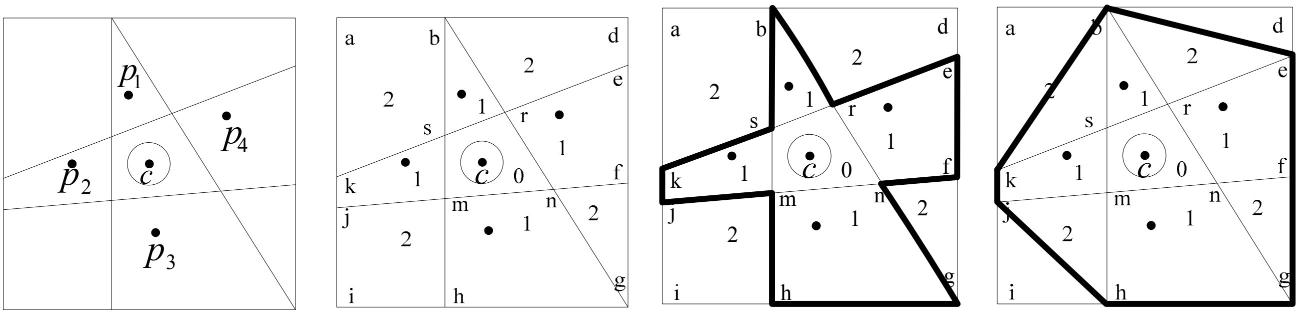
[1]Korn F, Muthukrishnan S. Influence sets based on reverse nearest neighbor queries. Special Interest Group of Data[J]. 2000, PP201-212

[2]Wu Wei,Chee Fei Yang,Chan Yong,Tan Kian-Lee. Continuous reverse k nearest neighbor monitoring. Proceedings of the 9th International Conference on Mobile Data Management[J].2008,PP132-139

[3] Tao Yu-fei,Papadias D,Lian X,et al.Multidimensional reverse kNN search. The VLDB Journal The International Journal on Very Large Data Bases[J]. 2007,PP293-415

[4]Wu Wei,Yang Fei,Chan Yong,Tan Kian-Lee. FINCH:Evaluating reverse k nearest neighbor queries on location data. Proceedings of the Very Large Data Bases[J].2008,PP1056-1067

[5]Wang Wei,Wang Wenping.An algorithm for finding the smallest circle containing all points in a given point set. Journal of Software[J].2000,PP1237-1240



(a) space partition (b) level of polygon (c) candidate area (d) search area

Figure 3. CINCH algotithm

Published by Atlantis Press, Paris, France.