Continuouos KNN Queries Based on Self-Adaptive Distance Threshold in Road Networks

Shasha Tian[1], Yanhong Li[1,*], Hong Wang[1], Wei She[2]

1 College of Computer Science
South-Central University for Nationalities
Wuhan, Hubei, China
2 *School of Mathematics and Statistic*
South-Central University for Nationalities
Wuhan, Hubei, China

***Abstract* — Continuous K Nearest Neighbor (CKNN) queries is one of the kernel problems to the Location Based Service (LBS). To obtain efficient query algorithm, a CKNN queries algorithm based on self-adaptive distance threshold (CKNN-SADT) is proposed by detailed study of CKNN. Firstly, an improved method to construct index structure based on PMR quad-tree is used in the CKNN-SADT algorithm. Secondly, the idea of self-adaptive distance threshold is proposed in the query algorithm. Through simulation experiments, it was found that the CKNN-SADT algorithm is superior to the Unique continuous search (UNICONS) algorithm in term of execution time, and whether the queries objects are sparse or not. The improved construction method of index structure and the self-adaptive distance threshold can also be used by other query algorithms.**

Keywords - self-adaptive; threshold; index structure; execution time

I. INTRODUCTION

Intelligent transportation is an important part of the intelligent city. Location based service (LBS) is a basis question to the intelligent transportation [1]. Many of the Location Based Service, LBSs, rely on the K nearest neighbor（KNN）queries which have attracted the attention of many scholars [2]. Static KNN queries can help citizens and tourists to find near hospital, bank, restaurant and tourist attractions, etc. With the development of mobile communication, the queries of static object can no longer meet public demand for LBS. More and more mobile users need efficient application to help them continuously. Now, the continuous KNN (CKNN) queries are coming into our lives and bring great convenience to people [3,4]. However, because of the complexity of road connections, it is difficult to computer the distance between query point and query objects in the road networks in comparison to Euclidean space. Moreover, due to the increasing number of mobile devices, more and more data are generated. The big data puts forward higher requests to the continuous KNN queries.

The remainder of this paper is organized as follows. Section 2 describes the recent work status quo about continuous KNN queries at home and abroad. Section 3 introduces the question description and index structure and describes an improved method to construct index structure based on PMR quadtree. Section 4 introduces the proposed CKNN queries method based on self-adaptive distance threshold (CKNN-SADT) firstly, and then presents some experiments to evaluate the performance of the query method. Finally, the conclusions are summarized in Section 5.

II. STATE OF THE ART

As an important variant of KNN queries, CKNN Queries has been studied deeply by many scholars and obtained many important results. Among these results, Kolahdouzan et al. proposed upper bound (UB) algorithm [5]. In the processing of the UB algorithm, the KNNs of all nodes on a path were found firstly, and then the intermediate split points were found for those adjacent, the KNNs of the split points using the KNNs of the surrounding node were computed finally. Because the shortest path length depended on the density of interesting objects too largely, the Unique continuous search (UNICONS) algorithm was proposed by Cho et al. to overcome the shortcoming [6]. In [7], the concept of “safe region” was introduced .If the query point was in the safe region, its KNNs was stable. In [8], Continuous visible K nearest neighbor query over moving objects was described.

The paper mainly researched the main technical matters of continuous KNN queries, and proposed a CKNN queries method based on self-adaptive distance threshold in road networks.

III. QUESTION DESCRIPTION AND INDEX STRUCTURE

A. *Question Description*

In this study, we research on the problem of CKNN queries in road networks. Figure 1 shows an example of general road networks, where q is query point, pi represents the query object, ni is the intersect node, the black fine lines denote the roads, the red bold line describes query path, and every road is divided into several segments using the intersect nodes. As shown in figure 1, when q reaches the point a, its 2NN is p1 and p3, and when it reaches the point b, its 2NN is p8 and p3.It means that query point q would face to different KNNs when its location is different. The task of CKNN queries is to find the KNNs of query point at any time and any location. For instance, after analyzing figure 1, we will obtain a C2NN result set composed by these tuples: < n1 , n2, 0.34,{ p1 , p3}>,< n1 , n2, 0.51,{ p3 , p8}>,< n1 , n2, 0.71,{ p3 , p8}>….where n1 and n2 denote the start and the end of the segment, 0.34 , 0.51 and 0.71 describe the location of q. Assuming the length from n1 to n2 is 1, "0.34" presents the distance between n1 and q is 0.34.

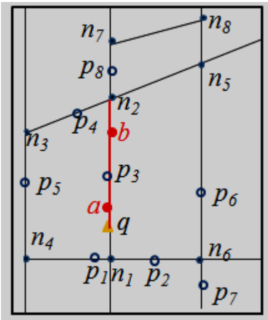


Figure 1. Example of CKNN queried in road networks.

B. Index Structure

In this paper, PMR quadtree is used to store the road networks information. Because of the shortcoming of the existing process in term of constructing PMR quedtree, an improved construction method is proposed.

C. Existing Method

In the existing method, the process of constructing the PMR quedtree is as follows. Firstly, a big unit is initialized to present all of the roads. And then all of the segments are inserted into the big unit. When the number of the segments exceeds a given threshold, the current unit is chopped into four identically sized units. For example, Figure 1 can be stored using the PMR quadtree in figure 2.

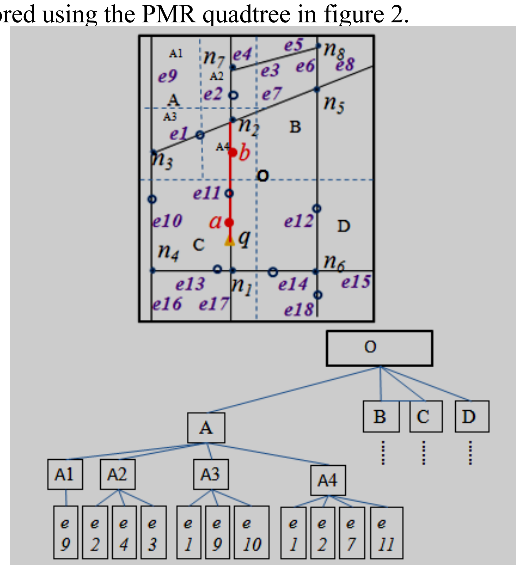


Figure 2. PMR quadtree with existing method.

As shown in Figure 2, the whole road networks are abstracted to a big unit O, and the threshold of segments is assumed to be 4. The segments are described by ei. Because of the number of the segments exceeds the threshold, O is divided into four identically sized units A, B, C, and D. After that, the number of segments is found still greater than 4 in the unit A, B and C, so these units are divided again. However, the method to construct the PMR quadtree is inefficient. From figure 2, we can find that the segments are stored repeatedly, such as e2, which is stored in unit A2 and unit A4.

D. Improved Construction Method

Based on the shortcoming of the existing method, an improved thought is proposed as follows:

Regulation 1: If a segment passes through several units, it is only stored in the unit including its left-first intersect node.

Regulation 2: If a segment passes through several units vertically, it is only stored in the unit including its top intersect node.

Regulation 1 is prior than regulation 2. Figure 1 is stored using the PMR quadtree with improved construction method and the result is showed in Figure 3.

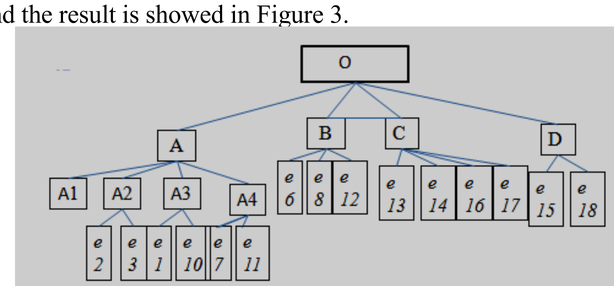


Figure 3. PMR quadtree with improved method.

Compared with previous method to construct the PMR quadtree, the proposed had two merits of low redundant storage size and low time cost.

IV. CKNN-SADT ALGORITHM AND EXPERIMENTAL ANALYSIS

A. Algorithm Description and Algorithm Analysis

Algorithm Description

The UNICONS algorithm cannot maintain the efficient operation when the value of K increases. Aiming at the weakness, CKNN-SADT algorithm was proposed in this study.

The precondition of the CKNN-SADT algorithm is that the shortest distance between any two intersect nodes have been calculated and stored in a look-up table. The shortest distance of any intersect nodes is denoted by Sdist (ni, nj). and can be output immediately in the process of algorithm execution.

Assuming m to be the density of query objects in the road networks, r equals to K/m. A circle centered at q with radius equal to r can help KNN queries to prune search space.

Lemma: In the circle centered at q with radius equal to r, if there are more than K query objects whose distance to q is lower than r, then the search space out of the circle can be pruned.

Proof: Since the radius of the center is r, the distance between the query objects out of the circle and q must be higher than r.

The steps of CKNN-SADT algorithm are described in detail as follows:

Step1: Draw a circle centered at q with radius equal to r to prune the query objects out of the circle.

Step2: Calculate the shortest distance between every query object pi (assume pi on the segment between nh and nj) in the circle and q (assume q on the segment between nx and ny). The shortest distance is denoted by Sdist (pi, q) and expressed as formula (1):

$$Sdist(p_i,q)=\min\begin{Bmatrix} dist(p_i,n_h)+Sdist(n_h,n_x)+dist(n_x,q), \\ dist(p_i,n_h)+Sdist(n_h,n_y)+dist(n_y,q), \\ dist(p_i,n_j)+Sdist(n_j,n_x)+dist(n_x,q), \\ dist(p_i,n_j)+Sdist(n_j,n_y)+dist(n_y,q). \end{Bmatrix} \quad (1)$$

Step3: Store Sdist (pi, q) for every query object in the circle. Use num to record the number of the query objects whose Sdist (pi, q) is lower than r and store the query objects into a candidate set S.

Step4: If num > K, the Sdist (pi ,q) of query objects in S is to sort in ascending order, and the K query objects corresponding the K smallest Sdist (pi ,q) is the KNNs at the moment.

Step5: If num < K, go to Step1 and change r as formula (2):

$$r=(1+K/num)\cdot r \quad (2)$$

The value of r can vary self-adaptively, so the algorithm is called CKNN queries based on self-adaptive distance threshold.

Step6: Update the circle and the query objects in the circle with the moving of the query point (Figure 4 shows an example). Repeat all of the steps and update S at every time.

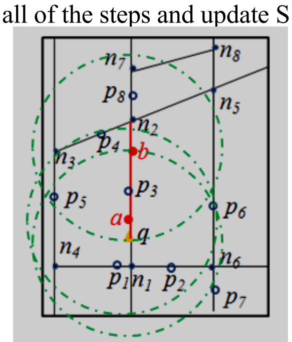


Figure 4. A example of updating the circle centered at q with radius equal to r

Step7: If query point reached the end of the query path, the algorithm terminates.

B. Algorithm Analysis

The algorithm realizes CKNN queries through updating the circle centered at q with radius equal to r, the query objects in the circle and Sdist (pi, q).The time cost of these operations is biggest in the process of the CKNN-SADT algorithm. However, because the Sdist (ni, nj) have been calculated and stored before starting the algorithm, the KNN queries at any time can be finished in polynomial time.

C. Experimental Analysis

For comparing the performance of the CKNN-SADT and UNICONS algorithm, the California city map is chosen in the experiments from the website [9]. All query objects, roads and intersect points are real-world data. The two algorithms are implemented in Java. Experiments are run on a Linux desktop machine with a 2.4GHz CPU and 512M memory. The query objects in [9] are described in the table I.

TABLE I DESCRIBTION OF QUERY OBJECTS

Query Objects	Objects Type	Objects Number
Hospital	30	835
Lake	33	2636
Park	44	6900
Shcool	50	11173

The experiments evaluated the execution time of the CKNN-SADT algorithm and UNICONS algorithm aiming at different densities of the query objects. The experimental results are showed in figure 5 and figure 6.

As shown in Figure 5 and 6, the execution time of CKNN-SADT algorithm is lower than UNICONS algorithm whether the query objects are sparse or not.

After comparing the Figure 5 and 6, it is apparent that the CKNN-SADT algorithm is better-performing when the query objects are dense.

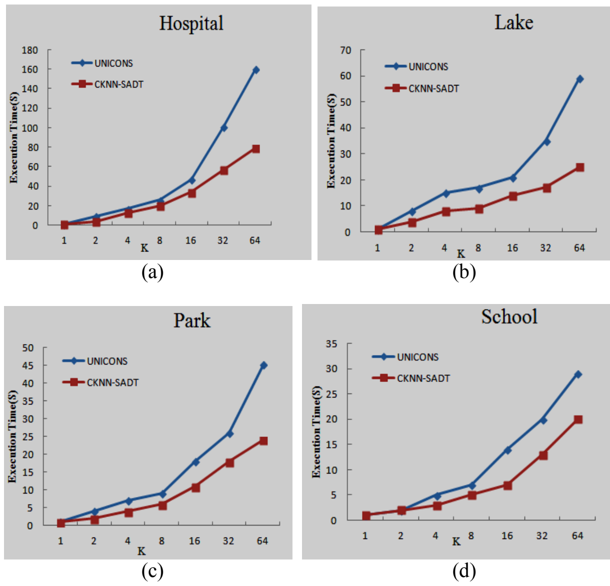
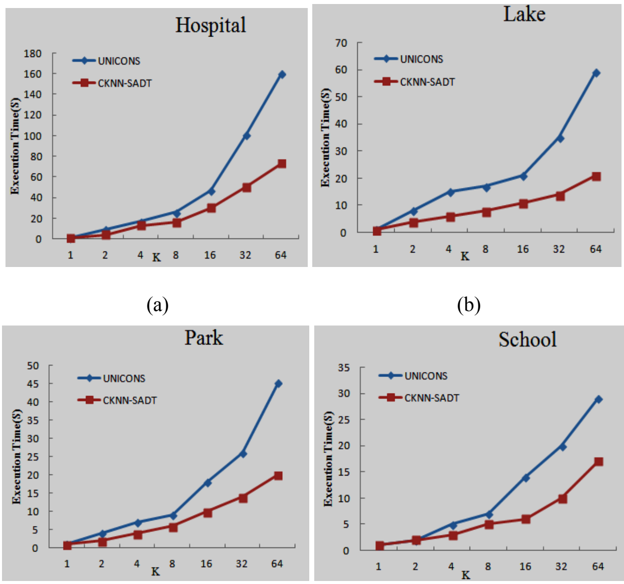


Figure 5. Execution time comparation between the CKNN-SADTA algorithm and the UNICONS algorithm when the query objects are sparse.



(a) (b) (c) (d)

Figure 6. Execution time comparation between the CKNN-SADTA algorithm and the UNICONS algorithm when the query objects are dense.

VI. CONCLUSION

This paper proposed a CKNN queries algorithm based on self-adaptive distance threshold in road networks, which have two main improvements. In the first, an improved method to construct index structure was proposed. The method only needs to store every segment once in the road networks, so the query speed can be increased and the storage space can be decreased. Secondly, r is obtained to be radius of a circle centered at q. And r is self-adaptive with the change of the density of the query objects and the distance between the query objects and q. In other words, the query space is variable with the change of actual road networks. Also, through simulating and implementing a actual query, it was found that the CKNN-SADTA algorithm has advantages over UNICONS algorithm in terms of algorithm speed. The results of this study have significant potential to support many other query algorithms. Further research based on the index structure and the process of CKNN-SADTA is necessary.

ACKNOWLEDGMENT

This work is supported by the Hubei Province Natural Science Foundations (No.2015CFC876) and the Basic Scientific Research funds of the Central University (CZQ14010).

REFERENCES

[1] Bohl O, Manouchehri S, Winand U, “Mobile information systems for the private everyday life”, Mobile Information Systems, vol.3, No.3, pp. 135-152, 2007.

[2] Gu Y, Zhang H, Wang Z, et al, “ Efficient moving k nearest neighbor queries over line segment objects”, World Wide Web,pp.1-25. 2015.

[3] Cho H J, “Continuous range k-nearest neighbor queries in vehicular ad hoc networks”, Journal of Systems and Software, vol.86, No.5, pp.1323-1332, 2013.

[4] Sistla A P, Wolfson O, Xu B, “Continuous nearest-neighbor queries with location uncertainty”, The VLDB Journal,vol.24, No.1, pp.25-50, 2015.

[5] Kolahdouzan M R, Shahabi C, “Alternative solutions for continuous k nearest neighbor queries in spatial network databases”, GeoInformatica, vol.9, No.4, pp.321-341, 2005.

[6] Cho H J, Chung C W, “An efficient and scalable approach to CNN queries in a road network”, Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, pp.865-876.2005.

[7] Li C, Gu Y, Qi J, et al, “A safe region based approach to moving KNN queries in obstructed space”, Knowledge and Information Systems, vol.4, No.2, pp.417-451,2015.

[8] Wang Y, Zhang R, Xu C, et al, “Continuous visible k nearest neighbor query on moving objects”, Information Systems, vol.44 pp.1-21, 2014.