

Continuous Reverse k-Nearest-Neighbor Query in Dynamic Road Network

Chen Wen

*School of Mathematics and Computer Science, Tongling College,
Tongling, P.R.China
tlxychenwen@163.com*

Abstract

Traditional location-based query measures the distance of all data points in various methods, but it is not applicable for the condition in which the weight of all sections in the road network are altered dynamically. In order to solve the new application problems brought by the real-time traffic status, a dynamic monitoring algorithm is designed in this paper, and it mainly updates the weight changes of road network with temporary network extension tree incrementally. On this basis, a continuous reverse K-nearest-neighbor (RKNN) query method is proposed, and it is proved to be efficient by the experimental results.

Keywords: *dynamic weight road network; continuous reverse k-nearest-neighbor; location-based services*

1. Introduction

With the extensive application of smart mobile device and wireless location technique, the number of users applying location service-based applications increases drastically. As a significant location query, Reverse k-nearest-neighbor query (RKNN) receives much attention, and it is mainly applied to search the spatial data points containing the query point in k-neighbor, for reflecting the great impact of query points on those spatial data points. RKNN has been widely applied in the spatial decision support, resource allocation and data mining, *etc.* Continuous reverse k-nearest-neighbor query is a relatively complicated query, and it can support advanced analysis and prediction, for instance, in seismic disaster relief, medical centers should be set for the disaster area. If users drive vehicles with full load of drugs, he may inquire those medical centers along the route from the starting point to the terminal. Apparently, these medical centers should be close enough to the route, namely the k-neighbor of medical points. Or the medical center can move drugs from other medical centers. In real life, the travel time may be changed dynamically due to various reasons, such as the traffic accident in the road network. Travel time is a method of measuring the distance between all data points in the road network, and the dynamic changes of travel time means the dynamic changes of weight in all sections of road network, which may make the query method of conventional road network inapplicable. In this paper, an effective continuous reverse k-nearest-neighbor query method based on the dynamic weight road network is designed according to the changes in real-time traffic stats.

2. Relevant Work

The application of Location Dependent Queries mainly includes Range Query, Nearest Neighbor Query, Reverse Nearest Neighbor Query and *etc.* In the recent years, the researchers have deeply studied the Location Dependent Queries, especially the Nearest Neighbor Query and Reverse Nearest Neighbor Query and proposed lots of query

processing technologies so as to effectively support the query problems for all kinds of spatial and spatiotemporal objects.

As one of the important type in the Location Dependent Query Processing Technologies, the Nearest Neighbor Query has become a key point and hotspot of the researches in the field of spatial and spatiotemporal database. With the wide and deep studies by researchers, a number of Nearest Neighbor Query Algorithms has been proposed. The Static Nearest Neighbor Query with static objects was firstly studied by people and Roussopoulos *et al* [1] firstly put forward the algorithm of Static Nearest Neighbor Query with static objects based on Euclidean Distance. And Chueng *et al* [2] further improved this algorithm to delete the pruning rule which not only cannot improve the pruning effect, but possesses the high computation complexity, thus reduce the computational cost. Seidl *et al* [3] proposed a kind of Optimal Multi-step Neighbor Algorithm suitable for the high dimensional data, which is the best high-dimension neighbor algorithm adopted by filter and refined methods so far. Belussi *et al* [4] studied how to apply the tree index in the Nearest Neighbor Query problem. In addition, Hjaltason *et al* [5] discussed the breadth-first-based Nearest Neighbor Query Algorithm. Henrich *et al* [6] proposed the Nearest Neighbor Query Algorithm combined with the tree index and two priority queues. Hjaltason *et al* [7] put forward a method which integrates the tree-index structure and the priority queue. Besides, Prasad *et al* [8] firstly discussed the Continuous Nearest Neighbor Query problem.

The methods to solve the Continuous Nearest Neighbor Query problem can be divided into two kinds: one is sample basis and the other is movement path basis. Song *et al* [9] proposed that the periodic sampling technology can be adopted to cope with the Continuous Nearest Neighbor Query problem. Tao *et al* [10] came up with a movement-path-based method, where the algorithm is utilized to effectively deal with the Time-parameterized Nearest Neighbor Query. However, this method needs to repeatedly submit and calculate the query result for several times so as to make the cost of query processing bigger. Tao *et al* [10] improved the query processing method proposed in the literature, overcame its original shortcomings and made it more effective. Soon afterwards, the people's research interest point was transferred to the Continuous Nearest Neighbor Query with mobile objects. Kollios *et al* [12] proposed the processing algorithm of mobile-object-based Nearest Neighbor Query. And Benetis *et al* [13] came up with a method to solve the problem of Nearest Neighbor Query with mobile objects. Glenn *et al* [14] put forward a kind of algorithm called Continuous Window Query Algorithm to solve this problem. Mokble *et al* [15] firstly proposed the frame to deal with the continuous query problem. In addition, Benetis *et al* [16] improved the algorithm put forward in the article in order to support the processing of K Nearest Neighbor Query problem. And Xiong *et al* [17] proposed the SEA-CNN algorithm with the excellent extensibility.

There are two kinds of RNN, namely Monochrome reverse nearest neighbor(MRNN) and Double - reverse nearest neighbor (BRNN). In MRNN, all mobile data objects and query objects are of the same type; while in DRNN, objects and queries are of two distinct types. The initial RKNN query algorithm is a precomputation-based method. At first, index structure, like RNN tree [18] and RDNN tree [19], *etc.* should be constructed, and it mainly realize static RNN/RkNN query through the query of index structure. In order to eliminate the precomputation, snapshot RNN occurs, and it mainly cut down the query space with some measures (known as filtration step), and then, candidate targets obtained should be refined, for determining if they are the RNN for query. There are two filtration means, namely six-section spatial reduction [20] and TPL reduction [21]. Reference [22] proposed a CRNN algorithm, which is an incremental and extensible method. Reference [23] put forward IGERN (incremental, general, CRNN query method), which mainly makes use of the TPL reduction strategy.

The above algorithms are applicable for the treatment of RNN problems in Euclidean space, rather than the road network space. Moreover, the movement of mobile target in practical application is always restricted to the specific road network structure, and the RNN query algorithm of mobile target in road network is quite significant. Reference [24] proposed the RNN query treatment problem in large scale pictures (here, road network is also the application of picture). They have managed the optional k MRNN and DRNN query in limited picture and unlimited picture, and RNN query optimized computing method and corresponding algorithm of RNN query are proposed on the basis of some characteristics of the picture. However, it is only applicable for the static RNN. Safar *et al.* [25] solved the RNN query on road network based on the NVD and road network extension. Although, the target and query proposed by the method is movable, it is actually a kind of snapshot RNN, and it cannot solve the continuous monitoring problem of RNN. Liang *et al.* [26] came up with the CRNN query treatment method for the road network mobile target. Reference [27] put forward a SRNN query algorithm for land surface. Li *et al.* [28] proposed an increasing and general mode to solve the continuous query problem in road network. Reference [29] came up with a BRC pruning algorithm by aiming at the filtration steps. Cheema *et al.* [30] considered that the present method is quite sensitive to the query point and location of mobile data, and proposed a CRNN query framework, for setting safe area of the query point and target.

3. Relevant Definition

Definition 1: link. Since the real road might be long, the road is segmented at the crossings, and the one-way road of each section is called a link.

Definition 2: real-time traffic information. The real-time traffic information includes the travel time of each link. In this paper, it is called the weight of road network.

Aiming at the storage of road network information, Dijkstra algorithm is employed, with the given query point q as a starting point, for extending the road network according to the topology structure. Therefore, the initial RKN and a temporary network extension tree (TNETree) containing the extended road (namely the link) are obtained, and related definitions are given as follows:

Definition 3 :TNETree, it records the optimal path reaching each extended link, requiring that the total travel time of reaching the extended link is less than the given extension upper limit. TNETree is bi-directional, and the leaf node is called the terminal link.

Definition 4: dominance relation. In order to make the expression much more clear, the previous link and follow-up link are applied to indicate the relationship between links. When a link has one or more previous links, there must be dominance relation between these links.

As shown in Figure. 1, link 1 and link 2 are the previous links of link 3 and link 4. If link 1, link 3, and link 4 have already been extended, and link 1 is the father of link 3 and link 4, when link 2 is extended, suppose the route from link 2 to link 3 and link 4 is better than the route from link 1 to link 3 and link 4, link 1 is no longer the father link of link 3 and link 4, and it will lose all sub-links. Link 3 and link 4 would become the sons of link 2, namely link 2 is the dominant link of link 1. Meanwhile, link 1 is the dominated link of link 2. There must be certain dominance relation between link 1 and link 2.

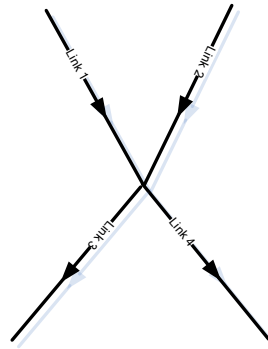


Figure 1. Link

4. CRKN Query of Dynamic Weight

The query process consists of three stages, including the initial stage, mobile stage and weight change incremental updating stage. When the receiver receives a query request, it will enter the initial query stage at first, and provide the RKN to the user through network extension, and establish temporary network extension tree. When users move along the roads, the server may update TNETree according to the effective regional theory, for guaranteeing that the RKN query result obtained in this stage is the latest. The previous two stages neglect the changes in weight. When the latest real-time traffic information is received, the query will enter the third stage directly, and get the latest RKN through updating the TNETree dominance relation.

4.1 Initial Stage of Query

The main task of this stage is to establish temporary network extension tree and inquire the initial RKN. Suppose the user launches the query at point q , the query would enter the initial query stage. Here, the result is noted as $RKNN(q)$, and the dominance relation mapping is noted as dom . The key of dom is any link, and its value is the extended previous link, including its parent link p and the dominated link of p . The sum of travel time of several links is noted as $totalTT$.

The steps of algorithm are shown as below:

- (1) The link of q is taken as the root node, and the root is added into the link treatment queue.
- (2) Extend the link of queue. When the queue is not empty, the first element L of the queue is taken as the current extended link, or it will turn to step (7).
- (3) When the link L contains the query target, if $RKNN(q).size < k$, L should be added into $RKNN(q)$.
- (4) If $L.totalTT$ is smaller than the greatest $totalTT$ in the queue, L should be applied to replace the link with the greatest $totalTT$ in the queue. And then, the upper limit should be updated to the greatest $totalTT$ in the $RKNN(q)$.
- (5) Check each follow-up link $nLink$ of L . If the $totalTT$ of $nLink$ is smaller than the upper limit, $nLink$ should be added into the queue, and it will turn to step (2).
- (6) If $nLink$ has not been extended yet, it should be added in TNETree as the sub-link of L , or it means that there is dominance relation between L and father node p' of $nLink$. If $L.totalTT < p'.totalTT$, it suggests that L governs p' , and the father node of $nLink$ will be replaced as L . If dom has no records of the $nLink$, the dominance relation will be recorded in dom , or corresponding dominance relationship of $nLink$ should be adjusted to guarantee that the dominate node is in the first position in dom .
- (7) If $RKNN(q).size < k$, the terminal link of TNETree will be added into queue, and the upper limit will become two times of the original limit, and it should turn to step (2).

4.2 Mobile Stage

When the query point moves to another link, this line should be kept as the sub-tree of the root node, and other nodes in the temporary network extension tree should be deleted according to the theory of effective area. And then, the terminal link of new TNETree should be found out and extended till RKN is found.

4.3 Weight Change Incremental Updating Stage

The weight change stage is an important component in CRKN query, and it mainly deals with the impact brought by the weight changes. In this stage, due to the change in link weight, the dominance relation may not be accurate. Therefore, TNETree may not be able to indicate the optimal route from the current starting point to the extended link. Consequently, the algorithm will go through the TNETree for inspecting the dominance relation, so as to update the TNETree for reflecting the optimal route in the current traffic status correctly.

The specific algorithm is shown as below:

- (1) Re-set the travel time of all links according to the real-time traffic status.
- (2) Add the root node of TNETree into the queue
- (3) If the queue is not empty, the first element L of the queue should be taken, and all sub-link of L should be added into the queue, or it will turn to step (8).
- (4) If L is in the dominance relation mapping dom, the value v of L in dom should be taken for finding out the tmpdom with the smallest totalTT.
- (5) If tmpdom is the father link of L, it means that it is not necessary to update the father link of L, or tmpdom will be regarded as the dominant link of all links and it will be set as the father link of L.
- (6) If RKNN(q) changes, the upper limit of TNETree will be updated as the greatest totalTT of RKNN (q).
- (7) Look for the 1st dominated link in each branch of the subtree of L, add queue for the convenience of inspection, and it should turn to step (3).
- (8) Look for the latest terminal link of TNETree and extend it, till all RKN is found.
- (9) When the weight changes, update the TNETree to the correct state, and the application of dominance relation may calculate and save the computing time

5. Experimental Results and Analysis

In this paper, the data set employed is the road network of San Francisco and California [31]. The contrasting algorithm employs the network extension algorithm proposed in reference [30], noted as algorithm b. and the algorithm of this paper is noted as a. Parameter k stands for the number of target data points obtained. The default is 5, and the increment is 5. To simulate the real traffic condition, about 80% of the section weight is altered randomly each time in the experiment. The algorithm of this paper is realized with C++, and it runs in Windows XP with E3400 2.59GHz processor and 2 G internal storage.

- (1) Figure. 2 (a) gives the impact of changes of k on cpu. With the increase of k, the cpu time of two algorithms increase, but the algorithm can modify the shortest route incrementally despite of the dynamic updating of road network weight. Moreover, EP algorithm should go through the entire road network nodes. Consequently, the cpu time of the latter is longer.
- (2) Figure. 2 (b) verifies the impact of the number of data points in road network on the cpu time. With the increase of data set, the available information point in the link increases, and the link went through reduces. The cpu time of the two algorithms reduces accordingly. The algorithm shows again the advantages of incremental

updating. Since there is no need to go through the entire road network, the cpu time is less than EP algorithm.

- (3) Figure. 2(c) verifies the impact of changes of the number of query points on the cpu in road network. With the increase of query points, more links should be treated, and the cpu time increases accordingly. Consequently, the algorithm of this paper is better than the contrasting algorithm.
- (4) Figure. 2 (d) verifies the impact of road network weight changes on cpu. In order to simulate the real traffic status, about 80% of the road section weight is altered randomly. At this moment, the advantages of the algorithm in this paper are evident, and its operating time is far shorter than that of the contrasting algorithm, reflecting that the algorithm can deal with the dynamic road network condition effectively.

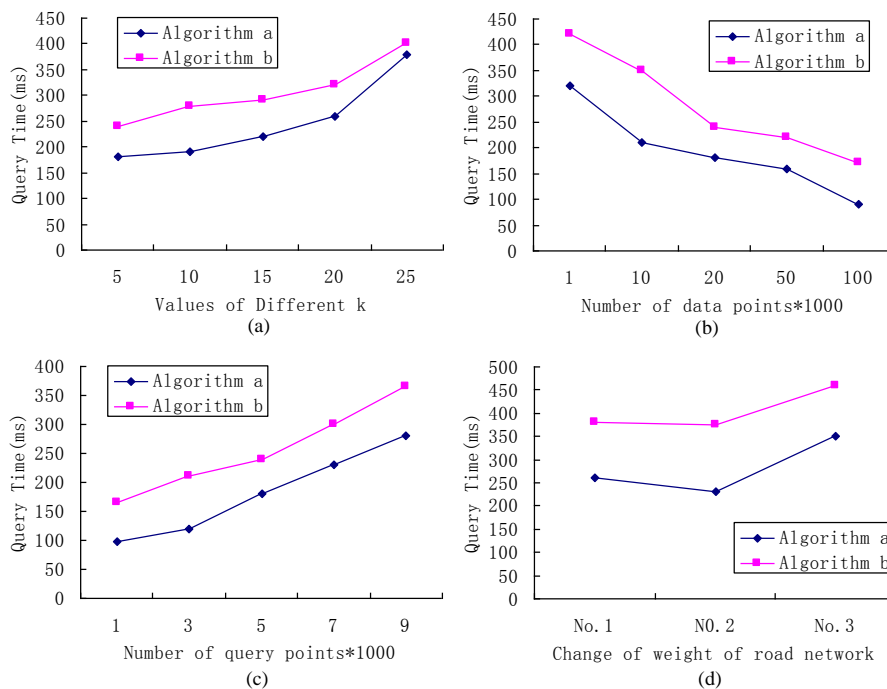


Figure 2. Experimental Results

It can be learnt from the analysis that the dynamic changes of road network, size of data set and distribution of data point have relatively huge impact on the performance of algorithm. When there are few query points, the algorithm is not applicable for processing the RNN query with few query points. Consequently, the algorithm should construct TNETree at the beginning, and it may consume time. But with the increase of query points, the incremental updating advantage will be more and more evident.

6. Conclusion

The research status of k-neighbor query and RKN query of the current road network is analyzed. On this basis, a dynamic monitoring algorithm is designed by aiming at the practical conditions of the dynamic changes of road network weight, and it solves the CRKNN query problems in dynamic road network effectively.

Acknowledgements

This work was supported by funds from Universities Key Fund of Anhui Province for Young Talents of China under Grant 2013SQRL082ZD, Natural Science Research Universities Key Project of Anhui Province of China under Grant KJ2014A256 and 2016 Anhui province colleges and universities outstanding young talent support program key projects of China under Grant gxyqZD2016319.

References

- [1] Roussopoulos N, Kelley S, Vincent F. Nearest neighbor queries[C]//ACM sigmod record. ACM, 1995, 24(2): 71-79.
- [2] Cheung K L, Fu A W C. Enhanced nearest neighbour search on the R-tree[J]. ACM SIGMOD Record, 1998, 27(3): 16-21.
- [3] Seidl T, Kriegel H P. Optimal multi-step k-nearest neighbor search[C]//ACM SIGMOD Record. ACM, 1998, 27(2): 154-165.
- [4] Belussi A, Bertino E, Catania B. Using spatial data access structures for filtering nearest neighbor queries[J]. Data & Knowledge Engineering, 2002, 40(1): 1-31.
- [5] Hjaltason G R, Samet H. Ranking in spatial databases[C]//Advances in Spatial Databases. Springer Berlin Heidelberg, 1995: 83-95.
- [6] Henrich A. A Distance Scan Algorithm for Spatial Access Structures[C]//ACM-GIS. 1994: 136-143.
- [7] Hjaltason G R, Samet H. Distance browsing in spatial databases[J]. ACM Transactions on Database Systems (TODS), 1999, 24(2): 265-318.
- [8] Sistla A P, Wolfson O, Chamberlain S, *et al.* Modeling and querying moving objects[C]//icde. IEEE, 1997: 422.
- [9] Song Z, Roussopoulos N. K-nearest neighbor search for moving query point[M]//Advances in Spatial and Temporal Databases. Springer Berlin Heidelberg, 2001: 79-96.
- [10] Tao Y, Papadias D. Time-parameterized queries in spatio-temporal databases[C]//Proceedings of the 2002 ACM SIGMOD international conference on Management of data. ACM, 2002: 334-345.
- [11] Tao Y, Papadias D, Shen Q. Continuous nearest neighbor search[C]//Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002: 287-298.
- [12] Kollios G, Gunopulos D, Tsotras V J. On indexing mobile objects[C]//Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 1999: 261-272.
- [13] Benetis R, Jensen C S, Karčiauskas G, *et al.* Nearest neighbor and reverse nearest neighbor queries for moving objects[C]//Database Engineering and Applications Symposium, 2002. Proceedings. International. IEEE, 2002: 44-53.
- [14] Iwerks G S, Samet H, Smith K. Continuous k-nearest neighbor queries for continuously moving points with updates[C]//Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003: 512-523.
- [15] Mokbel M F, Xiong X, Aref W G. Sina: Scalable incremental processing of continuous queries in spatio-temporal databases[C]//Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, 2004: 623-634.
- [16] Benetis R, Jensen C S, Karčiauskas G, *et al.* Nearest and reverse nearest neighbor queries for moving objects[J]. The VLDB Journal, 2006, 15(3): 229-249.
- [17] Xiong X, Mokbel M F, Aref W G. Sea-cnn: Scalable processing of continuous k-nearest neighbor queries in spatio-temporal databases[C]//Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on. IEEE, 2005: 643-654.
- [18] Korn F, Muthukrishnan S. Influence sets based on reverse nearest neighbor queries[C]//ACM SIGMOD Record. ACM, 2000, 29(2): 201-212.
- [19] Yang C, Lin K I. An index structure for efficient reverse nearest neighbor queries[C]//Data Engineering, 2001. Proceedings. 17th International Conference on. IEEE, 2001: 485-492.
- [20] Stanoi I, Agrawal D, El Abbadi A. Reverse Nearest Neighbor Queries for Dynamic Databases[C]//ACM SIGMOD workshop on research issues in data mining and knowledge discovery. 2000: 44-53.
- [21] Tao Y, Papadias D, Lian X. Reverse kNN search in arbitrary dimensionality[C]//Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. VLDB Endowment, 2004: 744-755.
- [22] Xia T, Zhang D. Continuous reverse nearest neighbor monitoring[C]//Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on. IEEE, 2006: 77-77.
- [23] Kang J M, Mokbel M F, Shekhar S, *et al.* Continuous evaluation of monochromatic and bichromatic reverse nearest neighbors[C]//Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on. IEEE, 2007: 806-815.
- [24] Yiu M L, Papadias D, Mamoulis N, *et al.* Reverse nearest neighbors in large graphs[J]. Knowledge and Data Engineering, IEEE Transactions on, 2006, 18(4): 540-553.

- [25] Safar M, Al-Saleh A. PINE Based RNN Queries in Road Networks[C]//GeoInfo. 2005: 356-367.
- [26] Sun H L, Jiang C, Liu J L, *et al.* Continuous reverse nearest neighbor queries on moving objects in road networks[C]//Web-Age Information Management, 2008. WAIM'08. The Ninth International Conference on. IEEE, 2008: 238-245.
- [27] Yan D, Zhao Z, Ng W. Monochromatic and bichromatic reverse nearest neighbor queries on land surfaces[C]//Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012: 942-951.
- [28] Li Yanhong, Li Guohui, Du Xiaokun. study on continuous bichromatic reverse k nearest neighbor query processing in road networks[J]. Computer Science, 2012, 39(11): 131-136.
- [29] Wang S, Lv Q, Liu D, *et al.* Efficient filter algorithms for reverse k-nearest neighbor query[M]//Web-Age Information Management. Springer Berlin Heidelberg, 2011: 18-30.
- [30] Cheema M A, Zhang W, Lin X, *et al.* Continuous reverse k nearest neighbors queries in euclidean space and in spatial networks[J]. The VLDB Journal—The International Journal on Very Large Data Bases, 2012, 21(1): 69-95.
- [31] Data set[EB/OL]. <http://www.cs.fus.edu/lifeifei/spatialDataset.htm>, 2013

Author



Chen Wen is an Associate Professor in the School of Mathematics and Computer Science, Tongling College, Tongling, P.R.China. He holds a master degree in Computer Science and Technology from the Anhui University, Anhui, P.R.China. His previous research areas include privacy preserving.