Generalized Vicinity Query Algorithm in Road Network Distance

Yutaka Ohsawa[1], Htoo Htoo[1] , and Myint Myint Sein[2]
Graduate School of Science and Engineering, Saitama University[1], Japan
University of Computer Studies[2], Yangon
ohsawa@mail.saitama-u.ac.jp htoohtoo@mail.saitama-u.ac.jp myintucsy@gmail.com

Abstract

Vicinity queries include set k nearest neighbor (SetkNN) query, ordered k nearest neighbor (OrderkNN) query, bi-chromatic reverse nearest neighbor (BRkNN) query and distance range query. In existing approaches, each type of queries has been individually studied because they are different in query criteria. Moreover, these types of queries take long processing time in calculating road network distances. To improve in the efficiency, this paper proposes vicinity query algorithm based on the higher-order network Voronoi region (NVR) which gives a unified procedure to run these different queries in an integrated framework although the query criteria of all individuals differ from each query type. Through extensive experiments, the proposed method significantly outperforms the existing works in terms of processing time by nearly two orders of magnitude.

1. Introduction

Rapid increase in demand of mobile users and availability of wireless network communication, the growth in the utilization of location based services (LBS) applications becomes increase. Consequently, efficient queries for LBSs that meet the satisfaction of users are necessitated in the real world. In the literature, several nearest neighbor queries (NN) have been studied firstly focusing on Euclidean distance. However, in real scenarios, queries on road networks is needed for LBSs. Since 2000, various *k*NN queries algorithm based on road network distance have been actively studied.

Given an interest objects set **P** and a query set **Q**, for each point **q** ∈ **Q,** the *k*NN query retrieves its *k* nearest neighbors from interest objects in **P**. The variations of *k*NN queries consist of set *k* nearest neighbor (Set*k*NN) query, ordered *k* nearest neighbor (Order*k*NN) query, bi-chromatic reverse nearest neighbor (BR*k*NN) query. However, each type of query holds a specific query criterion. Existing approaches take long processing time in road network distance calculation for *k*NN queries. On the other hand, for fast kNN retrieval on large road networks, efficient data structures for spatial data such as R-tree, network Voronoi diagram have also been actively studied.

In this paper, we propose vicinity query algorithm based on the higher–order network Voronoi region (NVR) which gives a unified procedure to run an individual Set*k*NN, Order*k*NN, BR*k*NN, and distance range query in an integrated framework. Moreover, we studied the on demand network Voronoi regions (NVR) generating method for large road networks. In the literature, Voronoi Diagram (VD) has been applied in several fields including geographical information system (GIS) and computer vision.

The Voronoi diagram (VD) based on Euclidean distance is called ordinary VD. The VD

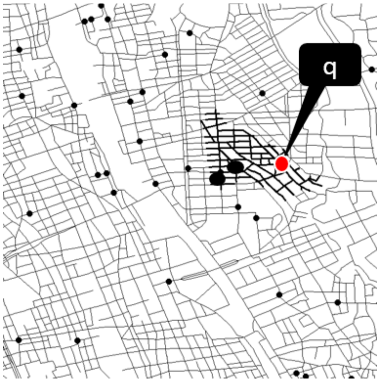

Figure 1. The Network Voronoi Region

is then extended to network Voronoi diagram (NVD) for the road network and practically used for the location based services (LBS).

For instance, when a set of interest objects *P* and a query point *q* are given, the nearest neighbor (NN) of *q* can be retrieved easily based on the NVD. As an alternative, by referring higher-order NVD, a range search for *k* nearest neighbor (*k*NN) can effectively be performed. However, generating higher-order NVD in advance is not feasible in practice. Moreover, NVD for the entire road network is not always necessary to be generated for the searching because the only partial region of the road network is required to search depending on the query criteria in most cases. The original existing approach takes drastic processing time.

Thus, in this paper, we adapted NVD generation concept to network Voronoi region (NVR) as shown in Figure 1. This figure shows the 2NN query result (black big circles) for a given query point *q*, and its on-demand NVR (a thick black lines region) where 2NN lie.

In existing approaches, the search NVR region is gradually enlarged in the road network distance starting from a given query point *q* by using Dijkstra's algorithm. In this node expansion, it is also verified whether every adjacent network node to be expanded lies the same region with the query point (q). In this case, if the verification result is true, adjacent nodes are expanded. Otherwise, the region expansion is terminated. When the search region is large or the query set is sparsely distributed, the verifying process in the road network distance takes long processing time. This fact degrades the performance. To improve in the performance of each query in terms of processing time, this paper proposes a fast algorithm by applying a single-source multi-targets A* algorithm (SSMTA*) [7], and introduces an integrated and adaptive framework for several vicinity queries.

The contributions of the paper are as follows.

(1) A fast algorithm for vicinity queries including set *k* nearest neighbor (Set*k*NN) query, ordered *k* nearest neighbor (Order*k*NN) query, bi-chromatic reverse nearest neighbor (BR*k*NN) query, and distance range query in road network distances is introduced.

(2) Through extensive experiments, the proposed method has a great efficiency in processing time comparing to existing works especially when query objects are sparsely distributed.

The rest of the paper organizes as follow. In Section 2, related works to vicinity queries and NVR are discussed. Section 3 discusses algorithms of the proposed method, and applies the method to various kinds of vicinity queries. Section 4 presents the performance evaluation of the proposed method. Section 5 concludes this paper and describes future works.

2. Related Work

Several queries similar to vicinity query described in this paper have been researched based on Voronoi Diagram [1]. For example, the region giving the same *k*NN query result set with unsorted order distance of the *k*NN result is called the set *k* Voronoi region and Voronoi region with *k*NN result of the sorted order distance is called ordered order-*k* Voronoi region. However, the result for several types of queries in the road network distance is apt to differ from the Euclidean distance.

On the other hand, the road network distance is suitable when the move of the query point is restricted on the road network. Consequently, the *k*NN query [2], R*k*NN query [3], and range query [4] in road network distances have been actively researched. However, the concept to expand search regions is basically implemented by Dijkstra's algorithm [6] in these queries and it takes long processing time.

The network Voronoi diagram (NVD) [8] was proposed to be adaptable to road networks. The generation method of the first order NVD was proposed by Erwig [8] and Hakini et al. [5]. The NVD is generated by expanding the region from a specified point s($\in$ S) while it is the nearest neighbor for expanding nodes. This can be done by a simultaneous Dijkstra's algorithm [6] starting from all query points as generators. However, this method requires long processing time when the number of the network nodes is large.

Furuta et al. [5] proposed algorithms to generate the higher order NVD. The principle of these algorithms is as same as the first order case described above. When the higher-order NVD for *k*NN in which *k* is more than one, is generated, all nodes in the road network is checked *k* times in node expansions. Therefore, this algorithm takes long processing time when *k* is large.

The application area of the NVD is wide. The *k*NN query algorithm using the first order NVD was proposed in [2]. Alternatively, the similar concept is applied to bichromatic reverse *k* nearest neighbor (Br*k*NN) query in [9].

On the other hand, Yiu et al. [3] proposed the Eager algorithm to search R*k*NN on the fly in road network distance. Cheema et al. [4] proposed a continuous R*k*NN query algorithm on road network distance. They also proposed the distance range query algorithm [10]. These algorithms gradually enlarge the search area from the query point similar to Dijkstra's algorithm checking whether the query criterion is satisfied. The most time consuming part in these algorithms is the checking process whether the query condition is satisfied.

This paper proposes a fast algorithm applicable to several types of vicinity queries.

3. Proposed Method

In this section, definitions for vicinity queries with a given interest object set **P** and a query object set **S** are described.

3.1 Vicinity Queries

Definition 1: SetkNN query *is a kNN query to find interest objects in* ***P*** *whose kNN in set* ***S*** *including the same set. The order of distances in kNN result is not considered.*

Definition 2: Order*k*NN query *is similar kNN query to SetkNN query in which the order of the distances in kNN result is considered.*

Definition 3: BRkNN query *is a query to find interest objects in* ***P*** *whose kNN in set* ***S*** *contains a specified query object in S.*

Definition 4: Distance range query *is a query to find query objects in* ***S*** *whose distances from a query point* ***q*** *are less than or equal to a distance* ***d*** *for the given distance and the query point.*

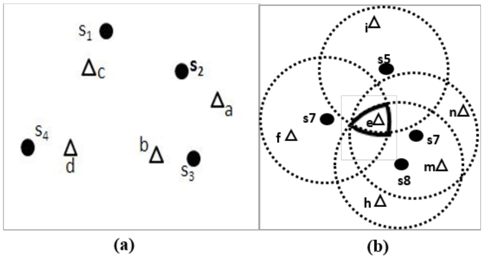

Figure 2. Example for vicinity queries

In Figure 2(a), the dark circles represent given query objects s_1 to s_4 in the set **S** and the triangles represent interest objects ***a*** to ***d*** in the set **P**. While considering for 2NN, the result set of each query type will be as shown in the Table 1. Similarly, Figure 2(b) shows the distance range query.

Table 1. Result set for Vicinity Queries

Query type	**Result Set**
Set*2*NN	{s1, s2} for **a** and **b**
Order*2*NN	{s2, s3} for **a** and {s3, s2} for **b**
Br*2*NN	{a, b, c}, s2 as a query point
Distance Range	{s5,s6,s7,s8} lie within NVR of **e**

As described above, these queries are called vicinity queries in this paper. When *VQ* for set *k*NN is denoted by VQ_{set}, for ordered *k*NN by VQ_{order} and for BR*k*NN by VQ_{br}, the relationship of these vicinity queries can be described as in the equation (1):

$$VQ\,order \subseteq VQ\,set \subseteq VQ\,br\,(1)$$

3.2 How to generate 1-order NVR

A large road network is considered and modeled as a directed graph ***(V, E, W)***, where ***V*** is a set of nodes, ***E*** is a set of edges (road segments) and ***W*** is a set of edge weights and **w** **(∈ W) ≥0** stands. In the rest of the paper, ***w*** is assumed as the length of edge.

In this section, we present how to determine NVR for 1NN, in other words, 1-order network Voronoi region, and node expansion process. For a given query point ***q*,** NVR (q) is a set of road segment in which **q (∈ S)** lies as the nearest object. An edge on **G** may belong to plural NVR, and the edge is exactly partitioned into plural regions. Figure 3 illustrates a road network
node. In this figure, white circles are road network nodes, black circles are query objects in **S**, and the numbers attached on road segments are distances. The area bounded by **✕** shows the 1-order network Voronoi region in which **q** is included as the nearest object when it is considered as a query point.

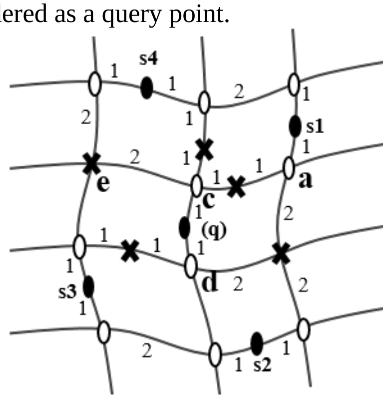

Figure 3. The 1-order NVR

The basic principle how to generate NVR region for vicinity queries is followed by two steps below:

(1) As the similar way in Dijkstra's, the search region is gradually expanded from a query point q to adjacent nodes.

(2) At every visited node in step (1), expanded nodes are verified whether they meet the query criterion or not.

In the above step (1), a best-first search is applied from a query point to a current node referring to priority queue (PQ). In the step (2) after verifying whether the current node meets the query criterion, all adjacent nodes to the current nodes are inserted into PQ. Then, the further search proceeds from these nodes ahead. Reversely, if the query criterion is not satisfied at the current node, node expansion from this node ahead is terminated.

In the next section, the algorithm for vicinity queries based on NVR region is discussed.

3.3 Vicinity Query Algorithm

The algorithm VQA describes how to generate a region (NVR) by expanding the search area gradually while the query criterion is satisfied.

The PQ in line 2 and 3 is the priority queue to control the region expansion. In line 4, a closed set (CS) is prepared for checked road segment to avoid duplicated checks.

Algorithm: Vicinity Query Algorithm(VQA)

```
1:  function VQA (q)
2:     PQ.enqueue(<d_N(n_1,q),n_1,q,l(n_1,q)>)
3:     PQ.enqueue(<d_N(n_2,q),n_2,q,l(n_2,q)>)
4:     CS ← Ø
5:     RS ← Ø
6:     T ← INITIALIZE (q)
7:     while PQ.size () > 0 do
8:        r ← PQ.deleteMin ()
9:        if CS contains r then
10:           continue;
11:       end if
12:       CS ← CS ∪ r.l
13:       if CHECK(r.n, T) then
14:          ns ← Adjacent Node (r.n)
15:          for all n ∈ ns do
16:          PQ ← PQ ∪ < r.c + d_N (r.n, n), n, n.r, r.l >
17:          end for
18:          RS ← RS ∪ ADDPONL (r.l)
19:       else
20:       RS ← ∪ POINTONLINKCHECK (r.l,q,T)
21:       end if
22:    end while
23:    return RS
24: end function
```

RS in line 5 is the result set of interest objects in ***P*** set. In line 6, INITIALIZE function is called to search NN of ***q***, and the result is assigned into ***T***. The function INITIALIZE is differently implemented based on the query type. Line 7 to 21 perform the following process. Initially, a record with the minimum cost is dequeued from PQ, and the road segment of the record *r.l* is checked whether it is already registered in the CS. If *r.l* is in the CS, the segment has already been checked, and the rest steps are skipped. In line 12, *r.l* is added into the closed set.

In line 13, the current node *r.n* is checked whether the node meets the query criterion. Here, the query condition is the NNs of the current node *r.n* are same as objects in ***T***. If the result of CHECK is true, the region is expanded. New records for all adjacent nodes to *r.n* are enqueued into PQ. Then, all *p* ***(∈P)*** on *r.l* are added into *RS* in line 18. For the false result by CHECK, line 20 checks whether each *p* ***(∈P)*** on *r.l* meets the query condition or not, and only interest objects satisfying the query condition are added into *RS*.

The algorithm VQA is applicable to a variety of *k*NN queries including set *k*NN, ordered *k*NN and BR*k*NN, and distance range query. To adapt these queries, three functions, INITIALIZE (*q*), CHECK and POINTONLINKCHECK are needed to prepare for passing specific parameters for an individual query.

3.4. Improvement by SSMTA*

The most time consuming step in Algorithm is at the CHECK function called for every node in region expansion. For a visiting node *n*, *k*NN candidates of ***n*** are searched in Euclidean distance, and these candidates are verified in the road network distance. To verify in the road network distance, the A* algorithm can be applied. However, in the general VQ, at least *k* numbers of objects are searched as targets at every node *n*. In such condition, even if original A* algorithm is fast in practice, the processing time becomes long due to repeated searching in adjacent regions.

To improve the efficiency in terms of processing time for the distance calculation in road network distances, the idea of the single–source multi–targets A* (SSMTA*) [7] algorithm is applied to the proposed method. The SSMTA* finds the shortest paths from a source node to multiple target nodes by changing target points sequentially in VQ queries avoiding duplicated expansions. The original A* algorithm finds the shortest path from a start point *s* to a target point *n* repeatedly every time the target point is changed. Comparing to it, the processing time is considerably reduced in the improved method. Moreover, the update process for all records in the priority queue is taken place in the memory and it does not take long processing time.

4. Experimental Results

To evaluate the performance of the proposed method, we conducted experiments comparing to existing works in which the original A* algorithm is used. In the proposed method, we applied SSMTA* algorithm in road network distance calculation. For simulation data in experiments, a real road map of Saitama city, Japan with 16,284 nodes, and 24,914 links is used. Density in these experiments represent the density of objects and Density 0.01 refers that an interest object exists once 100 road network links. The proposed method (Propose) and the comparison method, the original A* (Origin) were implemented in Java and evaluated on a PC with Intel Corei7-4770 CPU (3.4 GHz) and 32 GB memory.

Figure 4 (a) shows the comparison in the processing time of VQset for the Set*k*NN query when the density of query objects set ***S*** is 0.005.

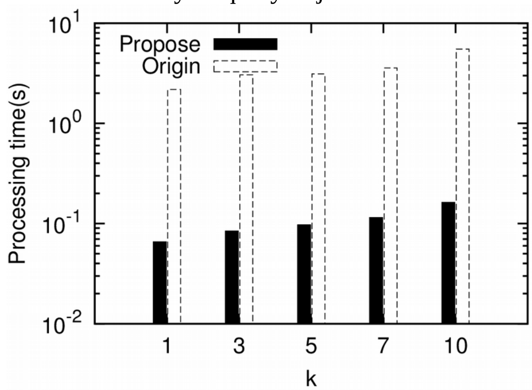

Figure 4 (a) VQset when the *density* is 0.005

In this figure, the processing time increases according to the increase of the value *k* in both methods. As shown in the figure, the proposed method is more than 10 times faster than the comparison method.

Figure 4(b) shows the processing time of VQset when *k* is fixed at 5 and the density of query objects are varied as shown in the figure.

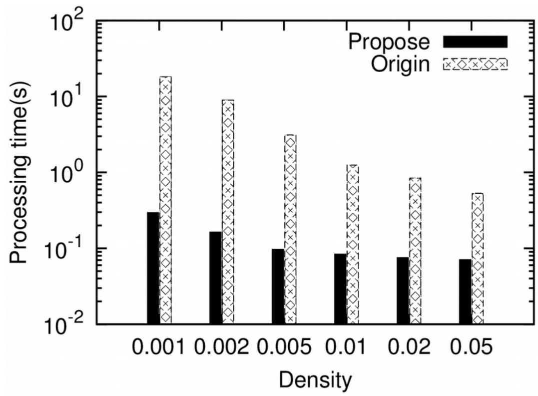

Figure 4 (b) VQset when *k* value is 5

In this result shown in Figure 4(b), the processing time decreases according to the increase of the density of interest objects because the region size for VQset also decreases.

Comparing two methods, the proposed method has 10 to 100 times faster in the processing time.

Figure 5 (a) and (b) compares the processing time for VQorder for Order*k*NN query between the proposed method and the origin method.

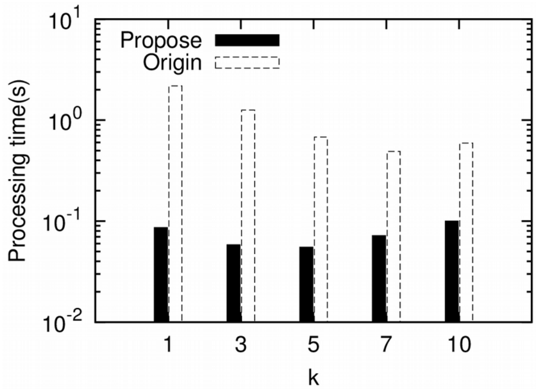

Figure 5 (a) VQorder when the *density* is 0.005

Figure 5 (a) shows the processing time varying the *k* value. In the origin method, the processing time once decreases according to the increase of the *k* value. Because the region size of VQorder decreases according to *k* value increase, and the number of times to invoke CHECK function also decreases.

For the result shown in Figure 5 (b), the value of *k* is fixed at 5. When the density increases, the processing time decreases. This is because the region size is monotonically decreases due to the increase of the density.

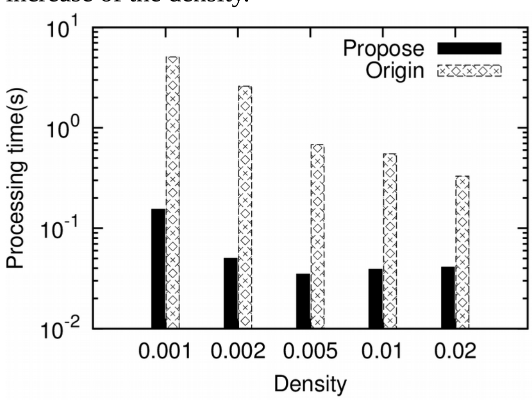

Figure 5 (b) VQorder when *k* value is 5

Figure 6 (a) shows the processing time for VQbr for BR*k*NN query. In this figure, the density is set to 0.005. Due to the increase in region size, this query needs long processing time than previous two queries (VQset and VQorder). The processing time of this query is most affected by the *k* value increase.

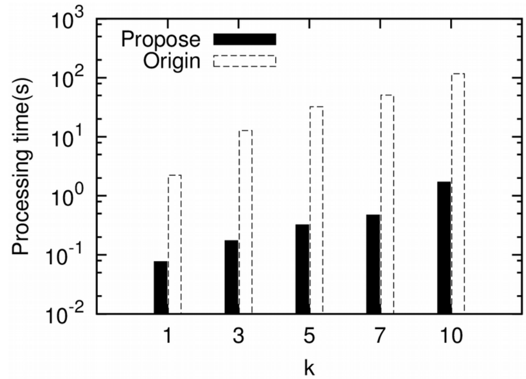

Figure. 6(a) VQbr when the *density* is 0.005

Figure 6(b) shows the comparison in processing time between the proposed method and the origin method by setting *k* value at 5 and varying the density of interest objects.

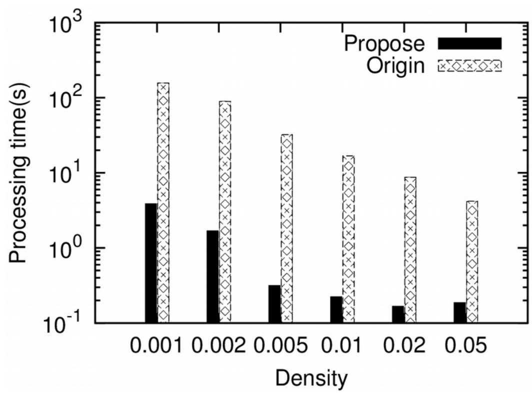

Figure 6 (b) VQbr when k value is 5

Figure 7 (a) shows the processing time VQdist for the distance range query. In this result, the distance range ***d*** is fixed at 1.5km varying the density of ***S***. In this result, the processing time increases when the density of ***S*** becomes denser. Contrary, the basic method takes less processing time when the density of ***S*** is denser.

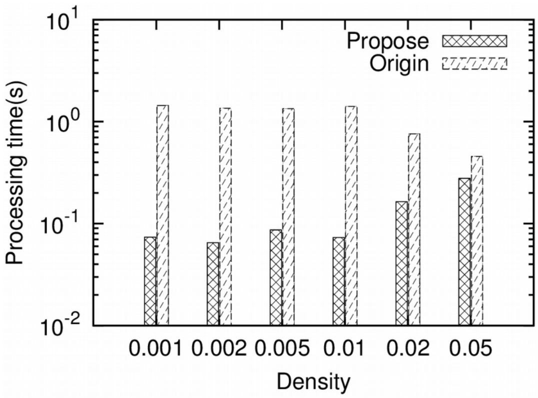

Figure. 7 (a) VQdist when *d* value is 1.5km

Figure 7 (b) shows the processing time for the distance range query varying the distance range ***d***. When the ***d*** value increases, the region becomes larger and the basic method takes long processing time. However, the proposed method reduces the processing time to less than one tenth of the origin method.

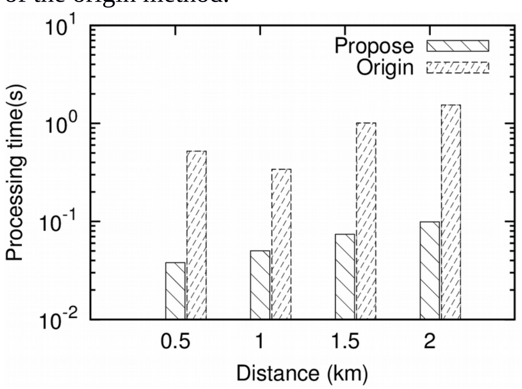

Figure. 7 (a) VQdist when the distance *d* varies

5. Conclusion

In this paper, a vicinity query algorithm in road network distance was proposed. It is an integrated framework adaptable for several vicinity query methods including *k*NN query, ordered *k*NN query, BR*k*NN query and distance range query. In the basic idea of the proposed algorithm, when a query point *q* is given, the search region is gradually expanded from *q* to its neighboring road network nodes by verifying whether these nodes meet the query condition or not. In the verification step, SSMTA* was applied to improve the performance for the road network distance verification. Besides, the concept of generating NVR region in road network distance on demand was introduced and applied to vicinity queries.

With extensive experiments, the proposed method outperformed existing works in terms of processing time. The proposed method is almost 100 times faster than the origin method especially when the density of distribution of data objects is sparse. Moreover, the proposed algorithm is suitable for queries for mobile objects within safe region in which the query result remains unchanged. Applying the proposed method to more complicated queries is future works.

References

[1] Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial Tesssellations, Concepts and Applications of Voronoi Diagrams, 2nd ed. John Wiley & Sons (2002)

[2] Kolahdouzan, M., Shahabi, C.: Voronoi-based K nearest neighbor search for spatial network databases. In: Proc. 30th VLDB. (2004) 840-851.

[3] Yiu, M.L., Papadias, D., Mamoulis, N., Tao, and Y.: Reverse nearest neighbor in large graphs. IEEE Transaction on Knowledge and Data Engineering 18(4) (2006) 1-14

[4] Cheema, M.A., Zhang, W., Lin, X., Zhang, Y., Li, and X.: Continuous reverse k nearest neighbors queries in Euclidean space and in spatial networks. VLDB Journal 21 (2012) 69-95

[5] Furuta, T., Suzuki, A., Inakawa, and K.: The *k*th nearest network Voronoi diagram and its application to districting problem of ambulance systems. Discussion Paper, No.0501, Center for Management Studies, Nanzan University (2005)

[6] Dijkstra, E.W.: A note on two problems in connection with graphs. Numeriche Mathematik 1 (1959) 269-271

[7] H.Htoo, Ohsawa, Student Member, "Incremental Single Source Multiple Target A* Algorithm" IEICE Transactions on Information and Systems E96-D (5) (2013)1043-105

[8] Erwig, M.: The graph Voronoi diagram with applications. Network 36(3) (2000) 156-163

[9] Tran, Q.T., Taniar, D., Safar, and M.: Bichromatic reverse systems neighbor search in mobile systems. IEEE System Journal 4(2) (2010) 230-242

[10] Cheema, M.A., Brankovic, L., Lin, X., Zhang, W., Wang, W.: Continuous monitoring of distance based range queries. IEEE Transactions on Knowledge and Data Engineering 23 (2011) 1182-1199