D-002

Fast Algorithm for Multi-type Nearest Neighbor Query

Aye Su Yee Win[1] Htoo Htoo[1] and Yutaka Ohsawa[1]

Graduate School of Science and Engineering, Saitama University[1]

1 Introduction

When a query point q and an arbitrary number of point of interest (POI) sets are given, the query to find the shortest route to visit POIs selected from each specified category set is called a multi-type nearest neighbor (MTNN) query. Though this type of query has been named variously according to the condition of query, Ma et al. [1] named it an MTNN query.

A similar query as the MTNN is a trip planning query (TPQ). In this query, the final destination of a trip is given, and then a minimum length route between the starting point (s) and the final destination (d) via POIs chosen each from the specified POI categories is found. The main difference between MTNN and TPQ is that the final destination is given as the same point (MTNN) or as a different point (TPQ) with the starting point.

There are three types of queries been proposed for trip planning query. One is specified a visiting order of the POI categories uniquely, and it is called as optimal sequenced route query (OSR) [2]. Another one is not specified the visiting order, and it is named as a trip planning query by Li et al. [3]. The last one is specified some partial visiting orders, and it is called a multi-rule partial sequenced route (MRPSR) query [4].

This paper proposes a fast algorithm to find an MTNN route in the road network distance.

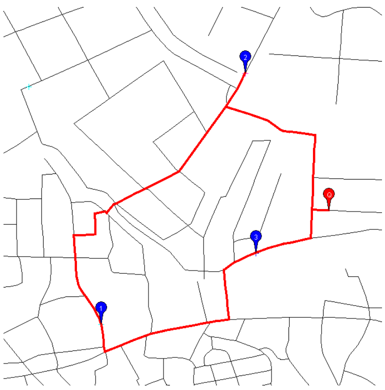

Fig. 1 Example of MTNN query

2 Proposed Method

The MTNN query can be considered as a special case of TPQ in which the start and destination are the same point. Therefore, algorithms developed for TPQ are applied to MTNN query. The authors have presented an algorithm for TPQ based on the incremental Euclidean restriction (IER) strategy [5]. This algorithm finds the shortest TPQ route in the Euclidean distance on the prerequisite that POI data sets are managed by R-trees,

The basic principle of the MTNN query algorithm presented in this paper is as follows.

(1) Search the shortest MTNN route in the Euclidean distance incrementally.

(2) Verify the road network distance of the candidate route.

(3) Repeat (1) and (2) while the length of the next candidate route is less than the current shortest route in the road network distance.

2.1 Search in Euclidean distance

Let the query point be q and the POI categories to be visited be $C_i(1 \leq i \leq M)$, here M is the number of categories to be visited. Incremental MTNN route search in the Euclidean distance, mentioned in the first step of the procedure, can be done as the following. First, the nearest POI in C_1 to q is searched. Let this POI be P_1^1, here P_i^j means the jth NN (nearest neighbor) to q in the category C_i. This NN query is performed incrementally. Next, the MTNN route starting from P_1^1 visiting each POI in $C_i(i \leq 2 \leq M)$ and finally q is searched. This route is called as a partial sequenced route (PSR) and it is denoted as $R(1, q)$. Then, after we add the path from q to P_1^1, we can obtain a total MTNN route. For the search to find $R(1, q)$, the method proposed in [5] can be used.

The route to visit P_1^1 may not always be the shortest route. Therefore, when P_1^k is found by the kNN query from q, P_1^{k+1} is searched and visited, and then the respective route is inserted into the priority queue that controls the best-first search process.

2.2 Verify in road network distance

The candidate of the shortest MTNN route can be obtained by the method described in the previous section. Next, the length in the road network distance is calculated to verify that the candidate route is truly the shortest.

For this operation, to find the road network distance between two points is necessary. When two terminal points are specified on the road network, the shortest path length on the road network can be obtained by A* algorithm or a materialized path view method efficiently.

In the MTNN routes produced by the Euclidean distance are apt to contain the same partial sequences. To avoid duplicate calculation of the road network distance between the same terminal pair of points, these points and the distance are recorded in a distance table. And then, when a distance calculation of a pair of points is required, the table is first checked. If these points are found in the table, the distance is simply returned. Otherwise, the distance is calculated and a new record is registered into the table. This materialization approach reduces the road network distance calculation cost drastically.

3 Experimental Results

We implemented the proposed algorithm and PNE [2] in Java and performed experiments to evaluate the processing time of these methods. The system used in this experiment was Intel Core i7 CPU (3.2GHz) with 32GB memory. The road map data we used was organized by 16,283 nodes and 24,914 road links. The POI data were generated by pseudo-random sequences.

From Fig.2 to Fig.4 show the processing time varying the number of visiting POI categories (M) from 3 to 5. The horizontal axis shows the density of the POI ($Prob$). The POI density was defined by the average number of the POI existing on a road network link. For example, $Prob = 0.01$ means the density that one POI exists over 100 road links.

When M is small, the processing time of PNE is quite fast (see Fig. 2), however, the processing time increases rapidly according to increasing number of M. Especially, when the POI density is high, PNE requires very long processing time. On the other hand, the proposed method can search the MTNN route faster (under 1s) even the POI density is high.

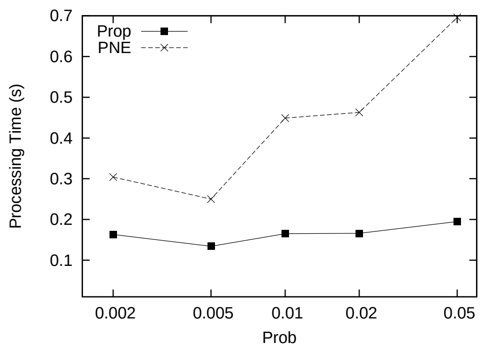

Fig. 2 M=3

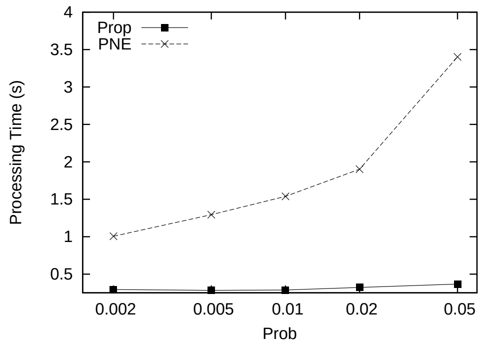

Fig. 3 M=4

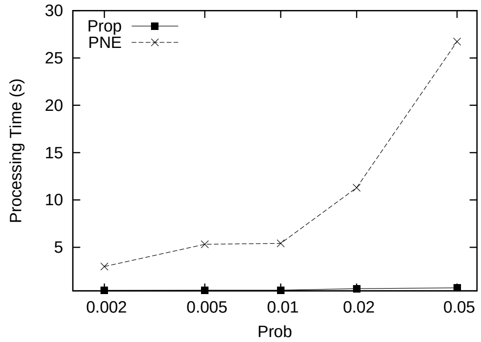

Fig. 4 M=5

4 Conclusion

This paper proposes an algorithm to find MTNN on road network distance using IER strategy. By comparing with a conventional method (PNE), we presented that the proposed method considerably outperforms with the conventional method.

References

[1] X. Ma, S. Shekhar and H. Xiong: "Multi-type nearest neighbor queries in road networks with time window constraints", ACM GIS'09, pp. 484–487 (2009).

[2] M. Sharifzadeh, M. Kalahdouzan and C. Shahabi: "The optimal sequenced route query", Technical report, Computer Science Department, University of Southern California (2005).

[3] F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios and S.-H. Teng: "On trip planning queries in spatial databases", Proc. SSTD 2005, pp. 273–290 (2005).

[4] H. Chen, W. S. Ku, M. T. Sun and R. Zimmermann: "The multi-rule partial sequenced route query", ACM GIS '08, pp. 65–74 (2008).

[5] Y. Ohsawa, H. Htoo, N. Sonehara and M. Sakauchi: "Sequenced route query in road network distance based on incremental euclidean restriction", DEXA 2012, LNCS 7446, pp. 484 – 491 (2012).