A New Similar Trajectory Search Algorithm Based on Spatio-Temporal Similarity Measure for Moving Objects in Road Networks

Young-Chang KIM†, Nonmember and Jae-Woo CHANG†(a), Member

SUMMARY The deployment of historical trajectories of moving objects has greatly increased for various applications in road networks. For instance, similar patterns of moving-object trajectories are very useful for designing the transportation network of a new city. In this paper, we define a spatio-temporal similarity measure based on a road network distance, rather than a Euclidean distance. We also propose a new similar trajectory search algorithm based on the spatio-temporal measure by using an efficient pruning mechanism. Finally, we show the efficiency of our algorithm, both in terms of retrieval accuracy and retrieval efficiency.

key words: similar trajectory search, spatio-temporal similarity measure, road network, moving objects

1. Introduction

Because of the recent evolution of different devices and technologies such as mobile devices, sensor networks and GPS technologies, a tremendous amount of spatio-temporal data about moving objects in road networks has been generated, in the form of trajectories. As a result, many researchers are interested in analyzing the trajectories of moving objects for different purposes[1], [2]. In this scenario, handling the movement of moving objects in road networks is a central issue. However, prior to analyzing the patterns of moving objects’ trajectories, it is necessary to search those trajectories which are similar to a given query trajectory.

There exists a small volume of work on similar trajectory search for moving objects by considering spatial network distance [3]–[5]. Hwang et al. [3] proposed a similar trajectory search algorithm in a road network, based on identical POIs (Points of interest) and TOIs (Times of interest) of the query trajectory. However, the algorithm fails to find those trajectories which are temporally or spatially very close to a given query but do not share its TOIs or POIs because it must share all POIs of the query trajectory. Chang et al. [4] proposed a signature-based trajectory search algorithm. They defined spatial similarity as an average shortest network distance between a query trajectory’s node and a data trajectory’s nodes as well as defined temporal similarity as a combination of a time range for 24 hours, day, and week. However, the algorithm only considers similar trajectories as those which share at least one common node with a query trajectory. In addition, Tiakas et al. [5] proposed another type of similarity measure and a similarity search algorithm using an M-tree to search for similar trajectories of moving objects in a spatial network. They defined a spatial similarity measure as an average ratio of the minimum distance and the maximum distance between a query’s node and a data trajectory’s node. However, if a minimum and maximum distance ratio is the same for a different pair of trajectories, the algorithm is unlikely to distinguish which pair is most similar to a query.

In order to solve these problems, we first define a spatio-temporal similarity measure based on a road network distance, rather than a Euclidean distance. To efficiently and accurately retrieve all similar trajectories, we also propose a new similar trajectory search algorithm based on the spatio-temporal measure, by using an efficient pruning mechanism. Compared with the work of Hwang et al. [3] and Chang et al. [4], our algorithm has no requirement that a data trajectory should share at least a common node with a query trajectory to be a similar one. Compared with the work of Tiakas et al. [5], which calculates nearly the same similarity for most of trajectories in a bi-directional spatial network, our algorithm provides a precise similarity measure by using a real network distance and thus is suitable for both a uni-directional and a bi-directional spatial network. In addition, our similar trajectory search algorithm can achieve good retrieval performance because it can prune out those trajectories which never become candidates for similar ones by using grid cell distances.

2. Spatio-Temporal Similarity Measure

In general, we can represent a spatial network in terms of a directed graph $G(V,E)$, where $V$ is the set of vertices (junctions or intersections of the road network) and $E$ is the set of edges (road segments) [5]. Hence, we can define a trajectory $T$ of a moving object in a spatial network as $T = \{(v_1,t_1), (v_2,t_2), \ldots, (v_n,t_n)\}$, where $n$ is the number of segments of the trajectory $T$, $v_i$ represents the $i$th node of trajectory $T$, and $t_i$ is the time stamp of the node $v_i$, where $1 \leq i \leq n$. We define both a spatial similarity measure and temporal similarity measure, which are either equal (same length) or unequal (different length) to a given query trajectory. We first assume two trajectories $T_a$ and $T_b$ without

Manuscript revised October 13, 2008.

The authors are with the Dept. of Computer Eng., Chonbuk National Univ., Chonju, Chonbuk 561–756, Korea.
a) E-mail: jwchang@chonbuk.ac.kr
DOI: 10.1587/transinf.E92.D.327

Copyright © 2009 The Institute of Electronics, Information and Communication Engineers
considering time. Both of them have the same length \( m \), i.e., \( T_a = (V_{a1}, V_{a2}, \ldots, V_{am}) \), \( T_b = (V_{b1}, V_{b2}, \ldots, V_{bm}) \). In a spatial network, two trajectories are similar if they are spatially close to each other. Since moving objects always traverse limited paths, the spatial closeness between their trajectories is expressed in terms of a spatial network distance, rather than a Euclidean distance. Therefore, the distance between two nodes is defined by the shortest network distance between them. In addition, those trajectories which are close to a query trajectory can be more similar than a trajectory sharing some segments with the query trajectory. Thus, the ratio of the shared portion over the total length of the trajectory should be used. As a result, the spatial distance between two trajectories is defined as follows:

**Definition 1.** The spatial distance \( D_{sub} \) between two trajectories, \( T_a \) and \( T_b \), with equal length \( m \) is defined as an average network distance between the corresponding nodes of the two trajectories. Where, \( D_{network} \) is the shortest real network distance between two nodes in the spatial network, the \( SharedEdgeRatio \) is the ratio of the length of the shared edge to the total length of the trajectory, \( \alpha \) is the weighted value for the \( SharedEdgeRatio \).

\[
D_{sub}(T_a, T_b) = \frac{1}{m} \sum_{i=1}^{m} |D_{network}(V_{ai}, V_{bi})| \\
\times \left( \frac{1}{1 + SharedEdgeRatio \times \alpha} \right)
\]

To calculate a spatial similarity between two trajectories with different lengths, we separate trajectories into a set of sub-trajectories with fixed size, with respect to the number of segments. Then, we find the closest sub-trajectory pairs for the two trajectories and average their distances to calculate a spatial similarity. The spatial similarity \( SS \) between two trajectories can be defined as follows:

**Definition 2.** The spatial similarity \( SS \) between two trajectories, \( T_a \) and \( T_b \), with different lengths \( n \) and \( m \), respectively, is defined as an average spatial distance of sub-trajectory pairs. Where, \( l \) is the length of sub-trajectory, \( 1 < l \leq n \) and \( 1 < l \leq m \).

\[
SS(T_a, T_b) = \frac{1}{n-l+1} \sum_{i=1}^{n-l+1} \text{MIN}(D_{sub}(T_{ai}, T_{bi}))
\]

In a spatial network, a time factor is as important as a spatial factor for representing patterns of moving objects. For example, we generally experience traffic jams during morning rush hour. Therefore, it is necessary to define a temporal similarity measure to search similar trajectories of moving objects. The temporal closeness between two trajectories is significant for defining a temporal similarity measure. Thus, we can define a temporal similarity in terms of the temporal distance between the corresponding nodes of two trajectories.

**Definition 3.** The temporal distance \( D_{time} \) between two trajectories, \( T_a = (t_{a1}, t_{a2}, \ldots, t_{am}) \) and \( T_b = (t_{b1}, t_{b2}, \ldots, t_{bm}) \), with equal length \( m \) is defined as an average of absolute time differences between the corresponding of the two trajectories.

\[
D_{time}(T_a, T_b) = \frac{1}{m} \sum_{i=1}^{m} |t_{ai} - t_{bi}|
\]

To define a temporal similarity measure for two different length trajectories, we use a temporal distance in the same manner as the spatial similarity measure. We define a temporal similarity measure \( TS \) as follows:

**Definition 4.** The temporal similarity \( TS \) between two trajectories, \( T_a \) and \( T_b \), with different length \( n \) and \( m \), respectively, is defined as an average temporal distance of sub-trajectory pairs from \( T_a \) and \( T_b \).

\[
TS(T_a, T_b) = \frac{1}{n-m+1} \sum_{i=1}^{n-m+1} D_{time}(T_{ai}, T_{bi})
\]

Now, we define a spatio-temporal similarity measure in terms of a combination of the spatial and temporal similarity measures. To make equivalence between two similarity measures, we multiply the average velocity of moving objects by the temporal similarity. Then, we combine both spatial and temporal similarity measures using two weights \( W_{net} \) and \( W_{time} \) for different types of applications. A spatio-temporal similarity measure \( STS \) can be defined as follows:

**Definition 5.** The spatio-temporal similarity \( STS \) between two trajectories, \( T_a \) and \( T_b \), is defined as the summation of the spatial and the temporal similarity, \( SS \) and \( TS \), with different weights. Where, \( W_{net} + W_{time} = 1 \) and \( V_{avg} \) is the average velocity of moving objects.

\[
STS(T_a, T_b) = W_{net} * SS(T_a, T_b) + W_{time} * TS(T_a, T_b) * V_{avg}
\]

3. **New Similar Trajectory Search Algorithm**

Though R-tree variant index structures are used widely in a spatial network, there are a couple of problems with using them to search similar trajectories of moving objects. First, when many objects traverse the common segments of a road network, R-tree variants undergo substantial splitting and merging because of frequent insertions and updates. Secondly, the searching capability of R-tree variants is degraded because of a large amount of overlapped areas among their MBRs. Finally, R-tree variants cause long time to compute the shortest network distance between two nodes. For an efficient algorithm, it is essential to reduce the cost of distance computations and avoid irrelevant trajectories. We design a grid-based index structure which can search the similar trajectories of moving objects very fast by using an efficient pruning mechanism (Fig. 1). A spatial network can be partitioned into a set of cells of equal size. The grid-based index structure consists of two components: hash table and grid information file. A hash table contains the pointers of all grid cells in memory for direct accesses to the grid cells. Because each trajectory is represented by a list of cells which a moving object has traversed, a grid
information file contains all the cell information on disk. Each element of cell information has a form: 
\(<\text{#ofMO}, \text{MOIDArray}, \#ofAdjCell, \text{AdjCellArray}>\), where \#ofMO is the total number of moving objects traversing nodes in a grid cell. MOIDArray is an array of moving object IDs, \#ofAdjCell is the number of adjacent cells which contain nodes connected to the cell, and AdjCellArray is an array of adjacent cells’ coordinates.

To retrieve the \( k \) most similar trajectories to a given query in an efficient way, we propose a new similar trajectory search algorithm based on our spatio-temporal similarity measure by using the grid-based index structure. The grid-based index structure provides an efficient pruning mechanism by using an average cell distance.

**Definition 6.** The cell distance \( \text{CellDist} \) between two cells, \( \text{Cell}_a \) and \( \text{Cell}_b \), is defined as the minimum distance between them.

\[
\text{CellDist}(\text{Cell}_a, \text{Cell}_b) = \text{MINDIST}(\text{Cell}_a, \text{Cell}_b)
\]

**Definition 7.** The average cell distance between two trajectories, \( T_a = (t_{a1}, t_{a2}, \ldots, t_{am}) \) and \( T_b = (t_{b1}, t_{b2}, \ldots, t_{bm}) \), is defined as the average of cell distances between a cell including \( t_{ai} \) and a cell having \( t_{bi} \) for all \( 1 \leq i < m \).

\[
\text{AvgCellDist}(T_a, T_b) = \frac{1}{m} \sum_{i=1}^{m} \text{CellDist}(\text{Cell}_{t_{ai}}, \text{Cell}_{t_{bi}})
\]

The algorithm uses an average cell distance, rather than an average real network distance, because a cell distance in the spatial network is always less than or equal to a network distance between two nodes of different cells. As a result, we can define a pruning policy by using average cell distance as follows:

**Definition 8.** Pruning policy: Let \( \text{AvgNetDist} \) assume to be the \( k \)th average network distance, in a sorted order, between a query trajectory and candidate data trajectories. We can prune out those trajectories which have \( \text{AvgCellDist} \) larger than the \( \text{AvgNetDist} \).

Figure 2 presents an example of our pruning mechanism. Let \( Q \) be a query trajectory and \( k \) is 1. First, we calculate a sort of average cell distances for all candidates \( T_a, T_b, \) and \( T_c \). Because \( T_a \) passes the cells \( c7, c8, c13, c14 \), an average cell distance between \( Q \) and \( T_a \), \( \text{AvgCellDist}(Q, T_a) = 1/4 \times (0 + 0 + 0 + 0) = 0 \), by using Definition 5. Next, we calculate the average network distance to the maximum of the \( k \)th candidate. In the example, we calculate the average network distance for the 1st candidate \( T_a \), i.e., \( \text{AvgNetDist}(Q, T_a) = 1/4(0 + 22 + 0 + 0) = 5.5 \). We use the \( \text{AvgNetDist} \) to prune out those trajectories with larger \( \text{AvgCellDist} \). Because the \( \text{AvgCellDist} (= 5) \) of \( T_c \) is less than the current \( \text{AvgNetDist} \), we calculate \( \text{AvgNetDist}(Q, T_c) = 1/4 \times (22 + 22 + 23 + 23) = 22.5 \). Its value is larger than the current \( \text{AvgNetDist} \), so we discard \( T_c \). The \( \text{AvgCellDist} (= 8.1) \) of \( T_b \) is larger than the \( \text{AvgNetDist} \), so we prune out \( T_b \).

Figure 3 presents our similarity search algorithm using our pruning mechanism. First, the algorithm obtains the grid cells traversed by a given query (line 1) and counts the number of trajectories in the cells (line 2). Secondly, if it cannot obtain \( k \) trajectories, it obtains the required number of \( k \) similar trajectories by expanding the closest adjacent cells (line 3–5). Thirdly, it reads all the trajectories of the candidate set from a data trajectory file (line 6). Fourthly, the algorithm calculates a spatio-temporal similarity of the \( k \)th trajectory and inserts it into a result set (line 7). The value of the \( k \)th trajectory is used as a distance for range searching. It obtains those trajectories which traverse the grid cells to satisfy the range search (lines 8–9). The average cell distance of each candidate is calculated and compared with the value of the \( k \)th trajectory (lines 10–12). If the average cell distance of a candidate is larger than the value of the \( k \)th trajectory, then the candidate is discarded. Otherwise, the algorithm calculates a real distance for the candidate and finally updates the value of the \( k \)th trajectory.
4. Performance Analysis

To analyze our similar trajectory search algorithm, we implemented it on Windows 2003 enterprise server using Visual studio .Net under HP ML 150 G3, Intel Xeon 3.0 GHz dual and 2 GB memory. The data sets used in our experiment are based on the road network of Pusan city, South Korea, which consists of 935 nodes and 1,800 edges. By using Pusan National University data generator [3], we generate five data sets which contain 1,000, 2,000, 3,000, 4,000 and 5,000 moving objects’ trajectories. The data sets have a non-uniform (i.e., skewed) distribution because they are generated to simulate real moving objects (cars) in the road network of Pusan city. Figure 4 shows a snapshot of moving objects in a part of Pusan city’s road network.

We use a query set being created by replacing 10–50% segment of the original data trajectory. The query set contains 1,000 queries, each varying from 10 to 50 segments in size. We set the size of grid cells as 100 meter, which is the average length of the segments of the road network in Pusan city in order to minimize the negative effect of skewed data. We compare the performance of our algorithm with that of Tiakas et al.’s algorithm [5] (in short Tiakas’s algorithm), in terms of accuracy and retrieval time. We exclude Hwang et al.’s algorithm [3] and Chang et al.’s one [4] for performance comparison because they have a restriction of trajectories sharing with one or all POIs of a query trajectory.

To evaluate the quality of our algorithm, we conduct experiments on queries to search \(k (=10)\) similar trajectories with different \(W_{net}/W_{time}\) values where \(W_{net}\) and \(W_{time}\) mean the weights of spatial and temporal similarity, respectively. Figure 5 shows retrieval accuracy. It is shown that our algorithm has about 97% accuracy while Tiakas’s algorithm has about 60% accuracy when both \(W_{net}\) and \(W_{time}\) are the same (= 0.5). This is because our algorithm uses a more precise spatio-temporal measure. The result shows the relatively little importance of \(W_{net}/W_{time}\) values on retrieval accuracy. This is because the spatial similarity measure dominates the temporal measure in most cases.

To show the retrieval efficiency of our algorithm, we measure response time for \(k (=10)\) similar trajectory queries with \(W_{net}/W_{time} = 1\). Figure 6 shows retrieval efficiency. For 1,000 data trajectories, our algorithm requires about 10 seconds in retrieval time while Tiakas’s algorithm requires about 68 seconds. For 5,000 data trajectories, our algorithm requires about 30 seconds, whereas Tiakas’s algorithm needs about 230 seconds. This is because our algorithm can prune out unnecessary trajectories by using an efficient pruning mechanism while Tiakas’s algorithm needs to search all sub-trajectories of the original trajectory. Figure 7 shows retrieval efficiency with respect to the length of a query trajectory. It is shown that retrieval time is increased as the length of the query trajectory is increased. When the length of a query trajectory is 50, the retrieval time of our algorithm is 230 seconds while that of Tiakas’s algorithm is about 295 seconds. As a result, it is shown that our algorithm achieves better retrieval efficiency than Tiakas’s one.

5. Conclusions and Future Work

To calculate the similarity of moving objects’ trajectories in road networks, we defined a spatio-temporal similarity measure. Based on it, we proposed a new algorithm to
search for similar trajectories for a given query trajectory. Performance analysis showed that our algorithm outperformed Tiakas’s one [5], in terms of both retrieval accuracy and retrieval time. This is because our algorithm can minimize the negative effect of skewed data by setting the grid cell size with an average length of edges in the spatial network and by providing an efficient pruning policy based on grid cell distances.

Main applications of our work are a design of new public transportation routes and an intelligent ridesharing. First, to find effective public transportation routes (e.g. buses) for a new town, our algorithm can be used to analyze the patterns of moving objects’ trajectories. Secondly, an intelligent rideshare application can be one possible solution to the ever increasing congestion problems of urban transportation networks [6]. Our algorithm can be used to find sharable routes (i.e., trajectories) for a set of commuters and suggests rideshare possibilities to them. Though our algorithm can minimize the negative effect of skewed data, it may be degraded on retrieval performance in case of highly skewed data since it uses uniform grids. As future work, it is needed to study on similar trajectory search algorithms using non-uniform grids, like hierarchical ones.

Acknowledgments

This work is financially supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. R01-2008-000-11002-0). And this work is also supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energie (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency.

References

[1] Y. Sakurai, M. Yoshikawa, and C. Faloutsos, “FTW: Fast similarity search under the time warping distance,” Proc. PODS, pp.326–337, 2005.
[2] D. Zeinalipour-Yazti, S. Lin, and D. Gunopulos, “Distributed spatio-temporal similarity search,” Proc. 15th ACM CIKM, pp.14–23, 2006.
[3] J.R. Hwang, H.Y. Kang, and K.J. Li, “Spatio-temporal similarity analysis between trajectories on road networks,” Proc. ER Workshops, pp.280–289, 2005.
[4] J.W. Chang, R. Bista, J.H. Kim, and Y.C. Kim, “A new trajectory search algorithm based on spatio-temporal similarity on spatial network,” Proc. IEEE CIT, pp.110–115, 2007.
[5] E. Tiakas, A.N. Papadopoulos, A. Nanopoulos, Y. Manolopoulos, D. Stojanovic, and S. Djordjevic-Kajan, “Trajectory similarity search in spatial networks,” Proc. IDEAS, pp.185–192, 2006.
[6] G. Gidofalvi and T.B. Pedersen, “Mining long, sharable patterns in trajectories of moving objects,” Proc. STDBM, pp.49–58, 2006.