Development of Route-based Data Generation Algorithm and Data Model for Moving Objects

Yong-Ki Kim, Cheol-Joo Chae

Abstract*: With the widespread of mobile phones and mobile telecommunication devices, recently, diverse location-based services are being provided. For location-based application services, an algorithm which stores a moving object's mobile data and generates indexing structure and experimental data to provide coordinate-, route- and trajectory-based application services is required. To execute conventional route- and trajectory-based queries, this study proposes a route-based data generation algorithm which converts coordinate-based data to route-based ones. It also suggests moving object data model for their execution. The trajectory data including the proposed a data model and routes of the converted moving objects are able to support diverse coordinate-, route- and trajectory-based query processing algorithms. In addition, it is able to provide diverse services to users quickly, using the data model. This study enables the provision of services which should be processed according to a moving object's flow of time in a quick and efficient fashion.

***Index Terms*: Location based service, Data generation, Moving objects, Index structure, Road network**

I. INTRODUCTION

With the widespread of mobile phones and mobile telecommunication devices, various forms of prototypes have been developed. Now, mobile phones are the necessity of life. With a rising interest in a location-based service (LBS), they have been used in diverse applications. An LBS is defined as a provision of user-wanted application services in real-time by combining a moving object's location information with the user's other information [1-4]. In the Republic of Korea, the number of smartphones reaches 48,659,934 as of 2017, showing 94% smartphone ownership rate as shown in table 1. In fact, the number of smartphones has increased by 1% annually from 2011 to 2017 [5]. With the development of diverse and rapid telecommunication technologies, the number of the mobile devices has risen quickly.

In general, to provide LBS, the efficient management of location information is required. Since location information is expressed in dots, it can be used or processed differently depending on a service environment. For example, let's say a general store offers a special discount coupon to nearby customers. Then, a user may find the customers within a close distance by pinning the store's location as a query point. In this case, a moving object with close ideal distance (the Euclidean distance) is being sought. In contrast, if a moving object is limited to move on the given spatial network only, it is required to explore in a way which definitely differs from ideal space-assumed explorations. For example, it's quite often that the nearest point in the ideal space 'Euclidean space' is not the nearest on a spatial network. In other words, provided that a user goes to the point of interest (POI) using actual road, network distance, not the distance in the Euclidean space, should be used. Therefore, results should be obtained by calculating network distance. The figure 1 illustrates an example of this case. The Euclidean distance from a query point is 60m in case of close POI1. If the distance to POI1 is calculated using actual network distance, however, the intersection (I1 and I2) should be passed through. Therefore, the distance from the user to POI1 becomes 120m. Hence, the Euclidean distance of POI2 on the actual network is 100m, farther than POI1. In contrast, the network distance is 110m, making it quicker. As described above, it is needed to express or process LBS differently according to an application service type or environment. A moving object such as a passenger car and bus moves along the limited road on a road network. Therefore, a conventional way of handling moving objects would not be efficient. Furthermore, since it moves over time, there should be indexing structure which handles large data efficiently for the appropriate management of moving objects.

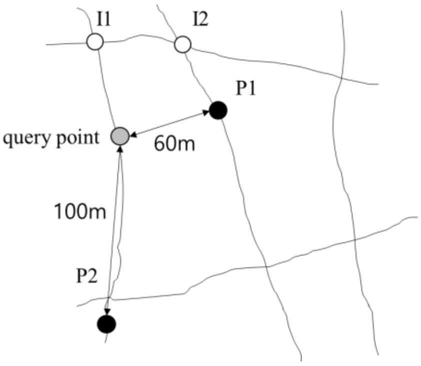


Fig. 1 An example of location based service

In terms of studies on LBS applications in a spatial network environment to provide diverse LBS-based application services, there are coordinate-, route and trajectory-based services. Many of these studies have been conducted and used in developing diverse application services. To perform such studies targeted to support LBS-based application services, there should be data needed for data processing on a spatial network. For this, multiple data exist in various spatial network environments. The data on such spatial network environments were generated based on a moving object's location. Even though such location-based

Revised Manuscript Received on May 22, 2019.

Yong-Ki Kim, Dept. of IT Convergence System Engineering, Vision College of Jeonju, Jeonju, Republic of Korea.

Cheol-Joo Chae, Dept. of General Education, Korea National College of Agriculture and Fisheries, Jeonju, Republic of Korea.

Table I. Korea's Smartphone penetration rate

		2013	2014	2015	2016	2017
Number	# of mobile devices	54,680,840	57,290,356	58,935,081	61,295,538	63,658,688
	# of cellular phone	51,645,203	52,844,612	53,660,658	54,899,789	56,047,339
	# of smart phone	37,516,572	40,698,151	43,667,799	46,418,474	48,659,934
	Population	51,141,463	51,327,916	51,529,338	51,696,216	51,778,544
Penetration Rate	Smart phone/mobile devices	68.6%	71.0%	74.1%	75.7%	76.4%
	Smart phone/mobile devices	72.6%	77.0%	81.4%	84.6%	86.8%
	Smart phone/mobile devices	73.4%	79.3%	84.7%	89.8%	94.0%
Growth Rate		-	8.5%	7.3%	6.3%	4.8%

data are able to provide diverse LBS, they are inappropriate in processing a moving object's movements. Because the data with a moving object's location only are generated, data movement over time cannot be expressed. To support route- or trajectory-based LBS applications, therefore, they should be converted into the data which can be processed with the passage of time, instead of being used as they were. Since LBS has to be processed quickly and provided in real-time, it would be possible to provide quick services if data which can change over time exist. Even so, the data for processing a moving object's trajectories aren't available. Therefore, this study proposes an data model needed to process trajectories on a road network and suggests trajectory data generation techniques.

This paper is consists of four parts: In chapter 2, current studies written for processing LBS in a spatial network environment are stated. In chapter 3, the proposed a data model and route based data generation algorithm are explained. In chapter 4, conclusions and future work are given.

II. RELATED WORK

The location-based application services using a moving object are classified into three types based on continuous characteristics in a moving object's time and space: i) coordinate-based query considering a moving object's static characteristics in time and space, ii) route-based query in consideration of continuous time and static space characteristics, iii) trajectory-based query considering continuous characteristics in time and space. Query types can be classified according to a moving object's continuous characteristics in time and space as follows:

First, a coordinate-based query is a query exploring a moving object or POI according to coordinates within the given time. Typical examples are a range query which searches POI within the distance 'R' from a moving object's position and a kNN query that searches the nearest POI from a moving object's position. Second, a route-based query is divided into a continuous k-NN query which calculates the nearest point when an objectives passes through the given route from the start to the destination and an in-route nearest neighbors (IRRN) query which estimates the nearest point when an object goes off of the route the least. Lastly, in a trajectory-based query, there is a similar trajectory search query which searches a moving object heading in a similar pattern with the given moving object's trajectory. The above queries can be summarized as follows:

- Find five nearest gas stations from the current location.
- Find a restaurant that a vehicle with a plan to move along the given route between two spots is able to visit by going off of the route the least.
- Find all moving objects with a similar pattern with the trajectory of Bus 88 in Seoul for the past 3 years.

A. Coordinate-based Services

In terms of coordinate-based queries, there are plenty of queries in addition to a study on a k-nearest neighbor (kNN) query processing algorithm. In this study, a kNN query is reviewed. Specifically, studies on INE[4], VN3[6], island[7] and S-GRID[8] are examined.

First, [4] proposed Euclidean space extension and spatial network extension techniques. In the IER, 'k' number of nearest points is searched by exploring R-tree as Euclidean distance. After sorting the results through the calculation of network distance, they are stored as a candidate set. After calculating network distance by searching the next nearest POI with Euclidean distance, it is included in the search results if network distance is smaller than kth POI in a candidate set. Otherwise, the followings are searched. In case the Euclidean distance of the POI to be searched is larger than that of kth POI in a candidate set, the algorithm is closed. Lastly, the INE technique searches a query-included edge. Then, edges are added to a separate storage space, and a network is being extended until 'k' number of nearest points on the edge is being found. In the acquired k-nearest POIs, their distance refers to distance on a spatial network. In case the distance until the newly added edge is larger than kth nearest network distance, the algorithm is being closed. The INE algorithm finds the nearest point, using Dijkstra's Algorithm basically. Therefore, in case POI density is lower than map data, search time increases.

Second, [6] proposed a VN3 technique in which a Voronoi diagram from an Euclidean environment is applied to a spatial network. The VN3 creates a Voronoi polygon using the given POI data and stores a Voronoi diagram by calculating inter-node and POI-node distances. After setting a Voronoi polygon which is close from the query point 'Q', the result POI is obtained, using the network distance of POI in a candidate zone. In the Voronoi diagram, then, inter-node and POI-node distances are already calculated. Therefore, it is able to calculate the distance between the query point and POI quickly.

Third, [7] proposed an island technique in which the distance from each node on a spatial network to the POI is first calculated and then stored. Specifically, the distance to the node which exists within a certain range is calculated, and POI information is saved in each node, generating islands. After finding nodes by extending a network from the query point, the distance to the POI is calculated using the node's islands, and then results are being found. The island technique is able to regulate a storage amount by adjusting and calculating a range for island formation so that it offers more flexible structure than VN3 in terms of POI and network data update. Both VN3 and island techniques are studies on a POI-based pre-computation technique which calculates and saves the distance from the POI to a network node. In the event of changes in POI's location information, inter-node and node-POI distances should be recalculated, showing update vulnerability. In VN3, search performances increase as the POI becomes less dense. Because network information needed for distance calculation increases, however, update performances rather drop. In the island technique as well, as the range 'R' increases, search performances could improve. Due to changes in the POI's location information, however, there will be a lot of nodes which require distance recalculation. Therefore, it is very disadvantageous for update functions. Furthermore, the above methods are not applicable to a moving object in which location information keeps changing.

Lastly, in [8], a spatial network is divided into 2D fixed grid cells, and network information is saved by cell. In case a node constituting network edges exists in different cells, a border point is inserted into the edge, saving it in two cells. Each grid cell is comprised of 3 components (vertex-edge, vertex-border, cell-border) and used in computing network distance. Even though S-GRID is able to find the POI fast based on network extension, it does not suggest an indexing structure for the POI and moving objects.

B. Route-based Services

In terms of a route-based query designed for spatial network database, the following studies are available: a study on a continuous kNN query processing algorithm in which a moving object moves along the given route and searches 'k' near POIs; an IRNN study designed to search the nearest POI by going off of the route the least.

First, [6] proposed a continuous kNN query processing algorithm for spatial network database. [6] suggested two algorithms: intersection examination (IE) designed for continuous kNN query processing using a network Voronoi diagram (NVD), upper bound algorithm (UBA). First, the IE searches 'k' number of POIs by performing a kNN query processing algorithm in all nodes on the moving object's route. In addition, it divides the POIs from each node into the POIs with increasing distance and those with decreasing distance and computes a split point, using the minimum distance between the continuous increasing POI and decreasing POI. Second, an UBA algorithm used the proposed threshold 'δ' value to reduce overhead that an IE algorithm should process a kNN query processing algorithm on the route node. First, 'k+1' number of POIs is searched in the start node, and they are divided into two different types: POIs with increasing distance, POIs with decreasing distance. Just like the IE algorithm, 'δ' value is calculated aft4er computing the minimum distance between the continuous increasing POIs and decreasing POIs. After proving the theory that the POI distance sequence does not change within the range of 'δ', search time of the kNN query processing algorithm was improved.

Second, [9] suggested an IRNN query and proposed 4 different algorithms. First, a simple graph-based method finds the nearest point after calculating the shortest path to the destination via each node's nearest point after finding each node's nearest point on the route. If the number of nodes increases, however, search costs rise. Second, a recursive spatial range query-based method finds the nearest point on a current route through the shortest path algorithm and then the nearest point on the next node through the range query, using the distance to the already found nearest point. In case the distance to the nearest point is lower, adjust the range to this distance and execute the range query in the next node. This method is lower than the first one in terms of search costs but generates the costs to execute a range query in each node. Third, a spatial distance join-based method was proposed to reduce the costs to execute a range query in each node. This method calculates the nearest point after estimating a query range through the Euclidean distance join of each node and POIs on the route, using two trees on the node and POI. Lastly, a pre-computed zone-based technique pre-calculates the network distance from a node to the POI after setting a service zone in each POI. If a moving object's route is given, therefore, this method is able to find the nearest point quickly by searching a service zone including the route.

C. Trajectory-based Services

A trajectory-based query designed for spatial network database is a query which targets a trajectory on the continuous past location saved while a moving object moves. As a study on the similar trajectory search designed to analyze a moving object's moving patterns, there are the studies written by [10] and [11]. First, [10] proposed a similar trajectory search algorithm which uses actual network distance and time distance between the POIs that the trajectories pass through in consideration of time and spatial characteristics for similar trajectory search in a spatial network. To reduce unnecessary trajectory search, in addition, 3 different methods were proposed: i) computation of the sum of time distance between common POIs after filtering the trajectories which do not include a query trajectory's POIs, ii) computation of the sum of spatial distance between the times of interest (TOIs) after searching trajectories with a time range such as a query trajectory was computed, iii) computation of the sum of the time-space distance, using the velocity of two trajectories after time and space-based filtering. Since query trajectories and those with common POIs are only targeted, however, trajectories may not be searched unless all POIs are included despite very similar trajectories in terms of spatial distance. Second, [11] proposed a similarity measurement method considering time-space characteristics and suggested an M-tree-based similar trajectory search algorithm for effective similar trajectory search. He defined spatial similarity as the percentage of maximum and minimum moving

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

distances between the nodes constituting a trajectory. He also explained time similarity as the ratio of maximum and minimum moving distances in the moving speed of two trajectories. Furthermore, the author used the weighted values of each similarity and defined them as 'time-space similarity'. In such maximum-minimum moving distance ratio-based similarity, however, most trajectories reveal same spatial similarity on a common two-way road network. In addition, moving object's velocity-based time similarity tends to find the trajectory of a moving object which moved at similar velocity with the trajectory that is more similar to that of a moving object which moved in a similar time zone.

III. DATA MODEL FOR MOVING OBTECTS AND TRAJECTORY DATA GENERATION

As stated in the above chapter, LBS is classified into coordinate-, route- and trajectory-based services. Among them, route- and trajectory-based services have an assumption that a moving object moves in the space, such route or trajectory along which a moving object moves should be handled. In this chapter, therefore, a data model for professing moving objects and a route data generation algorithm are proposed.

A. Data Model

A road network is comprised of nodes and edges. A node is expressed in x-y coordinates while an edge is characterized by two nodes. Therefore, a road network can be modeled with nodes, edges and connection information. Specifically, it can be stated as follows: Network = (V, E), V=(x, y), E=(V_i, V_j). Meanwhile, a moving object's location information on the road network consists of coordinates and time. Since the object moves according to the time axis, a route from '1' to 'N' can be '{(x_1, y_1, t_1), (x_2, y_2, t_2), ... , (x_{i-1}, y_{i-1}, t_{i-1}), (x_i, y_i, t_i)}', using (1) below:

$$Data(Object) = \{(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_{i-1}, y_{i-1}, t_{i-1}), (x_i, y_i, t_i)\} \quad (1)$$

Then, a moving object's location at the time of 'i' is '(x_i, y_i, t_i)'.

Such moving object's trajectory data were expressed, using coordinates and time in the Euclidean environment. However, since trajectory data on a motor (e.g., bus, etc.) road network pass by edges, they should be mapped to make sure that coordinates are positioned on such edges. Then, each moving object's route data are expressed as stated in (2) below. Here, route-based data are named, 'trajectory'.

$$Trajectory = \{E_1, E_2, E_3, \dots, E_{k-2}, E_{k-1}, E_k\} \quad (2)$$

Here, 'k' is smaller or equal to the time 'i' before mapping because trajectories at more than one point in time can be located on the same edge.

The starting point (x_1, y_1, t_1) of coordinate-based trajectories prior to their conversion to edge-based ones is matched with a dot on the first edge (E_1). In addition, the destination (x_i, y_i, t_i) is positioned on a dot on the last edge (E_k). The coordinate-based trajectory information between the start and destination points passes through a trajectory between edge E_k at each edge E_1.

B. Trajectory Data Generation Algorithm

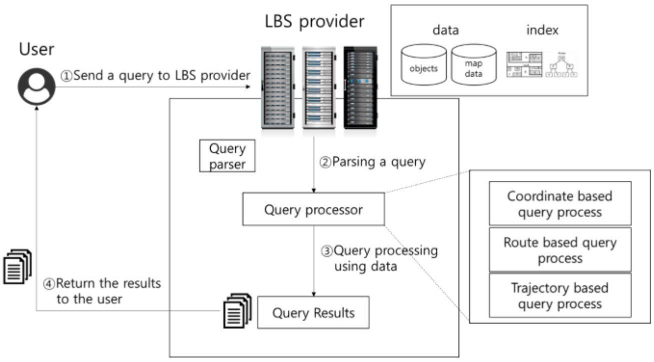

User
①Send a query to LBS provider
LBS provider
data
index
objects
map data
Query parser
②Parsing a query
Query processor
③Query processing using data
④Return the results to the user
Query Results
Coordinate based query process
Route based query process
Trajectory based query process

Fig. 2 The entire flow of Location based service

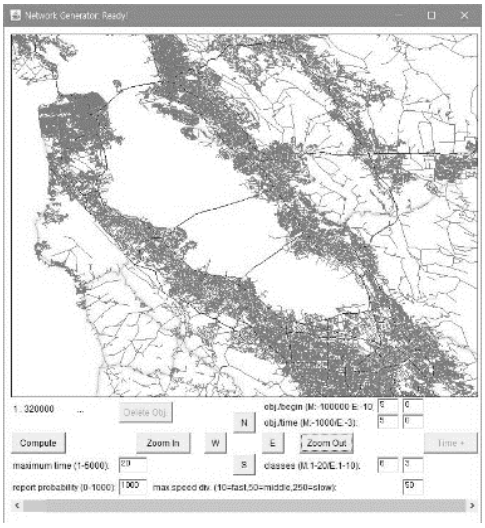
In actual location-based application services, a query on user-wanted services is received and executed by the LBS provider, and the results are returned back to a user. The LBS process is illustrated in figure 2. Location-based application services are provided as follows. First, a user gives a query to the LBS provider. Then, the LBS provider analyzes the query type and selects an algorithm. To provide results to the user, the algorithm is executed, using the query, data in the database and indexing structure. After that, the LBS provider provides the results to the user.

The route data which are available in actual applications should be converted into edge-based route data, not those comprised of coordinate-based moving object's points. Therefore, the LBS provider should have route-based data. The overall storage structure designed to provide coordinate-, route- and trajectory-based services is as follows: To save network information, nodes and edges are indexed, using B+Tree and R*Tree. Each R*Tree/B+Tree terminal node has a node/edge storage address and basic metadata. To express network connection, edge information includes adjacency information connected to neighboring nodes. Furthermore, trajectory data are saved in a continuous data format according to the moving flow and connected with edge R*Tree for connection with a storage network. Therefore, the proposed storage structure is able to index coordinate-based network information, and each trajectory is positioned at the terminal node of edge R*Tree, and edge information indicates trajectory data.

The storage structure-based trajectory data generation algorithm is illustrated in the figure 3. To generate route based trajectory data, a moving object's points are mapped with edges in advance. Then, the mapped edges are connected. A set of the connected edges can be organized differently from the number of a moving object's coordinates. In case a moving object is fast, the total number of edges can increase. If the object is slow, on the contrary, the total edge number can be lower than the number of coordinates. Then, road network connection information in addition to the speed and orientation of a moving object's trajectory data should be considered. Otherwise, a moving object's inter-point connection could be incorrectly executed. This study has revealed that it is able to generate edge-based trajectory data with the proposed generation algorithm after extracting data through the conventional moving object generation platform and converting route-based trajectory data.

```
Input : coordinate based trajectory
Output : edge based trajectory
For each point Time t from trajectory (x, y, t)
  {
     Find nearest edge from point (x_i, y_i, t_i)
     Selected Edge ← E_i
  }
For Selected Edge and Its next Edge
  {
     R ← Find route from to in accordance with cost of route
     Result ← R
  }
Return Result;
```

Fig. 3 Route-based Data Generation Algorithm



1 : 320000 ...
obj./begin (M:-100000 E:-10: 5 0
obj./time (M:-1000/E:-3): 5 0
maximum time (1-5000): 20
classes (M:1-20/E:1-10): 6 3
report probability (0-1000): 1000
max speed div. (10=fast,50=middle,250=slow): 50

Fig. 4 Data generation by moving object generator

IV. CONCLUSION

Even though there are a lot of actual data on a moving object for data processing on a road network, it's been impossible to find information needed for processing trajectory data. Since a moving object's coordinates are adopted in using route/trajectory-based services, quick response has been unexpected. Therefore, this study proposed storage and indexing structure for trajectory data processing in a road network environment and suggested an algorithm which generates edge-based moving object trajectory data with coordinate-based ones. This study also confirmed that edge-based trajectory data can be generated through moving object data and proposal algorithm on the San Francisco road map data. There should be further studies on storage structure and query processing algorithm in distributed environments to provide diverse route/trajectory-based services more quickly. Furthermore, the utility of the proposed system needs to be tested by performing studies on the application of trajectory data, using the algorithm proposed in this study.

ACKNOWLEDGMENT

This work has supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No.2018R1C1B6005863).

REFERENCES

1. S. Shekhar, S. Chawla, S. Ravada, A. Fetterer, X. Liu, C. T. Lu, "Spatial databases-accomplishments and research needs," IEEE transactions on knowledge and data engineering, Vol. 11, No. 1, 1999, pp. 45-55.
2. L. Speičvcys, C. S. Jensen, A. Kligys, "Computational data modeling for network-constrained moving objects," In Proceedings of the 11th ACM international symposium on Advances in geographic information systems, 2003, pp. 118-125.
3. Y. Oh, H. Park, Y. Min, "Understanding location-based service application connectedness: Model development and cross-validation," Computers in Human Behavior, Vol. 94, 2019, pp. 82-91.
4. D. Papadias, J. Zhang, N. Mamoulis, Y. Tao, "Query processing in spatial network databases," In Proceedings of the 29th international conference on Very large data bases, Vol. 29, 2003, pp. 802-813.
5. Korea's Smartphone penetration rate. Available: https://www.netmanias.com/ko/mobile-statistics/mobile-connections-in-korea/1335/
6. M. R. Kolahdouzan, C. Shahabi, "Continuous K Nearest Neighbor Queries in Spatial Network Databases," In Proc. of STDBM, 2004, pp. 33-40.
7. X. Huang, C. S. Jensen, S. Saltenis, "The Islands Approach to Nearest Neighbor Querying in Spatial Networks," In Proc. of SSTD, 2005, pp.73-90.
8. X. Huang, C. S. Jensen, H. Lu, S. Saltenis, "S-GRID: A Versatile Approach to Efficient Query Processing in Spatial Networks," In Proc. of SSTD, 2007, pp. 93-111.
9. S. Shekhar, J. Yoo, "Processing In-Route Nearest Neighbor Queries: A Comparison of Alternative Approaches," In Proc. of ACM GIS, 2003, pp. 9-16.
10. J. Hwang, H. Kang, K. Li, "Spatio-Temporal Similarity Analysis Between Trajectories on Road Networks", International Conference on Conceptual Modeling, 2005, pp. 280-289.
11. E. Tiakas, A. N. Papadopoulos, A. Nanopoulos, Y. Manolopoulos, D. Stojanovic, S. Djordjevic-Kajan, "Trajectory Similarity Search in Spatial Networks," In Proc. of IDEAS, 2006, pp. 185-192.

AUTHORS PROFILE

Yong-Ki Kim is a professor in the Department of IT Convergence Engineering, Vision College of Jeonju, South Korea from 2016. He received the B.S., M.S., and Ph. D degrees in Chonbuk National University in 2002, 2005, and 2011, respectively. During 2012–2015, he was a senior researcher in Korea Institute of Science and Technology Information. His research interests include spatial database, data mining, and data privacy and artificial intelligence

Cheol-Joo Chae received his Ph.D. degree in computer engineering from Hannam University, Daejeon, Korea, in 2009. From 2009 to 2013 he was a senior member of the research staff in the ETRI, Daejeon, Korea, where he did research on network communication system. From 2013 to 2016 he was a senior member of the research staff in the KISTI, Daejeon, Korea, where he did research on information system. Since then, he has been a Professor with Korea National College of Agriculture and Fisheries.