International Journal of Control and Automation
Vol.6, No.5 (2013), pp.1-8
http://dx.doi.org/10.14257/ijca.2013.6.5.01

Design of a Shape File Index for Fast Map Matching in Electric Vehicle Services

Junghoon Lee and Gyung-Leen Park

Jeju National University, Republic of Korea
{jhlee, glark}@jejunu.ac.kr

Abstract

Mainly targeting at a small-city level road network, this paper builds an in-memory index for fast map matching, one of the most fundamental building blocks for vehicle information services. For the ESRI shape file sequentially storing road objects such as intersections and links, our indexing scheme obtains the offset of each link and creates the corresponding record consisting of link id, two end points, and the offset. Along with node entries, each of which includes all emanating links for a node, this index makes it possible to traverse the shape file just like Dijkstra's shortest path algorithm, while all index traversal steps are done within memory. The previous location of a vehicle gives the start point of a map match process, significantly reducing the number of links to check if they include a specific point. This scheme particularly helps us to build an analyzer application for a series of spatio-temporal streams such as battery discharge dynamics.

Keywords: s*mart transportation, electric vehicle, map matching, graph index, road network*

1. Introduction

[1]According to the smart grid vision, electric vehicles, or EVs in short, are one of the most important elements in the future transport system [1]. If current gasoline-powered vehicles are replaced by EVs, greenhouse gas emissions will be much reduced. The deployment of EVs will avoid burning fossil fuels and basically take advantage of cheap nuclear energy. Moreover, many renewable energy sources such as wind, sunlight, and the like, can be used to charge EV batteries. Besides those environmental benefits, EVs can be more easily combined with information and communication technologies due to their digital nature in engine control and vehicle management. Hence, a variety of intelligent computer applications can be developed and embedded in in-vehicle computers for efficient and convenient driving [2]. In addition, current vehicle networks are sure to enrich vehicle information services with inter-vehicle and vehicle-to-infrastructure cooperation [3].

Due to their eco-friendliness, many cities, especially having many natural attractions, are much interested in EVs and trying to accelerate their fast deployment. However, there are some obstacles for this effort. Most of all, stemmed from the capacity limitation in EV batteries, the driving distance is practically at most 100 *km* and it takes 30 ~ 40 minutes to charge an EV even with a fast charger [4]. With slow chargers, it takes about 6 ~ 7 hours. Fast chargers may shorten the battery life and can fill

---

[1] Prof. Gyung-Leen Park is the corresponding author.
This research was supported by the 2013 scientific promotion program funded by Jeju National University

ISSN: 2005-4297 IJCA

electricity by just up to 80 % of the full battery capacity. The battery discharge is affected by so many factors such as road shapes, altitude changes, drivers' behaviors, air conditioner operations, and the like. Sometimes, drivers worry that they cannot reach their destinations. Hence, it is necessary to estimate current battery remaining, or interchangeably SoC (State of Charge), and the reachable distance on the way to the destination [5]. If the current SoC is not enough, the driver must have his or her EV charged.

In the mean time, many vehicle services can be developed by a computer application running on in-vehicle computers such as telematics devices, just as many other smart grid components take advantage of computational intelligence and communication networks [6]. Moreover, GPS receivers make the location information available to the application. With this, many location-dependent applications can be developed. The most essential application is definitely the vehicle navigation which finds the route to a destination from the current position. Basically, by the current coordinate of an EV, it is possible to know the road segment the EV is currently moving on. The process of binding the coordinate to a road segment is map matching. For a given coordinate, mainly specified by longitude and latitude in the world-wide coordinate system, the map matching procedure searches the set of road segments to find the match. Here, each road segment is represented by a series of coordinates accounting for the shape of roads.

For a fast moving EV, map matching speed is important to give more room for the execution of other complex and time-consuming applications such as SoC analyzers and inference engines [7]. Basically, the efficiency of map matching processes is subject to how the road segments are organized and thus how many comparisons are needed to find the match [8]. A link having a complex shape consists of hundreds of line segments and the same number of comparisons is needed to check whether an observation point belongs to the link. So, if we have to investigate many links, the computation time can be too much. For a moving object which creates a spatio-temporal trajectory, the current location is not far away from the previous location. Hence, it is reasonable to start the map matching procedure from the road segment to which the past coordinate is matched. In addition, the number of links to investigate can be further cut down by narrowing the search scope only to the links along the road network on which the EV can move.

Generally, spatial objects are specified by their coordinates in the 2-dimension space, and can be arranged by their locations and connectivity from a specific reference point. The spatial index can much improve the search speed, taking advantage of this order. Existing index schemes are developed for general-purpose location-based applications, not mainly concerning the road network topology and handling both static and dynamic objects. If we just focus on the trajectory analysis, the graph style index can prevent the search scope from deviating from the actual moving road segment of an EV. In this regard, this paper designs a complementary index for fast map matching, aiming at providing an efficient EV service framework capable of hosting many sophisticated applications. As an extended version of our previous work [9], this paper further includes related work and more details on design details.

The rest of the paper is organized as follows: Section 2 reviews related work, focusing on specialized map matching and indexing. After our system configuration is specified in Section3, Section 4 proposes a graph-based index scheme. Finally, Section 5 concludes this paper along with a brief introduction of future work.

2. Related Work

Online map matching is very important for real-time location-based applications. According to [10], most online map matching algorithms localize the search area based on fixed sliding window and fixed depth recursive look-ahead strategies. Their work proposes an optimal localization strategy called the variable sliding window, which divides the input trajectory points into subproblems and finally finds a global solution by means of a sequential estimation of each local matching result. In this algorithm, candidate paths are searched first according to their likelihood calculated by a hidden Markov machine model. Each hidden state in the Markov chain is associated with an emission probability. It quantifies the likelihood that the point is on the specific link and the link closer to the point has higher emission probability. The scoring function selects the link match by the momentum change function and the distance discrepancy function. After all, this algorithm can achieve both accuracy preservation and response time reduction.

As an interesting example of map matching procedure for a series of locations having low sampling rate, [11] proposes a compromised global search scheme. Its main challenge is to overcome the difficulty in applying incremental map matching techniques which take advantage of the correlation with the previous point and the last matched link. If the sampling period is large, the distance between two consecutive observation points can increase, making it difficult to narrow the local search area. Even worse, the exact area containing the observation point can be excluded from the search area. The authors propose a kind of a global matching scheme called ST-matching. It considers not only the spatial features on geometry and topology but also temporal constraints imposed on the trajectory. This spatio-temporal analysis creates a candidate graph consisting of nodes representing the pair of candidate link and an observation point as well as links representing the dependency between each point-to-link match. In this graph, the well-known shortest path algorithm can find the best matching sequence.

As for an indexing scheme for road networks, [12] proposes a two-level index structure consisting of route overlay and association directory. Basically, the route overlay manages the physical network architecture and interestingly some shortcuts, which can be given by map providers. Various shortcuts can coexist for different requirements from different applications. The association directory associates objects and object abstracts to the actual road network. Here, content providers specify the mapping information from respective objects to nodes, links, and specific area. Moreover, the directory facilitates flexible object types and network updates. Area hierarchy reduces index overhead especially for range queries and nearest neighbor queries. This framework allows efficient processing of location dependent spatial queries on the target road network by means of separating the network from spatial objects, exploiting search space pruning techniques, and permitting different distance metrics.

3. System Configuration

Figure 1 illustrates our system configuration. First, the EV information acquisition module reads the current status of the electric control unit (ECU) and the battery management unit (BMS) of an EV. In EVs, the ECU continuously monitors the status of electric system inside the vehicle and decides an appropriate control action. Besides the classic control objects such as brake control modules, the ECU is required to deal with battery devices in cooperation with the BMS. The BMS is the most important part in EVs, as the battery is the engine of EVs [13]. The BMS prevents every cell in the battery from being overcharged or undercharged. In addition, it balances the battery cells during charge to extend the battery life and driving range. The ECU sends the collected information to in-vehicle display units such as a dash board or a telematics device. Moreover, on-board diagnosis (OBD) devices scan this status information and send to another analysis module via CAN (Control Area Network).

Our main concern lies in SoC, but it is not open to general application developers. EV manufacturers have no obligation to open it, but SoC is displayed in the dash board. By a negotiation with them, SoC may be partially available. At least, our computer application can read current battery remaining by taking pictures of the dash board and recognizing digits. In addition, the GPS receiver keeps providing the current coordinate to the application. For each second, the GPS receiver reports RMC (Recommended Minimum Data) and GGA (Global Positioning System Fix Data) records according to the NMEA (National Marine Electronics Association) interface. We can obtain the stream of SoC records associated with a spatial stamp consisting of longitude, latitude, and altitude. At this stage, our research team gets the SoC data from an EV business and telecommunication company. However, it must be pointed out that BMS-reported battery remaining may have an error of up to 15 %. As the accuracy improvement is out of scope of this paper, we just assume that the battery information is always correct.

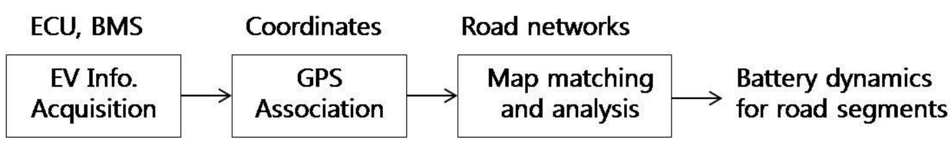


Figure 1. System configuration

A road network is comprised of intersections and links connecting them. EVs can move only along the links, each of which has its own shape. While just a single point is enough to represent an intersection, a link needs a series of points to trace its shape. Our system is targeting at Jeju city, Republic of Korea [14]. As a middle-level city, currently it has about 18,000 intersections and 27,000 links. Jeju area is surrounded by coastal line of about 200 *km*. This city is ambitiously accelerating the large deployment of EVs as a part of smart grid model city enterprise. In addition, the road network has significant terrain diversity and it is possible to build a battery discharge model for each road type. That is, from the terrain aspects, this city embraces mountain roads, coastal road, and urban roads. Here, the discharge model can benefit from many artificial intelligence techniques [15].

4. Indexing Scheme

A road network is stored in ESRI shape files. Sometimes, the network is converted to a spatial database records for better retrieval and management. However, for the small or middle city-level road networks, it is possible to build a complementary in-memory index for efficient map matching, relieving a burden of maintaining a complex and expensive database system. The size of its link shape file depends on the pattern of road layout. Figure 2 shows the actual road shape of Jeju city. In downtown area, the road segment is generally straight, possibly specified just by 2 points. On the contrary, in the mountain and seaside area, the roads are winding and needs hundreds of points. A single link is observed to have 600 marking points. In shape file, each link record is stored sequentially. The shape file can have a general-purpose index, but it can support an efficient move-forward and backward. We implement a Window-based shape file viewer which reads and plots road shapes sequentially. It can be zoomed-in, zoomed-out, and panned as shown in Figure 2.

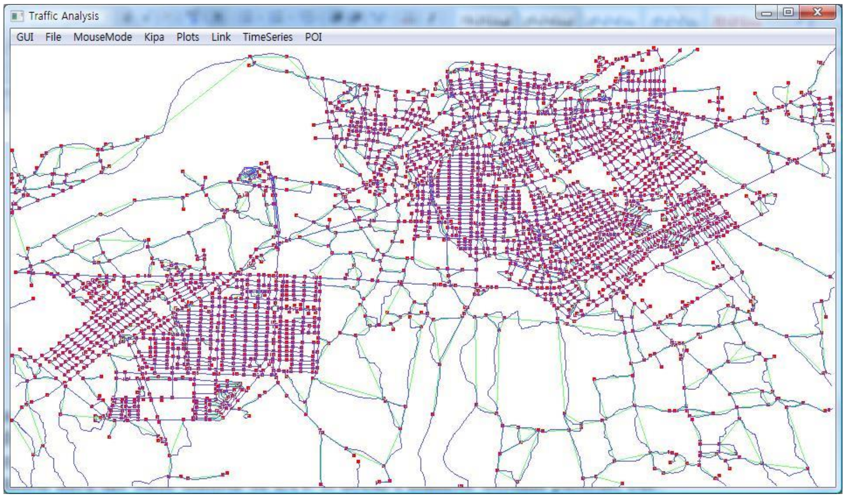


Figure 2. Road layout

Our design exploits an in-memory graph which includes intersections and links as shown in Figure 3. Each intersection, or node, has the set of links emanating from it. Actually, the size of a node record is not so large, so the entire node records can be loaded into memory. On the contrary, each link has two end nodes and offset in the shape file. The size of a link record is different by the number of marking points in the link. That is, each link record is not stored in fixed position, so the shape fine permits only a sequential access. According to the shape file format, a link record begins with the coordinates of its bounding box, namely, x-max, x-min, y-max, and y-min. Here, x and y means longitude and latitude for simplicity. If the match point is included in this boundary, it is necessary to check if the point is on a line segment of the link one by one. Otherwise, we can skip this link. For a line segment, the distance from the observation point is calculated and if it is less than the small bound, the link match is found.

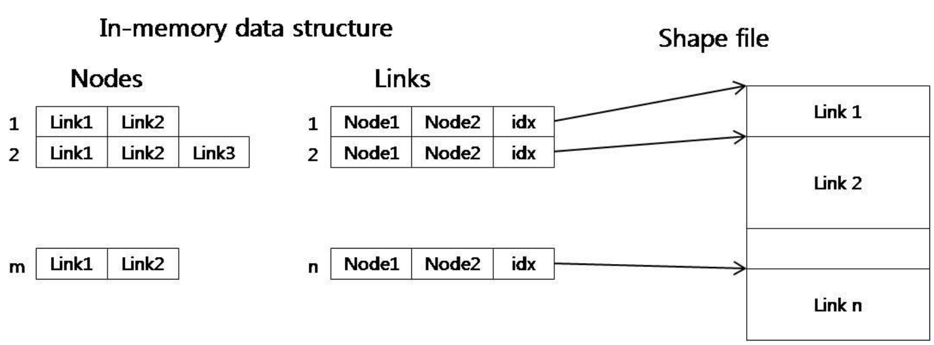
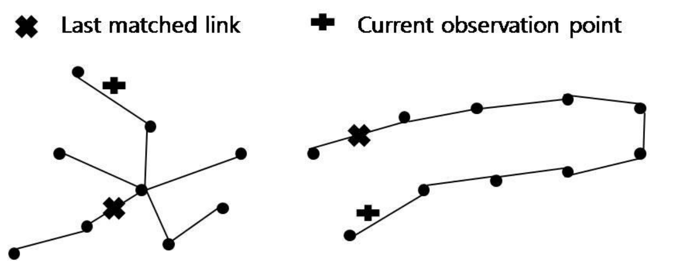


Figure 3. Index architecture

The graph index allows the map matching procedure to traverse the shape file not sequentially but according to the connection architecture of the road network, taking advantage of existing graph search algorithms such as Dijkstra's and bread-first schemes. That is, if the procedure investigates a link but does not find a match, it will proceed to the adjacent links. The graph index contains the two end points of a link, so the connected links can be easily found following nodes and their links. As the in-memory index includes the offset in the shape file, the map matching procedure can randomly access the file using the *SetFilePointer* system call on Windows operation systems. Practically, adjacent links are stored in the shape file not far away from each other. As operating systems read records from the disk system by the block, it is highly likely that additional disk access is not necessary. Finally, the real-time SoC information can be associated with the road network even in the fast moving vehicles.

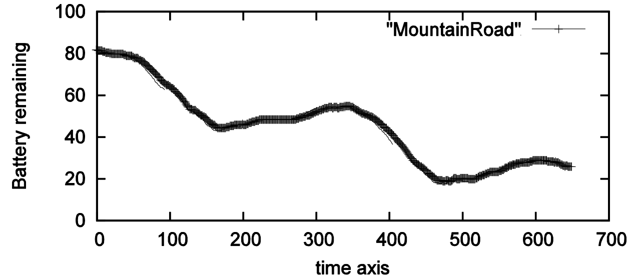
Figure 4 shows pros and cons of our graph index. Here, the road shape details are omitted and just the link connections are plotted. An x mark represents the link matched to the previous observation point. From this link, the map matching procedure extends the search tree until it finds the match. The two end-points of the x-marked link will be the starting points of the search if the heading information is not available. Two level extensions from the current observation point marked by + can find the matched link in Figure 4(a). On the contrary, in Figure 4(b), the two observation points are quite close in Euclidean distance, but the network distance is large, making us follow the graph index several times. In this case, area-based filtering looks better, but a single spatial query needs a quite much time overhead.



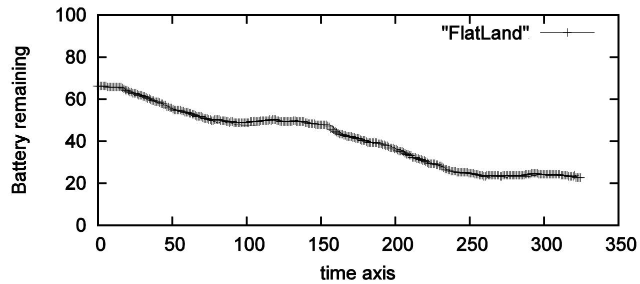
(a) Normal case (b) inefficiency in graph index

Figure 4. Index-based matching

Figure 5 plots the change in the remaining amount according to the driving distance. Here, the time interval between two observation points ranges from 10 to 30 seconds. Some measurements are lost in delivering to the analysis module and some points fail to match a link. The y-axis denotes battery remaining and it corresponds to the reachable distance with current battery amount. Two graphs start from different initial values, and the difference from the initial amount is important for respective cases. First, Figure 5(a) shows the battery when an EV drives along the mountain road having steep up-slopes and down-slopes. Remaining battery increases without external charging, when the EV moves down, due to regenerative brake energy. On the contrary, battery remaining reduces almost uniformly when an EV drives on flat area as shown in Figure 5(b). How to quantify this change and map the road characteristics to the change will be net next problem [16].



(a) driving mountain area



(b) driving plain area

Figure 5. Battery remaining dynamics

5. Conclusions

Smart transportation is an important area of the smart grid system, pursuing energy efficiency in the transport system mainly with EVs and their charging infrastructure. For developing intelligent services, it is necessary to build an EV-specific data processing framework, such as a battery discharge model along the roads of different features. As an essential building block for such applications, this paper has designed a graph-style index scheme, targeting at online map matching. To speed up the search in the sequential shape file, this index consists of nodes and links along with inter-referencing pointers, while each link has an offset within the shape file. This index allows a fast traversal along the road network just like the shortest path algorithm. From the starting point, which is the last matched link, the index gradually extends the search tree until it finds a link match from the starting point. In the mean time, the offset field allows us to avoid unnecessary segment-by-segment on-the-line tests, significantly enhancing the map matching speed.

As future work, we are planning to develop a SoC dynamics model based on the collected SoC change for different types of routes such as urban, mountain, and coast roads. Currently, our research time is collecting the stream of SoC change along the representative roads in Jeju city area.

References

[1] E. Sortomme, M. Hindi, S. MacPherson and S. Venkata, “Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses”, IEEE Transactions on Smart Grid, **(2012)**, pp. 198-205.

[2] A. Ipakchi and F. Albuyeh, “Grid of the future”, IEEE Power & Energy Magazine, **(2009)**, pp. 52-62.

[3] C. Chen, K. Nagananda, G. Xiong, S. Kishore and L. Snyder, “A communication-based appliance scheduling scheme for consumer-premise energy management systems”, IEEE Transactions on Smart Grid, vol. 4, no. 1, **(2013)**, pp. 56-65.

[4] C. Botsford and A. Szczepanek, "Fast charging vs. slow charging: Pros and cons for the new age of electric vehicles", International Battery Hybrid Fuel Cell Electric Vehicle Symposium, **(2009)**.
[5] O. Trembly and L. Dessaing, "Experimental validation of a battery dynamic model for EV applications", World Electric Vehicle Journal, vol. 3, **(2009)**.
[6] S. Ramchurn, R. Vytelingum, A. Rogers and N. Jennings, "Putting the 'Smarts' into the smart grid: A grand challenge for artificial intelligence", Communications of The ACM, vol. 55, no. 4, **(2012)**, pp. 86-97.
[7] J. Lee, H. Kim and G. Park, "Integration of battery charging to tour schedule generation for an EV-based rent-a-car business", Lecture Notes on Computer Science, vol. 7332, **(2012)**, pp. 399-406.
[8] J. Lee, G. Park, S. Kim and C. Park, "Monitoring-based temporal prediction of power entities in smart grid cities", ACM Research in Applied Computation Symposium, **(2012)**, pp. 371-375.
[9] J. Lee, J. Kang, H. Park and H. Kim, "Design of an in-memory index for fast map matching in electric vehicle services", Proc. of Ubiquitous Computing and Multimedia Applications, **(2013)**.
[10] C. Goh, J, Dauwels, N. Mitrovic, M. Asif, A. Oran and P. Jaillet, "Online map-matching based on hidden Markov model for real-time traffic sensing application", 15-th International IEEE Conference on Intelligent Transport Systems, **(2012)**, pp. 776-781.
[11] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories", ACM Int'l Workshop on Advances in Geographic Information Systems, **(2009)**, pp. 352-361.
[12] K. Lee, W. Lee and B. Zheng, "Fast object search on road networks", ACM 12th International Conference on Extending Database Technology, **(2009)**, pp. 1018-1029.
[13] H. Wang, Y. Liu, H. Fu and G. Li, "Estimation of state of charge of batteries for electric vehicles", International Journal of Control and Automation, vol. 6, no. 2, **(2013)**, pp. 185-194
[14] J. Lee, G. Park, Y. Kim, E. Kim and I. Lee, "Wind speed modeling based on artificial neural networks for Jeju area", International Journal of Control and Automation, vol. 5, no. 2, **(2012)**, pp. 73-80.
[15] J. Rigol, C. Jarvis and N. Stuart, "Artificial neural networks as a tool for spatial interpolation", International Journal of Geographical Information Science, vol. 15, no. 4, **(2001)**, pp. 323-343.
[16] Z. Chem, S, Qiu, A. Marsur and Y. Murphey, "Battery state of charge estimation based on a combined model of extended Kalman filter and neural network", International Joint Conference on Neural Networks, **(2011)**, pp. 2156-2163.

Authors

Junghoon Lee

He received the B.S., M.S., and Ph.D. from Dept. of computer engineering, Seoul National University, Korea. From 1990 to 1992, and also in 1996, he was a senior research engineer at Lab. of optical telecommunication, Daewoo Telecom, Korea. In 1997, he joined the Dept. of computer science and statistics in Jeju National University. From 2003 to 2005 and in 2009, he was a visiting scholar at Dept. of computer sciences, University of Texas at Austin. His research interests include real-time communication and smart grid.

Gyung-Leen Park

He received B.S. in Dept. of computer science from Chung-Ang University. He received M.S. and Ph.D. from computer science and engineering department at the University of Texas at Arlington, respectively. His research interests include scheduling in parallel and distributed systems, mobile computing, and vehicular telematics. In 1997, he was an assistant professor at the University of Texas at Arlington. In 1998, he joined the Dept. of computer science and statistics at Jeju National University, Korea. Currently, he is the director of the Smart Grid Research Center, Jeju National University.