ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 21-04-24
j.inst.Korean.electr.electron.eng.Vol.25,No.4,756~761,December 2021
https://doi.org/10.7471/ikeee.2021.25.4.756
173

Dynamic Route Guidance via Road Network Matching and Public Transportation Data

Hoa-Hung Nguyen*, Han-You Jeong*★

Abstract

Dynamic route guidance (DRG) finds the fastest path from a source to a destination location considering the real-time congestion information. In Korea, the traffic state information is available by the public transportation data (PTD) which is indexed on top of the node-link map (NLM). While the NLM is the authoritative low-detailed road network for major roads only, the OpenStreetMap road network (ORN) supports not only a high-detailed road network but also a few open-source routing engines, such as OSRM and Valhalla. In this paper, we propose a DRG framework based on road network matching between the NLM and ORN. This framework regularly retrieves the NLM-indexed PTD to construct a historical speed profile which is then mapped to ORN. Next, we extend the Valhalla routing engine to support dynamic routing based on the historical speed profile. The numerical results at the Yeoui-do island with collected 11-month PTD show that our DRG framework reduces the travel time up to 15.24 % and improves the estimation accuracy of travel time more than 5 times.

Key words : Dynamic route guidance; road network matching; node-link map; openstreetmap; intelligent transportation systems

Ⅰ. Introduction

Dynamic route guidance (DRG) is an emerging new intelligent transportation systems (ITS) application. The DRG provides the fastest path from a source to a destination location. Contrary to the static route guidance, the DRG requires real-time traffic state information for each road, such as the average speed of vehicles. Therefore, the availability of traffic state information is the key factor of DRG.

In Korea, the traffic state information is periodically updated through public transportation data (PTD) [1]. The road of PTD indexed by the node-link map (NLM), the authoritative road network for exchanging ITS information in Korea [2]. However, the NLM represents the road objects and their interconnectivity for major roads only. Moreover, there is no open-source software packages that supports many ITS applications, such as automotive navigation and autonomous driving. Therefore, the NLM itself is not suitable for the DRG.

Openstreetmap (OSM) is a collaborative project to create a free editable geographic information system (GIS) [3]. The OSM road network (ORN) represents the detailed geometry of all street and

* Dept. of Electrical Engineering, Pusan National University
★ Corresponding author
E-mail : hyjeong@pusan.ac.kr Tel : +82-51-510-7332
※ Acknowledgment
This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(2019R1I1A3A01060890).
Manuscript received Dec. 7, 2021; revised Dec. 30, 2021; accepted Dec. 31, 2021.

road networks. Furthermore, there are a few routing engines operating on top of ORN, such as OSRM and Valhalla [4]. However, there is no infrastructure to provide real-time traffic state information in ORN.

Taking into account the above limitations, we propose a DRG framework that integrates the NLM-indexed PTD and the extended open-source routing engine via road network matching (RNM) between NLM and ORN [5]. The RNM is a solution to data association problem between two road networks, where a set of road objects in one road network is mapped into another set of road objects in the other road network. Based on the PTD collected every five minutes over 11 months, the numerical results show that the proposed DRG reduces the travel time up to 15.24 % and improves the estimation accuracy of travel time more than 5 times. To the best of our knowledge, this is the first work to provide the DRG by integrating RNM, PTD, and an open-source route setting engine in a Korean road environment.

The rest of this paper is organized as follows. In section 2, we present our DRG framework. Then, the numerical results of our DRG framework are discussed in section 3. Finally, we conclude this paper in section 4.

II. DRG Framework

Fig. 1 shows the block diagram of the DRG framework consisting of three layers. The traffic state processing layer consists of the PTD crawler and historical speed profile generator to retrieve and generate time- dependent historical speed profile of each road. The RNM adaptation layer is composed of index conversion and ORN extension blocks. The former converts the NLM-indexed PTD to ORN-indexed speed profile using the RNM, whereas the latter provides the detailed information to precisely match road objects. Since the OSRM supports a static routing only, the dynamic routing layer extends the Valhalla routing engine so that it can support time-dependent cost metric. The road network and speed file generators creates graph topology and the cost metric of the Vahalla engine.

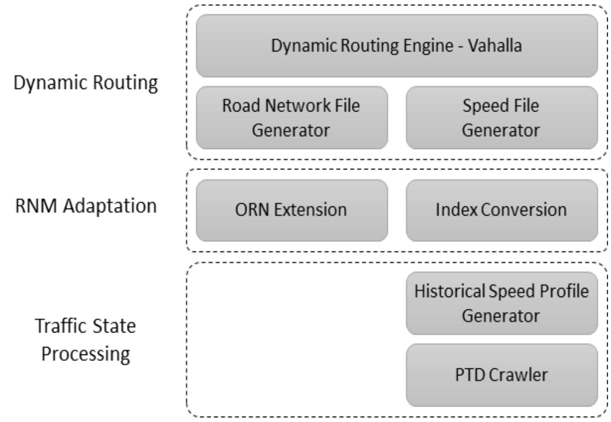

Fig. 1. Block diagram of DRG framework.

2.1 PTD Processing

In Korea, the National Transportation Information Center (NTIC) collects all ITS information about traffic, construction, incidents, and CCTV information and open this information to the public via representational state transfer (REST) APIs [1]. To enable the DRG, we focus on the public traffic information which includes the road speed which is updated every 5 minutes.

To obtain this speed data, our PTD crawler periodically queries the speed profile request to NTIC. The request period is set as 2.5 minutes which is half of the traffic information update period to make sure that no information is missed. The traffic information of a link contains the *roadsectionid, avgspeed,* and *generatedate* which are its NLM index, average speed, and the generation date, respectively.

The DRG needs the speed data of all the roads in the current and near-future time points. We crawled the speed profile of all roads in Yeoui-do island over 11 months which is called the historical speed profile. Since the speed profile shows different pattern depending on time-of-day and day-of-week, the historical speed profile is created to capture these patterns. The collected

PTD is categorized by the timeslot of a week (TSoW) of *generatedate*. The duration of timeslot is selected to the updated period of PTD, i.e. 5 minutes. As a result, there are 2,016 TSoWs, where TSoW 0 starts at 12:00 AM on Sunday.

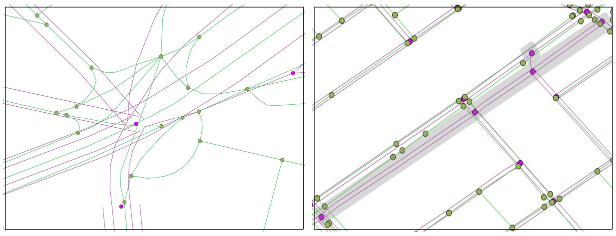

Fig. 2. LoD difference (left), and missing correspondents (right) between NLM (purple) and ORN (green).

2.2 RNM Adaptation

Both road networks, i.e. NLM and ORN, show a significant dissimilarity in terms of the representational level of details (LoD) of a complex intersection, and missing corresponding objects as shown in Fig. 2. Our RNM approach [5] can successfully group a complex ORN intersection to a supernode that can address the LoD difference. The missing correspondents is resolved by inserting a new subgraph to ORN.

An RNM solution generates the mapping between NLM links and OSM edges. Using this mapping, the NLM-indexed PTD can be converted to OSM-indexed speed profile. Only a few geographical and topological NLM node attributes are sufficient to fill out the attribute of new ORN nodes generated to address the missing correspondents. Similarly, the attributes of a new ORN edge can be filled out by using the rule in [6].

2.3 Dynamic Routing

In this section, we extend the Valhalla routing engine to support the DRG. Valhalla is a low-memory routing engine that divides the road network into a tile hierarchy. Valhalla uses the A^* algorithm, a heuristic least-cost path algorithm with low computational complexity [7]. We first convert the ORN and the historical speed profile to the input of Valhalla. Then, we modify the A^* algorithm to support time-dependent cost metric.

2.3.1 *Road Network File Generator*

The first input of Valhalla is the information about road network. Valhalla converts the ORN to its own graph representation $G_V = (N_V, E_V)$. Next, this graph is then divided into tile files depending on its geospatial location.

The second input of Valhalla is the speed profile of each ORN edge. The original Valhalla also has a functionality to convert the speed information of ORN edge into Vahalla cost metric. To support 2,016 TSoW speeds for each edge, we modify the speed file generator so that it can periodically overwrite the Vahalla speed file.

2.3.2 *Dynamic Routing Engine*

Given the road network file and speed file, we extend the A^* algorithm of Vahalla to support DRG. In this section, we first describe the original Valhalla, and ttwo DRG approaches called *snapshot* and *time-dependent*.

The A^* algorithm finds the shortest path by greedily expanding the set of traversed nodes based on the low-bound estimation of sub-path to the destination as shown in **Algorithm 1**. The cost of low-bound estimation of sub-path is represented by the function $f(n_i)$ at line 11, in which $g(n_i)$ is the fastest travel time from the heuristic function that estimates the lower-bound cost from n_i to the destination. In Valhalla, $h(n_i)$ is computed as the travel time over the Euclidean distance from the intermediate node (n_i) to the destination node (n_d) with the maximum speed of road network.

In the original A^* algorithm, for a given source node and node n_i, $g(n_i)$ are constant. However, if the vehicle speed on each edge changes according to TSoW, function $g(n_i)$ should also change accordingly. To obtain the correct speed profile in multiple TSoW cost metrics, the departure

Algorithm 1. A^* algorithm.

Input: node set N_V, edge cost function $c(.,.)$, heuristic function $h(.)$, source node n_s, destination node n_d.
Output: shortest path from n_s to n_d

1. Set $g(n_s) := \infty$ and $f(n_i) := \infty, (\forall n_i \in N_V)$
2. Add n_s to open set N_O
3. Set $prev(n_s) := \varnothing, g(n_s) := 0, f(n_s) := h(n_s)$
4. **while** $N_O \neq \varnothing$
5.   Get $n_i \in N_O$ with lowest $f(n_i)$
6.   **if** $n_i = n_d$, reconstruct path with $prev(.)$ and return
7.   **for** each neighbor n_j of n_i
8.   **if** $g(n_i) + c(n_i, n_j) < g(n_j)$
9.   Set $prev(n_j) := n_i$
10.   Set $g(n_j) := g(n_i) + c(n_i, n_j)$
11.   Set $f(n_j) := g(n_j) + h(n_j)$
12.   **if** $n_j \notin N_O$, add n_j to N_O

time t_s at source node is used to compute the shortest path to the destination [8]. Particularly, given departure time t_s, function $g(n_j)$ at line 10 is computed as follows:

$$g(n_j) = g(n_i) + c(n_i, n_j, t_s) \tag{1}$$

where $c(n_i, n_j, t_s)$ is the cost of the edge between n_i and n_j at time t_s. $c(.,.,t_s)$ is a snapshot of the historical speed profile at the TSoW interval which departure time t_s belongs to. We call this approach the snapshot scheme.

The time-dependent scheme considers the fact that TSoW can change during the travel to the destination. To address this change, function $g(n_j)$ can be calculated as follows:

$$g(n_j) = g(n_i) + c(n_i, n_j, t_s + g(n_i)) \tag{2}$$

Notice that the start time at each ORN edge depends on the travel time of all previous ORN edges along the path. To address this issue, we simulate the TSoW change at each edge as suggested in [8]. **Algorithm 2** shows the algorithm to calculate the edge travel time. The idea is sequentially computing the travel distance l_m in TSoWs until the vehicle leaves the edge based on the constant-speed model. This edge travel time is used to compute the equation in (2).

Algorithm 2. Edge travel time algorithm.

Input: edge length l, TSoW speed v_m
Output: edge travel time

1. Set $l_k = v_k * (t_k - (t_s + g(n_i)))$ and $m = k$
1. **while** $l_m < l$
1.   Set $m = m + 1$
1.   Set $l_m = l_m + v_m * T_{TSoW}$
1. Set $c(n_i, n_j, t_s + g(n_i)) = t_{m+1} - v_m * (l_m - l) - (t_s + g(n_i))$

Table 1. Departing time interval.

Traffic Condition	Day of Week	Time of Days
Free flow	All days of week	2:00–5:00
Changing	Weekdays	20:00–22:00
Congestion	Weekdays	17:00–20:00
Peak	Friday	18:30–19:00

III. Numerical Results

We investigate the performance of three routing schemes at Yeoui-do island, Korea's autonomous vehicle testing site. It covers an area of 3.5 km X 2.8 km. The traffic data is collected from July 2017 to June 2018. The first 80 % of collected data is used for the generation of historical speed profile, and the rest for the test scenario.

We compare the performance of three routing schemes: (1) original Valhalla, (2) snapshot, and (3) time-dependent scheme. We generate 1,000 routing requests each of which has a random source and destination node, and a random request time. For each request, the Euclidean distance between source and destination node is at least 2 Km. To observe the impact of traffic state, we choose four TSoW intervals with different traffic conditions as shown in Table 1.

Fig. 3 shows the average travel time of four TSoWs. As the traffic becomes more congested, the travel time increases. Two DRG schemes always have a smaller travel time than the original Valhalla scheme. Especially, the gap between DRG and static routing becomes higher with the level of traffic congestion. The time-dependent scheme improves the travel time up to 15.24 % compared to the original Valhalla scheme. On the other hand, there is no clear difference between snapshot and time-dependent scheme. From this result, we conclude that the key difference comes from the ability to adapt to a longer time scale change of traffic congestion, rather than a short time scale traffic change.

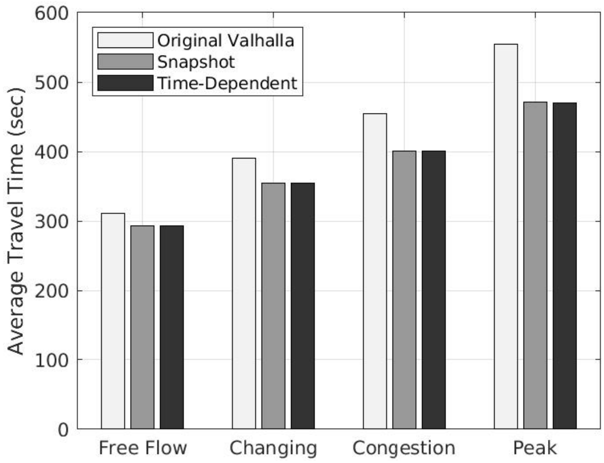

Fig. 3. Average travel time.

Fig. 4 shows the RMSE of vehicle travel time in four traffic conditions. We observe that the DRG schemes can accurately estimate the travel time, at least 5 times better than the original Vahalla scheme. This is because the original Valhalla uses a constant speed metric regardless of traffic situation: as a result, its RMSE increases as road traffic becomes more congested.

IV. Conclusion

In this paper, we present the DRG framework based on the RNM between NLM and ORN. The collected NLM-indexed PTD are aggregated to historical speed profile, and then converted to ORN cost metric. The Valhalla routing engine is extended to support the snapshot and time-dependent A^* algorithm. The result at Yeoui-do island shows that the travel time of DRG scheme is improved up to 15.24 %, and the estimation accuracy of travel time is improved at least 5 times of the original Valhalla scheme.

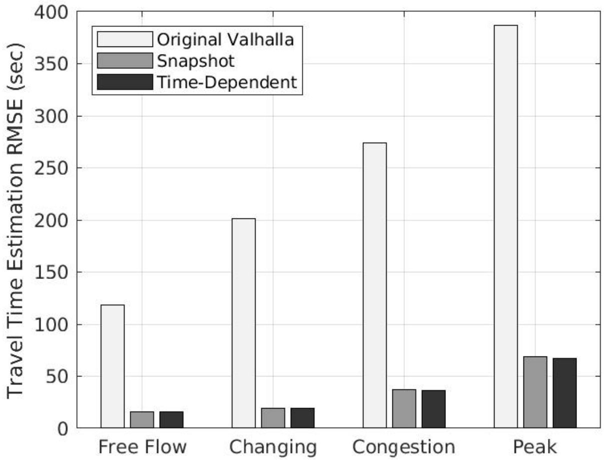

Fig. 4. RMSE of travel time.

References

[1] Korean National Transport Information Center, “Public Traffic Information Service,” http://openapi.its.go.kr/

[2] Korean National Transport Information Center, “Node Link Map,” http://www.its.go.kr/nodelink/

[3] OpenStreetMap Contributors, “OpenStreetMap,” http://www.openstreetmap.org/

[4] D. Gearhart, G. Knisely, K. Kreiser, K. DiLuca, and D. Nesbitt, “Valhalla,” https://github.com/valhalla

[5] H.-H. Nguyen, H.-Y. Jeong, “An Area Partitioning and Subgraph Growing Approach to the Conflation of Road Networks,” http://vialab.pusan.ac.kr/lib/exe/fetch.php?media=Publication:apsg_vialab.pdf

[6] OpenStreetMap. Ko: Map Features. https://wiki.openstreetmap.org/wiki/Ko:Map_Features

[7] P. Hart, N. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Trans. Syst. Sci. Cybern.,* vol.4, no.2, pp.100-107, 1968.
DOI: 10.1109/TSSC.1968.300136

[8] L. Zhao, T. Ohshima, and H. Nagamochi, “A∗

algorithm for the time-dependent shortest path problem," in *Proc. of the 11th Japan-Korea Joint Workshop on Algorithms and Computation (WAAC08)*, pp.36-43, 2008.

BIOGRAPHY

Hoa-Hung Nguyen (Member)

2009 : BS degree in Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Vietnam.
2013 : MS degree in Department of Big Data, Pusan National University.
2020 : PhD degree in School of Electrical, Electronic and Computer Engineering, Pusan National University.
2020~Present: Postdoctoral Researcher, Pusan National University

Han-You Jeong (Member)

1998 : BS degree in Electrical Engineering, Seoul National University.
2000 : MS degree in Electrical and Computer Engineering, Seoul National University.
2005 : PhD degree in Electrical and Computer Engineering, Seoul National University.
2005~2007 : Senior Engineer, Samsung Electronics.
2008 : Postdoctoral Researcher, University of Minnesota, USA
2015 : Visiting Scholar, University of Paderborn, Germany
2008~Present: Professor, Pusan National University