A Fastest Route Planning for LBS based on Traffic Prediction

YONG-BIN KANG, SUNG-SOO KIM
Telematics·USN Research Division
Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon.
REPUBLIC OF KOREA

Abstract: This paper proposes a new route plan on the basis of traffic prediction for finding a fastest route to a given destination in traffic network. So far, lots of traffic prediction systems were introduced to help drivers. Previous works were done mainly on providing restricted route services which depend on only cumulative traffic velocities. For this reason, we consider both real-time and cumulative traffic information together to obtain more accurate future traffic information. In location-based services, the traffic network is needed to solve certain constrains, such as turns problems and provide method for avoiding traffic congestions. To guide a fastest route service in such a complicated network, we first construct a linear dual graph from a traffic network. Then, we propose main algorithmic approaches which are developed by Kalman Filter and cumulative traffic patterns to predict a much better quality of future traffic information by combining real-time with cumulative traffic conditions. Finally, we adopt Dijkstra's shortest path algorithm to minimize the travel time with generating a fastest cost function. Experimental results show that this approach is highly efficient in route plan than previously used ways by only cumulative approaches. This approach is supposed to proceed convenience for drivers and develop a quality of navigation service in telematics.

Key-Words: Fastest Route Planning, Traffic Prediction, Kalman Filter, Cumulative Traffic Patterns, Linear Dual Graph, LBS

1 Introduction

1.1 Motivation

Recently, location-based service(LBS) have been increasingly providing a variety of services to our real-life, along with developments in the areas of location positioning, wireless communication and automotive technologies. Also, LBS are considered to be a highly essential technologies in telematics services, which are heavily dependent on the vehicle's location information. In particular, navigation service in LBS must be capable of facilitating driver's convenience and safety by providing more accurate traffic information of near future in traffic network. In addition, this service should continuously find and notify all the current and future traffic conditions to conduct a fastest route for drivers with avoiding congested areas.

In traffic network, where the travel time changes dynamically, the correctness of the fastest route plan very much depends upon the correctness of the cost model of the route. Typical route cost model based on static information that, in general, is heavily weighted by the only cumulative velocities. Unfortunately, as many drivers in traffic areas are aware, this cost model cannot entirely be appropriate for determining fastest route, because real-time condition of routes such as the severity of congestion should be play an important role. For this reason, it is required to provide an effective fastest route service of truly valuable future traffic information reflecting on real-time traffic conditions.

1.2 Related Works

The analysis of traffic network is one of the many application areas where the computation of shortest paths is one of the most fundamental problems. The route selection problem in traffic network involves finding an optimal route from a starting point to a destination on a road map[10]. Even those traffic conditions, such as the distribution of congestion, change during driving, the route should be reevaluated before the car reaches the next intersection. Since no "best" algorithm exists for every kind of transportation problem on traffic network, research in this field has recently moved to the design and the implementation of "heuristic" shortest path algorithm[1]. Much of the focus has been on the choice and implementation of efficient data structures[6].

There is one more interesting related work by Chon et al.[3]. Chon et al. presents a time-dependent shortest path algorithm that the trajectory of moving objects are determined by monitoring the real-time network conditions. In the study of Chon et al., they show that the average travel time of moving objects has been markedly reduced by considering dynamically changing information of moving object with variation of time.

In this paper, we do not attempt to solve or propose yet another variation of the shortest path problems. The major contribution of this work is to present a new route plan is that based on the traffic prediction reflecting on dynamic changing of real-time and patterns of cumulative traffic information.

1.3 Overview of Route Planning

Major procedure for our determination of fastest route planning is as follows (see also Fig.1).

Step 1: Construction of a linear dual graph
Construct a linear dual graph D_T [4][9] from a given data of traffic network G_T, which is represented by schema of node and edge. D_T is a fundamental structure for determining fastest route in our approach.

Step 2: Prediction of future traffic velocity
Predict the future traffic velocity $V_{P(t+d)}$, which is evaluated by real-time velocity $V_{R(t)}$ and cumulative traffic patterns $V_{C(t-d)}$ at time t, where d denote an interval index of given periods of time.

Step 3: Cost definition for fastest routing service
A cost function $\omega(e_k)$ of each edge e_k in D_T is formed by using $V_{P(t+d)}$ and traffic topology T_G.

Step 4: Performance of fastest route service
Dijkstra's shortest method is performed to guarantee the fastest route plan using $\omega(e_k)$.

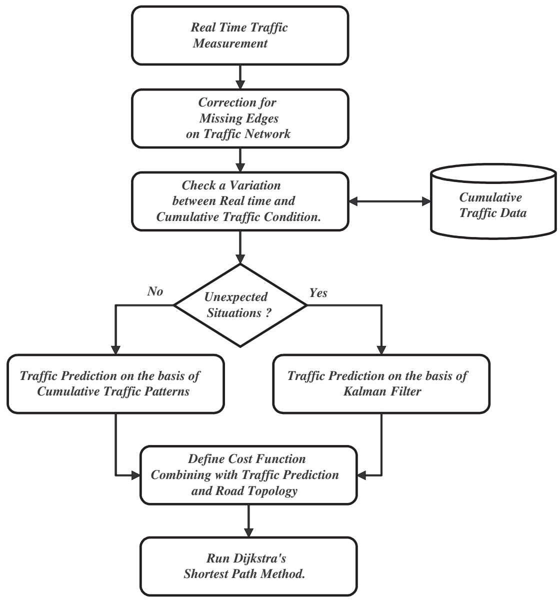

Fig. 1: Overview of our proposed traffic prediction system.

2 Route Planning using Linear Dual Graph

In most of traffic networks, the route planning for car navigation or public transport should consider no-left-turn, P-turn, U-Turn and other turn problems to find a minimal travel time cost. In particular, in urban environment, turning left is often forbidden in order to minimize congestions and, when allowed, traffic lights and counter flow cause an extra travel cost itinerary between two nodes.

The common approaches to handle these problems are *node expansion*[8] and linear dual graph method[4]. The node expansion approach builds up the expanded network, G_e, which is obtained by highlighting each movement in the intersections by means of dummy nodes and edges, where the costs of the dummy edges are the penalties. The major disadvantage of this approach is that the resulting network G_e is significantly larger than the original graph G.

Generally, a directed graph G is consists of a set of nodes N and a set of edges E connecting the nodes, $G(N, E)$. Given a graph $G(N, E)$, the linear dual graph D $=\mathcal{L}(G)$ has node set $N(D) = E(G)$ and edge set $E(D) = \{ab : a, b \in N(D)$, the head of a coincides with the tail of $b\}$. The detail definition of a linear dual graph is described in[8](see also Fig.2):

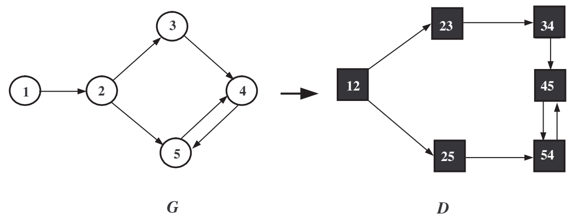

Fig. 2: A digraph G and its linear dual graph $D = \mathcal{L}(G)$

In determining the performance of the approaches, it is useful to summarize their storage requirements. The node expansion method requires $\delta_{max}|N_G|$ nodes for the primal graph G since any node n expands into $\delta(n)$ nodes (where, δ_{max} denotes maximum degree of node). This method also requires the number of edges in G, $|E_G|$, plus the number of path of length 2, $\frac{\delta_{max}}{2}|E_G|$[7]. The *boundary node* is a node which has a single-source and single-target in G. The linear dual graph (D) method requires the number of nodes which $|E_G|$ adds to the number of boundary nodes in G, $|N'_G|$. The storage requirement for the edges in D is the number of edges which $\frac{\delta_{max}}{2}|E_G|$ plus $|N'_G|$ (see Table 1).

Method	$\|N\|$	$\|E\|$
Node expansion (G_e)	$\delta_{max}\|N_G\|$	$(1+\frac{\delta_{max}}{2})\|E_G\|$
Dual graph (D)	$\|E_G\| + \|N'_G\|$	$\frac{\delta_{max}}{2}\|E_G\| + 2\|N'_G\|$

Table 1: Estimation of upper limits for node expansion and linear dual graph. (N'_G denotes boundary nodes of a primal graph G and δ_{max} denotes maximum degree of node)

Based on these facts, we adopt the linear dual graph technique as a conceptual model for our route specification on G_T.

3 Traffic Prediction

To predict $V_{P(t+d)}$, we measure not only $V_{C(t-d)}$ simply, but through $V_{R(t)}$ and $V_{C(t-d)}$. If $V_{R(t)}$ is similar to that of the of regular cases then, the result of $V_{P(t+d)}$ will be obtained by the difference between $V_{R(t)}$ and $V_{C(t-d)}$. Otherwise, we apply a method on the basis of Kalman Filter for finding a practically useful convergence by combining updates with $V_{R(t)}$ and $V_{C(t-d)}$. To find a minimal travel time cost, it is important to properly calculate $\omega(e_k)$ from D_T. This $\omega(e_k)$ is determined with results of $V_{C(t-d)}$ and T_G.

3.1 Real-Time Measurement and Correction

To predict the accurate traffic condition, we first measure $V_{R(t)}$ in G_T. This measurement can be possible by AVI(Automatic Vehicle Identification) systems or the diffusion of car navigation devices. In the results of these measurement, however, it may contain missing edges in G_T which were not measured correctly. Two kinds of missing edges can be measured according to the extent of these edges - (a) *one specific missing edge* and (b) *all missing edges can be found in a specific region.*

Under these cases, appropriate correction processes are required to be done to obtain the more suitable measurements. Correction of missing edges are made by deriving the average velocity of adjacent edges except edges which permit U-turn for the case of (a). In the case of (b), $V_{C(t)}$ of all adjacent edges in a specific region are treated as the correction velocity, because all edges in this region are missing edges. As already noted, $V_{C(t)}$ means the cumulative traffic patterns, which represent a cumulative average traffic velocity at current time (t).

Fig.3 shows an example of the case of (a) that the correction process can be applied when a missing edge obtained. Let $v(\overrightarrow{a,b})$ be a velocity of two adjacent node from a toward b.

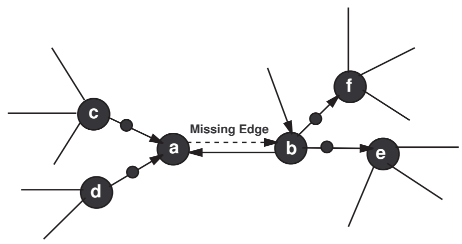

Fig. 3: An example of a missing edge and its correction process.

It is seen from Fig.3 that the average velocity of those edges $v(\overrightarrow{c,a})$, $v(\overrightarrow{d,a})$, $v(\overrightarrow{b,f})$, $v(\overrightarrow{b,e})$ is treated as correction velocity for $v(\overrightarrow{a,b})$.

3.2 Detection of Unexpected Situations

Particularly, it should be taken notice of that unexpected situations(e.g., traffic congestion, road construction, traffic accident) are found to accurately provide these situations to drivers. To find out such edges, a comparison method between $V_{R(t)}$ and $V_{C(t)}$ was required.

Let $V_{R(t)}(e_k)$ be $V_{R(t)}$ and $V_{C(t)}(e_k)$ be $V_{C(t)}$, where a connected edge e_k in G_T. Then, unexpected situations for all e_k can be detected by the following expression,

$$| \; V_{R(t)}(e_k) - V_{C(t)}(e_k) \; | \quad > \delta \text{ , where}$$

$$\delta = \alpha * \sqrt{\frac{\sum_{d=1}^{n}(V_{C(t)}(e_k) - V_{C(t-d)}(e_k))^2}{n}}$$

where δ is adjusting threshold which is believed as boundary of unexpected traffic situations and constant α represent average velocity deviation within a given total periods of time n.

In this present work, there are two possible traffic prediction models will be defined differently, in the case of when these unexpected situations appeared or not.

3.3 Traffic Prediction based on Cumulative Traffic Patterns

In this section, we describe the method that predict the future traffic condition, which is applied when a deviation between $V_{R(t)}$ and $V_{C(t)}$ can be regarded as permitable small.

Based on the facts that have been observed so far, traffic patterns which were classified by the same periods of days and times have been shown quite similar aspect. Therefore, we attempt to define the prediction velocity on each e_k at prediction time $(t+d)$ as a future value by using the following equation,

$$V_{P(t+d)}(e_k) = V_{R(t)}(e_k) \pm f(e_k), \text{where}$$

$$f(e_k) = \beta * \sqrt{\frac{\sum_{d=0}^{m}(V_{R(t)}(e_k))^2 - (V_{C(t-d)}(e_k))^2}{m}}$$

where β is average velocity deviation within a given total period of time m. The result of this evaluation turned out to be proceed smoothly and provide desired prediction velocity.

3.4 Traffic Prediction based on Kalman Filter

In this section, we derive the Kalman filter as the optimal velocity predictor in D_T for a discrete time-varying system. The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements[2].

The input to the system consists of a sequence of unknown K-dimensional real-valued input column vectors $u_j (j = 1, 2, ...)$. A sequence of hidden N-dimensional real-valued state vectors $x_j (j = 1, 2, ...)$ are meant to represent the internal dynamics of the system. Input vector u_j and state vector x_j combine linearly to produce the next state vector:

$$x_{j+1} = A_j x_m + B_j u_j$$

with a measurement $z \in \Re^M$ that is

$$z_j = H x_j + v_j$$

A_j is the $N \times N$ state transition matrix at time j and B_j is an $N \times K$ matrix that maps the input into the state space.

In our research case, element values of matrix A are velocity values of G_T. The input vectors may also be interpreted as state transition noise vectors. The input vector can be decomposed as $B_j u_j + \omega_j$, where ω_j is possible noise. We simply exploit initial matrix A by initializing the $V_{C(t)}$. The matrix $M \times N$ matrix H in the measurement equation relates the state to the measurement z_j. In practice H might change with each time step or measurement so we adopt the average value of the cumulative velocities for each road link. Fig.4 shows the prediction process by using Kalman filter. The detailed prediction procedures are beyond the scope of this paper. So, we will not cover detail two steps of the Kalman filter.

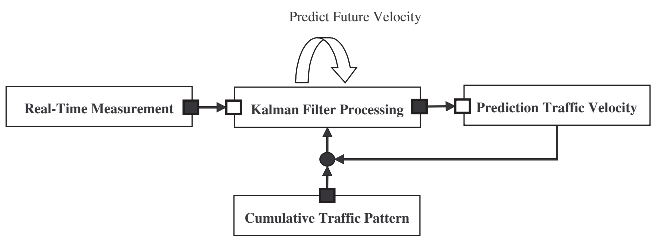

Fig. 4: The process of prediction by using Kalman filter's iterative solution.

4 Fastest Route Planning

The *route service* is a network-accessible service that determines travel routes and navigation information between two or more points. The OpenLS route determination service is one of the OpenLS service specification for route service[5]. In this specification, the route preferences are defined as follows:

(1) *"Fastest"* - Minimize the travel time

(2) *"Shortest"* - Minimize the travel distance

(3) *"Easiest"* - Minimize the driving difficulty

(4) *"Pedestrian"* - Best route by foot

(5) *"PublicTransportation"* - Best route by public transportation.

In this work, we only describe *"Fastest"* preference case. To provide this routing service, the weight cost of between two adjacent nodes to be calculated. The cost function was defined by using $V_{P(t+d)}(e_k)$, which indicates predication traffic velocity of (e_k), road classes c_k and the number of traffic-lane m.

Thus, the cost function of (e_k) can be expressed by the following expression,

$$\omega(e_k) = c_k \cdot \lambda, \text{where}$$
$$\lambda = \frac{1}{V_{P(t+d)}(e_k) \cdot m}, 0 < \lambda \leq 1$$

After then, Dijkstra algorithm is conducted to search fastest path on D_T.

5 Experiment

In order to evaluate the accuracy of the proposed approach, we performed a comparative analysis on an actual traffic network, Kang-Nam area where consists of about 10,000 nodes in Seoul, Korea. In this experiment, accumulated data which include cumulative traffic velocities were prepared for six months. The three prediction method was evaluated for comparative analysis as follows:

(1) *"Method A"* - by the only cumulative traffic data

(2) *"Method B"* - by the real-time and the cumulative traffic data

(3) *"Method C"* - by the real-time and the cumulative traffic data(with Kalman filtering)

To analyze the accuracy of these cases, the average deviation errors on all the edges are given by the following expression,

$$\varepsilon(t) = \frac{\sum_{k=1}^{n} \mid V_{R(t)}(e_k) - V_{P(t)}(e_k) \mid}{n}$$

, which denotes a average value of the difference between the actual and the prediction velocity at time t.

We measured a traffic network in every 5 minutes intervals during hours(08:00 $\sim$ 09:00) to obtain a real-time velocities and predicted in every 5 minutes and the length of prediction time were 3 peak hours(08:05 $\sim$ 11:05). The experimental results are obtained through investigating for 4 weeks at the 95% confidence interval.

AVERAGE DEVIATION OF ERROR $\varepsilon(t)$: 5 KM
METHOD A : 85.3%
METHOD B : 86.2%
METHOD C : 89.2%

On the *"Method C"*, Kalman filter was applied when $\varepsilon(t) > 10$km and more accurate results are obtained comparing with *"Method A"* and *"Method B"* as can be seen the above results. Fig.5 shows the corresponding results of our experiments, which illustrates the rates of edges(0 $\sim$ 25%) in the average deviation of errors between the actual and the prediction velocities(1km $\sim$ 20km).

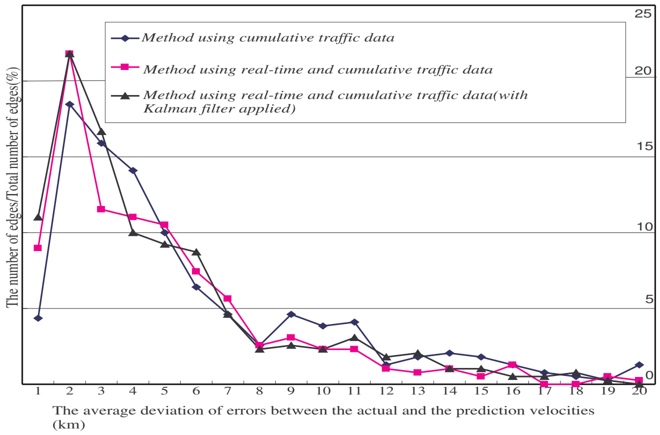

Fig. 5: Comparison results of the three prediction method.

As you can see in this graph, the rates for *"Method C"* were always at higher level(about 12%, 23%, 17%) than the others, *"Method B"*(about 8%, 23%, 12%) and *"Method A"*(about 4%, 18%, 16%) between 1km to 3km, during the lowest scope of errors, while nearly the same for the rest of scope of errors(4km $\sim$ 20km). These indicate that the rate of edges for *"Method C"* gives us minimal deviation error between the actual and prediction velocities and can provide higher trustworthy prediction model than the two others, *"Method A"* and *"Method B"*. Fig.6 shows a snapshot from navigation device, providing traffic prediction service that was proposed by our approach. The traffic prediction service flow is that firstly, a user selects traffic prediction service menu during the navigation and then a user sets desirous future time for traffic prediction service, finally, the user can preview the result of predicted traffic information as texts or on the map.

Fig. 6: A snapshot of navigation device simulated with our approach.

From these experimental results, we expect that our proposed method could be efficient for future traffic prediction and can be integrated with conventional method.

6 Conclusion

In this paper, we propose a new route plan to guide a fastest route from a given destination, which based on future traffic prediction. The data structure of linear dual graph is our primal graph that permit to minimize travel time cost and solve the turn problem that may be occurred at the intersection area. To predict the future traffic condition, we first measured real-time traffic condition for reflecting on dynamic changing of current traffic flows. The traffic prediction model on the basis of real-time and cumulative traffic patterns was proposed to provide more realistic and accurate traffic information to drivers. If the measured real-time velocity was permitted within satisfiable threshold, traffic prediction was determined by adopting the cumulative traffic patterns. Otherwise, Kalman filtering method was adopted in case of the traffic congestion or unexpected circumstances. The results of our works are supposed to provide driver with truly valuable traffic information of near future. Furthermore, this work may also be extended to the fields in the services of LBS solution and telematics.

References:

[1] Chang Wook Ahn, R. S. Ramakrishna, A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations, *IEEE Transaction on Evolutionary Computation*, pp. 566-579, Vol. 6, No. 6., 2002.

[2] Greg Welch et al., An Introduction of the Kalman Filter, *The SIGGRAPH 2001 Course Note 8*, 2001.

[3] H. D. Chon, D. Agrawal and A. E. Abbadi, FATEST:Finding A Time dependent Shortest Path, *Proceedings of the 4th International Conference on Mobile Data Management*, 2003.

[4] J. Anez, T. de la Barra, B. Perez, Dual Graph Representation of Transport Networks, In *Transportation Research*, Vol. 30, No. 3, pp. 209-216. 1996.

[5] OpenLS Initiative, OpenLS Route Determination Service Specification, *OpenGIS Interoperability Program Report* (OGC 02-090), Open GIS Consortium Inc., 11 November 2002.

[6] Stefano Pallottino, Maria Grazia Scutella, Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects, Technical Report, Univ. of Pisa, 1998.

[7] Stephan Winter, Weighting the Path Continuation in Route Planning, In *Proceedings of 9th ACM International Symposium on Advances in Geographic Information Systems*, pp. 173-176, 2001.

[8] Stephan Winter, Route Specifications with a Linear Dual Graph, In *Proceedings of Advances in Spatial Data Handling*, pp.329-338, 2002.

[9] Stephan Winter, Modeling Costs of Turns in Route Planning, *GeoInformatica*, Vol. 6, No. 4, pp. 345-360, 2002.

[10] T. Nakamura et al., Head Start Route Selecting Algorithm for Reducing Driver's Waiting Time, *The Second World Congress on Intelligent Transport System*, pp.2031-2036, 1995.