Finding shortest path to non-stop intersection for future transportation

Oanh Tran Thi Kim, Jun Hyeong Lee, Ji Yeon Kim, Choong Seon Hong*
Department of Computer Engineering, Kyung Hee University
{ttkoanh, wnsgud3014, nabillee, cshong}@khu.ac.kr

Abstract

Shortest path query is an important problem and has been well studied in static graphs. However, in practice, there are many factors that influence cost of edges such as weather, time of day, and vehicle type. Then, the costs of edges in graphs always change over time. Therefore, finding shortest path needs to be considered not only distance but also factors affecting road speed. In this paper, we addressed the problem of computing fastest paths underlying future transportation with traffic speed patterns to non-stop intersections. A system model is proposed to show how a finding path processing is performed in future transportation. By defining the path cost, we propose an efficient finding path algorithm to compute a cost-optimal path.

Key word: shortest path, future transportation, traffic management system

1. Introduction

Information technology (IT) has transformed many industries, from education to health care to government, and is now in the early stages of transforming transportation systems. It enables elements within the transportation system—vehicles, roads, traffic lights, message signs, etc. to become intelligent by embedding them with microchips and sensors and empowering them to communicate with each other through wireless technologies. In the leading nations in the world, Intelligent Transportation Systems Information (ITS) bring significant improvement in transportation system performance, including reduced congestion and increased safety and traveler convenience. Intelligent transportation systems include a wide and growing suite of technologies and applications such as real-time traffic information systems, in-car navigation (telematics) systems, vehicle-to-infrastructure integration (VII), vehicle-to-vehicle integration (V2V), adaptive traffic signal control, ramp metering, electronic toll collection, congestion pricing, fee-based express (HOT) lanes, vehicle usage-based mileage fees, and vehicle collision avoidance technologies.

South Korea is one of the world's leaders in in intelligent transportation systems based on the importance ascribed to ITS at the highest levels of government, the number of citizens benefitting from use of an impressive range of opera- tionally deployed ITS applications, and the maturity of those applications. South Korea's strengths in several ITS application ar- eas make it a world leader in intelligent transportation systems. These strengths include: 1) real-time traffic information provision, 2) advanced public transportation information systems, and 3) electronic fare payment and electronic toll collection [1]. South Korea's National ITS Service addresses: traffic operations and management, electronic payments, information integration and dissemination, public transport quality enhancement, enhanced safety and automated driving, efficient commercial vehicles, and pollution control. A central mission of the National ITS Service is to create a network of traffic systems that facilitate interactions and interconnection between South Korea's large cities.

In this paper, we put the finding fastest path problem into perspective of ITS. In South Korea, the Seoul Urban Expressway Traffic Management System can gathers an analyzes traffic information by using cuttting-edge video detectors, Dedicated Short Range Communications (DSRC), and CCTVs, and then provides live traffic information through electric road signs, the Internet, and smartphones. So we can get factors affecting road speed, such as weather, time of day, and traffic density. These traffic data are often more useful than the simple Euclidean distance-based computation to choose these routes.

Most existing work on path computation has been focused on the shortest-path problem. Dijkstra's algorithm [1-2] and A*search [3] are often used to find the shortest path in various real-life applications such as robot navigation, routing problems and games. A*, with a consistent heuristic, is usually faster than Dijkstra, in practice, for finding the shortest path between two nodes, but Dijkstra algorithm has the advantage of calculating all distances from one node to all other nodes on the graph, in a single run. Nevertheless, these two algorithms may be inefficient in terms of computation time for large-scale grid environments. Also, they just focus on Euclidean distance as main parameter to decide the shortest path. With more advanced, we will take additional conditions, such as weather forecast, or road construction/closure information to improves trip duration.

The remainder of this paper is organized as follows. The overview of future transportation and some algorithms about finding shortest path are presented in

Section 2. Finally, Section 3 highlights the conclusion.

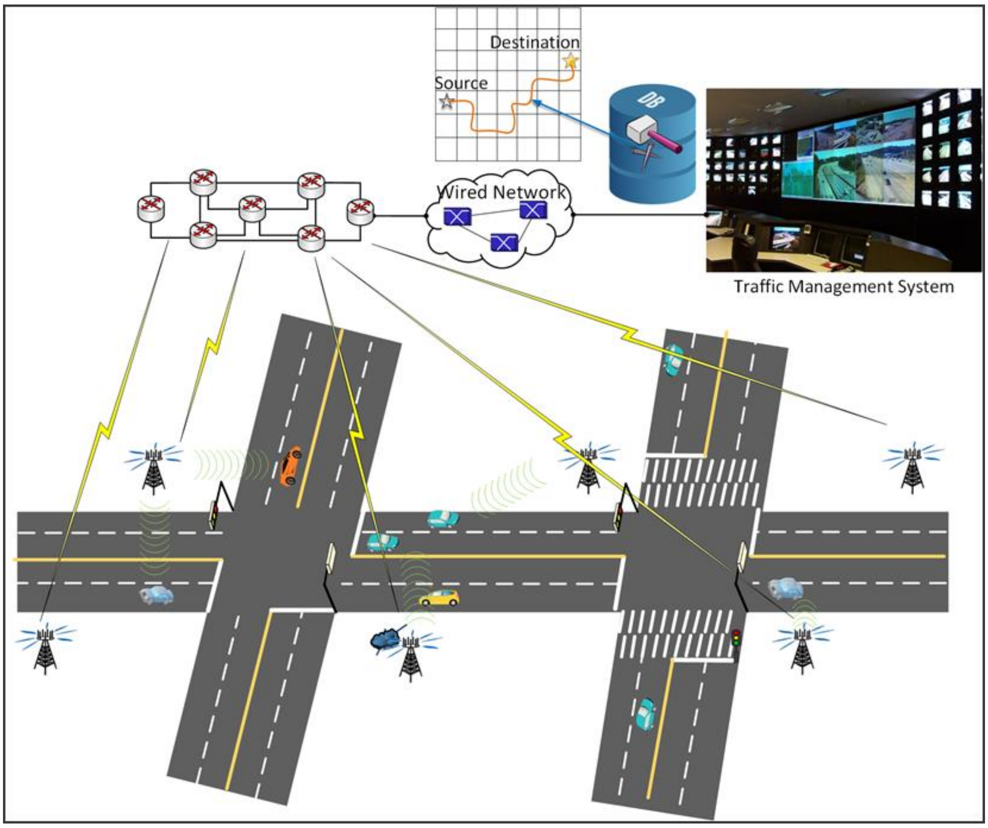

Fig1. System model: Finding shortest path for future transportation

2. Finding shortest path underlying future transportation

A system mode, finding shortest path underlying future transportation, is illustrated in Fig.1. In future transportation, all of things on the roads are managed and analyzed at TMS. Thus, in our model, we can monitoring traffic situations and facility conditions. We also can improve traffic strategies, respond to traffic situations, and analyze traffic statistics. In addition, each RSU acts as a link between cars and TMS. Then, when a car starts a journey, it will send finding path request to Roadside Unit (RSU). Each RSU will communicate with TMS to get shortest path from car's source to car's destination. At TMS, when it receive a finding path request from RSU, a processing is occurred. By mining data at its database, the result is transferred to car through RSU.

Next, an algorithm is presented in details next part to demonstrate how the finding shortest path is processed under our model. The algorithms is based on A^* algorithm. However, underlying our model, finding shortest path not only consider about distance but also take additional conditions such as traffic flow, weather conditions, and traffic situations. These information is retrieved by mining data at ITS's data house. Therefore, it make the result of our proposal is more reliable and effective. Pertaining to the details of the proposal, basic definition and notations are defined as follows.

Definition 2.1. A road network is a directed graph G(V, E), where V is a set of vertices representing road intersections and terminal points, and E is a set of edges representing road segments each connecting two vertices.

We assume that road networks are partitioned into cells by intersections.

Definition 2.2. A speed pattern is a tuple of the form (edge_id, t_start, t_end,(d_1, d_2, . . . , d_k): m), where edge id is an edge, (t_start, t_end) is a time interval, each d_i is a value for speed factor D_i, and m is an aggregate function computed on edge speed.

Speed patterns are obtained through a processing of mining traffic data [5].

An example of speed patterns are presented in Table 1. In the example we list edge speeds for three conditions: *time- of-day, D1 = weather, and D2 = vehicle-type.*

Table 1: Speed Pattern for a Particular Edge

Time	Weather	Vehicle	Speed
1am-8am	Good	Car	65 mph
1am-8am	Bad	Car	45 mph
8am-10am	Good	Car	40 mph
8am-10am	Bad	Car	25 mph

Problem Statement. Given a road network G(V, E), a set of speed patterns S, and a query q ← (s, e, start time), compute a best route q_r between nodes s and e starting from s at time start time, such that q_r contains a small number of intersection.

In our finding path algorithm, this is done by calculating a cost of each cell, which is called path cost, denoted f_{cost}.

Path cost: For each cell, $f_{cost} = G + H$ where:

- G is the intersection cost from the start cell **s** to the current cell. So for a cell adjacent to the start cell S, this would be 1, but this will increase as we get farther away from the start cell.
- H is the time cost to move from the current cell to the destination cell.

$H = \mathit{Distance}_{(\text{current cell, destination cell})})/\mathit{Speed\ pattern}_{(\text{current cell, destination cell})}$

The Algorithm: Finding shortest path underlying future transportation

open list: is used to write down all the cells that are being considered to find the shortest path.

closed list: is used to write down the cell that does not have to consider it again.

A car will find the shortest path by repeating the following steps:

1) Get the cell on the **open list** which has the lowest cost. Let's call this cell **s**.
2) Remove **s** from the **open list** and add **s** to the **closed list**.
3) For each cell C in **s**'s walkable adjacent tiles:
    A. *If* C *is in the* **closed list**: Ignore it.
    B. *If* C *is not in the* **open list**: Add it and compute its cost.
    C. *If* C *is already in the* **open list**: Check if the f_{cost} is lower when we use the current generated path to get there. If it is, update its cost and update its parent as well.

An example is shown in Fig.2.

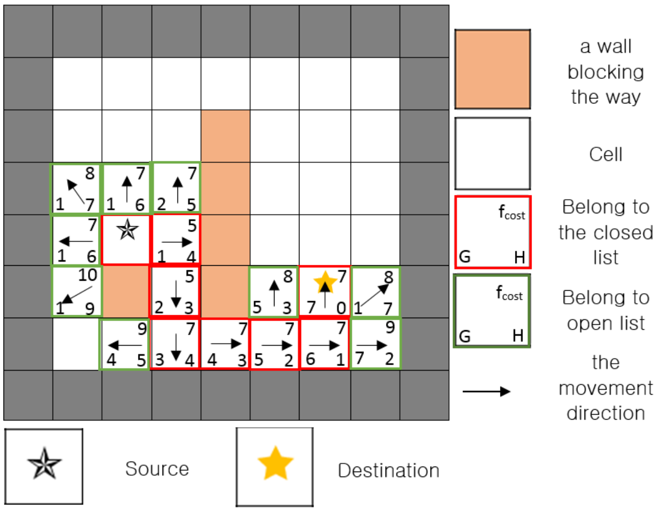

Fig 2. Example of finding shorstest path by using our algorithm

3. Conclusion

In this paper, we addressed the problem of computing fastest paths underlying future transportation with traffic speed patterns to non-stop intersections. A system model is proposed to show how a finding path processing is performed in future transportation. By defining the path cost, we propose an efficient finding path algorithm to compute a cost-optimal path.

4. Acknowledgement

This work was supported by the ICT R&D program of MSIP/IITP. [R-20150224-000386, A Development of Software Technology for the Generation of Detailed Map Progressively Updated and the Driving Situation Awareness based on Cloud Computing] *Dr. CS Hong is the corresponding author.

5. Reference

[1] Ezell, Stephen. "Explaining international it application leadership: Intelligent transportation systems. The Information Technology and Innovation Foundation (ITIF)." Washington DC (2010): 1.

[2] DijkstraEW(1959), A note on two problems in connexion with graphs. NumerMath 1(1):269–271

[3] Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination ofminimum cost paths. IEEETrans Syst Sci Cybern 4:100–107

[4] ChoubeyN, GuptaMBK(2013) Analysis ofworking ofDijkstra and A* to obtain optimal path. Int J Comput SciManag Res 2:1898–1904

[5] Gonzalez, Hector, et al. Adaptive fastest path computation on a road network: a traffic mining approach. Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment, 2007.

[6] A. D. Zhu, H. Ma, X. Xiao, S. Luo, Y. Tang, and S. Zhou. Shortest path and distance queries on road networks: towards bridging theory and practice. In SIGMOD Conference, pages 857–868, 2013.

[7] J. Gao, R. Jin, J. Zhou, J. X. Yu, X. Jiang, and T.Wang. Relational approach for shortest path discovery over large graphs. In PVLDB, volume 5, pages 358–369, 2012.

[8] J. Cheng, Y. Ke, S. Chu, and C. Cheng. Efficient processing of distance queries in large graphs: a vertex cover approach. In SIGMOD Conference, pages 457–468, 2012.