Towards Online Shortest Path Computation

Abstract

The online shortest path problem aims at computing the shortest path based on live traffic circumstances. This is very important in modern car navigation systems as it helps drivers to make sensible decisions. To our best knowledge, there is no efficient system/solution that can offer affordable costs at both client and server sides for online shortest path computation. Unfortunately, the conventional client-server architecture scales poorly with the number of clients. A promising approach is to let the server collect live traffic information and then broadcast them over radio or wireless network. This approach has excellent scalability with the number of clients. Thus, we develop a new framework called live traffic index (LTI) which enables drivers to quickly and effectively collect the live traffic information on the broadcasting channel. An impressive result is that the driver can compute/update their shortest path result by receiving only a small fraction of the index. Our experimental study shows that LTI is robust to various parameters and it offers relatively short tune-in cost (at client side), fast query response time (at client side), small broadcast size (at server side), and light maintenance time (at server side) for online shortest path problem.

Existing System

SHORTEST path computation is an important function in modern car navigation systems and has been extensively studied. This function helps a driver to figure out the best route from his current position to destination. Typically, the shortest path is computed by offline data pre-stored in the navigation systems and the weight (travel time) of the road edges is estimated by the road distance or historical data. Unfortunately, road traffic circumstances change over time. Without live traffic circumstances, the route returned by the navigation system is no longer guaranteed an accurate result. Suppose that we are driving from Lord & Taylor (label A) to Mt Vernon Hotel Museum (label B) in Manhattan, NY. Those old navigation systems would suggest a route based on the pre-stored distance information. Note that this route passes through four road maintenance operations (indicated by maintenance icons) and one traffic congested road (indicated by a red line). In fact, if we take traffic circumstances into account.

Nowadays, several online services provide live traffic data (by analyzing collected data from road sensors, traffic cameras, and crowdsourcing techniques), such as Google-Map, Navteq, INRIX Traffic Information Provider, and TomTom NV, etc. These systems can calculate the snapshot shortest path queries based on current live traffic data; however, they do not report routes to drivers continuously due to high operating costs. Answering the shortest paths on the live traffic data can be viewed as a continuous monitoring problem in spatial databases, which is termed online shortest paths computation (OSP) in this work. To the best of our knowledge, this problem has not received much attention and the costs of answering such continuous queries vary hugely in different system architectures.
Disadvantages of Existing System

1. High Cost
2. No accuracy due to heavy traffic

Proposed System

Typical client-server architecture can be used to answer shortest path queries on live traffic data. In this case, the navigation system typically sends the shortest path query to the service provider and waits the result back from the provider (called result transmission model). However, given the rapid growth of mobile devices and services, this model is facing scalability limitations in terms of network bandwidth and server loading. According to the Cisco Visual Networking Index forecast, global mobile traffic in 2010 was 237 petabytes per month and it grew by 2.6-fold in 2010, nearly tripling for the third year in a row. Based on a telecommunication expert, the world’s cellular networks need to provide 100 times the capacity in 2015 when compared to the networks in 2011. Furthermore, live traffic are updated frequently as these data can be collected by using crowdsourcing techniques (e.g., anonymous traffic data from Google map users on certain mobile devices). As such, huge communication cost will be spent on sending result paths on the this model. Obviously, the client-server architecture will soon become impractical in dealing with massive live traffic in near future. Ku et al. Raise the same concern in their work which processes spatial queries in wireless broadcast environments based on Euclidean distance metric.

Advantages of Proposed System

1. Accurate routing
2. Alternate routing

Example

(a) Shortest route using pre-stored weights
(b) Shortest route using live traffic (by LTI)
SYSTEM REQUIREMENTS:

SOFTWARE REQUIREMENTS:

• Operating system : Windows XP.
• Coding Language : Java.
• Data Base : MY SQL