NANYANG TECHNOLOGICAL UNIVERSITY

SCE16-0446
Time-Dependent Shortest Path Queries on Mobile Devices

Submitted in Partial Fulfillment of the Requirements
for the Bachelor of Computer Science
of the Nanyang Technological University

by

Wei Yumou

School of Computer Science and Engineering
2017

SCE16-0446

Time-Dependent Shortest Path Queries on Mobile Devices

by

Wei Yumou

Submitted to the School of Computer Science and Engineering
on 27 March 2017, in partial fulfillment of the
requirements for the degree of
Bachelor of Computer Science

Abstract

In this thesis, I designed and implemented a compiler which performs optimizations that reduce the number of low-level floating point operations necessary for a specific task; this involves the optimization of chains of floating point operations as well as the implementation of a "fixed" point data type that allows some floating point operations to simulated with integer arithmetic. The source language of the compiler is a subset of C, and the destination language is assembly language for a micro-floating point CPU. An instruction-level simulator of the CPU was written to allow testing of the code. A series of test pieces of codes was compiled, both with and without optimization, to determine how effective these optimizations were.

FYP Supervisor: Xiao Xiaokui
Title: Associate Professor, Assistant Chair (Strategic Research)

Acknowledgments

I would like to express my special thanks of gratitude to my FYP supervisor (Assoc Prof. Xiao Xiaokui) who gave me the golden opportunity to do this wonderful project on the topic (GPS Trajectory Mining), which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them. Secondly i would also like to thank my parents and friends who helped me a lot in finalising this project within the limited time frame.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

Finding a practically shortest route on a large road network in a metropolis is not only of algorithmic interests, but also of economic and environmental values. Less travelling time means less fuel consumptions and less carbon emissions. However, finding shortest routes can be challenging, especially when the road traffic is known to be *time-dependent* or *dynamic*, namely, when the road conditions change with respect to time. It may take 10 minutes on average to traverse a particular road at 10a.m, but it is possible that the expected travel time increases to 20 minutes at 5p.m. Moreover, two different roads may have different time-varying patterns. For instance, one road may have a peak traveling time at 12p.m but the other may have two peaks at 8 a.m. and 6 p.m., respectively.

The formal definition of a dynamic road network is given as follows.

Definition 1 (*Dynamic road network*)**.** A dynamic road network is a weighted, directed graph $G = (V, E)$ where E represents a set of road segments and V denotes the set of intersections of these road segments. It has a weight function $w : E, t \to \mathbb{R}$, where t represents an instant in time.

With the definition of a dynamic road network, the generalised time-dependent shortest path problem can be formally defined as follows.

Definition 2 (*Generalised time-dependent shortest path problem*)**.** In a dynamic road network $G = (V, E)$, given a source node u, a destination node v and a departure time t from u, find a path p that satisfies:

$$w(p) = \delta(u, v) = \begin{cases} \min\left\{w(p) : u \overset{p}{\rightsquigarrow} v\right\} & \text{if there is a path from } u \text{ to } v, \\ \infty & \text{otherwise.} \end{cases} \tag{1.1}$$

where $w(p)$ is the weight of the path p and defined as sum of the weights of its constituent edges, and $\delta(u, v)$ is called the **shortest-path weight** from u to v.

A typical Bellman-Ford[1] or Dijkstra's algorithm[2] for finding shortest paths assume the cost of traversing each edge in the abstract graph is constant with respect to time and therefore, do not work on time-dependent road networks without appropriate modifications. Fortunately, most online mapping services such as Google Maps or Apple Maps are able to recommend shortest routes by incorporating real-time traffic information. This project seeks to investigate an alternative approach of finding shortest routes on a dynamic road network based on mining a GPS[1] trajectory database aggregated from thousands of taxis in Beijing, China.

Chapter two defines several important concepts used in this project.

Chapter three gives an overview of the GPS-based approach.

Chapter four describes the data pre-processing and presents a novel neural network-based outliner detection method.

Chapter five elaborates the process of building a landmark graph and estimating the expected travel time of each landmark graph edge.

Chapter six introduces the method of evaluating the estimated tralvel time from

[1] Global Positioning System

Chapter five.

1.1 Motivations for mining taxi GPS trajectories

Taxi drivers or any experienced car drivers, more often than not, possess some *implicit* knowledge or intuitions about which route from source u to destination v is the best in terms of travelling time at a particular moment. Such knowledge or intuitions come from everyday experiences. For example, a taxi driver may observe that there are always traffic jams during 6 p.m. to 7 p.m. on a particular street and hence avoids travelling on that street during that period whenever possible. But observations of this kind, albeit valuable, are just too subtle to be captured by any general algorithms and oftentimes, even the drivers themselves may not be aware of that.

However, mining their GPS trajectories can reveal such knowledge to some extent. In a metropolis such as Beijing or New York, taxi drivers are required by regulations to install GPS devices on their cars and to send time-stamped GPS information to a central reporting agency periodically for management and security reasons. Such information typically includes latitudes, longitudes, instantaneous speeds and heading directions. Therefore, the GPS data is readily available and little effort is needed to collect it. By means of mapping to a real road network all GPS data points of a particular taxi during a specific period of time, a GPS trajectory can be obtained to represent the driver's intelligence.

1.2 Practical limitations

There are some practical limitations that are worth mentioning.

Arbitrary sources and destinations

In a typical map-query use case, a user is able to select an arbitrary source to start with and an arbitrary destination to go to. But this may not be possible in the GPS-based approach, since the taxi GPS trajectories do not necessarily cover every part of a city's map. It is likely that there are no trajectories passing through the source and the destination.

Low sampling rate

Taxis report their locations to the central reporting agency at a relatively low rate to conserve energy. For the data set used in this project, the expected sampling interval is one minute. But oftentimes, the GPS device may not be working properly or may be occasionally shut down due to various reasons, which causes the actual sampling interval to fluctuate.

Even if the sampling interval *is* strictly kept at one minute, for a taxi moving at a typical speed of $60km/h$, it means the distance between two consecutive sample points is $1km$. Such a large distance increases the uncertainty of the *exact* trajectory that the taxi has moved along. The picture[6] below demonstrates a problem caused by low sampling rate and long inter-sample-point distance.

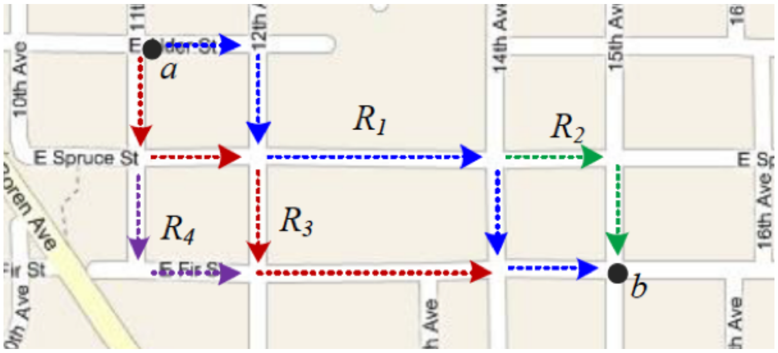

Figure 1-1: Example of low sampling rate problem

The taxi is known to have traversed from point a to point b. But there are four possible trajectories from a to b. The exact route cannot be determined without additional information in this case.

Limited GPS accuracy

After decades of development, the GPS service has achieved great accuracy, but it is still not completely error-free. A report[4] in 2015 showed that GPS-enabled smartphones typically have an accuracy of 5 metres *under open sky*. But in a metropolis like Beijing, the actual accuracy may be lower than this value due to the reflection of signals amongst high buildings. Moreover, the data set used in this project was collected in 2009 when GPS devices had lower accuracy than that of today's.

The limited accuracy in GPS devices makes the exact mapping from a GPS data point to a street impossible. In Beijing, there is usually a side road running in parallel with a main road. Due to that limited accuracy, the taxi might be *actually* on the side road but the GPS data point is shown on the main road, or vice versa.

1.3 Related work

The incentive for carrying out this project comes from a similar project[6], and similar procedures are followed in this project but with some modifications.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Preliminary Data Processing

2.1 Data Collecting and Cleaning

The taxi GPS data used in this project is collected from the Computational Sensing Lab[7] at Tsinghua University, Beijing, China. The data set contains approximately 83 million time-stamped taxi GPS records collected from 8,602 taxis in Beijing, from 1 May 2009 to 30 May 2009. The original data set consists of seven fields as shown in Table 2.1. The GPS coordinates in the data set are defined in the WGS-84[1] standard coordinate system, which is the reference coordinate system used by the GPS.

The original data set comes in a binary file format. After the data was decoded and imported into a MySQL database, the first step in cleaning data was to delete all records with zero value in the SPEED field, since when a taxi is stationary it yields no valuable information about the *trajectory* it is moving along. While being stationary could be due to a traffic jam, this kind of information is well captured by the time difference between the previous *non-stationary* data point and the next *non-stationary* data point.

[1] World Geodetic System

Field	**Explanation**
CUID	ID for each taxi
UNIX_EPOCH	Unix timestamp in milliseconds since 1 January 1970
GPS_LONG	Longitude encoded in WGS-84 multiplied by 10^5
GPS_LAT	Latitude encoded in WGS-84 multiplied by 10^5
HEAD	Heading direction in degrees with 0 denoting North
SPEED	Instantaneous speed in metres/second (m/s)
OCCUPIED	Binary indicator of whether the taxi is hired (1) or not (0)

Table 2.1: Fields in the original data set

Moreover, all records must have a *unique* pair of CUID and UNIX_EPOCH fields, since it is not possible for a taxi to appear in two different locations at the same moment in time. This is possibly due to some errors in aggregating the original data set.

2.2 Reverse Geocoding

After the data set was cleaned, the next step was to map each GPS data point to a road segment, which is also known as *reverse geocoding.* A number of algorithms[3] have been proposed for this purpose, but most of them require an additional GIS[2] database of the road network in Beijing. This project adopted an alternative strategy which leverages on the existing public API[3] for reverse geocoding.

Currently, a number of online mapping platforms provide reverse geocoding services as part of their developer APIs. Amongst others, Google Maps and Baidu Maps offer relatively stable and fast reverse geocoding services. However, due to the "China

[2] Geographic Information System

[3] Application Programming Interface

Field	**Explanation**
DataUnitID	Nominal primary key for each record
BD09_LONG	Longitude encoded in BD-09 format
BD09_LAT	Latitude encoded in BD-09 format
STREET	Street name

Table 2.2: Additional fields added to data set

GPS shift problem"[5] where coordinates encoded in WGS-84 standard are required by regulations to be shifted by a large and variable amount when displayed on a street map, Google Maps is not able to display a GPS data point correctly.

Baidu Maps, on the other hand, has been using their own coordinate system called BD-09 which is an improved version of the Chinese official coordinate system, GCJ-02. Baidu provides a set of APIs to convert WGS-84 coordinates into BD-09 ones. To store the converted coordinates as well as the street names obtained from reverse geocoding, four new fields were added to the original data set as shown in Table 2.2.

In order to use Baidu APIs for reverse geocoding, the following system architecture was set up as shown in Figure 2-1. The Apache HTTP server hides the MySQL database and sends HTTP POST request to Baidu Maps Web API to get converted coordinates. Then it updates the database through PHP *mysqli* utilitity.

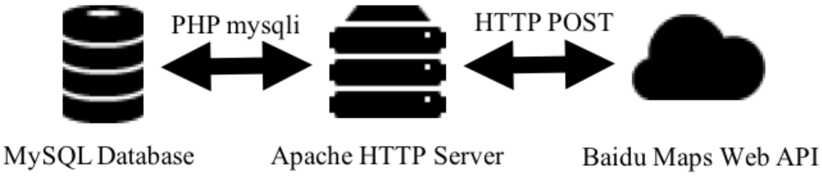

Figure 2-1: Basic system architecture

After the coordinates were converted from WGS-84 format to BD-09 format, Baidu

Maps Web API was used to reverse geocode all GPS data points. However, the system architecture was slightly changed, to accommodate the change in technology used. For reverse geocoding, AJAX[4] was used to communicate with the Baidu Maps Web API for speed and unlimited number of requests per day. Therefore, one additional layer was added to the existing system architecture as shown in Figure 2-2.

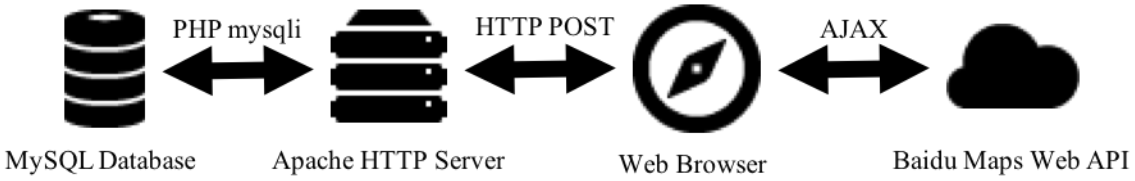

Figure 2-2: Augmented system architecture

Executed in a web browser environment, AJAX sent HTTP POST requests to the Apache HTTP server to fetch the converted coordinates in BD-09 format which were subsequently sent to the Baidu Maps Web API server *asynchronously* via HTTP GET requests. Once the server responded with the name of the road segment, AJAX updated the database by sending another HTTP POST request to the Apache HTTP server. The asynchronous nature of this architecture, however, caused a few problems which are addressed in Section 2.3.

2.3 Outlier detection

[4] Asynchronous JavaScript and XML

Appendix A

Tables

Table A.1: Armadillos

Armadillos	are
our	friends

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Figures

Figure B-1: Armadillo slaying lawyer.

Figure B-2: Armadillo eradicating national debt.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, third edition edition, 2009.

[2] E. W. Dijkstra. A note on two problems in connections with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[3] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *ACM SIGSPATIAL GIS 2009*. ACM SIGSPATIAL GIS 2009, November 2009.

[4] Frank van Diggelen and Per Enge. The world's first gps mooc and worldwide laboratory using smartphones. In *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, pages 361–369, Tampa, Florida, September 2015.

[5] Wikipedia. Restrictions on geographic data in china, March 2017.

[6] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: Driving directions based on taxi trajectories. ACM SIGSPATIAL GIS 2010, November 2010.

[7] Bing Zhu, Peter Huang, Leo Guibas, and Lin Zhang. Urban population migration pattern mining based on taxi trajectories. In *Mobile Sensing Workshop at CPSWeek 2013*, Philadelphia, April 2013.