AN ENERGY-SAVING APPROACH FOR REAL-TIME HIGHWAY TRAFFIC ESTIMATION USING GPS-ENABLED SMARTPHONES

A Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Daxiao Liu

December 2014

AN ENERGY-SAVING APPROACH FOR REAL-TIME HIGHWAY TRAFFIC ESTIMATION USING GPS-ENABLED SMARTPHONES

Daxiao Liu

APPROVED:

Albert M.K. Cheng
Dept. of Computer Science

Weidong(Larry) Shi
Dept. of Computer Science

Jingmei Qiu
Dept. of Mathematics

Dean, College of Natural Sciences and Mathematics

AN ENERGY-SAVING APPROACH FOR REAL-TIME HIGHWAY TRAFFIC ESTIMATION USING GPS-ENABLED SMARTPHONES

An Abstract of a Thesis
Presented to
the Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

By
Daxiao Liu
December 2014

Abstract

This paper presents a microscopic traffic estimation algorithm for smartphones by employing their built-in probes such as GPS and acceleration sensors to increase the accuracy of real-time traffic condition estimation without significantly increasing the smartphones' energy consumption. In this approach, real-time traffic data is collected through the smartphones of participating users traveling on urban roads. A new reporting algorithm is provided on the clients' side to minimize the amount of time the smartphone maintains connection to the server. Based on the data received from each individual smartphone, real-time traffic conditions and the level of service (LOS) are estimated on the server side by applying the Kalman Filtering algorithm and link aggregating speed algorithm. An iOS application is developed to work as a sample client side smartphone node. Simulations of three different traffic scenario are also created to evaluate the performance of the algorithm. Simulation results show that the proposed algorithm requires less energy usage than existing methods without sacrificing the accuracy of real-time traffic estimations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

As the number of vehicles on the road increases, real-time traffic information can be utilized to help drivers reduce travel time by avoiding congested areas [10, 11, 12, 13]. Among numerous methods for collecting real-time traffic information, using smartphone probes has been in the forefront in the past five years. Compared with traditional methods which employ fixed-location sensing infrastructures, such as inductive loop detectors, radars, and video cameras, using GPS-enabled smartphones has decreased cost and increased scalability. Compared with other new technologies which use floating traffic probes, such as LiDAR and Dedicated Vehicle Probe (PVs) in taxi and public buses, using GPS-enabled smartphones has higher feasibility on the market and lower requirement on equipment installation and maintenance. Realizing this, traffic engineers have started to explore this area [4, 5, 8, 9, 15, 19].

Lovell [15] studied the accuracy of speed measurement from cell phones, Herrera [8] went a step further and collected 8 hours of data on a 10-mile stretch of I-880 and found that a 2-3% penetration of cell phones in the driver population is enough to provide accurate measurements of the velocity of the traffic flow. Kalman Filtering (KF) has widely been adopted as a traffic estimation model [4]. Meanwhile the UC-Berkeley wireless group released the Mobile Millennium [17] to monitor Northern California area based on smartphone probes. Waze [21], a successful commercial mobile application, provided turn-by-turn navigation and also collected real-time traffic information. However, when running Mobile Millennium and Waze on smartphones, a major concern is their increased energy consumption. Smartphones have limited battery and limited 4G network coverage. From the users' perspectives, energy consumption and 4G network usage are important factors determining the practicality of the traffic estimation.

Based on Kalman Filtering [1, 4, 16], this thesis proposes a new energy-saving algorithm to address these issues. To accurately measure the power consumption of smartphones, Zhang et al [23] described a Regression-Based Finite Power State approach. For each category such as CPU, Wi-Fi, Audio, LCD, GPS, and Cellular, a corresponding coefficient was estimated for different states. One remarkable observation made by Zhang et al. is that given fixed channel and packet rates, the packet size did not influence power consumption. Pathak et al. [18] also adopted a Finite Power State Model, but their emphasis was more from a system's perspective which included adding coefficients for file-read, file-save, and so on. Cui et al. [2] presented a survey on current energy efficient modeling in mobile cloud computing. In his

Wi-Fi power management section, he summarized that "the main idea of reducing transmission energy is to keep the WNIC in low power state (sleep state for Wi-Fi or idle state for cellular) as long as possible" and "Burst is better than scattered data flow". Our proposed algorithm meets Cui's recommendations by sending more burst data flow than sending scattered data flow. In section 1.2, this thesis reviews the traditional traffic flow model and traffic estimation method Kalman Filtering. In section 1.3, this thesis summarizes the widely accepted power consumption methods and equations of smartphones.

1.2 Traffic Estimation

1.2.1 Traffic Flow Model and Numerical Discretization

As implied in the previous section, traditional traffic estimation methods can be categorized as two different approaches. One approach adopts Kalman Filtering. This includes some variations such as Ensemble Kalman Filtering. The other approach employs a statistical perspective to track and predict traffic conditions [5, 12, 13]. While the Kalman FiItering approach makes the basic assumption that the moving vehicle's velocity undergoes perturbations which can be modeled by zero-mean Gaussian noise [4], the statistical learning approach emphasizes achieving the maximum probability of each vehicle's driving trajectory to estimate and predict the traffic conditions. Although the statistical learning approach might perform better than the Kalman Filtering approach on predicting the future traffic condition, it also generates a larger computational burden on the server. In this thesis, the author adopts Kalman Filtering as the traffic estimation model.

Traditional highway traffic theory uses Lighthill-Whiteham-Richards (LWR) [14] to model traffic on a highway as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{1.1}$$

where $q(x,t)$ and $\rho(x,t)$ respectively denote the flow of vehicles and their density at location x and time t [3]. Additionally, the velocity field on the highway is denoted as $v(x,t)$. Equation 1.1 expresses the conservation of mass for a fluid density ρ and of flux q, and is considered relevant to modeling traffic on a highway [22]. In traffic theory, traffic flow and traffic density have the empirical relation shown in Equation 1.2:

$$q(x,t) = Q(\rho(x,t)) \tag{1.2}$$

where Q is the flux function which is assumed to be independent of time and space. One widely accepted flux function is Greenshields flux function [6] which expresses a linear function between ρ and v as:

$$v = v_{max}(1 - \frac{\rho}{\rho_{max}}) \tag{1.3}$$

In Equation 1.3, v_{max} and ρ_{max} respectively denotes the maximal velocity and the maximal density allowed by the model. After inserting Equation 1.3 into Equation 1.1, one can rewrite the partial differential equation of LWR on density as a partial differential equation of LWR on velocity as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(R(v)) = 0 \tag{1.4}$$

where $R(v) = (v)^2 - v_{max}v$. Equation 1.4 provides a relationship between traffic density and velocity with respect to continuous flow state. In the author's research, a Godunov numerical scheme is used to numerically discretize current equation for practical use [15]. This scheme is also employed by the famous velocity cell transmission model (CTM-v) [14]. Given M space cells $I_i(0 \le i \le M)$ of length $\Delta x = \frac{b-a}{M}$ and N time steps $J_n(0 \le n \le N)$, we call v_i^n a discrete value of v on $I_i * J_n$. Based on the Godunov scheme, at each tilmestep v_i^{n+1} is computed from the previous time step by the following formula:

$$v_i^{n+1} = v_i^n - \frac{\Delta t}{\Delta x}(g(v_i^n, v_{i+1}^n) - g(v_{i-1}^n, v_i^n)) \tag{1.5}$$

where the numerical flow g is defined as follows:

$$g(v_1, v_2) = \begin{cases} R(v_2) & \text{if } v_1 \le v_2 \le v_c \\ R(v_c) & \text{if } v_1 \le v_c \le v_2 \\ R(v_1) & \text{if } v_c \le v_1 \le v_2 \\ max(R(v_1), R(v_2)) & \text{if } v_1 \ge v_2 \end{cases} \tag{1.6}$$

with v_c defined as the minimum of the convex flux function from the Equation 1.4. As implied by Equation 1.5 and 1.6, this model is a nonlinear dynamic system[7, 22]. Defining the state of the system $v^n = [v_0^n, v_1^n, ..., v_m^n]$ as the vector of velocities in all cells at timestep n, the state dynamics of the system can be written as

$$v^{n+1} = M[v^n] + \eta^n \tag{1.7}$$

In Equation 1.7, the η^n is the modeling error.

1.2.2 Kalman Filtering

Kalman Filtering is an algorithm that uses a series of measurements observed over time. This contains noise (random variations) and other inaccuracies, and produces estimates of unknown variables [16]. The algorithm contains two steps: prediction step and update step. Before the first step, an initialization distribution is needed to start the algorithm. In the author's research, a prior Gaussian distribution is assumed to mean the velocity of the highways is smooth and no shocks exists along the highway. Then a prediction of $\hat{v}_k^n$ of v_k^n is made from Equation 1.7. The mean is obtained through

$$v^n = \frac{1}{K}\sum_{k=1}^{K} \hat{v}_k^n \tag{1.8}$$

and the covariance matrix P of the predicted state is computed as:

$$P^n = \frac{1}{K-1} E^n (E^n)^T \tag{1.9}$$

where matrix E^n is defined as:

$$E^n = [\hat{v}_1^n - v^n, ..., \hat{v}_k^n - v^n] \tag{1.10}$$

Then the updated matrix G^n can be computed from the following equation:

$$G^n = P^n (H^n)^T [H^n P^n (H^n)^T + R^n]^{-1} \tag{1.11}$$

Finally, we get the updated v_k^n

$$v_k^n = \hat{v}_k^n + G^n [y^n - H^n \hat{v}_k^n + \xi_k^n] \tag{1.12}$$

where H^n is the discrete cells on the highway and ξ^n is the Gaussian observation noise [17, 22]. In the author's research, ξ^n accounts for the GPS position and speed error. Although the author conducted some data mapping and processing, those errors still exists.

1.3 Energy Consumption

Researchers have proposed numerous power models for portable embedded systems such as smartphones [2, 23, 24]. Two important network parameters to derive the model are data rate and channel rate. The model is derived by exchanging the fixed TCP packets between the smartphone and the server. In Zhang's paper [23], two power consumption equations are derived, shown in Equation 1.13 and Equation 1.14.

$$\beta_{WiFi_h} = 710mW + \beta_{cr}(R_{Channel}) * R_{data} \tag{1.13}$$

where $\beta_{cr}(R_{Channel})$ is calculated in Equation 1.14;

$$\beta_{cr}(R_{Channel}) = 48 - 0.768 * R_{Channel} \tag{1.14}$$

Here, $R_{Channel}$ refers to uplink channel rate. It varied as 11Mbps, 36Mbps, 48Mbps, and 54Mbps. During the experiments, the author assumes that the channel rate stayed the same.

Chapter 2

System Overview

The main components of the system are GPS-enabled smartphones as the clients and a server. At the client side, an iOS application is developed to record vehicles' driving speed, current location, and driving directions. At the server side, the traffic estimation algorithm is running in order to reconstruct the real-time traffic condition from the gathered real-time traffic data. Due to limited resources, a simulation-based framework is created on the server side to evaluate the algorithm's performance.

2.1 Server Side

2.1.1 Simulation-based Framework

A Simulation-based framework is designed to emulate the smartphone-based real-time urban traffic estimation shown in Figure 2.1. The framework consists of three parts: traffic simulation generator, traffic estimation evaluation, and energy consumption estimation. Traffic simulation generator is designed to simulate the urban road network and the corresponding traffics. It generates a large numbers of vehicles at a given time and provides us the "ground truth" traffic states [19]. According to the predefined penetration rate, 5% - 7% of vehicles of generated vehicles are chosen as GPS-enabled smartphone probes. The penetration rate is predefined by the author based on previous recommendation [9, 20]. At the second step, the Kalman Filtering (KF) is implemented to track each vehicle and filter out undesired position/ speed estimates. After that, the real-time traffic is estimated by the calculated average speed. Finally, with a threshold technique, we use the estimated average link speeds to provide the estimated level of traffic service (LOS). Different traffic condition levels are presented as different colors on road segments map.

2.1.2 Traffic Modeling

The microscopic road traffic simulation package "Simulation of Urban Mobility" (SUMO) is employed to model the real-time traffic on the urban road network. To generate traffic, we use an open source map website called OpenStreetMap (OSM). OSM provides SUMO map data; next, we used two functions provided by SUMO NETCONVERT and DFROUTER. NETCONVERT imports a map layer which contains all points and edges information into the SUMO format. DFROUTER generates random routes and emits vehicles into the network. Even though 30 percent of random routes generated by DFROUTER would cause an exception when you do traffic assignments, DFROUTER is still a very useful tool providing random routes.

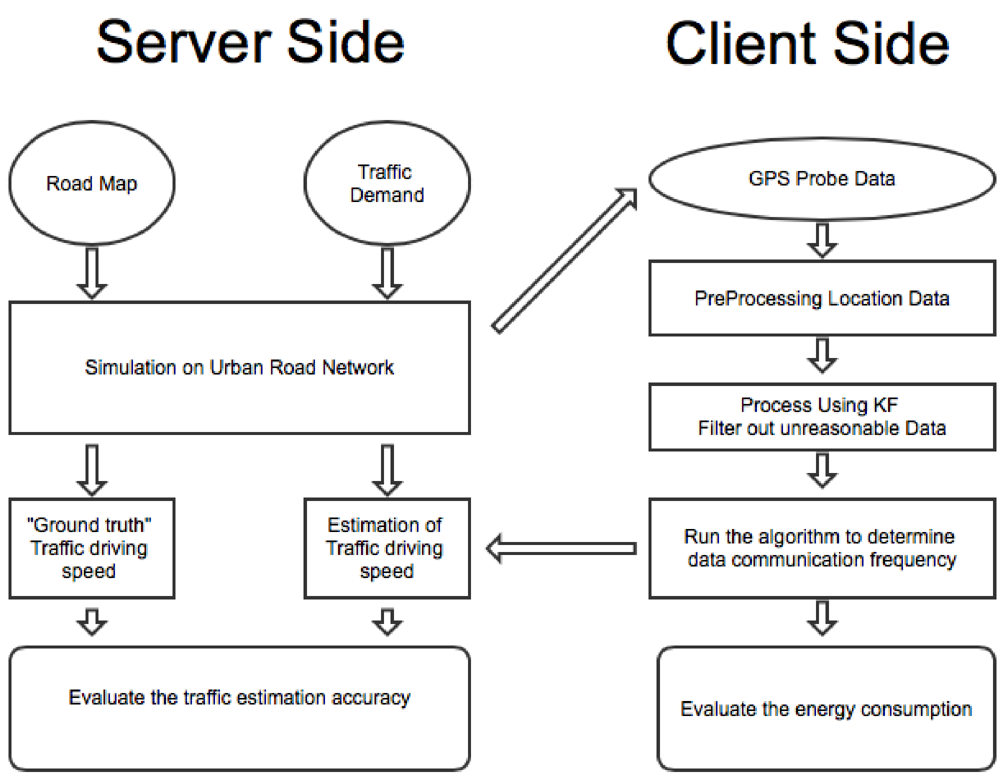

Figure 2.1: System Framework Overview

As a conclusion, SUMO generated two datasets. One dataset is the "ground truth" traffic condition. The other dataset consists of individual vehicles which is used as GPS probe to estimate the real-time traffic information.

2.1.3 System Design Parameters

From related works, two key factors that might influence the system performance are sample size and the sampling frequency [20]. These sampling-related issues have been discussed extensively in many previous studies. For the penetration rate, similar conclusion were drawn from previous tests [9, 19, 20]. Probe-based system could be expected to work well for freeways with penetration rates ranging from 3%-5%. Another issue with probe-based systems is the sampling frequency. Typical GPS receivers receive location updates roughly every 1-3 seconds. This frequency of data collection would cause network congestions. Tao proposed a sampling interval of 10-20 seconds [19] as a reasonable and effective sampling frequency. In the author's research, fixed sampling frequency is not a necessity. This paper intends to use an algorithm to determine its sampling frequency. That will be discussed in Chapter 3.

2.2 Client Side

In general, any GPS-enabled portable electronic device could be considered as a client. To simplify the research, the author only develops an iOS-based smartphone application with a successful version on iOS 7. Multiple GPS-enabled smartphones which are using an Android system don't belong to our research case in this thesis. The design of this application includes two parts: user interface and back-end database. The basic principle of designing the user interface is simplicity and clarity. In this project, the application is constantly running on an iPhone to collect traffic data and receive push-up notices. From a user's view, it is simple, as no extra operations are needed. The main screen is a map which periodically updates your current location. The user may also view setting options and change settings after clicking the information icon on the top menu bar. The other part is the back-end

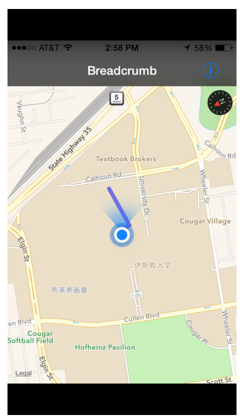

Figure 2.2: Application View

database. The author is adopting Apple self-provided core data to serve as a back-end database. Core data organizes data as a relational entity attribute model, which is able to serialize data into XML, binary, or SQLite on demand. As it is provided by Apple, the performance is stable. Detail documentation allows easy integration and it meets current requirements. All communication between the server database and the client is through with the standard JSON format. JSON, also known as

JavaScript Object Notation, is a widely used open standard format to transmit data objects consisting of attribute-value pairs. Similarly, like XML, it is used primarily to transmit data between a server and web applications. There is a sample JSON data shown in Figure 2.3. In this JSON data, UserDefinedID is the identification number


```
{
  "User_DefinedID": "19230311",
  "Timestamp": "1415320922"
  "Time Interval": "20",
  "Location": "29.6840229, -95.3842392, N29° 41.0414', W095° 23.0544'",
  "Address": {
    "streetAddress": "2750 Holly Hall Drive",
    "city": "Houston",
    "state": "TX",
    "postalCode": "77054-1300"
  },
  "Speed limit": 35.00,
  "Driving Speed": 21.63,
  "Standard deviation": 0.00
  "Direction": North

}
```


Figure 2.3: JSON Sample Representation

each smartphone sends to the server to identify its identity. It is an 8-digit number generated by a random number generator, Timestamp is a current time following the unix format; time interval tells the server the sampling frequency in seconds; location is the coordinates from the GPS; the address is obtained after reverse geocoding the coordinates; speed limit is derived from analyzing the address; driving speed is calculated by dividing the driving distance with the time interval; and direction is obtained after analyzing coordinates changes. Driving speed and standard deviation is described in Chapter 3.

Chapter 3

Methods

3.1 Traffic Generation

In order to simulate traffic on the server, a successful configuration of SUMO (simulation of urban mobility) is required. An overview of input files of SUMO network is illustrated in Figure 3.1. Node file, Link file, Link property file, and Lane connection file provide SUMO the background map framework. The traffic demand file provides SUMO the traffic information. In the author's research, the map information is obtained through the open source website OpenStreetMap; the traffic demand is generated by the DFROUTER, provided by SUMO and enhanced by author's python code. The author designed three different traffic scenario: peak time without accident and non-peak time with and without accident. In each traffic scenario, different numbers of vehicles are generated following a Gaussian distribution.

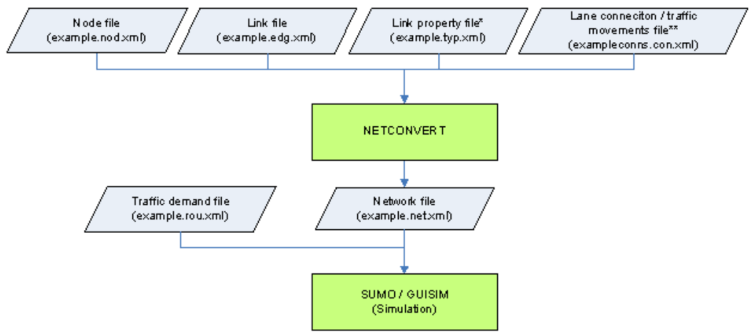

Figure 3.1: Required Input Files to Configure a Network in SUMO

3.2 Probe Sampling Interval Algorithm

As mentioned in Section 2.1.3, sampling frequency may affect the system performance. In the author's research, an algorithm is proposed to determine the frequency dynamically. The pseudo code is listed as follows:

In Line 2, timeInterval refers to the shortest sampling time frequency. In the author's algorithm, this procedure is run once every 20 second. In Line 3, reverse geocoding is a built-in function of the Xcode Library. Its functionality is to get its location information from a geographical coordinates. Sometimes, due to GPS's accuracy issue or systematic error, an extra data mapping procedure is required to pinpoint the location. Due to those coordinates being simulation-generated, this type of error dramatically decreased. In Line 5, a queue is used to store the speed and location information data for recent previous iterations. After the queue reaches its capacity, Lines 6 to 15 automatically calculate the mean driving speed and the standard deviation. In Line 9, the smartphone determines whether or not to send

Algorithm 1 Probe Sampling Interval algorithm

1: **procedure** MyProcedure
2:     **for** `each timeInterval` **do**
3:         *Call Reverse Geocoding*
4:         $V_{speed} \leftarrow$ *vehicle driving speed*
5:         *queue pushes* V_{speed}
6:         **if** *queue achieves its capacity* **then**
7:             $V_{mean} \leftarrow \frac{1}{n} \sum V$
8:             $\Theta_{sd} \leftarrow \frac{1}{n^2} \sum (V_{mean} - V)^2$
9:             **if** $(V_{mean} \leq \frac{3}{4} V_{speedLimit})$ *or* $(\Theta_{sd} \geq \frac{1}{6} V_{speedLimit})$ **then**
10:                 *Send stored data to server*
11:                 *Empty the queue*
12:             **end if**
13:         **else**
14:             *Update the queue*
15:         **end if**
16:     **end for**
17: **end procedure**

these stored traffic information to the server. If the mean driving speed limit is less than 75% of speed limit or if the standard deviation of the driving speed changes dramatically, the smartphone sends the stored traffic data to the server. Both conditions imply some traffic change. Otherwise, in Line 14, the algorithm keeps updating the queue.

Chapter 4

Results

4.1 Highway US59 Test Study

In order to test this algorithm, a total of 10.3 miles of Houston urban road is chosen as a case study as shown in Figure 4.1. The road is divided into 7 links. This includes local streets with a speed limit of 35 mph, a highway entry with meters, a highway exit with a speed limit of 35 mph, and four highway US59 segments with a speed limit of 60 mph.

The map network file is obtained from OpenStreetMap. For simplicity, highways and location streets are set to be one-way. Also, different traffic scenarios such as morning rush hours, noon rush hours, and non-rush hours are employed to evaluate the algorithm's performance. In order to achieve specific traffic equilibrium in different traffic conditions, different number of vehicles are generated with random trips.

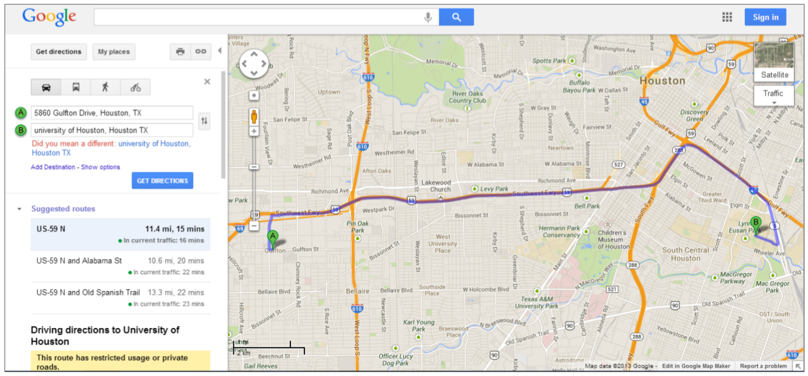

Figure 4.1: A Case Study Overview Map

These trips are generated using the author's code together with SUMO's procedure DFROUTER. The objective of author's code is to clean up some random unreachable travel created by DFROUTER. Among the generated vehicles, a 5% penetration rate is applied. After estimating the average link speed on the server side, the traffic level of service (LOS) estimation is compared with the actual average link speeds in a one-hour frame. We classified the estimated traveling speed into four different traffic condition LOS, based on the Highway Capacity Manual (HCM). Level A is smooth traffic if the highway link speed is above 50 mph or the local street link speed is higher than 30 mph; level B is good traffic if the highway link speed is between 40 mph and 50 mph or the local street link speed is between 20 mph and 30 mph; level C is medium traffic if the highway link speed is between 30 mph and 40 mph or the local street link speed is between 10 mph and 20 mph; and level D is congested traffic if the highway link speed is less than 30 mph or the local street link speed is less than 10 mph. The estimation result is shown in the Table 4.1.

Table 4.1 shows that the proposed algorithm has a reasonable agreement from this case study with a 90% successful estimation of the level of traffic services. As for the two error estimations, the estimated link speed is close to the actual link speed with an average 5 mph difference. One thing worth noting is that in Table 4.1 at Link 7, there is one N/A shown as the estimated link speed. The reason for this is that as all probe vehicles exiting the highway are driving at a high speed and not much data was sent to the server. In such a case, the server makes an assumption that there is smooth traffic on that link.

Link—Mile	Description	Actual —Estimated Average Link Speeds; Actual LOS — Estimated LOS		
		MorningRush (7 am)	Noon Rush (12 pm)	Normal (10 am)
#1 — 0.8	Renwick(L)	18.57 — 15.74; C — C	**12.47 — 8.38; C — D**	25.72 — 26.80; B — B
#2 — 0.5	US59 Entry(L)	12.24 — 10.78; C — C	8.34 — 6. 80; D — D	**20.66 — 14.27; B — C**
#3 — 1.8	US59 & I-610(H)	45. 34 — 46.7; B — B	37.22 — 38.36; C — C	57.71 — 58.44; A — A
#4 — 3.5	US59(H)	32.41 — 33.72; C — C	41.56 — 40.73; B — B	58.35 — 59.21; A — A
#5 — 1.2	US59 & I-45(H)	37.24 — 36.83; C — C	20.11 — 28.36; D — D	54.63 — 56.77; A — A
#6 — 1.8	US59(H)	46.11 — 47.20; B — B	43.42 — 40.18; B — B	57.78 — 58.73; A — A
#7 — 0.6	US59 Exit(L)	34.85 — 32.15; A — A	34.56 — 33.71; A — A	35.21 — N/A; A — A

Table 4.1: US59 Actual Link Speed vs Estimated Link Speed

Algorithm	Traffic Data file Size (Kb)	Cellular and data (Kb)	Frequency (times/hr)	Energy consumption (excluding CPU, LCD Audio) Unit(mW)	Percentage
Previous	4	10	20 sec/time = 180	1573.46	100.00%
Proposed (worst case)	12	18	2 min/time = 30	1224.76	77.84%
Proposed (Average)	12	18	Rush time: 28 Normal time: 14	1190.44 1136.20	75.86% 72.21%

Table 4.2: US59 Energy Consumption Comparison

In this simulation, each smartphone updates the estimated level of traffic service every five minutes. This part of the energy consumption is not included in our simulation results. The simulation result shown good performance of our approach. Only 2 of 21 estimations are inaccurate and the energy consumption is only 70%, compared with traditional algorithms at non-rush hours, and about 78% at rush hours.

4.2 Highway TX288 Test Study

In order to achieve a more convincing conclusion, a total of 13.8 miles of Houston urban road is chosen as another case study as shown in Figure 4.2. The road is divided into 4 links. This includes local streets with a speed limit of 35 mph, and three highway TX288 segments with a speed limit of 60 mph. TX288 is one of the most congested highways in Houston area.

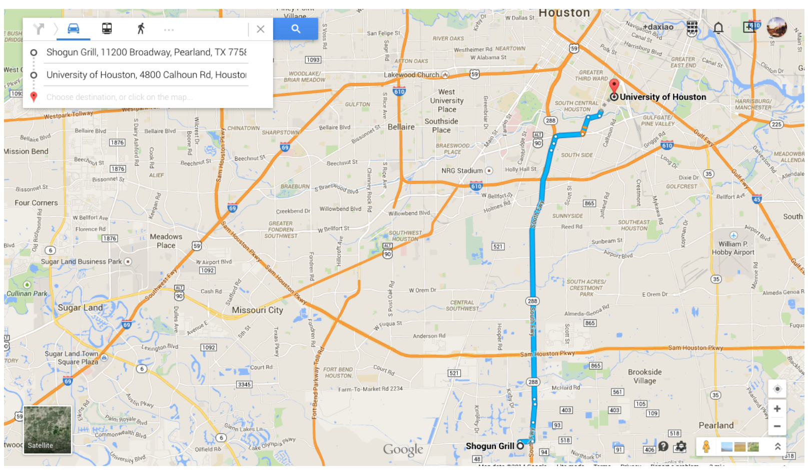

Figure 4.2: Another Case Study Overview Map

Different traffic scenarios such as morning rush hours, noon rush hours, and non-rush hours are employed to evaluate the algorithm's performance. After estimating the average link speed on the server side, the traffic LOS estimation is compared with the actual average link speeds in a one-hour frame. The estimation result is shown in the Table 4.3.

Link—Mile	Description	Actual —Estimated Average Link Speeds; Actual LOS — Estimated LOS		
		MorningRush (7 am)	Noon Rush (12 pm)	Normal (10 am)
#1 — 3.4	TX288 (H)	32.80 — 34.76; C — C	22.18 — 22.46; D — D	**52.53 — 46.77; A — B**
#2 — 6.6	TX288 & Sam Houston(H)	25.16 — 21.09; D — D	12.15 — 11. 50; D — D	**50.47 — 47.82; A — B**
#3 — 2.8	TX288 & I-610(H)	36.88 — 32.55; C — C	8.73 — 9.01; D — D	53.70 — 56.46; A — A
#4 — 1.1	Old Spanish(L)	11.13 — 12.49; C — C	9.36 — 9.52; D — D	21.13 — 22.49; B — B

Table 4.3: TX288 Actual Link Speed vs Estimated Link Speed

Table 4.3 shows that the proposed algorithm has a reasonable agreement from this case study with a 87.5% successful estimation of the level of traffic services. As for the two error estimations, comparing with the actual link speed, the estimated link speed is 5 mph less.

Algorithm	Traffic Data file Size (Kb)	Cellular and data (Kb)	Frequency (times/hr)	Energy consumption (excluding CPU, LCD Audio) Unit(mW)	Percentage
Previous	2	5	20 sec/time = 180	1556.85	100.00%
Proposed (worst case)	5	12	2 min/time = 30	1483.99	95.32%
Proposed (Average)	5	12	Rush time: 28 Normal time: 14	1290.44 1236.20	82.86% 79.39%

Table 4.4: TX288 Energy Consumption Comparison

In this simulation, each smartphone updates the estimated level of traffic service every five minutes. This part of the energy consumption is not included in our simulation results. The simulation result also show good performance of our approach. Only 2 of 16 estimations are inaccurate, and the energy consumption is only 80% compared with traditional algorithms at non-rush hour and about 95.32% at rush hour. At rush hour, the proposed algorithm's performance is similar to traditional algorithms.

Chapter 5

Conclusions

This paper presents a energy-saving approach for microscopic traffic estimation using GPS-enabled smartphones. The simulation results show that the accuracy of real-time traffic has not been sacrificed and the energy consumption is around 80% of the energy consumption compared with traditional algorithm. An iOS application is also developed for real experiments. Future work will consider evaluating this algorithm's performance on applying a different traffic estimation algorithm such as a statistical learning approach other than current employed Kalman Filtering approach. Also, experiments with smartphones-equipped real vehicles are being planned to test our algorithm.

Bibliography

[1] Burgers, G., Jan van Leeuwen, P., & Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. Monthly Weather Review, 126(6), 1719-1724.

[2] Cui, Yong, Xiao Ma, Hongyi Wang, Ivan Stojmenovic, & Jiangchuan Liu. (2013). A survey of energy efficient wireless transmission and modeling in mobile cloud computing. Mobile Networks and Applications 18, 1, 148-155.

[3] Daganzo, C. F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological, 28(4), 269-287.

[4] Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research: Oceans (1978-2012), 99(C5), 10143-10162.

[5] Greenshields, B. D., Channing, W., & Miller, H. (1935). A study of traffic capacity. In Highway research board proceedings (Vol. 1935). National Research Council (USA), Highway Research Board.

[6] Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Matematicheskii Sbornik, 89(3), 271-306.

[7] Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., Bayen, & A. M. (2010). Evaluation of traffic data obtained via GPS- enabled mobile phones: The Mobile Century field experiment. Transportation Research Part C: Emerging Technologies, 18(4), 568-583.

[8] Liu, D., & Cheng , A.. (2013). An energy-saving approach for real-time highway traffic estimation using gps-enabled smartphones (2013). IEEE Real-Time Systems Symposium, Work-in-Progress Proceedings.

[9] Liu, K., Lim, H. B., Frazzoli, E., Ji, H., & Lee, V. (2014). Improving positioning accuracy using GPS pseudorange measurements for cooperative vehicular localization. IEEE Transactions on Vehicular Technology, 63(6), pp. 2544-2556.

[10] Liu, K., Lee, V., Ng, J. K., Son, S. H., & Sha, E. H. M. (2014). Scheduling temporal data with dynamic snapshot consistency requirement in vehicular cyber-physical systems. ACM Transactions on Embedded Computing Systems (TECS), 13(5s), 163.

[11] Liu, K., Chan, E., Lee, V., Kapitanova, K., & Son, S. H. (2013). Design and evaluation of token-based reservation for a roadway system. Transportation Research Part C: Emerging Technologies, 26, 184-202.

[12] Liu, K., & Lee, V. (2012). Adaptive data dissemination for time-constrained messages in dynamic vehicular networks. Transportation research part C: emerging technologies, 21(1), 214-229.

[13] Lighthill, M. J., & Whitham, G. B. (1955). On kinematic waves. II. A theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 229(1178), 317-345.

[14] Lovell, D. J. (2001). Accuracy of speed measurements from cellular phone vehicle location systems, Journal of Intelligent Transportation Systems, 6(4), 303-325.

[15] Kalman Filtering, Wikipedia, http://en.wikipedia.org/wiki/Kalman_filter

[16] Mobile Millennium, http://traffic.berkeley.edu/

[17] Pathak, A., Hu, Y. C., Zhang, M., Bahl, P., Wang, & Y. M. (2011). Fine-grained power modeling for smartphones using system call tracing. In Proceedings of the Sixth Conference on Computer systems ACM 153-168.

[18] Tao, S., Manolopoulos, V., Rodriguez Duenas, & S., Rusu, A. (2012). Real-time urban traffic state estimation with A-GPS mobile phones as probes. Journal of Transportation Technologies, 2(01), 22.

[19] Wang, Y., Papageorgiou, M., & Messmer, A. 2008. Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing. Transportation Research Part A: Policy and Practice, 42(10), 1340-1358.

[20] Waze software, Wikipedia, http://en.wikipedia.org/wiki/Waze

[21] Work, D. B., Tossavainen, O. P., Blandin, S., Bayen, A. M., Iwuchukwu, T., & Tracton, K. (2008). An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In Decision and Control, 47th IEEE Conference on (pp. 5062-5068). IEEE

[22] Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R. P., Mao, Z. M., & Yang, L. (2010). Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In Proceedings of the Eighth IEEE/ACM/IFIP International conference on Hardware/Software Code Sign and System Synthesis (pp. 105-114). ACM.

[23] Zhang, X., Onieva, E., Perallos, A., Osaba, E., & Lee, V. (2014). Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction. Transportation Research Part C: Emerging Technologies, 43, 127-142.