The following publication Z. He, D. Zhang, J. Cao, X. Liu, X. Fan and C. Xu, "Exploiting Real-Time Traffic Light Scheduling with Taxi Traces," 2016 45th International Conference on Parallel Processing (ICPP), Philadelphia, PA, USA, 2016, pp. 314-323 is available at https://doi.org/10.1109/ICPP.2016.43.

Exploiting Real-time Traffic Light Scheduling with Taxi Traces

Zongjian He, Daqiang Zhang
School of Software Engineering
Tongji University
Shanghai, China
E-mail: {hezongjian, dqzhang}@tongji.edu.cn

Xiaopeng Fan, Chengzhong Xu
Institute of Advanced Technology
Chinese Academy of Sciences
Shenzhen, China
E-mail: {xp.fan, cz.xu}@siat.ac.cn

Jiannong Cao, Xuefeng Liu
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
E-mail: {csjcao, csxfliu}@comp.polyu.edu.hk

***Abstract*—Traffic lights in urban area can significantly influence the efficiency and effectiveness of transportation. The real-time scheduling information of traffic lights is fundamentally important for many intelligent transportation applications, such as shortest-time navigation and green driving advisory. However, existing traffic light scheduling identification systems either entail dedicated infrastructures or depend on specialized traffic traces, which hinders the popularity and real world deployment. Differently, we propose to identify real-time traffic light scheduling by analyzing taxi traces that are widely accessible from taxi companies. The key idea is to exploit the periodicity in traffic patterns, which is directly affected by traffic lights. We also develop advanced algorithms to identify red/green lights duration and signal change time. We evaluate our solution using over one billion taxi records from Shenzhen, China. The evaluation results validate the effectiveness of our system.**

***Index Terms*—data analysis; intelligent traffic; traffic light; signal processing;**

I. Introduction

Public traffic has become an essential problem for many urban cities around the world. A huge amount of time and money has been wasted by traffic congestions. According to the US urban mobility report, a total financial cost of $121 billion are wasted by bad traffic in 2012 and it is keep increasing [1]. Meanwhile, vehicle emissions also constitute the majority of the air pollution [2]. Therefore, providing better traffic control is of great importance for both economic and environmental concerns.

Currently, the primary approach to control the traffic flow is through traffic lights, which allows vehicles pass the road intersections alternatively. Therefore, the scheduling of traffic light significantly influences the efficiency and effectiveness of public traffic.

To improve the traffic light scheduling, adaptive traffic light control systems [3] have been proposed. However, upgrading current traffic lights requires huge amount of cost and effort. We argue that rather than controlling the traffic lights, just knowing the real-time scheduling of them can potentially bring lots of benefits for both individuals and communities. For individuals, current navigation systems can utilize the information to bypass red light ahead. Optimal suggestions can also be provided to drivers to pass the intersections smoothly [4] [5]. For communities, the transportation researchers can investigate the correlation between traffic light scheduling and traffic flow, and then make optimization accordingly. The promising autonomous driving technique like Google driverless car can also be enhanced by knowing the real-time traffic light scheduling [6].

The key technical hurdle of realizing those systems is how to obtain the real-time traffic light scheduling information. The real-time status of the traffic lights can not be directly accessible in US and other countries [7]. Alternatively, researchers have proposed to utilize vehicular networks to communicate between vehicles and traffic lights to obtain such information [8] [9] [10]. However, the heavy infrastructures costs including network adapter installation and software upgrade hinder the practicability of the approach. In addition, vision based approaches utilize the car-mounted cameras or smartphone cameras and image processing technique to obtain current traffic light status [6] [11]. But this approach is limited to line-of-sight range and may get interfered by the environment like weather and light. Recently, crowdsourcing based approaches have attracted much attentions due to the low data collection cost. Sensory data such as acceleration and GPS trajectories from car-borne sensors or in-vehicle smartphones can effectively be utilized to estimate traffic light scheduling. Unfortunately, existing approaches either rely on high frequency sampling (e.g., 1HZ), which may raise user privacy and network overhead concerns [5] [12] [13], or depend on statically routed buses [14], which may be restricted by the temporal and spatial coverage of the trajectories.

In this work, we propose to identify the real-time traffic light scheduling information by analyzing crowdsourced taxi GPS traces. The advantages of using taxi traces include the widely data availability without installing extra devices, and the broad coverage of the taxi trajectories. However, the analysis of taxi traces entails several challenges. First of all, the taxi traces are usually updated at a low sampling rate (e.g., once per 30 seconds). It is impossible to identify events such as acceleration or deceleration for a specific vehicle. Hence, existing approaches designed for high frequency sampling data can not be directly employed. Next, due to the urban driving scenarios, the GPS data inevitably incur localization errors for up to 100 meters [15], which prevents us from identifying the sequence of vehicles in a waiting queue. In addition, the movement of a taxi is also affected by other context. E.g., the stochastic on and off of passengers, ambient vehicles, etc. These interferences also need to be carefully handled. Besides, not all traffic lights have fixed scheduling, some of them are adaptively adjusted according to busy/free hour. It is essential to identify the traffic light scheduling changes. Finally, taxi traces are not uniformly distributed for all city regions at all time. The unbalanced data further increase the difficulty of data analysis.

To tackle these challenges, we propose a data analytic approach for real-time traffic light scheduling identification. The key idea is to exploit the *periodicity* of traffic patterns and reveal its correlation with traffic lights. To the best of our knowledge, this is the first traffic light scheduling identification solution for low-frequency, irregular, and unbalanced traffic traces. We are also the first to investigate the scheduling change of traffic lights. Our key contributions can be summarized as follows:

- We perform data analysis to the taxi traces, and figure out several fundamental statistical characteristics and patterns. These characteristics not only inspire our solutions, but also can be potentially used by other similar traffic analysis applications.
- We design novel algorithms to identify the scheduling of a single traffic light in real-time, including the identification of cycle length and signal change.
- We also develop a system to identify the scheduling change of a traffic light by continuously monitoring the cycle length.
- We conduct extensive evaluation using real taxi traces from Shenzhen, China. Evaluation results validate the effectiveness of our solution.

The rest of the paper is organized as follows: In Section II, we analyze the the taxi traces and describe some exclusive characteristics. In Section III, we briefly introduce the overview of the system. Following that, section IV describes data preprocessing. Section V VI and VII present the main parts of the algorithm. I.e., cycle length, signal change, and scheduling change identification. Section VIII describes the evaluation and results. Related work are summarized in Section IX. Finally, Section X concludes this paper.

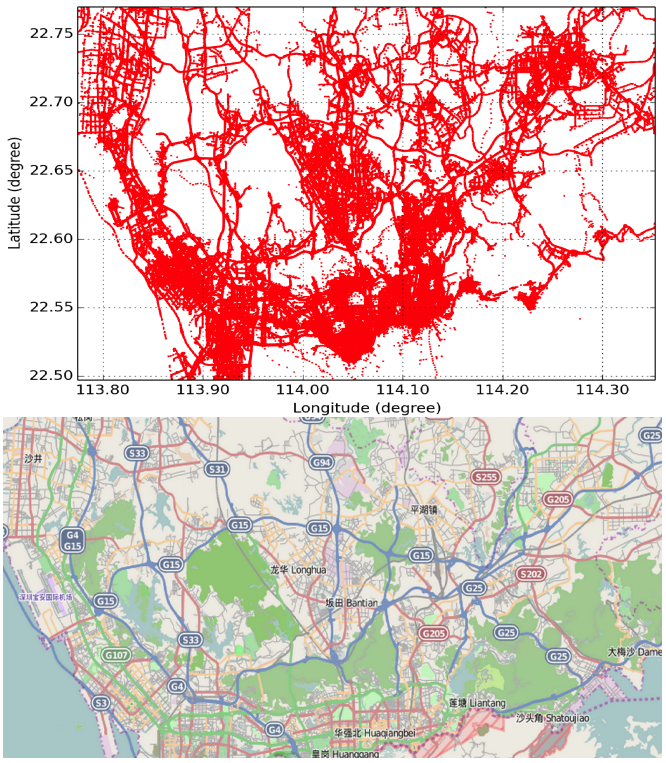

Fig. 1. Comparison between aggregated plot of Shenzhen taxi updates between 8 am and 11 am in December 5, 2014 and Openstreetmap data

TABLE I
DATA FORMAT OF TAXI TRACE

Index	Description	Format
1	Car plate number	STRING
2	Longitude	longitude $\times$1000000
3	Latitude	latitude $\times$1000000
4	Report time	YYYY-MM-DD HH:mm:ss
5	Onboard device ID	NUMBER
6	Driving speed	km/h
7	Car heading	degree to north, clockwise
8	GPS condition	0: unavailable; 1: available
9	Overspeed warning	1: overspeed
10	SIM card number	STRING
11	Passenger condition	0: vacant; 1: occupied
12	Taxi body color	yellow, blue, etc

II. DESCRIPTION OF TAXI TRACE

The results of this research is highly dependent on the crowdsourced taxi traces from Shenzhen, China. It is regulated that every taxi has to install an onboard computer with GPS and cellular network modules, which will upload the taxi's latest location information to the data center periodically [16].

There are 12 fields in the uploaded report and the detailed data format is described in Table I. In this research, we mainly use 5 of them (id, time, longitude, latitude, speed). GPS condition, passenger condition, and car heading are also used, but only for outliers filtering. Fig. 1 illustrates the aggregated taxi traces for 3 hours in a day, and the comparison with real road networks.

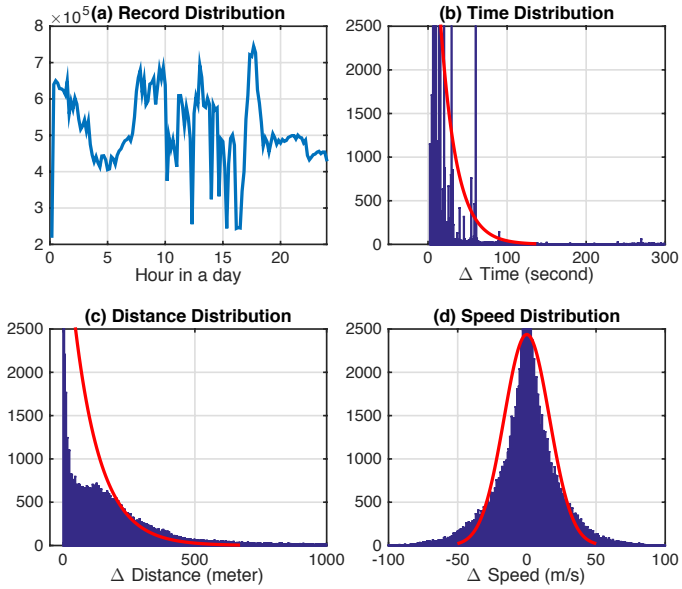

Fig. 2. Statical analysis of taxi trace: (a) Number of records distribution during a day. (b-d) Time, distance and speed differences between consecutive records update

Judging from the taxi plate number, there are over 28,000 taxis in total. Everyday, they update approximately 80 million records, which take about 10 GB of storage. To have a better understanding of such a huge amount of data, statistical analysis is conducted. The results are shown in Fig. 2. Fig. 2(a) describes the number of records in each time of a day, divided into 10 minutes time slot. We can see that the trace can cover entire 24 hours, but the number of record is unbalanced, caused by driver shifting, etc. On average, there are 52,000 records per minute. Fig. 2(b) is the distribution of update frequency. Each taxi updates at a fixed frequency. The interval may range from 5 seconds to over 100 seconds. Several typical frequencies can be easily observed from the plot. E.g., 15s, 30s and 60s. Other long or short intervals may be caused by packet loss or network delay. The mean update interval is 20.41 second, and the standard deviation is 20.54. Fig. 2(c) shows the traveled distance among consecutive update. We can see that lots of taxis remain at the same position between two consecutive updates. This is mainly caused by the waiting of red lights. Because the data update frequency is usually less than the red light duration. This observation also motivates the design of our red light identification algorithm. For moving taxis, most of them can drive 50 to 500 meters during one update interval, and the mean update distance is 100.69 meters. Fig. 2(d) is the speed differences between consecutive update. A positive value indicates that the taxi is accelerating while negative means decelerating. The distribution fits normal distribution well with $\mu = 0$ and $\sigma = 40$, which again confirms the observation that many taxis stay still during consecutive updates.

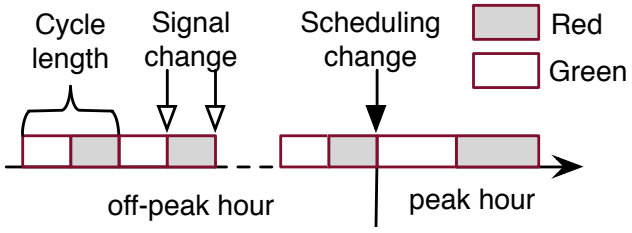

Fig. 3. Scheduling parameters of a traffic light

III. System Overview

According to our on-site interview with Shenzhen traffic police, there are three categories of traffic lights on roads, they are:

1) **Static scheduling**: The length of red and green lights are static and never change according to time of day or current traffic flow. The majority of the traffic lights belong to this category.
2) **Pre-programmed dynamic scheduling**: These traffic lights have multiple scheduling policies. E.g., two different policies are used for off-peak hours and peak hours. The scheduling varies only according to current time of day. This kind of traffic light is usually used in downtown.
3) **Manual scheduling**: For traffic lights at arterial roads where congestions occur frequently during peak hour, on-site traffic policemen will manually control the scheduling according to the traffic flow. When these traffic lights are not manually controlled, they work similar as pre-programmed traffic lights.

In this research, our system is able to detect the traffic light scheduling change. Therefore, our system can be applied to identify the scheduling of the first two categories.

To identify the exact scheduling of a traffic light, three parameters need to be figured out. They are: cycle length, signal change time, and scheduling change time. These parameters are illustrated in Fig. 3. Cycle length is the time duration of two consecutive red and green lights. Signal change time is the time point when the traffic light changes from red to green, and vice versa. Scheduling change time is only applicable to those pre-programmed dynamic scheduling traffic lights. It indicates the time when traffic light scheduling policy changes. E.g., peak hour / off-peak hour switching. Yellow light is not included in the figure, as it usually follows green light and lasts for 3-5 seconds. In this research, we simply treat the yellow lights as red ones according to traffic regulation.

Fig. 4 depicts the key components and flow chart of the system. First, the collected taxi updates are partitioned into small blocks according to the closest traffic lights. Then, for each traffic light, the cycle length is firstly determined. Next, in signal change identification, the length of red light and green light, and the exact time when the light change will be detected. Finally, the system keeps on monitoring the

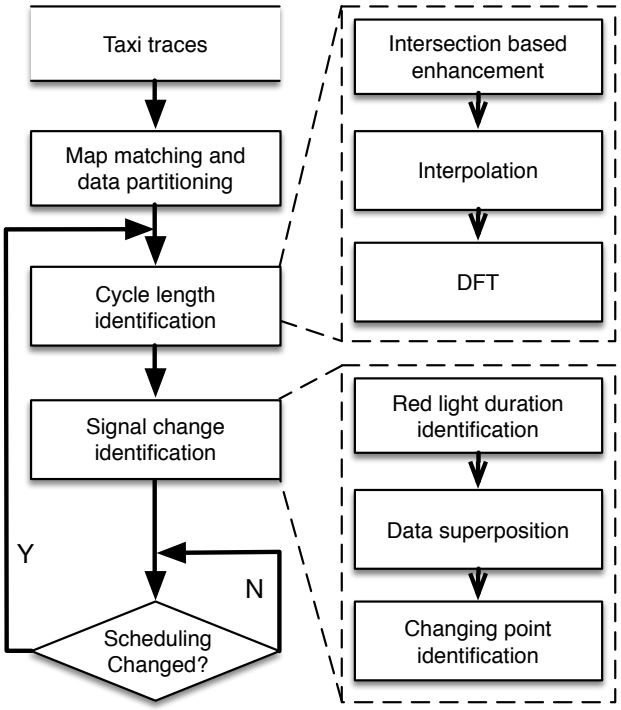

Fig. 4. System flow chart

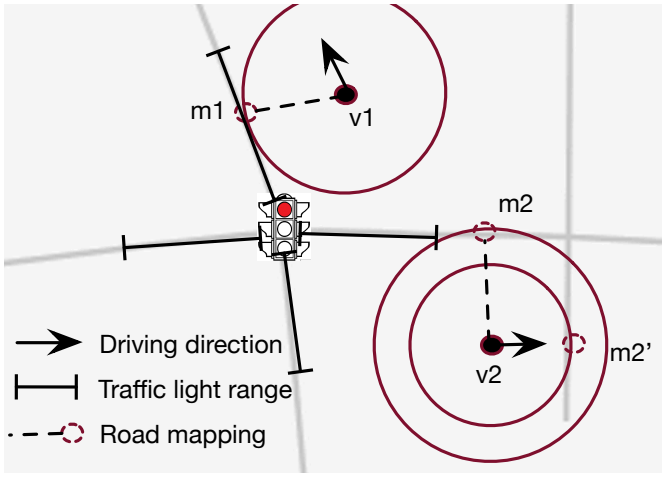

Fig. 5. Map matching and data partitioning

scheduling of the traffic light. When the scheduling changes, the system is able to determine the changing time.

IV. Data Preprocessing

The raw taxi trace can not be directly analyzed, as it contains outliers and miscellaneous errors. Data preprocessing is necessary to eliminate these errors and simplify the subsequent data analysis. In this work, map matching and data partitioning are applied to the raw data.

Map matching eliminates the GPS sampling errors and place the discrete GPS points onto a road segment on digital map. We utilize OpenStreetMap [17] for digital map service. Accurate low sampling GPS data matching [18] is extremely difficult. We only use current car position and driving direction to match GPS points, as depicted in Fig. 5. Normally, GPS points are matched to the nearest road segment, e.g., $v1$ is mapped to $m1$. There is only one exception, the driving direction is in conflict with the road orientation. In this case, the nearest intersection is replaced by the next one with the same orientation. E.g., $v2$ is matched to $m2$ rather than the nearest $m2'$. Since a traffic light at a road intersection only controls the taxis on the nearest segments, data can be simply partitioned into different parts according to the nearest traffic light. Then, the traffic light scheduling identification algorithm for different traffic lights can be easily paralleled. In the following sections, we will discuss how to identify the traffic light scheduling of a single traffic light.

V. Cycle Length Identification

Cycle length identification determines the length of an entire red and green cycle. The key idea of our cycle length identification algorithm is to treat the nearby traffic speed as a periodic signal, whose frequency is the same as the traffic light. By analyzing the traffic speed in frequency domain, the periodicity and cycle length can be reconstructed.

A. Frequency Domain Analysis

It is not easy to identify the frequency of taxi speed directly, as the raw taxi traces are updated in low frequency and contain potential errors. Fig. 6(a) shows the raw traces recorded at a road intersection for a minute. From this figure, we can find out that the traces are not continuous in time domain. Moreover, it is also possible that multiple taxis report their locations at the same time and same road segment. From the signal's point of view, these problem can be treated as data missing and data redundancy, which must be corrected before transforming the signal into frequency domain. Therefore, for the first step, we use interpolation to construct the missing data points. If there is more than one records in a second (e.g., the 10th, 20th second), we utilize the mean value as the interpolating input data point. Spline interpolation is adopted to obtain a smoother signal. This process is illustrated in Fig. 6(b). Notice that the interpolated traffic speed may have negative values, which are infeasible in real world. However, the purpose of interpolation is only to obtain the frequency of speed, and negative values will not affect the frequency of signal. Therefore, we just leave them along without special treatment.

Next, we use Discrete Fourier Transformation (DFT) to obtain the frequency of the traffic speed. To obtain the frequency domain information, a time period of data (e.g., the past 30 minutes) is required as the input. We denote the time domain input as X_n, which contains N data items in total. According to DFT equation, the frequency domain representation x_n is obtained by:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N}, \quad n \in \mathbb{Z} \tag{1}$$

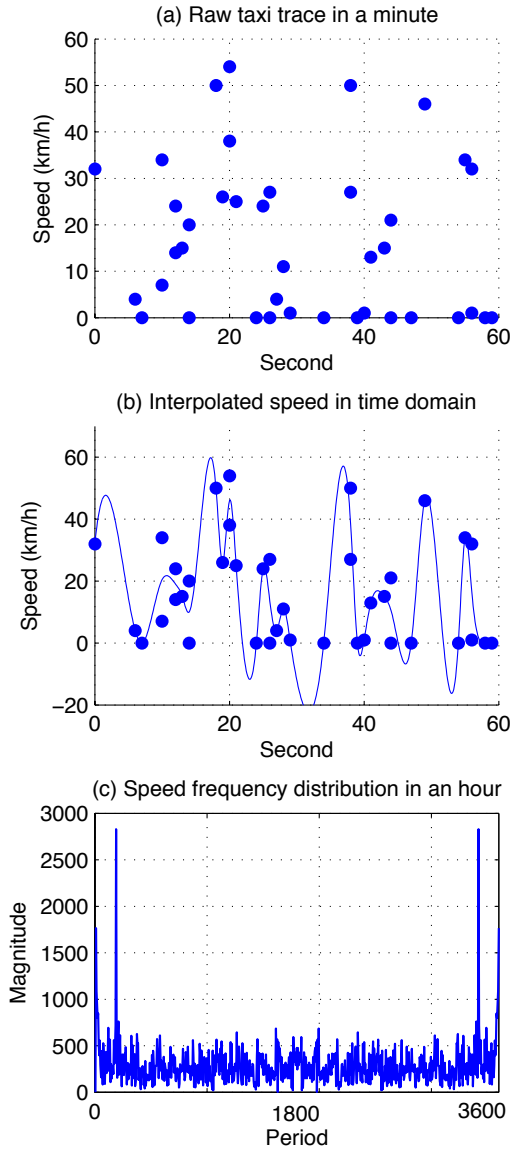

Fig. 6. Traffic light periodicity identification

Then, we traverse the frequency data x_n to find out the one with the largest magnitude. And this frequency can be regarded as the scheduling frequency of traffic light. Due to the nature of DFT, there are two symmetric peaks in the results. The real cycle length can be calculated using:

$$l = \frac{N}{\text{argmax}_n(|x_n|)}, \quad n \in [0, N/2] \tag{2}$$

For example, Fig. 6(c) depicts the magnitude of traffic speed in frequency domain. We can easily find out that index 37 has the largest magnitude, which means there are 37 traffic scheduling cycles in that hour. Then, the cycle length can be calculated by: $60 \times 60/37 \approx 97$ seconds. While the ground truth of the traffic light cycle length is 98 seconds.

B. Intersection based Enhancement

The data sparsity, which is caused by low frequency taxi update, is a big challenge to aforementioned algorithm, especially for some minor roads where the taxi trajectories seldom cover. In these cases, the interpolated taxi speed may contain remarkable errors and may lead to unacceptable inaccuracy in frequency domain analysis (as depicted in Fig. 7(a)). To overcome this difficult, more meaningful input data points are required.

Fortunately, based on the statistics and observations, the cycle lengths of all traffic lights that are installed at the same crossroad intersection are also the same (although the length of red and green lights may vary). This fact motivates us to use the taxi traces on the perpendicular road at the same intersection to enhance our frequency domain analysis algorithm.

Fig. 7(a) illustrates the interpolated speed of two perpendicular roads (North-South and East-West) at the same intersection in 5 minutes. Due to the low update frequency, there are only 3 data points in every minute approximately. Using the data from either north-south or east-west direction have difficulty to reconstruct the traffic light cycle length. It is well-known that cars in N-S direction and E-W direction move forward alternatively. Therefore, to use one direction to enhance the other, we can simply "mirror" the sampled speed. In this work, we obtain the mirrored speed using mean speed as line of symmetry. Specifically, the enhanced speed at time t, denoted by v_t^e is calculated using:

$$v_t^e = \begin{cases} v_t & v_t \neq \emptyset \\ \max(0, 2 \times \bar{v} - v_t^p) & v_t = \emptyset \wedge v_t^p \neq \emptyset \\ \emptyset & Otherwise \end{cases} \tag{3}$$

where v_t is the sampling at the direction that has more data points, v_t^p is the perpendicular direction, $\bar{v}$ is the mean speed of the intersection, and $v_t \neq \emptyset$ means time t has no data sampling. Fig. 7(b) shows the result of mirrored N-S direction, and Fig. 7(c) is the interpolation of enhanced data points from both directions. By applying data enhancement, more data points are added to the DFT input, and more accurate results can be achieved.

VI. Signal Change Identification

Only cycle length can not describe the scheduling of a traffic light. In this section, we will go further to find out the length of red and green light, and also at what time the signal changes. The key idea is to utilize the cycle length obtained from the previous step to merge data from multiple cycles into one. And then sufficient data can be accumulated to identify the length of red light and signal change time using accumulated statistical patterns.

A. Red light duration identification

Fig. 8(a) shows a typical trace when a taxi meets a red light. Based on our on-site observation of 36 traffic lights, the mean red light duration is 91.7s, which is 4.5 times longer

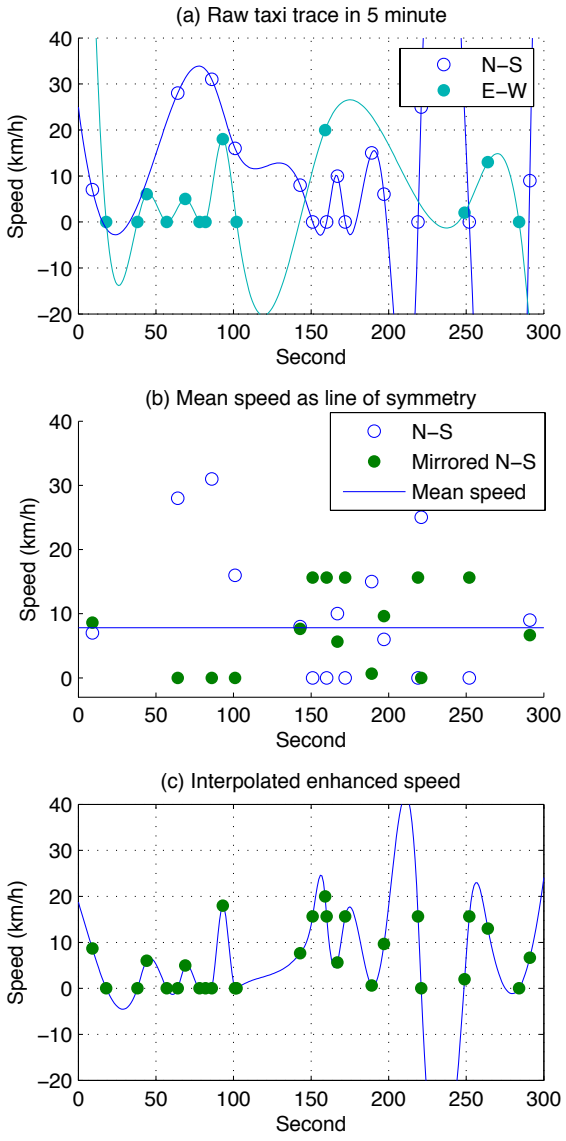

Fig. 7. Intersection based enhancement

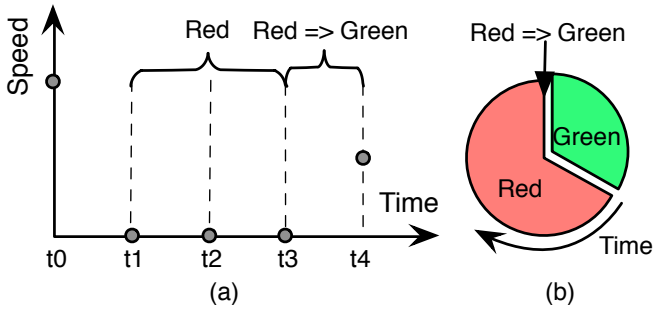

Fig. 8. Principle of signal change identification: (a) Typical trace when a taxi meets red light. Use longest stop duration to determine red light (b) Green and red light classification

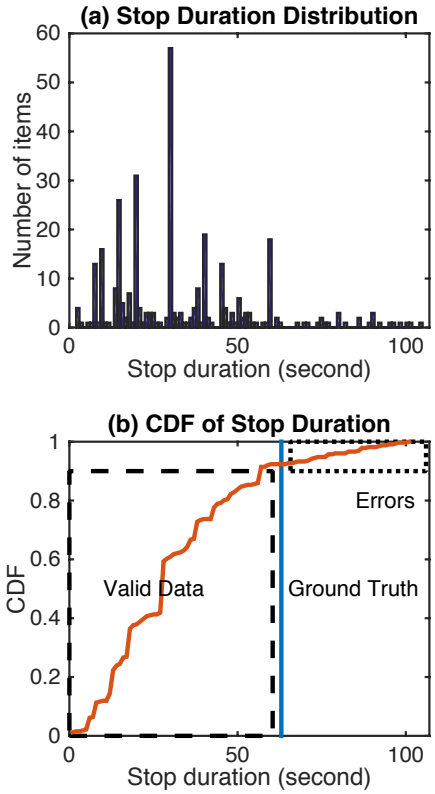

Fig. 9. Use the longest stop duration to determine the length of red light: Cycle length is 106s. Valid data takes 3 times of the mean sample interval (20.14×3), and errors takes 2 times of it (20.14×2). The ground truth is 63s

than the mean taxi update duration (20.14s). Therefore, it is very likely that at least two updates are recorded during the taxi's waiting in front of a red light. This observation is further validated by the statistical results in Fig. 2(c), which reveals that 42.66% of the taxis are stopped during two consecutive updates. Therefore, we can utilize this observation to design the red light duration identification algorithm. The basic idea is to find out the longest stop duration before a red light, and treat the longest stop duration as the length of red light.

In practice, taxis may also stop when the traffic light is green, in which case the longest stop duration may be longer than the actual red light length. For example, taxis may stop temporarily to pick up or drop off passengers. These errors must be eliminated to ensure the accuracy of prediction. In this research, we use two approaches to remove the errors: 1) stop durations that are longer than the traffic light cycle are directly dropped. 2). If the passenger condition (index 11 in Table I) in the taxi trace changes, the record is also discarded. However, these approaches can not remove all errors. Fig. 9(a) shows the stop duration distribution of a road intersection. The ground truth of corresponding red light duration is 63s. We can see that there are still errors in the data and it is not obvious to find the actual red light duration.

To determine the length of red light accurately, we need to distinguish the errors from valid data. Since the errors usually take a small portion of the data ($< 10\%$), the key design of this step is also to utilize the mean sample interval. Valid data

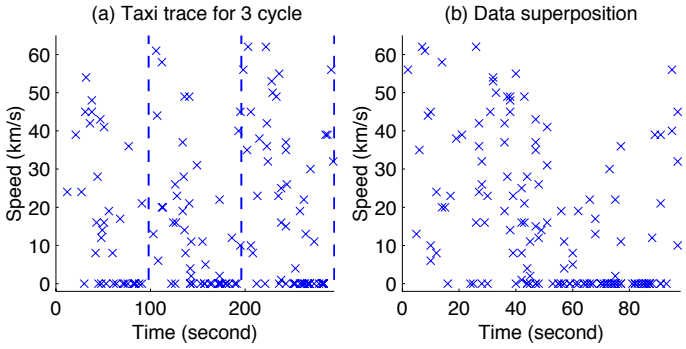

Fig. 10. Data superposition: merge taxi traces from 3 consecutive cycles into one cycle

are always located on the left side of the figure because the durations are shorter, while errors are on the right side. So we divide the entire cycle length into multiple mean sample intervals, and classify each sample interval into error or valid data according to the number of records in it. In this way, we can find the "border interval", which either belongs to valid data or errors. We then obtain the stop duration by calculating the weighted average of border interval, using the number of records as weight. Fig. 9(b) shows the CDF of stop duration, and the result of the algorithm.

Since we have already estimated the cycle length and red light duration, the green light duration can be easily calculated by subtracting red light duration from the entire cycle length, as in Fig. 8(b).

B. Data Superposition

The next step is to determine what time the traffic signal changes. Again, data sparsity is one of the major challenges. Even if we know the duration of the cycle, red light and green light, it is still difficult to distinguish between red lights and green ones (e.g., in Fig. 10(a)). To tackle this challenge, we design a data superposition algorithm, which merges the data from multiple cycles into one. In this way, sufficient samples are obtained. The idea is to separate data into multiple parts, and the length of each part equals to the cycle length. Then we plot all data into one cycle length with new index as old index modulo cycle length. Notice that, data superposition will keep the relative index of data within a cycle. Hence the signal change time within a cycle also remains unchanged.

Fig. 10 shows how data superposition algorithm works. Fig. 10(a) depicts taxi traces for 3 consecutive cycles, whose cycle length is 98s, red light length is 39s, and green is 59s. Due to data sparsity, it is not easy to directly distinguish red and green lights. Fig. 10(b) is the superposed data. In this figure, with more data available, the red and green pattern can be easier discovered. E.g., 50s – 80s are likely to be red, while 0s – 20s tend to be green.

C. Changing Point identification

The last step is to classify the data points within a cycle into red or green light. In other words, we need to find out what time the traffic signal changes. To achieve this goal, we

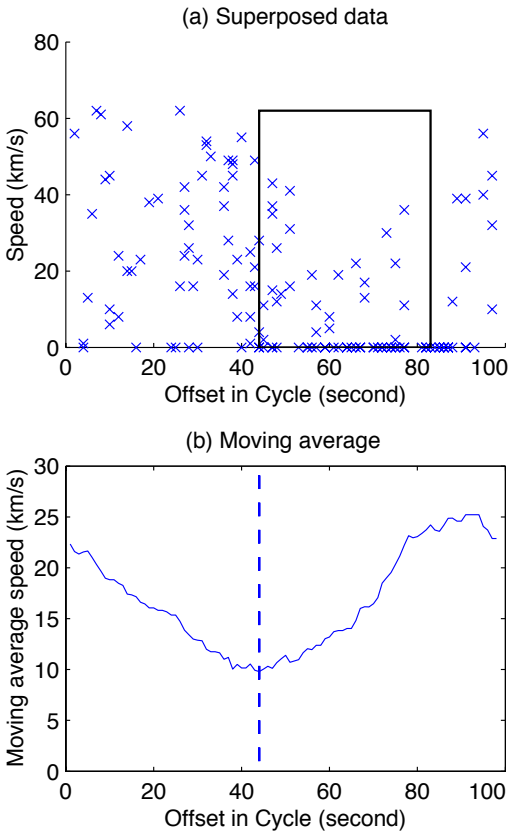

Fig. 11. Use sliding window to determine signal change: cycle length 98s, red light length 39s, and green length 59.

rely on the observation that when the traffic light turns red, all cars begin to decelerate and stop gradually. More and more cars are stopped in the waiting queue. Therefore, the mean speed of all the cars in the waiting queue keeps decreasing. At the time when the traffic light turns green, the mean speed will reach the minimum.

To reproduce this procedure, we design a sliding window based moving average to find the signal changing point. The key design is to use red light duration as sliding window to calculate the moving average of taxi speed using convolution operation. Fig. 11(b) shows the results of moving average speed. Then, we can easily find out the time spot with the minimum taxi speed, which can be treated as signal change time. The rectangle in Fig. 11(a) shows the identified red light duration. The identified signal change time (from green to red) is at 44s, while the ground truth is 41s.

VII. Scheduling Change Identification

Till now, we have introduced the entire procedure of identifying the scheduling parameter of a single traffic light. As we have discussed before, not all traffic lights are statically scheduled. Therefore, the scheduling parameters can not be identified once and used forever. Our system must have the ability to discover at what time the traffic scheduling

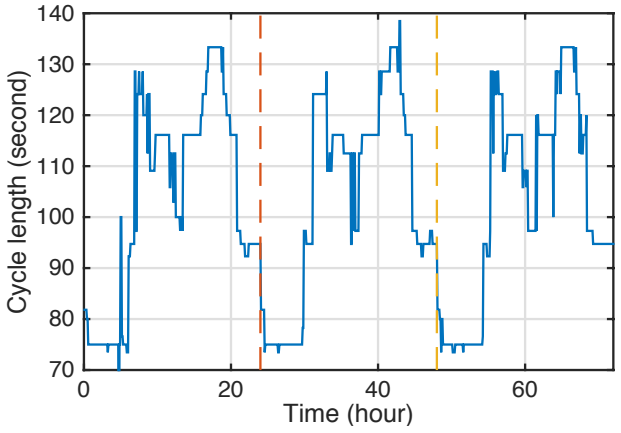

Fig. 12. Continuous monitoring of the cycle length for 3 days

is changed, although the traffic light scheduling change is generally a rare event.

In this work, our system keeps on monitoring the traffic light by calculating the cycle length every 5 minutes. Fig. 12 shows the continuous cycle length change in three days (May 21–24, 2014). Despite some obvious outliers, we can easily find the pattern for different time of day. Moreover, this traffic light uses similar scheduling policy at the same time of different day. This observation provides us insight to utilize historical traffic light scheduling to correct the identification of current scheduling.

VIII. Evaluation

The accuracy of the system is of great importance for real-world applications. To evaluate the performance of the system and demonstrate its potential application, we have conducted two types of experiments: First, we record the scheduling of some traffic lights by on-site observation, and compare our identified value with the recorded ground truth. Second, we develop a car navigation application with traffic simulator, which can avoid possible red lights in route recommendation, and compare the performance with traditional shortest-time navigation.

A. On-site ground truth recording

Since neither the traffic management office nor the taxi companies has the ground truth of the traffic light scheduling, we have to monitor and record the ground truth by ourselves. As on-site observation is a time consuming task, we have to select some typical road intersections to monitor. In this experiment, 36 traffic lights from 9 intersections are selected. The 9 intersections have covered both the busiest intersection and minor roads that taxis seldom visit. Table II lists the intersections, their locations and the number of records per hour. From this table, we can see that the car flow of the busiest intersection (ID 1) is 25x larger than the idlest one (ID 5), which further validate that the data are highly unbalanced. These intersections are monitored for over 8 days (May 20–25, 2014 and Dec 05–06, 2014) to obtain the scheduling ground

TABLE II
On-site data collected traffic lights

ID	Road Name	Geo Location	No. of Records Per Hour
1	ShenNan WenJin	114.125, 22.547	5071
2	FuHua FuTian	114.072, 22.538	1638
3	FuHua ZhongXinSi	114.053, 22.538	1039
4	SunGang BaoAn	114.104, 22.558	1863
5	BaGua BaGuaSan	114.094, 22.564	198
6	ShenNan BeiDou	114.129, 22.548	1687
7	HongLi HuangGang	114.068, 22.551	2178
8	FuHua ZhongXinWu	114.056, 22.537	708
9	FuZhong JinTian	114.058, 22.547	266

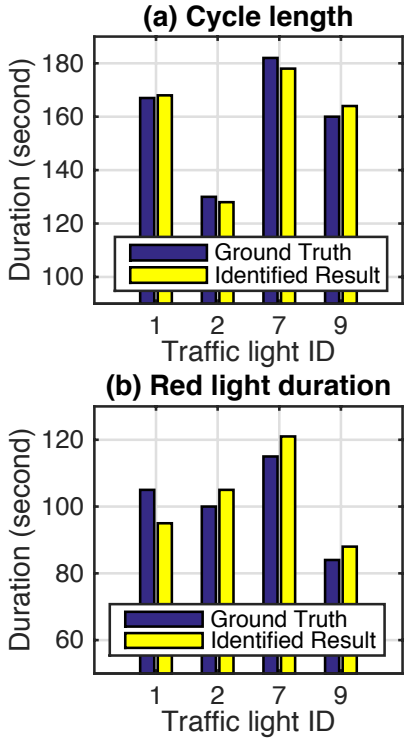

Fig. 13. Ground truth v.s. identified values at 15:22 Dec 05, 2014

truth. To simplify the analysis, we simply treat all yellow lights as red ones in the following experiments.

Fig. 13 shows the comparison between recorded ground truth and the system identified scheduling parameter for a randomly selected time point (15:22 Dec 05, 2014). From these figures, we can find out that the errors for both cycle length and red light duration are less than 5 seconds on average.

To obtain statistical results about the system performance,

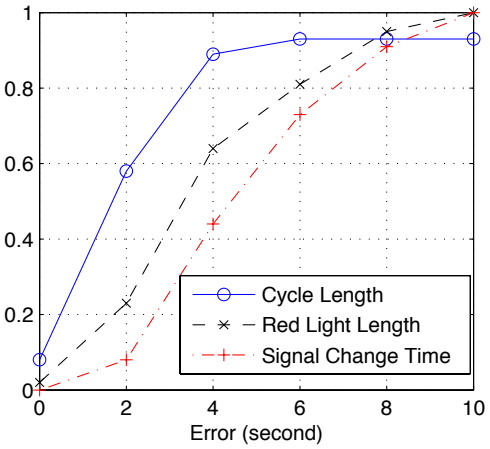

Fig. 14. CDF of identification errors

we randomly select multiple time spots and repeat the traffic light identification algorithm for all traffic lights for over 1,000 times. Fig. 14 illustrates the cumulative distribution of the identification errors for cycle length, red light length, and signal change time. The CDF curve of cycle length reveals that the cycle length identification algorithm is either very accurate, or has notable errors. About 7% of the results may have errors larger than 10 seconds, which is intolerable in real world applications. This is caused by the nature of the algorithm, since it analyzes the periodicity in frequency domain, and the strong magnitude is not necessarily continuous in frequency domain. Fortunately, since the errors are significant, recognizing and correcting the errors is not difficult. E.g., in Fig. 12, the errors can be easily identified. For red light length and signal change time errors, about 80% of the errors are within 6 seconds. Considering that the yellow lights usually last for as long as 5 seconds, the results are very promising.

B. Simulation based application

To exhibit the benefits of the traffic light identification technique, we have developed a demo application that can utilize the identified real-time traffic light scheduling to improve the navigation efficiency. The application is developed using SUMO [19] simulator, which can simulate traffic lights and traffic flows, and provides APIs to control them at runtime. Fig. 15 shows the road topology we use in the application, and the length of shortest road segment is 1km. Traffic lights are placed on each intersection. To simplify the case, the traffic lights cycle length are randomly picked from 120s to 300s. The red and green lights have the same duration.

The objective of the application is to provide the shortest-time navigation. Conventional shortest-time navigations only consider the real-time traffic speed, if traffic light scheduling is available, some red lights are also avoidable. Designing a red light bypassing algorithm itself is not trivial [5]. Our demo application adopts a simple strategy to bypass red lights. We simply enumerate all the possible trajectories from source to destination and calculate the total traveling time, which includes both driving and waiting time, and choose the minimum one. The strategy is updated whenever the car meets an intersection. The strategy is relatively easy to be implemented, but the complexity is not polynomial-time. Therefore, it can not be applied to large-scaled real road network.

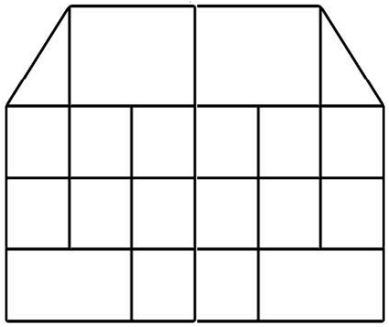

Fig. 15. Simulation road topology

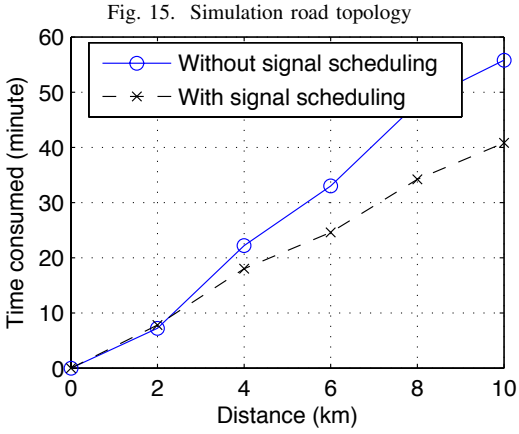

Fig. 16. Shortest-time navigation performance comparison

The performance comparison is illustrated in Fig. 16. From this figure, we can see that when the total navigation distance is small, the improvement is not obvious. Because red light bypassing usually increases the driving distance, which also increases the total navigation time. With the increase of navigation distance, the advantage of using real-time traffic light scheduling is obvious. Overall, about 15% driving time can be saved. This result reveals that the technique has promising applications in real world.

IX. Related Work

Crowdsourcing based traffic light scheduling identification is rapidly developing due to the low cost in data collection. In this approach, participants contribute their privately owned data from vehicles or smartphones to infer the traffic light scheduling. The data includes acceleration, location, speed, and etc. CityDrive [5] collects data from in-vehicle smartphone's accelerometer, magnetometer and GPS to infer the traffic light phase and cycle. Similar idea can be found in iTrip [12]. The traffic light changing phase can be inferred with vehicle acceleration or deceleration events. Differently, Kerper's approach [13] only leverages the speed profile of vehicles to find the difference pattern of red and green lights. However, all of these approaches assume high frequency data

collection rate like 2Hz or 1Hz. Differently, in our taxi traces, every taxi only reports their condition once or twice per minutes. Hence, the speed variation pattern is impossible to be found in our case.

Towards the most related work, Fayazi et al. [14] analyzes bus traces for traffic signal estimation. Similar to our taxi data, the bus traces are also updated with a low frequency (every 200m). The key different between taxi and bus traces is that buses always travel with predefined trajectories and stop at fixed bus-stops, which can be utilized to find the correlation with traffic light. But it does not hold for the case of taxi, which is much chaotic and unbalanced. In addition, the temporal and spatial coverage of the bus traces can also limit the application of the systems.

Vision based traffic light identification is also popular because it is natural. Levinson et al. [6] introduce an algorithm that can effectively identify the states of traffic light. Roters et al. [20] propose another solution to use video from smartphones to recognize the traffic light. Besides traffic signal recognition, SignalGuru [11] takes a step forward. It relies on a series of smartphones mounted on the windshield of cars to collect the image of traffic light ahead. Based on the collaboratively collected vision information, traffic light patterns can be deduced. However, the vision based technologies are restricted within line-of-sight range. Obstacles in front may cause interferences. In addition, the performance may be affected by the environment factors, such as light and weather.

Researchers have also investigated using vehicular ad-hoc networks (VANETs) to obtain the traffic light status [9]. Tielert et al. [8] studies to send traffic light information directly to vehicle using single hop communication. Evaluations show that the system can help to reduce fuel consumption by 8%. Differently, Alsabaan et al. [21] propose to combine vehicle-to-infrastructure and vehicle-to-vehicle communication to deliver traffic light information. Messages are propagated in multi-hop fashion. These approaches can obtain accurate traffic signal results. However, the high cost of upgrading existing infrastructures may confine the practicability.

X. Conclusion

Knowing the pattern and scheduling of traffic signal can bring new opportunities to many intelligent transportation systems. In this research, we have adopted a data analytic approach using massive taxi traces to recognize traffic lights scheduling patterns. At the beginning, we presented some statistical features of the taxi trace. Then, we developed a system to identify the traffic light scheduling. The system utilizes DFT and frequency analysis to find out the periodicity of the traffic pattern, which is also the cycle length of the traffic light. Then, we investigate the longest stop duration to identify the length of red light. Finally, we developed a moving average based algorithm to find the signal change time. To tackle the data sparsity issue, we have designed two approaches. One is time domain interpolation, and the other is cycle based data superposition. To validate the system, we have conducted extensive experiments using taxi traces. The evaluation results confirm the effectiveness of our system and algorithms.

References

[1] Texas A&M Mobility Institute, "Annual urban mobility report," 2012, [accessed 20-January-2015]. [Online]. Available: http://mobility.tamu.edu/ums/

[2] R. Flagan and J. Seinfeld, *Fundamentals of Air Pollution Engineering*, ser. Dover Civil and Mechanical Engineering. Dover Publications, 2013.

[3] B. Zhou, J. Cao, and H. Wu, "Adaptive traffic light control of multiple intersections in wsn-based its," in *Vehicular Technology Conference (VTC Spring)*, 2011, pp. 1–5.

[4] G. Mahler and A. Vahidi, "An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2516–2523, 2014.

[5] Y. Zhao, Y. Zhang, T. Yu, T. Liu, X. Wang, X. Tian, and X. Liu, "CityDrive: A map-generating and speed-optimizing driving system," in *IEEE INFOCOM*, 2014, pp. 1986–1994.

[6] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5784–5791.

[7] National Transportation Operations Coalition, "National traffic signal report card," 2015, [accessed 20-January-2015]. [Online]. Available: http://www.ite.org/reportcard/

[8] T. Tielert, M. Killat, H. Hartenstein, R. Luz, S. Hausberger, and T. Benz, "The impact of traffic-light-to-vehicle communication on fuel consumption and emissions," in *Internet of Things (IOT)*, 2010, pp. 1–8.

[9] W. Niebel, O. Bley, and R. Ebendt, "Evaluation of microsimulated traffic light optimisation using v2i technology," in *Telematics in the Transport Environment*, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2012, vol. 329, pp. 18–25.

[10] C. Barba, M. Mateos, P. Soto, A. Mezher, and M. Igartua, "Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights," in *Intelligent Vehicles Symposium (IV)*, 2012, pp. 902–907.

[11] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "SignalGuru: Leveraging mobile phones for collaborative traffic signal schedule advisory," in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2011, pp. 127–140.

[12] J. Zheng, J. Cao, Z. He, and X. Liu, "iTrip: Traffic signal prediction using smartphone based community sensing," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 2944–2949.

[13] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve, "Learning traffic light phase schedules from velocity profiles in the cloud," in *International Conference on New Technologies, Mobility and Security (NTMS)*, 2012, pp. 1–5.

[14] S. Fayazi, A. Vahidi, G. Mahler, and A. Winckler, "Traffic signal phase and timing estimation from low-frequency transit bus data," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2014.

[15] H. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 145–162, 1999.

[16] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He, "Exploring human mobility with multi-source data at extremely large metropolitan scales," in *International Conference on Mobile Computing and Networking (MobiCom)*, 2014, pp. 201–212.

[17] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[18] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *International Conference on Advances in Geographic Information Systems (GIS)*, 2009, pp. 352–361.

[19] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo - simulation of urban mobility: An overview," in *International Conference on Advances in System Simulation (SIMUL)*, 2011.

[20] J. Roters, X. Jiang, and K. Rothaus, "Recognition of traffic lights in live video streams on mobile devices," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1497–1511, 2011.

[21] M. Alsabaan, K. Naik, and T. Khalifa, "Optimization of fuel cost and emissions using v2v communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1449–1461, 2013.