A Participatory Urban Traffic Monitoring System: The Power of Bus Riders

Zhidan Liu, *Member, IEEE*, Shiqi Jiang, Pengfei Zhou, *Member, IEEE*, and Mo Li, *Member, IEEE*

***Abstract*— This paper presents a participatory sensing-based urban traffic monitoring system. Different from existing works that heavily rely on intrusive sensing or full cooperation from probe vehicles, our system exploits the power of participatory sensing and crowdsources the traffic sensing tasks to bus riders' mobile phones. The bus riders are information source providers and, meanwhile, major consumers of the final traffic output. The system takes public buses as dummy probes to detect road traffic conditions, and collects the minimum set of cellular data together with some lightweight sensing hints from the bus riders' mobile phones. Based on the crowdsourced data from participants, the system recovers the bus travel information and further derives the instant traffic conditions of roads covered by bus routes. The real-world experiments with a prototype implementation demonstrate the feasibility of our system, which achieves accurate and fine-grained traffic estimation with modest sensing and computation overhead at the crowd.**

***Index Terms*— Urban traffic monitoring, participatory sensing, bus systems, bus riders, cellular signal.**

I. Introduction

REAL-TIME and comprehensive urban traffic information benefit urban citizens' daily life and improve the efficiency of urban transportation. Tremendous efforts have been made to explore the accurate, efficient, and inexpensive urban traffic monitoring approaches. People used to widely deploy infrastructural devices, e.g, inductive loop detectors [22] and traffic cameras [25], at roadsides to detect instant traffic conditions. These conventional approaches, however, incur substantial deploying and maintenance costs, which thus greatly limits the road coverage. Recently people take the roving vehicles, e.g., taxis, on roads as probes to detect the instant traffic conditions [18], [36], [40]. Although these passive probing methods can extract on-site information from probes freely roving in the city with lightweight cost, they still cannot provide complete traffic estimations for the whole road network due to insufficient probes. In addition, as probe vehicles are usually managed by transit companies or agencies, it requires substantial efforts for obtaining the traffic data.

In this paper, we present a participatory sensing based urban traffic monitoring system, which takes the public buses as probes to sample the instant road traffic conditions. In urban cities, the public buses cover most roads with a high coverage ratio of the whole road network (see more details in Section III). Instead of requesting the GPS traces from any third parity, our system relies on the help of bus riders and crowdsources the traffic sensing tasks to their commodity mobile phones. The mobile phones automatically collect real-time traffic sensing data and anonymously upload the data to a backend server, which is responsible for processing and analyzing the uploaded information from different buses. Bus travel information are extracted and general travel speeds at different road segments are estimated to generate the traffic map of roads covered by public buses. Our system is fully built on the bus riders' commodity mobile phones with low computation and energy costs, which can encourage wide participation for large service coverage. The lightweight design allows the immediate adoption of our system to other cities.

Despite these advantages, the realization of such a participatory urban monitoring system encounters a set of challenges which call for practical and effective solutions to cope with. First, accurately and efficiently tracking bus trip is non-trivial. Considering the practical requirements of a participatory sensing system, energy-hungry GPS sensor is undesirable. Instead, we prefer the cellular signal together with several lightweight sensing hints, e.g., audio and acceleration signals, from the mobile phones to detect and identify the bus trip information. However, the cellular signals only provide rough location references which are insufficient for precise vehicle tracking [17], [37]. By leveraging the fact that public buses travel along determined routes and stop at known bus stations, we propose a novel method that exploits the invariant locations and cellular attributes of bus stations to build a location mapping between the physical space and the cellular space. Second, the sensing data collected from bus riders are complicated and noisy even with errors. To guarantee accurate traffic estimation, we clean the sensing data at individual mobile phone and propose some clustering and aggregation methods at the backend server to process and analyze the joint data from all participants. Third, it is difficult to build and maintain a cellular fingerprint database which stores the cellular signatures for all bus stops. The offline construction and maintenance of such a database requires intensive manual workload and affects the ease of system deployment. In this work, we further exploit the power of bus riders and propose an online method. Specifically, we bootstrap the database from a small set of bus stops with manually collected cellular data and crowdsource the full database construction to bus riders.

We detail and integrate all above techniques for a complete urban traffic monitoring system, and implement a prototype

Manuscript received August 17, 2016; revised October 31, 2016 and December 13, 2016; accepted January 3, 2017. Date of publication January 31, 2017; date of current version October 3, 2017. This work was supported in part by the Singapore MOE ArRF Tier 1 under Grant M4011159 and Grant M4011726 and in part by the NTU NAP under Grant M4080738.020. The Associate Editor for this paper was R. Malekian.

The authors are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: liuzhidan@ntu.edu.sg; sjiang004@ntu.edu.sg; pfzhou@ntu.edu.sg; limo@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2650215

system on the Android platform and a laboratory server. To evaluate the performance of our system, we conduct extensive experiments with 8 bus routes in a $\sim 28\,km^2$ region in Singapore. During the 2-month experiments, our system has received data from 122 participants and derived instant traffic map of the studied area. Experimental results demonstrate the feasibility and effectiveness of our system in practice. The system overhead is also carefully investigated.

The rest of this paper is organized as follows. We review the related work in Section II. The system design is motivated and detailed in Section III and Section IV, respectively. In Section V, we describe our crowdsourced online database construction. The evaluation results are reported in Section VI. Finally we conclude this paper and discuss future works in Section VII.

II. Related Work

A. Traffic Monitoring

There are numerous approaches proposed to derive traffic conditions of urban cities. People used to deploy plenty of intrusive infrastructures, e.g., inductive loop detectors [22] and traffic cameras [25], to measure spot traffic speeds, which incurs huge deployment and maintenance costs and thus provides limited coverage. Recently researchers resort to using the GPS traces collected from probe vehicles for low-cost traffic monitoring. In the transportation domain, some operational systems adopting the Automatic Vehicle Location (AVL) system have been developed for automobile traveling time prediction [9], [29], freeway traffic condition measure [10], and general travel condition inference from bus information [23]. In the computer science domain, the traffic data from probe vehicles, e.g., taxis and buses, are leveraged to develop various systems for traffic condition estimation and prediction [1], [2], [18], [19], [35], [36], [40]. For example, both [1] and [19] build advanced models to capture the spatial-temporal correlation among road traffic conditions for accurate traffic prediction. To overcome the data sparsity issue in the data-driven solutions, Asif *et al.* [2] use tensor decomposition to recover the missing traffic conditions from available traffic data. Liu *et al.* [18] propose an approach that exploits the mined road network correlation for real-time traffic estimation with instant taxi data. Zhu *et al.* [40] make use of singular value decomposition technique for missing traffic condition recovery from traffic data. These traffic data also enable other applications, e.g., traffic volume estimation [3], route traveling time prediction [28], [34], taxi traveling fare estimation [4]. These works heavily rely on the cooperations with particular companies or transit agencies to access the large amount of traffic data, which are prohibitively obtained without permit. Our work significantly differs from existing works by crowdsourcing traffic sensing tasks to bus riders. Rather than relying on special infrastructure devices or cooperation with any third parties, we encourage participatory efforts from bus riders to collect traffic data and derive the traffic conditions through careful data processing and analysis.

B. Tracking and Localization

Object tracking and localization have been extensively studied in recent years. Thiagarajan *et al.* [26] present a crowdsourced alternative that exploits the sensors on mobile phones, e.g., GPS, WiFi and accelerometer sensor, for transit tracking. CTrack [27] maps the vehicle trajectory in an energy-efficient manner by using the cell tower signals and various sensors on mobile phones. Zhou *et al.* [39] leverage the sensing signals from mobile phones of bus riders to track bus movement and predict bus arrival time. Our work is fundamentally different from the work in [39] that traffic estimation focuses on the processing and analysis of the bus trip data collected from various buses rather than classifying and inspecting specific bus route in the problem of bus arrival time prediction. EasyTracker [6] can automatically track, map, and predict the arrival time of transits by carefully analyzing GPS traces collected from mobile phones installed on vehicles. VeTrack [38] can timely track a vehicle's location based on only inertial sensors of mobile phones for indoor environment. Musa and Eriksson [21] propose to passively track mobile phones by exploiting available WiFi signals. Previous works [6], [27], [28] extensively use the hidden Markov model to map GPS data with road network to derive vehicle's trajectory, which introduces huge computation overhead for probability calculations and the best candidate searching. Different from those works on accurate vehicle tracking, our system only requires accurate bus stop identification to map traffic estimation on the roads. Thus we propose the cellular signal based bus stop identification and mapping methods that effectively satisfy the requirements of participatory sensing systems on the high scalability and lightweight deployment overhead.

C. Participatory Sensing

People-centric mobile computing has inspired the development of many participatory platforms and applications [8]. For example, GreenGPS [12] calculates the fuel consumption on city streets based on the participatory sensing data, and thus enables the fuel-efficient navigation service for drivers. CrowdAtlas [33] automates map updating based on the participants' traveling trajectories. Ear-phone [24] relies on participatory sensing to generate the urban noise map. Ganti *et al.* [11] collect traffic data from a roving sensor network of more than 2,000 taxis and present the data analysis results. Some works [7], [31], [32] propose sensing schemes to support practical and energy-efficient mobile phone sensing. As a middleware, Pogo [7] provides fine-grained user-level control for volunteering mobile phone users to preserve their privacy. ARTSense [31] is a framework for solving the problem of "trust without identity" in participatory sensing networks based on a privacy-preserving provenance model. Wang *et al.* [32] propose an Markov-optimal sensor sampling policy for mobile applications and services to save the energy of mobile phones. For the participatory sensing systems, it is significantly important to incentivize users' active and reliable participation. Existing incentive mechanisms are reviewed and discussed in [13]. Those works have addressed many open problems on practical participatory sensing systems and are parallel with our work.

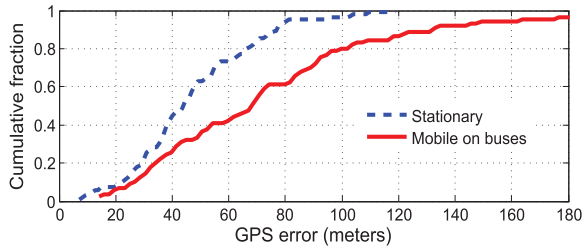


Fig. 1. GPS localization errors in downtown Singapore.

III. Motivation

Although there exists many works about traffic monitoring, the intrusive sensing approaches usually incur huge infrastructure costs and the probe vehicle based approaches are usually limited by data availability. To get rid of these disadvantages, we prefer a participatory sensing based urban traffic monitoring system that resorts to public buses for probing real-time traffic conditions. Specifically, we fundamentally decompose the traffic sensing tasks from the running buses to bus riders' mobile phones. The bus riders themselves contribute the primary traffic sensing data and meanwhile consume the final traffic output. Such a low-cost and flexible system can be easily adopted in other cities with slight modification.

The public bus network covers most of the roads in a city to provide convenient commuting for citizens, which thus provides good coverage of traffic monitoring. For example, the road network coverage ratio by bus routes is as high as 75% in Seattle[1] and London,[2] 79% in Singapore[3] and New York.[4] Taking the Jurong West area of size $\sim 28\,km^2$ in Singapore, shown in Fig. 2(a), as a concrete example, we find that $\sim$80% roads in the area are covered by more than 20 bus routes. Therefore, once we track the movements of these buses, we can map down the probed traffic conditions for this area.

The straightforward approach for vehicle tracking is using GPS sensors [26]–[28], however, it may not be a good choice due to the considerations of energy consumption and localization accuracy. First, GPS device is energy aggressive, that sorely discourages user participation due to the limited battery capacity of commodity mobile phones. We have measured the energy consumption of GPS sensor on Google Nexus One mobile phone using the Monsoon power monitor. The measurement shows that continuous GPS tracking incurs as high as $300\,mW$ energy consumption (see details in Section VI-E). Second, GPS suffers from large localization error in the downtown area due to the complicated surroundings, e.g., dense and high buildings. It is even worse when the mobile phones are placed inside buses where the GPS signal is further attenuated. Our measurement proves this phenomenon, as shown in Fig. 1, that the average errors are as high as $41\,m$ and $68\,m$ when the phone is at the bus stops (i.e., "stationary" in Fig. 1) and on a moving bus (i.e., "Mobile on buses" in Fig. 1), respectively. Such a phenomenon is common in practice and has been reported in other works [14], [20], [26].

[1]Seattle bus service. http://www.seattle.gov/html/citizen/bus.htm/.

[2]Bus Transport in London. http://www.tfl.gov.uk/.

[3]Bus transport in Singapore. http://en.wikipedia.org/wiki/Bus_transport_in_Singapore.

[4]New York city bus system. http://www.ny.com/transportation/buses/.

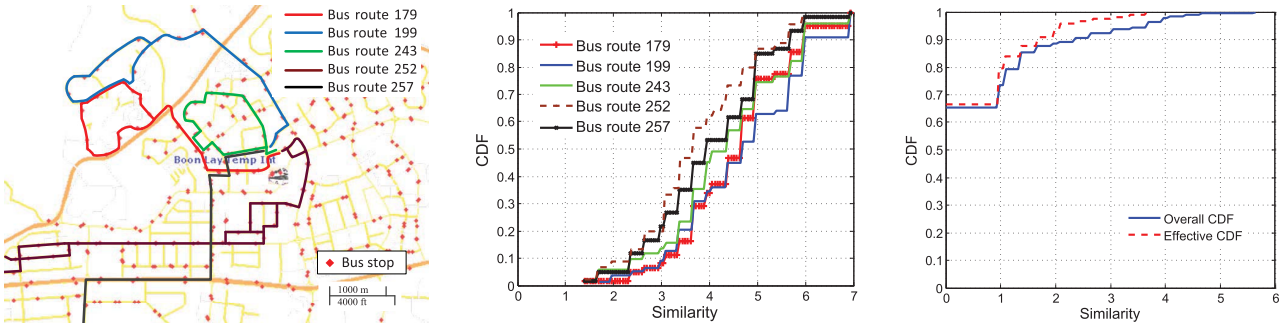
Although modern buses are equipped with GPS devices for management purpose, the collected data are held by transit agency and not available for the public. Despite possibly public open of these data in the future, the traffic data collected from bus riders can still be a complementary to the public open data for more comprehensive urban traffic monitoring.

Compared to energy-hungry GPS sensors, cellular signal from mobile phones is more energy-friendly and widely available [16], which makes it a better sensing hint for vehicle tracking. Meanwhile cellular signal outperforms other possible wireless signals, e.g., WiFi [30], for location references due to the following advantages. First, mobile phones always keep the cellular module working to support persistent telecommunication services. Thus the marginal energy consumption of collecting cellular signals is negligible. Much extra energy, however, will be consumed for scanning other wireless signals like WiFi [27], [41]. Second, cell towers are widely deployed to provide the complete coverage of the entire city while other wireless signals are usually sporadically available with poor coverage in outdoor areas. Third, unlike other transient wireless signal sources, e.g., WiFi hotspots, the cellular signal sources are much more consistent over time, which makes the cellular signature database more stable and easier to maintain.

The typical coverage of a cell tower in the urban area is about $200 \sim 900\,m^2$. Thus the cellular signals provide only rough location references and are insufficient for instant and accurate bus tracking. The fact that public buses strictly follow the determined bus routes and stop at the known bus stations, however, provides us an opportunity to relax the requirement of precise bus tracking. As Fig. 2(a) depicts, more than 100 bus stops densely distribute over the area and spontaneously divide the roads covered by bus routes into small road segments. The precise locations of the bus stops and how bus routes operate over those bus stops are public information and can be readily obtained from the web. Thus we transform the precise bus tracking problem to the bus status detection and bus stop identification among all possible bus stops. Based on the bus status information, we can recover the bus movements and estimate the traffic conditions on the road segments in between bus stops. The final traffic map can be derived by assembling the traffic estimations of all covered road segments. To accurately identify bus stops, we need to collect the cellular signals for all bus stops as cellular fingerprints, and later use these fingerprints to match bus stops in cellular space with the cellular signals uploaded from bus riders' mobile phones.

We conduct some preliminary experiments to explore the feasibility and effectiveness of using cellular signals as fingerprints to distinguish different bus stops in the cellular space. We experiment with 5 bus routes (i.e., bus route 179, 199, 243, 252, and 257 in the region as shown in Fig. 2(a)) and extensively measure the cell tower signals at 86 bus stops. Two scenarios are considered during the cellular signal measurements: *stationary* when we stand at the bus stops and *mobile* when we pass by the stops on a bus. Our data collection covers various weather conditions and different time of a day.

The mobile phone can perceive signals from multiple nearby cell towers at one time, and will connect to the one with the strongest signal strength. In general, 4$\sim$7 cell towers can



(a) (b) (c)

Fig. 2. Similarity measurement of bus stop cellular fingerprints. (a) Measured bus routes. (b) Similarity of the cellular fingerprints collected at the same bus stop. (c) Similarity of the cellular fingerprints collected at different bus stops.

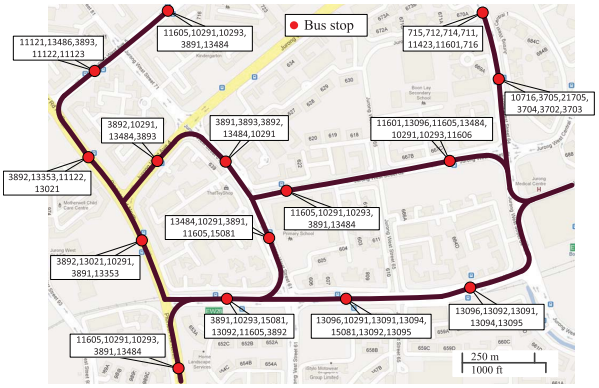


Fig. 3. An example area with the cellular fingerprints.

be captured by the mobile phones at each bus stop during our measurements. For each bus stop, we order the visible cell tower IDs in descending order of their Received Signal Strengths (RSS) and employ such an ordered ID set as the fingerprint for bus stops in the cellular space. Fig. 3 presents an example where we label the cellular fingerprints for 15 bus stops in the area. We find that there exists obvious differences among the cellular fingerprints of different bus stops.

We investigate the stability of such a cellular fingerprint by analyzing the similarities of cell ID sets collected at the same bus stop in different runs under various time and weather conditions. A modified Smith-Waterman matching algorithm (see details in Section IV-E) is used to calculate the similarities, where higher scores mean higher similarities. We present the statistics of self-similarity scores for all bus stops of the 5 routes in Fig. 2(b), which suggests that the similarity score between the cell ID sets collected at the same bus stop is quite high. The similarity scores of 90% cases are higher than 3 and more than 50% cases are higher than 4. The results shows that the cell ID sets are sufficiently stable to signature bus stops at different conditions.

We also investigate the differential ability of the cellular fingerprint by analyzing the similarities of cell ID sets collected from different bus stops. We summarize the results in Fig. 2(c). We can see that the similarity scores of $>70\%$ bus stops are 0 (no common cell IDs at all) and $>90\%$ bus stops have similarity scores lower than 2. We carefully inspect the cases with similarity scores higher than 3 and find that they are from the cell ID sets of two bus stops at opposite sides of the two-way roads. These bus stops can be viewed as the same bus stop in terms of location reference, which will not degrade the performance of our system. As the uploaded traffic data are time-stamped, we can infer the moving direction of the target bus and distinguish the right bus stop for traffic estimation. After such processing, we plot the effective CDF in Fig. 2(c) and find that $>94\%$ bus stops now have similarity scores lower than 2. The results validate the effectiveness of using cellular signals to distinguish different bus stops.

The above study demonstrates that we are able to design a participatory sensing based traffic monitoring system which relies on the power of bus riders' mobile phones. This system requires no special devices or cooperation with transit agency, and can work independently only with the help of bus riders. The energy-efficient sensing and automatic data collection of the system leave negligible overhead to the crowd and their mobile phones, which will thus encourage wide participation.

IV. System Design

The system consists of two major components, i.e., online/offline data collection and trajectory mapping for traffic estimation, as sketched in Fig. 4. In the following subsections, we will elaborate each component in details.

A. Data Collection

As depicted in Fig. 4 (top), the system input mainly comes from the following three data sources.

1) Bus Riders: The bus riders serve as the traffic probes along bus routes and are the major information sources of our system. Once the users are detected on buses, our system will automatically start the online data collection from mobile phones. A beep detection approach similar with that in [39] is applied to detect whether the user is on a public bus or not. Nowadays IC card systems are worldwide adopted by bus operators, e.g., ORCA in Seattle, EZ-link in Singapore, Oyster in London, and MetroCard in New York, to automatically collect transit fees. Typically the bus riders tap their IC cards on the card readers to pay their fees when getting on or off the bus at some bus stop, and meanwhile the card readers generate a unique beep, which is always consist of audio signals in specific frequencies, e.g., a combination of $1\,kHz$ and $3\,kHz$ in Singapore and $2.4\,kHz$ in London. To efficiently detect such a beep signal, we prefer the Goertzel algorithm [5] instead of Fast Fourier Transform (FFT) used

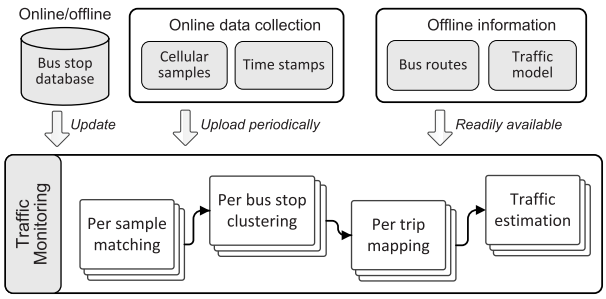


Fig. 4. System architecture and workflow.

in [39] to extract specific frequencies rather than all frequencies based on the prior knowledge of frequency components in the beep. The Goertzel algorithm performs tone detection using much less CPU computation than FFT, and thus significantly saves energy. To detect the beeps, we measure and normalize the signal strength of several interested frequency bands. If the signal strength of the frequency bands obviously jumps and exceeds an empirical threshold (e.g., three standard deviation of the signal strength), the beep detection is confirmed. To enhance the robustness of beep detection, we further adopt a standard sliding window averaging method with window size $300\,ms$ to filter out possible noises. The "sound" hints are widely adopted in IC card systems to let passengers be aware of the transactions. Thus our system can work in other cities with presetting the frequency components in audio signal. Even without such "sound" hints, our system can still detect the on-bus status using some inertial sensors based transportation mode detection methods [15].

The mobile phone will automatically start recording a bus trip after a confirmed beep detection. For each thereafter detected beep event, the mobile phone records the set of visible cell tower signals with an attached timestamp. Thus the sensing data on the mobile phone corresponds to a sequence of timestamped cellular samples along the bus trip. These data are uploaded to the backend server anonymously and periodically. The mobile phone terminates current trip if no beep is detected for $\Delta = 10\,mins$, which implies the user has got off the bus. Once new beeps are thereafter detected, the mobile phone will start recording and uploading another independent bus trip. The system parameter Δ can be adjusted for different cities. To detect possible traffic jams between two consecutive bus stops where bus travel time exceeds Δ, our system tracks the trip status of each anonymous mobile phone. Specifically, if two consecutive cellular samples are collected from the same bus route, we treat they belong to the same trip; otherwise, we think the user starts a new trip.

For the beep detection, we continuously filter out the noisy beep detections (e.g., the beeps in rapid train stations using the same IC card systems or the beeps false positively detected by the users waiting for other buses at the bus stops) by analyzing the readings of accelerometers on mobile phones to distinguish human mobility. A simple threshold based method is enough to filter out the noisy beeps. In general, buses usually travel with frequent acceleration changes while rapid trains are operated more smoothly. Similarly, the readings collected when the user is walking or standing at the bus stop are much smoother than those collected on a moving bus.

2) Bus Stop Database: We assume at this moment that there is an offline built database which stores the cellular fingerprints for all bus stops. The backend server relies on these cellular fingerprints to identify the corresponding bus stops for each uploaded cellular sample. Later we will show that the bus stop database construction can be crowdsourced to bus riders in an online manner (detailed in Section V), which significantly saves the manual workload in war-driving the bus stops.

3) Bus Routes and Traffic Model: The bus operators publicly publish the bus operational route information on the web, which implies constraints on how bus stops are passed. We will make use of such information for trajectory mapping later. In addition, some mature and classical traffic models [9], [10], [23], which describe the transformation between travel speeds of buses and general vehicles, are available for us to derive the general traffic conditions from buses. Therefore, although we reply on public buses to probe traffic conditions, our system reports the general travel speeds that are useful for all vehicles.

B. Trajectory Mapping

As depicted in Fig. 4 (bottom), we take the bus stops as landmarks and match the uploaded cellular samples to map the bus trajectories. For the received sequence of cellular samples from each independent trip, the backend server will recover the bus trip information by identifying the passing by bus stops. The server will do three levels of mapping to enhance the accuracy of trajectory mapping.

1) Per Sample Matching: For a typical IC card system, the card readers are only enabled when the buses arrive at the bus stops, where bus riders pay their transit fees by tapping IC cards. The beep, detected by bus riders' mobile phones, thus indicates the bus arrival at a bus stop, and the collected cellular sample at that time corresponds to a particular bus stop.

To identify one bus stop for each cellular sample of one trip, we match it with the signature sets stored in the fingerprint database. We adopt the modified Smith-Waterman algorithm [39] to measure the similarity of different cell ID sets. This algorithm focuses on the orders rather than the absolute RSS values of cell towers, which thus tolerates possible variances of RSS values due to different conditions (e.g., on/off buses, weather, time, etc.). The backend server reorganizes the cell IDs in a set in a descending order according to their RSS values. For the cell ID set of each cellular sample, the algorithm compares all possible segments to determine the optimal alignment with one cellular signature in the fingerprint database, and assigns different weighs to the matching results of *match*, *mismatch* and *gap*. As the penalty cost for gaps and mismatches will significantly affect the matching performance, we conduct simulations to select the best penalty cost by varying its value from -0.1 to -0.9. Simulation shows that -0.3 as the penalty cost achieves the best matching accuracy. We illustrate this algorithm with an example in Table I where the uploaded cellular sample with cell ID set as $c_{upload} = \langle 1, 2, 3, 4, 5\rangle$ is compared with one cellular fingerprint $c_{database} = \langle 1, 7, 3, 5\rangle$. The

TABLE I

BUS STOP MATCHING INSTANCE FOR ONE CELLULAR SAMPLE WHICH CONTAINS 5 CELL TOWER IDS IN THE SET

c_{upload}	1	2	3	4	5	Match	Gap	Mismatch	$\sum$
$c_{database}$	1	7	3		5				
Match	**1**	×	**3**	-	**5**	3	1	1	2.4
Score	+1	-0.3	+1	-0.3	+1				

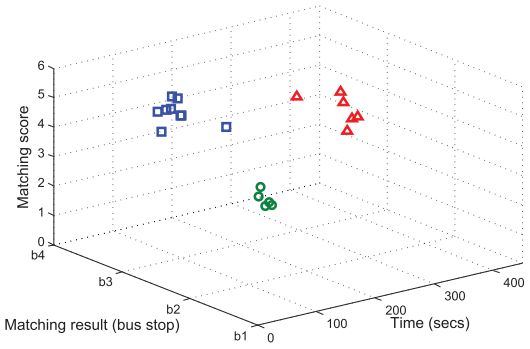


Fig. 5. A sample co-clustering example. 20 cellular samples are clustered into 3 groups corresponding to 3 bus stops.

algorithm finally scores 2.4 by aggregating 3 matches, 1 gap and 1 mismatch.

The matching algorithm runs over all bus stop candidates in the database, and selects one bus stop with the highest similarity score. To guarantee the accuracy of trajectory mapping, we further filter out possible noise cellular samples whose highest matching scores with candidate bus stops are lower than a threshold. We empirically set the threshold as 2 according to our measurement results in Fig. 2(c). Thus all cellular samples with low highest similarity scores are discarded with no further processing. If more than one bus stops are matched with one cellular sample, we select the one with a larger number of common cell IDs. Such a tie-breaker setting can effectively determine one bus stop for each valid cellular sample according to our practical tests. Besides, if the cellular sample $e(x)$ finally matches the bus stop fingerprint $b(y)$ we denote $M(e(x), b(y)) = 1$, otherwise $M(e(x), b(y)) = 0$.

2) Per Bus Stop Clustering: Generally a number of passengers board and alight at a bus stop when the bus arrives, which triggers multiple beeps. As a result, the mobile phones of bus riders can collect and upload multiple cellular samples at one bus stop to the backend server. Such redundant information allows us to further improve the bus stop identification accuracy. We can group the cellular samples according to their matched bus stops and timestamps, and then identify the bus stop for each closely clustered cellular samples with more confidence.

Given a sequence $E = \{e_1, e_2, \ldots, e_m\}$ of m cellular samples with attached timestamps $T = \{t_1, t_2, \ldots, t_m\}$, we can obtain their corresponding bus stops $\{b_1, b_2, \ldots, b_m\}$ with similarity scores $\{s_1, s_2, \ldots, s_m\}$. Fig. 5 depicts an actual sequence of cellular samples collected from one mobile phone. We can observe a clear clustering effect in the space enabled by three dimensions of time, bus stop, and matching score. The cellular samples collected at 3 different bus stops are clustered into 3 groups in the space. In the co-clustering algorithm, for cellular samples corresponding to the same bus stop, we

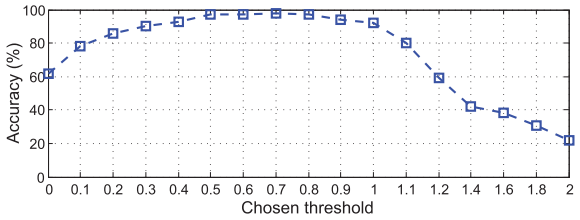


Fig. 6. Co-clustering accuracy with various ε values.

denote the maximum mutual similarity score as s_0 and the maximum possible time interval between their timestamps as t_0, which are empirically set as 7 and 30 *seconds* in our system, respectively. For any two cellular samples e_i and e_j, we weigh their matching relationship as

$$L(e_i, e_j) = \begin{cases} \dfrac{s_0 - |s_j - s_i|}{s_0}, & \text{if } b_i = b_j \\ 0, & \text{otherwise.} \end{cases}$$

Considering the timestamp information, we put e_i and e_j into the same cluster if

$$\frac{t_0 - |t_j - t_i|}{t_0} + L(e_i, e_j) > \varepsilon, \tag{1}$$

where ε is a threshold to verdict whether e_i and e_j belong to the same cluster. Only when two cellular samples are collected close in time and have approximate similarity scores, they will be grouped into the same cluster. We conduct experimental trial with bus route 243 in Singapore to study the impact of ε on the clustering accuracy. We vary the value of ε from 0 to 2 with a step length of 0.1 and plot the results in Fig. 6. In principle, the parameter ε in range of $0.3 \sim 1.0$ achieves reasonably high clustering accuracy, while other settings result in degraded clustering performance with mis-classifications. We finally choose $\varepsilon = 0.6$ in the system implementation.

By applying the co-clustering on all cellular samples, we finally derive a sequence of n clusters $\{C_1, C_2, \ldots, C_n\}$. Each cluster C_i should correspond to one bus stop. Due to the noises, however, the cellular samples from the same cluster may match different bus stops. To guarantee the accuracy of bus stop identification, each cluster C_i is thus temporarily associated with several potential bus stop candidates, as demonstrated in Fig. 7. In practice, however, most clusters only have one bus stop candidate according to our experiments.

3) Per Trip Mapping: The pre-configured bus routes greatly constrain the possible sequences or combinations of bus stops that can be visited by a specific bus. Such constraints help us filter out the impossible bus stop candidates and finally match each cluster of cellular samples to a sole bus stop. Fig. 7 depicts a concrete example where a sequence of n clusters are derived from all cellular samples. Each cluster $C_k (k = 1, 2, \ldots, n)$ includes E_k cellular samples $\{e_k(1), e_k(2), \ldots, e_k(E_k)\}$ and B_k bus stop candidates $\{b_k(1), b_k(2), \ldots, b_k(B_k)\}$. We assign each bus stop candidate $b_k(i)$ a probability $p_k(i) = \frac{\sum_{j=1}^{E_k} M(e_k(j), b_k(i))}{E_k}$ and an average similarity $\overline{s}_k(i) = \frac{\sum_{j=1}^{E_k} [M(e_k(j), b_k(i)) \cdot Sim(e_k(j), b_k(i))]}{\sum_{j=1}^{E_k} M(e_k(j), b_k(i))}$. Now we try to find out a segment from one bus route or the possible

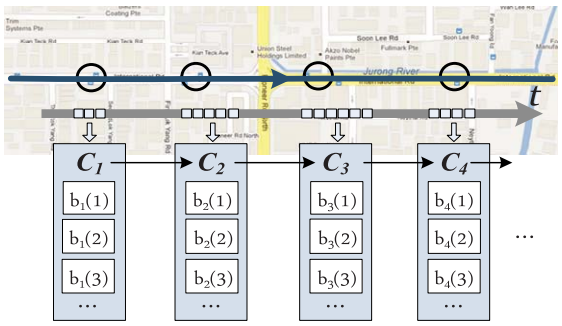


Fig. 7. Bus stop identification with a sequence of clusters. Each cluster contains multiple bus stop candidates.

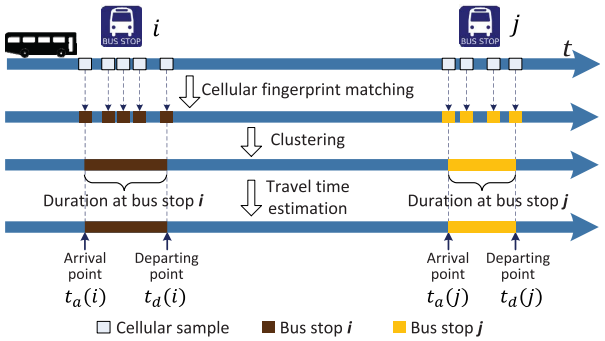


Fig. 8. Bus stop clustering and time duration at bus stops of an example trip. 5 samples are collected at stop i and 4 samples are collected at stop j, which are clustered into 2 groups. The bus arrival time and departing time at bus stops are estimated and used for travel time estimation.

concatenation of multiple different bus routes that best matches the current trip and then derive the most "correct" bus stop for each cellular sample cluster.

For any two bus stops x and y, we denote their order relationship in bus routes as $R(x, y) = 1$ if y is later visited by buses after passing by x in some bus route, $R(x, y) = 0$ if $x = y$, and $R(x, y) = -1$ for the rest. As some cellular sample cluster may match more than one bus stops, we thus can derive a set of possible bus stop sequences $S = \{S_1, S_2, \ldots, S_N\}$, where $N = \prod_{k=1}^{n} B_k$. Each S_j represents a sequence of n bus stops as $\{b_1(S_j(1)), b_2(S_j(2)), \ldots, b_n(S_j(n))\}$. We determine the bus stop sequence best matched with current trip using the maximum likelihood estimation as

$$S^* = \arg\max_{S_{j:1\sim N}} \{p_1(S_j(1)) \cdot \overline{s}_1(S_j(1)) + \sum_{i=2}^{n} [p_i(S_j(i)) \cdot \overline{s}_i(S_j(i)) \cdot R(b_{i-1}(S_j(i-1)), b_i(S_j(i)))]\}, \quad (2)$$

where we weigh S_j using both the probabilities $p_i(S_j(i))$ and average similarities $\overline{s}_i(S_j(i))$. The output S^* finally recovers the trajectory of current trip in the form of the best matched bus stop sequence, which also determines the most likely bus stop for each cellular sample cluster on the trajectory.

C. Traffic Estimation

In this subsection, we will describe how we make use of the trajectory mapping results to estimate traffic conditions of road segments in between bus stops on the trajectories.

Based on the matched bus stop sequences, we first purge the cellular samples by filtering the noise samples which falsely match other bus stops. Then by ordering the cellular samples in the same cluster according to their timestamps, we can extract the arrival time and departing time of one bus at the corresponding bus stop. Fig. 8 illustrates an example with a set of cellular samples from the same trip. These cellular samples are classified into 2 clusters corresponding to two bus stops, i.e., i and j. Based on the cellular sample cluster for bus stop i, we can extract the corresponding arrival time $t_a(i)$ and departing time $t_d(i)$. The same information can be derived for bus stop j. Then we estimate the bus travel time between stop i and j as $t_{ij} = t_a(j) - t_d(i)$. In practice, some bus stops may be skipped by the buses if no passengers board or alight, which results in information missing at these bus stops. In such cases, our system automatically treats the adjacent road segments as one and estimate the travel time on the combined road segment.

The travel time on a road segment by public buses may not accurately reflect the practical traffic condition. There exists a gap between the travel time of general automobiles ($\mathcal{T}_A$) and that of buses ($\mathcal{T}_B$) on the same road segment. The difference arises mainly due to the special operations of buses, e.g., frequent stopping at bus stops for passengers boarding and alighting, repetitive accelerations and decelerations from and to bus stops. Besides, there is a natural difference between buses and general automobiles due to their different operating abilities and speed limits. Fortunately, the relationship between the two kinds of travel times has been well explored in the transportation domain [9], [10], [23]. We have tested different models, both linear and non-linear models. The experimental results suggest that compared to non-linear models, the linear models are good enough for the travel time transformation, which are easy to learn without complex parameter settings and overfitting issues. Thus we use one classical linear traffic model proposed in [9] to estimate $\mathcal{T}_A$ from $\mathcal{T}_B$:

$$\mathcal{T}_A = a + b \times \mathcal{T}_B, \quad (3)$$

where $a = \frac{road\ length}{free\ travel\ speed}$ represents the average travel time by a typical automobile on the road segment when there is little or no traffic, and b represents the traffic congestion effect on $\mathcal{T}_A$ (as measured by the travel time of public buses). Based on historical traffic data, we can learn the parameter b using ordinary least-square technique. According to our experiments, we find that the best values of b for different road segments fall in a narrow range $[0.13, 0.18]$. In our system, we set $b = 0.15$ for all road segments for simplicity. The average travel speed of general automobiles on the road segment can thus be estimated as $v_A = \frac{road\ length}{\mathcal{T}_A}$. Although some cities may design dedicated lanes for a few special buses, e.g., bus rapid transit, there still are many ordinary buses traveling on the roads. To guarantee the correctness of our estimations, our system can filter out the trip reports from the special buses by comparing the mapped bus trajectory with special bus routes.

For each road segment, it may be simultaneously covered by multiple bus routes, which leads to several speed estimations as our system collects trip reports from massive mobile phones on various buses. Thus our system adopts a Bayesian method [23] to continuously update the traffic

estimation by carefully combining previous estimation and current estimation from new data input. With the variance of previous average speed $\bar{v}_0$ as σ_0^2 and the variance of new average speed $\bar{v}$ as σ^2, the updated speed estimation is normal with average speed $\bar{v}_{new}$ and variance σ_{new}^2 as

$$\bar{v}_{new} = \frac{\frac{\bar{v}_0}{\sigma_0^2} + \frac{\bar{v}}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}}, \quad \sigma_{new}^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}}, \tag{4}$$

which uses the inverse of estimation variance to weigh previous estimation and new estimation. The updating procedure produces sequential travel speed estimations with newly received traffic samplings from bus riders.

We combine the traffic estimations of all road segments to derive a complete traffic map. In particular we weight the overlapping road segments in combining their estimations. Say two road segments, AC and BC, share the common part IC where I is the intersection point of the two segments. When combining the traffic conditions of AC and BC, we divide them into AI, BI and IC. We weight $\bar{v}_{AC}$ and $\bar{v}_{BC}$ based on the position of I to derive the speed estimation $\bar{v}_{IC}$, i.e.,

$$\bar{v}_{IC} = \frac{\alpha \times \bar{v}_{AC} + \beta \times \bar{v}_{BC}}{\alpha + \beta},$$

where $\alpha = \frac{d_{IC}}{d_{AC}}$, $\beta = \frac{d_{IC}}{d_{BC}}$ and d_{ij} is the road length between i and j. Meanwhile the travel speed on AI and BI, $\bar{v}_{AI}$ and $\bar{v}_{BI}$, can be calculated as

$$\bar{v}_{AI} = \frac{d_{AI}}{\frac{d_{AC}}{\bar{v}_{AC}} - \frac{d_{IC}}{\bar{v}_{IC}}}, \quad \bar{v}_{BI} = \frac{d_{BI}}{\frac{d_{BC}}{\bar{v}_{BC}} - \frac{d_{IC}}{\bar{v}_{IC}}}.$$

The system will update the travel speed estimations on all road segments with a period of $T = 15\,mins$, which is a sufficiently fine-grained and adopted by many previous works for traffic estimation [35], [36], [40].

V. Online Database Construction

Till now we assume there exists a cellular fingerprint database for all bus stops, which is supposed built offline with burdensome war-driving. In this section, we show that we can online construct the database from participatory sensing data.

As a matter of fact, the bus movement is always constrained by its bus route and meanwhile each bus route can be sparsely represented by only a few of its all bus stops. If we pre-know the fingerprints of such a small initial set of bus stops, we can map some uploaded sequences of cellular samples to certain bus routes, and then automatically label the unknown bus stops. This idea is feasible due to following observations. First, according to our measurement study result in Fig. 2(c), different bus stops are usually highly distinguishable on their cellular fingerprints, i.e., the chance of matching a sequence of cellular samples to one particular bus route is fairly large. Second, each bus route has its own unique road segments (e.g., in Fig. 2(a)) and a few (2 or 3) representative bus stops on the segments can uniquely identify the entire route.

Incorporating this idea, we can bootstrap the database from a small set of bus stops for which we manually collect their cellular fingerprints and grow the database with each received sequence of cellular samples from bus riders. Considering

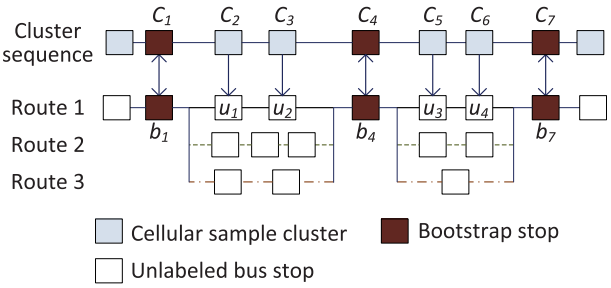


Fig. 9. An example of the construction process. The cellular fingerprints of the unlabeled bus stop u_1, u_2, u_3, u_4 are inferred from the cluster sequence. The pre-known bootstrap bus stop b_1, b_2, b_3 are used as references during the construction.

that more than one cellular samples can be detected at a bus stop, we classify an uploaded sequence of n cellular samples $E = \{e_1, e_2, \ldots, e_n\}$ into different clusters based on Eq. (1) in Section IV-B.2, while we weigh the relationship between two cellular samples e_i and e_j as $L(e_i, e_j) = \frac{Ma(e_i,e_j)}{s_0}$, where $Ma(e_i, e_j)$ is their matching score calculated using the modified Smith-Waterman algorithm [39] and s_0 is the maximum possible similarity score. Each cluster finally corresponds to an actual bus stop. Then a set of clusters $C = \{C_1, C_2, \ldots, C_m\}$ is generated. We denote a total of E_k cellular samples in cluster C_k as $\{e_k(1), e_k(2), \ldots, e_k(E_k)\}$. Assume we know the cellular fingerprints of N pre-known bootstrap bus stops $R = \{b_1, b_2, \ldots, b_N\}$. We denote cluster C_k is matched with bus stop b_i if

$$(\forall\ e_k(j) \in C_k)\ Ma(e_k(j), b_i) > \epsilon, \tag{5}$$

where ϵ is a parameter to guarantee the rigid matching between cellular sample and bus stop fingerprint.

We then follow two steps to find the correspondence between the clusters and the unlabeled bus stops. First, we match all the clusters of C to the bootstrap bus stops R to find a total of h matched clusters $\{C_{x_1}, C_{x_2}, \ldots, C_{x_h}\} \subseteq C$. As an example depicted in Fig. 9, $\{C_1, C_4, C_7\}$ are 3 matched clusters. Then we find the bus route B that passes all the matched bus stops $R_X = \{b_{x_1}, b_{x_2}, \ldots, b_{x_h}\}$ ($\{b_1, b_4, b_7\}$ in Fig. 9) and correspond the unlabeled bus stops $\{u_1, u_2, u_3, u_4\}$ to the sample clusters $\{C_2, C_3, C_5, C_6\}$ respectively. Although not usual cases, there could be more than one bus routes that contain the matched bus stops (e.g., route 1-3 in Fig. 9). In such a case, we match the most possible bus route according to the number of unlabeled bus stops $\{u_1, u_2, \ldots, u_l\}$ between every pair of matched bus stops (e.g., $b_1 \sim b_4$ and $b_4 \sim b_7$ in Fig. 9). We denote the number of bus stops on bus route B between stop b_i and stop b_j as $N_B(b_i, b_j)$. In the candidate bus routes $\{B_1, B_2, \ldots, B_w\}$ sharing the matched bus stops R_X, the most possible bus route is estimated as

$$\begin{aligned} B^* &= \arg\min_{B_{j:1\sim w}} W(B_j) \\ &= \arg\min_{B_{j:1\sim w}} \sum_{i=1}^{h-1} \left| \frac{N_{B_j}(b_{x_{i+1}}, b_{x_i}) - (x_{i+1} - x_i)}{x_{i+1} - x_i} \right|, \end{aligned} \tag{6}$$

where we weigh a candidate bus route using the difference between its number of unlabeled bus stops and the number of clusters of the uploaded data. If an unlabeled bus stop u is

corresponded to a cluster C_k, we set the fingerprint of u as the cellular sample e_k^* which has the highest matching score with the rest samples in C_k.

$$e_k^* = \arg \max_{e_k(j:1\sim E_k)} \frac{\sum_{i=1}^{E_k} Ma(e_k(i), e_k(i))}{E_k}. \quad (7)$$

This online database construction method exploits the essential information redundancy of bus stops included in each bus route. In Section IV-B.3, we utilize such information redundancy to filter the noises in mapping the cellular sample clusters to bus stops. Here because we are building the cellular fingerprint database that will serve as a basis for following trajectory mapping, we set much stricter criteria to ensure we only make use of the data sources with "good" quality. In particular, we rely on the following three rules to filter "bad" data sources. (I) Cluster C_k is determined as a "bad" cluster if $(\exists e_k(i), e_k(j) \in C_k), Ma(e_k(i), e_k(j)) < 3$, i.e., the samples in the cluster are not very similar to each other. (II) We set $\epsilon = 3$ in Eq. (5) to guarantee the matching accuracy of cluster C_k and bus stop b_i. (III) Unlike in Section IV-B.3, we strictly match the cellular sequences to individual bus route but not the combinations of bus stops to ensure the data quality. We do not use the best matching route B^* derived in Eq. (6) if $(\exists 1 \leq j \leq w, B_j \neq B^*) \frac{W(B_j)}{W(B^*)} < 2$, i.e., B^* must be an obvious match to be included.

This online database construction process is performed for several rounds. The bus stops labeled in previous rounds are treated as bootstrap stops for the next round. The growing process ends until there is no new bus stops labeled in a new round. The initial bootstrap bus stops can be selected as those most distinguishable ones of different bus routes from an online bus route map. For one particular bus route, the bus stop that is not shared by other bus routes can be considered as a bootstrap bus stop. For the initial set of bus stops, we manually measure their cellular signatures and label them.

VI. Implementation and Evaluation

In this section, we evaluate the performance of our system based on a prototype implementation on Android platform. We first introduce the implementation details and experimental settings. Then we present the evaluation results of the bus stop detection and identification methods, and analyze the traffic estimation results. Finally, we evaluate the online database construction method and investigate system overhead.

A. Experiment Settings

We have implemented the bus trip data collection App on Android platform (with Android version 4.0.3). Controlled experiments are conducted with three types of mobile phones, i.e., HTC Sensation XE, HTC Desire S, and Google Nexus One. These mobile phones are common phones equipped with accelerometer sensors and support 16-bit 44.1 kHz audio signal sampling from microphones. Their memory and CPU capacity are powerful enough for the light computation and bus trip sensing involved in our application. The phone types of the participants are more diverse. The HTC and Samsung phones dominate. As our system requires no particular hardwares, thus the proposed method could be easily implemented on other OS and hardware platforms, e.g., Apple iPhones and Windows Phones. In the future, we will implement our systems in iOS system due to the large market share of iPhones. We believe at that time more participants can join in the traffic sensing with our App, and provide better traffic condition estimations. We also implement the backend bus trip data processing and analysis services in Java executing on a server, i.e., the DELL Precision WorkStation T3500, for our experiments.

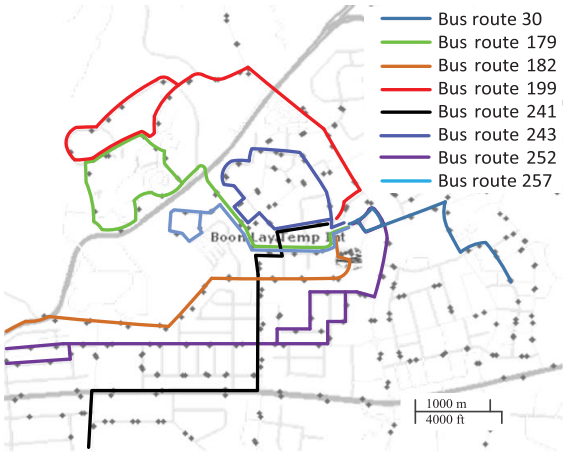


Fig. 10. 8 concerned bus routes in the $\sim 28\,km^2$ testing area in our experiments.

In Singapore, the public bus transit system serves millions of citizens every day and covers most parts of the road network [39]. The bus services are primarily provided by SBS Transit[5] and SMRT Corporation,[6] both of which are commercial transit companies in Singapore. Fig. 10 demonstrates our experiment area with a size of about $7\,km \times 4\,km$, which owns more than 20 bus routes with periodically transit bus services covering most of the roads in this area. Our experiments concern on 8 bus routes, including bus route 179, 182, 199, 241, 243, 252, 257 and partial part of route 30, as shown in Fig. 10. The selected bus routes cover a great portion of the roads in the area and thus can provide fine-grained traffic estimation results. In the testing area, we conduct various experiments to evaluate our system and methods. All the experiments lasted for more than 2 months.

1) Data Collection: The input of our system includes the cellular fingerprints of bus stops and the real-time bus trip sensing data collected from mobile phones of bus riders. The experiments involve the following two kinds of data resources:

a) Manual collection: For the 8 experimental bus routes, we manually collect the cellular fingerprints for their bus stops. Specifically, multiple cellular samples are primitively collected for each bus stop and we only keep the sample with the highest similarity score with the others as the bus stop cellular fingerprint stored in the database. We use these cellular samples with high-quality for the feasibility study of system design in Section III and for the performance evaluation of bus stop detection and identification methods in this section.

[5]SBS Transit in Singapore. http://www.sbstransit.com.sg/.
[6]SMRT Corporation in Singapore. https://www.smrt.com.sg/.

TABLE II
BEEP DETECTION RATIO

Scenario	1m	2m	3m	4m	5m	Second floor
In hand	98.3	96.6	92.8	91.3	86.1	78.3
In bag	96.0	94.2	92.5	87.7	74.0	68.5

TABLE III
BUS STOP IDENTIFICATION ERROR

Route	total	errors	error rate	1 stop error	2 stops error
30	58	3	3.45%	3	0
182	121	8	6.61%	5	2
199	93	5	5.38%	4	1
241	80	6	7.50%	5	1

b) Participatory collection: We have 122 participants in total, mainly undergraduate students and volunteers, involved in our experiments to contribute their real-time bus trip information to our system. The participants installed our data collection App[7] on their mobile phones to collect bus trip data and periodically upload the sensory data through WiFi or 3G to our backend server. Due to the small number of participants in the first month, we have received limited bus trip data from bus riders. The initially collected data mainly come from the popular bus routes. Therefore, we encouraged the participants with vouchers to intensively take buses from the 8 selected bus routes for 9 days, which provides richer sensory data for a comprehensive performance evaluation of our system. We also investigate the performance of our system in both sparse and intensive data collection scenarios (measured with number of bus stops) in Section VI-C.

B. Bus Stop Detection and Identification Performance

Bus stops are detected according to the beeps and identified by fingerprint matching. The length of a typical bus in Singapore is $10 \sim 12\,m$ and the width is about $2.5\,m$. There are 4 card readers placed at two sides of the front and back doors. We do experiments at different locations on the bus to test the beep detection accuracy. Some of the public transit buses are double-decker so we also conduct the audio detection experiments on the second floor. For all scenarios, we consider that mobile phones may be placed inside bags or held in hand.

We test the audio detection method more than 40 times at each scenario and summarize the average detection ratio in Table II. The detection ratio is above 90% when the distance to card readers is within $4\,m$, even when the mobile phones are placed inside bags. As the distance increases, the detection ratio decreases. For the passengers seated at the second floor of the double-decker bus, the average detection ratio are about 78% in hand and 69% in bags, respectively. Notice that our approach essentially tolerates some beeping missing events.

To understand the accuracy of our bus stop identification algorithm, we collect massive cellular samples at the bus stops for 8 rounds for all bus routes. For a specific bus route, we take the cellular signals collected in one of the 8 rounds as bus stop fingerprints (which should be stored in database), and view the cellular signals of the rest 7 rounds as testing data to identify the bus stops. We calculate the bus stop identification error and present the statistical results for 4 bus routes in Table III. We omit the other 4 bus routes in Table III as they have similar results. For all the 4 bus routes, the bus identification error is quite small, i.e., $<8\%$. Among the 4 bus routes, we find that our algorithm performs the worst on bus route 241 with the largest identification error as 7.50%. We carefully analyze 80 cellular signal sets of bus route 241 and find that there are 6 mis-identified cases, among which 5 cases are 1 bus stop away from the actual bus stop and only 1 case with 2 bus stop error. In general, our bus identification algorithm works well with rare mis-identification cases.

[7]Now it has been put up on Google Play (Jurong Bus Traffic).

C. Traffic Estimation Performance

In this subsection, we evaluate our traffic estimation method with sensory data collected from participants' mobile phones. Although we rely on traffic data collected from buses, we have transformed the bus travel information to the general travel speeds according to the method in Section IV-C. The following analysis is based on the finally transformed travel speeds.

We run our traffic estimation method with the sensory data of one day when the most participants are encouraged to intensively take buses and demonstrate 3 snapshots of the traffic maps for 8:30AM, 15:00PM and 19:00PM of that day, respectively, in Fig. 11 (top). We classify the travel speeds of automobiles into 5 levels as shown in Fig. 11(a). From Fig. 11 (top), we find that the traffic speeds of most roads in the studied area fall in level of $30-50\,km/h$. In general, the traffic conditions of roads vary considerably. Taking the traffic map for 8:30AM shown in Fig. 11(a) as an example, we find that road segments of the left and bottom area have the best traffic conditions with traffic speeds higher than $50\,km/h$, while road segments in the middle area have the worst traffic condition with traffic speeds about $20km/h$. In addition, we find the traffic conditions also temporally vary as shown by the traffic maps of the 3 time points. In general, road segments in the studied area are in good traffic conditions at 15:00PM (Fig. 11(b)) while their traffic conditions become worse at 19:00PM (Fig. 11(c)) which belongs to the evening peak hours in Singapore. However, we still find few road segments at 15:00PM are in poor traffic conditions with traffic speed $<20\,km/h$. For the traffic maps at 8:30AM and 19:00PM, although there are many low-speed road segments, their speed distributions are very different from each other. For the low-speed road segments at 8:30AM as shown in Fig. 11(a), we find they are close to each other in 2 main roads which lead heavy traffics to the highway in the early peak. In contrary, the congested road segments at 19:00PM are more dispersed. The results in Fig. 11 (top) prove that the bus network indeed provides high road coverage for fine-grained traffic map. Even only 8 bus routes can cover $>50\%$ major roads of the studied area. Compared with Google traffic map for the area, i.e., Fig. 11(d), our system provides much higher road coverage ratio. With traffic data collected by participants from more bus routes, our system will derive more complete traffic conditions.

We further test the system performance with the 19:00PM data of the same day using only 70% and 50% bus stop references and plot the results in Fig. 11(e) and (f), respectively. The bus stop references are randomly selected.

(a) 8:30AM. (b) 15:00PM. (c) 19:00PM.

(d) Google traffic map. (e) 19:00PM with 70% stops. (f) 19:00PM with 50% stops.

Fig. 11. Traffic map at different time of a day (a)-(c); Google traffic map (d); generated with partial bus stop references (e)-(f).

When we use 70% bus stop references, the estimated traffic conditions do not degrade much and we can see that the overall traffic fidelity is similar to that referencing all bus stops in Fig. 11(c). When the fraction of used bus stops drops to 50%, the estimated traffic map shown in Fig. 11(f) becomes rougher. With fewer bus stops as landmarks, the roads are segmented in a more coarse-grained manner, i.e., longer road segments. As a result, the traffic conditions in Fig. 11(f) are not as fine-grained as in Fig. 11(c). Even that, both Fig. 11(d) and Fig. 11(f) still provide us the general traffic information.

We compare our estimated traffic speeds with official traffic speeds acquired from the Land Transport Authority (LTA) of Singapore.[8] The LTA accumulates traffic data from traffic reports of more than 10,000 roving taxis and other data sources including traffic cameras and inductive loop detectors within a time slot of 15 $mins$ and derives the real-time average traffic speeds. The obtained official traffic speeds fully cover the experiment days and the testing area. We also compare our results with Google Maps' traffic data. Google Maps provide the live traffic visualization on the maps but they only give 4 coarse traffic levels (i.e., *very slow*, *slow*, *normal*, and *fast*) instead of providing detailed road traffic speed to the end users. In addition to that, there is no interface[9] for public access of Google Maps' real-time traffic speed data. Therefore, we manually obtain the coarse congestion indicator data from Google Maps that cover the experiment days and testing roads, and then compare them with our estimation

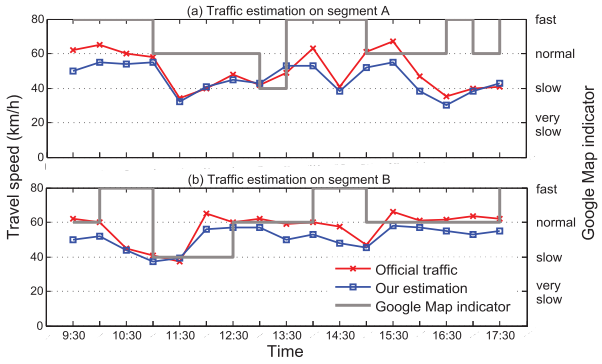


Fig. 12. Traffic estimation compared with official traffic data and Google Maps' indicator.

results. We pick 2 typical road segments (A and B as depicted in Fig. 11(d)) and plot their corresponding traffic speeds for the time period from 9:30AM to 17:30PM on one experiment day from three sources for comparison in Fig. 12. Specifically, we compare our estimated traffic speed of automobiles (v_A), traffic speed (v_T) from LTA, and the Google Maps' indicators on the two road segments. For v_A and v_T, we plot their average speeds with a time window of 15 $mins$. Fig. 12 shows that Google Maps only provide rough traffic levels while both v_A and v_T can provide more fine-grained information.

By comparing v_A and v_T in Fig. 12, we find some interesting relationships between them. We categorize the traffic speeds of v_A into 3 levels, i.e., low-speed ($<45\,km/h$), medium-speed ($40 \sim 50\,km/h$) and high-speed ($>50\,km/h$), for the following comparison between v_A and v_T. When v_A falls in the low speed level, we can see v_A matches v_T well.

[8]LTA of Singapore. http://www.lta.gov.sg.

[9]Although Google Maps Directions APIs enable the users to query the travel routes and travel durations, these APIs do not provide the detailed traffic speed of any specific road segment.

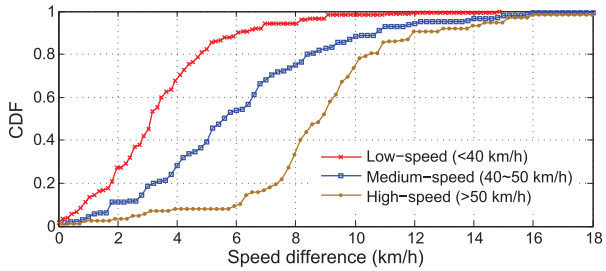


Fig. 13. Statistics of speed difference Δv between v_A and v_T.

When v_A is in the high speed level, a gap exists between v_A and v_T. The reason could be that v_A and v_T are derived from bus travel data and taxi travel data, which behave quite differently. Buses are normally slower while taxis travel more aggressively. Though we have transformed bus information to general automobile speed v_A, the taxi speed v_T is still much higher in light traffic scenario. However, both v_A and v_T have the similar variation pattern as shown in Fig. 12.

We further explore the relationship between v_A and v_T with all the 2-month experiment data. We calculate the speed difference Δv between v_T and v_A for all time durations and road segments of the studied area when both values are available. We still categorize the traffic speeds into 3 levels according to v_A and plot the statistics separately in Fig. 13. The results show that Δv is the smallest (about $3 \sim 5$) for low-speed level and the largest (about $8 \sim 12$) for high-speed level. For medium-speed level, Δv is more disperse in the range of $0 \sim 12$. Fig. 13 suggests that v_A is an effective measure for traffic conditions, especially indicative for heavy traffics and congestions which generally result in low traffic speeds.

D. Online Database Construction Performance

We evaluate the crowdsourcing online construction method for cellular fingerprint database by randomly setting different fractions of bus stops as bootstrap stops and growing up other bus stop fingerprints. The measured bus stop fingerprints of the 8 experimental bus routes are used as ground truths. The uploaded data from participants during the 2-month experiment period are used to feed our online database construction algorithm. Every discovered bus stop is compared with the ground truth bus stop and if the similarity of their fingerprints are larger than threshold 3 we treat it correct. For each setting, we run 100 independent experiments and report the statistical result of discovery ratios. For each run, a set of initial bootstrap bus stops is randomly selected. The algorithm terminates when there is no new discovered bus stops. We record the running rounds and report the average running rounds for each setting.

Fig. 14 plots the average discovery ratio, which is calculated as the ratio between the number of correctly fingerprinted bus stops and the total number of bus stops, and the number of running rounds to achieve the converged discovery ratio. We vary the fraction of bootstrap bus stops from 2% to 60%. We can see from Fig. 14 that when the fraction of bootstrap bus stops is small, the discovery ratio increases significantly with the growth of the fraction. The fingerprints of 71% bus stops are discovered from the participatory sensing data with 20% bootstrap bus stops. When the fraction of bootstrap bus stops further increases, the increase of discovery ratio becomes slow. Besides, we find that when the fraction is 10%-20%, the algorithm runs for the most rounds to be converged (see the "# of rounds" in Fig. 14). In other cases, however, the algorithm terminates earlier, which implies a small number of bus stops are the representative ones. We can discover the fingerprints of a majority of the bus stops with more than 25% bootstrap bus stops. Since the participatory data used for the online database construction process in our system is limited due to the small number of participants, we believe that our system can achieve much better performance if more participants are involved.

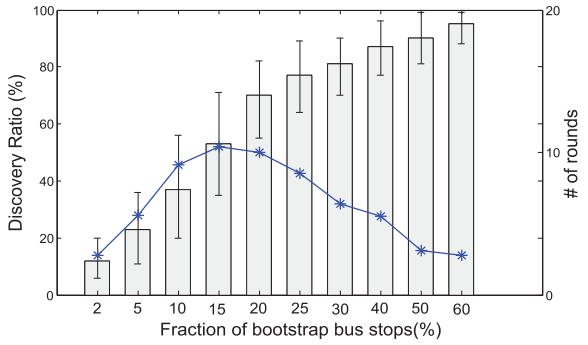
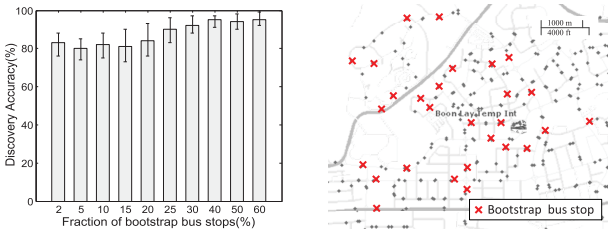


Fig. 14. Database construction performance.



(a) (b)

Fig. 15. Discovery accuracy of the online database construction and an instance distribution of the bootstrap bus stops. (a) Discovery accuracy for different factions of bootstrap bus stops. (b) An instance distribution for the online database construction.

The discovery accuracy is summarized in Fig. 15(a), where we find that the overall accuracy is higher than 80%. If the bootstrap bus stops are properly chosen, we can achieve higher discovery ratio and accuracy even with much lower fraction of bootstrap bus stops. When the bootstrap fraction is 20%, we show an empirically optimal bus stop distribution case in Fig. 15(b), where the red cross are the bootstrap bus stops. We get 87% discovery ratio and 91% accuracy grown up database from those bootstrap bus stops. This bus stop distribution is more dispersed compared with other cases, and contributes to making better use of the participatory sensing data. This observation can help us to determine the appropriate bootstrap bus stops which are exclusive bus stops of certain bus routes and are dispersedly distributed in space.

E. System Overhead

The Goertzel algorithm [5] dominates the computation overhead on mobile phones. In principle, the Goertzel algorithm is computationally efficient than the FFT algorithm for frequency extraction. The complexity of Goertzel algorithm and FFT is

TABLE IV
POWER CONSUMPTION COMPARISON (IN mW)

Sensor settings	HTC Sensation	Nexus One
No sensors	71 (6)	84 (5)
Cellular 1 Hz	72 (6)	85 (8)
GPS 0.05 Hz	304 (32)	333 (41)
Cellular+Mic(Goertzel)	182 (20)	196 (22)
GPS+Mic(Goertzel)	447 (45)	443 (57)
Cellular+Mic(Goertzel)+Acc	191 (24)	203 (25)

$O(K_g NM)$ and $O(K_f N \log N)$, respectively, where K_g and K_f are the "cost of operation per unit", M is the number of measured frequencies, and N is the sampling values. In general, the factor K_f is often much larger than K_g [5]. Thus when M is smaller than $\log N$, Goertzel algorithm significantly outperforms FFT. By setting the sampling rate of microphone as $8\,kHz$ for bus detection, the Goertzel algorithm saves more than $60\,mW$ power than FFT for data collection.

We measure the power consumption of two types of mobile phones (i.e., HTC Sensation and Nexus One) under different sensor settings using the Monsoon power monitor. For each setting, we switch off the mobile phone screen, and record the consumed energy over a period of $10\,mins$ and calculate the average power consumption as $\frac{energy}{time}$. Both average power consumption and relative standard deviation (in the parentheses) are reported in Table IV. We take the power consumption when no sensors are activated as the baseline. We can see that sampling cellular signals consume negligible power when compared to the baseline, e.g., 72 mW and 71 mW for HTC phone, respectively. Since mobile phones always maintain connections to nearby cell towers to support telephone calls and SMS service and thus sample the cellular signals at a high frequency, i.e., 1 Hz. As a result, our system takes the almost free lunch and actually introduces marginal energy consumption. The average power consumption for GPS tracking at a sampling rate of 0.05 Hz is as high as $304\,mW$ for HTC and $333\,mW$ for Nexus One. Sampling one GPS signal every 20 seconds is already very low for the vehicle tracking [28]. The trip data collection via cellular signals only consumes $182\,mW$ for HTC and $196\,mW$ for Nexus One in total. The power consumption, however, is as high as $\sim 450\,mW$ if we replace cellular signal with GPS signal for the bus trip tracking. Notice that the microphone on the mobile phone has to be kept always on for bus detection no matter the cellular signal or GPS signal is used for vehicle tracking. As our system uses accelerometer on mobile phones to continuously filter out noisy data, we also include the power consumption of accelerometer. The overall power consumption of data collection using our App becomes $191\,mW$ for HTC and $203\,mW$ for Nexus One, with slight increase.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents the design, implementation and evaluation of a participatory urban traffic monitoring system, which leverages public buses to probe traffic conditions. The system decomposes the traffic sensing tasks to participatory bus riders by utilizing lightweight sensing resources from their mobile phones, and exploits bus route constraints and bus stop references to derive the traffic map. We implement our system and conduct comprehensive evaluation of 2-month period in Singapore. The results demonstrate the effectiveness and feasibility of our system for urban traffic monitoring.

As future works, we plan to derive the complete traffic of a region from traffic conditions of road segments covered by bus routes. In addition, we would like to design an appropriate incentive mechanism to encourage more bus riders' participation for consistent and good performance. The bus drivers could be encouraged as the initial users to bootstrap our system. Besides, we have published our App on Google Play and we expect more experimental studies in other areas.

REFERENCES

[1] M. T. Asif *et al.*, "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, Apr. 2014.

[2] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1816–1825, Jul. 2016.

[3] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *Proc. ACM SenSys*, 2012, pp. 141–154.

[4] R. K. Balan, K. X. Nguyen, and L. Jiang, "Real-time trip information service for a large taxi fleet," in *Proc. ACM MobiSys*, 2011, pp. 99–112.

[5] R. Beck, A. G. Dempster, and I. Kale, "Finite-precision Goertzel filters used for signal tone detection," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 7, pp. 691–700, Jul. 2001.

[6] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, "EasyTracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones," in *Proc. ACM SenSys*, 2011, pp. 68–81.

[7] N. Brouwers and K. Langendoen, "Pogo, a middleware for mobile phone sensing," in *Proc. Middleware*, 2012, pp. 21–40.

[8] A. T. Campbell *et al.*, "The rise of people-centric sensing," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 12–21, Jul./Aug. 2008.

[9] P. Chakroborty and S. Kikuchi, "Using bus travel time data to estimate travel times on urban corridors," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1870, pp. 18–25, Jan. 2004.

[10] B. Coifman and S. Kim, "Using transit vehicles to measure freeway traffic conditions," in *Proc. AATT*, 2006, pp. 13–16.

[11] R. Ganti, I. Mohomed, R. Raghavendra, and A. Ranganathan, "Analysis of data from a taxi cab participatory sensor network," in *Proc. Int. Conf. Mobile Ubiquitous Syst., Comput., Netw., Services*, 2011, pp. 197–208.

[12] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "GreenGPS: A participatory sensing fuel-efficient maps application," in *Proc. ACM MobiSys*, 2010, pp. 151–164.

[13] H. Gao *et al.*, "A survey of incentive mechanisms for participatory sensing," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 918–943, 2nd Quart., 2015.

[14] W. Hedgecock, M. Maroti, J. Sallai, P. Volgyesi, and A. Ledeczi, "High-accuracy differential tracking of low-cost GPS receivers," in *Proc. ACM MobiSys*, 2013, pp. 221–234.

[15] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proc. ACM SenSys*, 2013, Art. no. 13.

[16] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2551–2572, Oct. 2015.

[17] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proc. ACM MobiSys*, 2010, pp. 285–298.

[18] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu, "Mining road network correlation for traffic estimation via compressive sensing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1880–1893, Jul. 2016.

[19] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[20] D. Maier and A. Kleiner, "Improved GPS sensor model for mobile robots in urban terrain," in *Proc. IEEE ICRA*, May 2010, pp. 4385–4390.

[21] A. Musa and J. Eriksson, "Tracking unmodified smartphones using Wi-Fi monitors," in *Proc. ACM SenSys*, 2012, pp. 281–294.

[22] S. Oh, S. Ritchie, and C. Oh, "Real-time traffic measurement from single loop inductive signatures," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1804, pp. 98–106, Jan. 2002.
[23] W. Pu, J. Lin, and L. Long, "Real-time estimation of urban street segment travel time using buses as speed probes," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2129, pp. 81–89, Dec. 2009.
[24] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. ACM/IEEE IPSN*, Apr. 2010, pp. 105–116.
[25] T. N. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 90–98, Jun. 2003.
[26] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones," in *Proc. ACM SenSys*, 2010, pp. 85–98.
[27] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. USENIX NSDI*, 2011, pp. 267–280.
[28] A. Thiagarajan *et al.*, "VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. ACM SenSys*, 2009, pp. 85–98.
[29] D. Tomaras, I. Boutsis, and V. Kalogeraki, "Travel time estimation in real-time using buses as speed probes," in *Proc. IEEE PerCom Workshops*, Mar. 2015, pp. 63–68.
[30] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with Wi-Fi!" *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2907–2920, Nov. 2016.
[31] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "ARTsense: Anonymous reputation and trust in participatory sensing," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2517–2525.
[32] Y. Wang, B. Krishnamachari, Q. Zhao, and M. Annavaram, "Markov-optimal sensing policy for user state estimation in mobile devices," in *Proc. ACM/IEEE IPSN*, Apr. 2010, pp. 268–278.
[33] Y. Wang, X. Liu, H. Wei, G. Forman, and Y. Zhu, "CrowdAtlas: Self-updating maps for cloud and personal use," in *Proc. ACM MobiSys*, 2013, pp. 469–470.
[34] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. ACM SIGKDD*, 2014, pp. 25–34.
[35] B. Yang, C. Guo, and C. S. Jensen, "Travel cost inference from sparse, spatio temporally correlated time series using Markov models," *VLDB Endowment*, vol. 6, no. 9, pp. 769–780, 2013.
[36] B. Yang, M. Kaul, and C. S. Jensen, "Using incomplete information for complete weight annotation of road networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1267–1279, May 2014.
[37] J. Yang, A. Varshavsky, H. Liu, Y. Chen, and M. Gruteser, "Accuracy characterization of cell tower localization," in *Proc. ACM UbiComp*, 2010, pp. 223–226.
[38] M. Zhao, T. Ye, R. Gao, F. Ye, Y. Wang, and G. Luo, "VeTrack: Real time vehicle tracking in uninstrumented indoor environments," in *Proc. ACM SenSys*, 2015, pp. 99–112.
[39] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proc. ACM MobiSys*, 2012, pp. 379–392.
[40] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.
[41] Y. Zou, G. Wang, K. Wu, and L. M. Ni, "SmartScanner: Know more in walls with your smartphone!" *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2865–2877, Nov. 2016.

Zhidan Liu (M'15) received the B.E. degree in computer science and technology from Northeastern University, Shenyang, China, in 2009 and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2014. He is a Research Fellow with Nanyang Technological University, Singapore. His research interests include distributed sensing and mobile computing, big data analytics, and urban computing.

Shiqi Jiang received the B.E. degree in computer science and technology from Zhejiang University, Hangzhou, China in 2013. He is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include mobile system, pervasive computing, localization, and cellular network communications.

Pengfei Zhou (M'11) received the B.E. degree from the Automation Department, Tsinghua University, Beijing, China, in 2009, and the Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2015. His research interests include mobile computing and systems, localization, and cellular network communications.

Mo Li (M'06) received the B.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2004 and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2009. He is an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include networked and distributed sensing, wireless and mobile, cyber-physical systems, smart city, and urban computing.