coSense: Collaborative Urban-Scale Vehicle Sensing Based on Heterogeneous Fleets

XIAOYANG XIE, Rutgers University
YU YANG, Rutgers University
ZHIHAN FANG, Rutgers University
GUANG WANG, Rutgers University
FAN ZHANG, Rutgers University
FAN ZHANG†, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
YUNHUAI LIU, Peking University, China
DESHENG ZHANG, Rutgers University, USA

The real-time vehicle sensing at urban scale is essential to various urban services. To date, most existing approaches rely on static infrastructures (e.g., traffic cameras) or mobile services (e.g., smartphone apps). However, these approaches are often inadequate for urban scale vehicle sensing at the individual level because of their static natures or low penetration rates. In this paper, we design a sensing system called coSense to utilize commercial vehicular fleets (e.g., taxis, buses, and trucks) for real-time vehicle sensing at urban scale, given (i) the availability of well-equipped commercial fleets sensing other vehicles by onboard cameras or peer-to-peer communication, and (ii) an increasing trend of connected vehicles and autonomous vehicles with periodical status broadcasts for safety applications. Compared to existing solutions based on cameras and smartphones, the key features of coSense are in its high penetration rates and transparent sensing for participating drivers. The key technical challenge we addressed is how to recover spatiotemporal sensing gaps by considering various mobility patterns of commercial vehicles with deep learning. We evaluate coSense with a preliminary road test and a large-scale trace-driven evaluation based on vehicular fleets in the Chinese city Shenzhen, including 14 thousand taxis, 13 thousand buses, 13 thousand trucks, and 10 thousand regular vehicles. We compare coSense to infrastructure and cellphone-based approaches, and the results show that we increase the sensing accuracy by 10.1% and 16.6% on average.

CCS Concepts: • **Networks** → **Sensor networks**; • **Information systems** → *Location based services*;

Additional Key Words and Phrases: Heterogeneous Fleets, Vehicle Sensing, Mobility Patterns

ACM Reference Format:
Xiaoyang Xie, Yu Yang, Zhihan Fang, Guang Wang, Fan Zhang, Fan Zhang, Yunhuai Liu, and Desheng Zhang. 2018. coSense: Collaborative Urban-Scale Vehicle Sensing Based on Heterogeneous Fleets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 196 (December 2018), 25 pages. https://doi.org/10.1145/3287074

196

† Also with Beidou Intelligent Technology Co., Ltd., Shenzhen, China

Authors' addresses: Xiaoyang Xie, Rutgers University, xx88@cs.rutgers.edu; Yu Yang, Rutgers University, yy388@cs.rutgers.edu; Zhihan Fang, Rutgers University, zhihan.fang@cs.rutgers.edu; Guang Wang, Rutgers University, gw255@cs.rutgers.edu; Fan Zhang, Rutgers University, Valiantzf@gmail.com; Fan Zhang, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong, 518055, China, zhangfan@siat.ac.cn; Yunhuai Liu, Peking University, Beijing, Beijing, China, yunhuai.liu@pku.edu.cn; Desheng Zhang, Rutgers University, New Brunswick, New Jersey, 08854, USA, desheng.zhang@cs.rutgers.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
2474-9567/2018/12-ART196 $15.00
https://doi.org/10.1145/3287074

1 INTRODUCTION

Nowadays, vehicles are essential components of our daily life, e.g., we have 1.2 billion vehicles in the world by 2015, and this number is projected to increase to 2 billion by 2035 [8]. This phenomenon is more obvious in urban areas, e.g., in New York City, there are 2.9 million vehicles entering the city every day [9]. All these vehicles in urban areas lead to various challenges, e.g., traffic congestion and energy consumption [29]. To address these challenges, it is essential to understand the real-time mobility patterns of these vehicles, e.g., real-time travel time and traffic volume by sensing urban-scale vehicles.

However, sensing urban-scale vehicles in real time is extremely challenging due to fine-grained temporal coverage (e.g., 10 seconds intervals), large-scale spatial coverage (e.g., all road segments), and high quantitative coverage (e.g., all vehicles), which requires a major investment of infrastructures. The existing approaches for vehicle sensing are mostly based on (i) static infrastructures, e.g., cameras [10] and RFID [16]; (ii) mobile services, e.g., vehicular manufacturers' services (e.g, OnStar [11] and Ford Sync [44]), smartphone apps (e.g., Google Maps [34]), navigators (e.g., Garmin [38]) or sensing devices from auto insurance companies [56]. However, the static infrastructure-based systems can only capture vehicles at limited locations with pre-deployed infrastructures, e.g., intersections with cameras or RFID readers; the mobile service based systems can only capture limited vehicles due to their low penetration rates at urban scale and huge effort to participate, e.g., constantly running smartphone apps and install additional devices in your vehicles [47][48][57]. Thus, they cannot enable urban-scale vehicle sensing in real time with the minimum efforts.

Recently, a new opportunity has emerged based on urban commercial fleet upgrades, which has the potential to enable transparent urban-scale vehicle sensing in real time. We are witnessing a surge of commercial fleets instrumented with sensing and communication capabilities [1], e.g., cellular connection, dashboard-mounted cameras and dedicated short range communications (DSRC), enabling real-time sensing and data uploading [40]. Further, there is a trend for connected vehicles or autonomous vehicles where all vehicles broadcast their status to nearby vehicles for safety applications [26] [4], e.g., using DSRC to broadcast basic safety messages to nearby vehicles within 100 to 300 meters at a frequency of 2 to 10 times per second including ID, speeds, and locations [14]. Finally, even though without periodical broadcasting, an uninstrumented vehicle can still be sensed by the dashboard-mounted cameras of commercial fleets [30]. Thus the key question we try to answer is that "can we utilize small-scale yet well-instrumented commercial vehicles to accurately understand real-time mobility patterns of large-scale yet minimally-instrumented regular vehicles (i.e., only requiring a vehicle to periodically broadcast basic information) or uninstrumented regular vehicles (i.e., being captured by other vehicles' cameras) with existing urban infrastructures?"

In this paper, we answer this question by designing an urban-scale system called coSense for collaborative fleet-oriented vehicle sensing and resultant applications. The core idea of coSense is to (i) collect regular vehicles' location data by various commercial vehicles through distributed sensing (e.g., cameras) or communications (e.g., DSRC-based broadcasting), and then (ii) consolidate these collected data to the cloud by real-time centralized communications (e.g., cellular-based uploading) to infer the detailed traces of regular vehicles. Different from the existing approaches (i.e., static infrastructures [10] or mobile services [47] [48]), coSense utilizes a *mobile infrastructure* approach based on existing commercial fleets potentially without additional investments. This is because (i) local broadcasting of regular vehicles is likely to become mandatory in the near future for safety applications [15], (ii) even without broadcasting, regular vehicles can still be captured by commercial vehicles with cameras in the scenario of connected vehicles or autonomous vehicles [30], and (iii) periodical centralized status uploading for commercial vehicles has already been mandatory for accounting in many cities [2]. In addition, the existing mobile approaches [47] [48] require vehicles to deploy sensing devices, e.g cellphone or GPS device, while coSense is **transparent** to the regular vehicles since it collects their data by commercial vehicles without the active participation of regular vehicle drivers. The key technical challenge in coSense is how to fuse heterogeneous commercial fleet data to recover urban-scale regular vehicles' traces in real time based on (i) diverse mobility patterns of fleets (i.e., random taxis, semi-random trucks, and regular buses), and (ii) contextual information (e.g., road maps, traffic speeds, and historical trips). In particular, the key contributions of the paper are as follows:

- To our knowledge, we conduct the first study on urban-scale vehicle sensing based on heterogeneous fleets. Our work advances the state-of-the-art vehicular investigations in two aspects: (i) very comprehensive vehicular systems, including taxis, buses, trucks, and regular vehicles from the same city, and (ii) detailed GPS traces from 50 thousand vehicles, more than 3% of all vehicles in the studied city. Our infrastructures and data are one or two orders of magnitude larger than existing academic systems (e.g., GreenGPS [18], VTrack [48], and CTrack [47]), which have the potential to advance our understanding on urban-scale vehicle sensing.
- We design a vehicle mobility inference model based on deep learning to infer real-time traces of regular vehicles. It utilizes diverse data from commercial vehicles to collaboratively infer locations of regular vehicles at fine-grained spatiotemporal partitions under the consent of drivers. The sensing results are further improved by real-world contextual constraints including mobility patterns of commercial vehicles, detailed road networks, real-time traffic speeds, and historical trips.
- We implement and evaluate coSense in Shenzhen with a preliminary road test and a trace-driven evaluation based on real-world data from 14 thousand taxis, 13 thousand buses, 13 thousand trucks, and 10 thousand regular vehicles. We evaluate coSense by comparing it to infrastructure-based and cellphone-based approaches, and the results show that we increase the sensing accuracy by 10.1% and 16.6% on average.
- Based on the inferred regular vehicles' traces, we present an application of coSense to estimate the urban-scale travel time between 491×491 region pairs in real time. Thanks to diverse mobility patterns of regular vehicles, we obtain travel time of 15% more region pairs than using commercial vehicles alone.

We organize the paper as follows. §2 gives our motivation. §3 describes the detail mobility data of four different vehicle fleets. §4, §5 and §6 give the design, implementation, and evaluation of the vehicle trace inference model. §7 describes a real-world potential application based on our model, followed by the discussion and related work in §8 and §9. §10 concludes the paper.

2 MOTIVATIONS

We show our motivations by investigating opportunities and challenges for commercial fleet-oriented vehicle sensing based on real-world fleet data in Shenzhen.

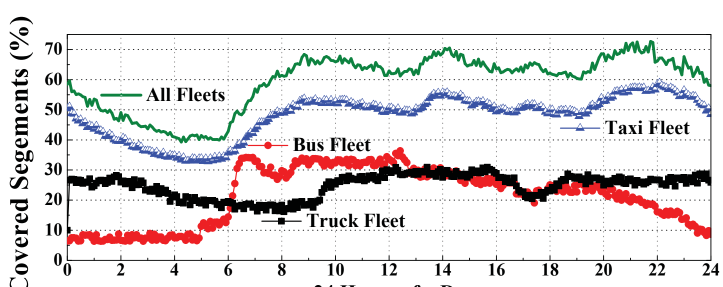

Fig. 1. Spatiotemporal Road Coverage

Opportunities for Fleet-oriented Sensing: The spatiotemporal coverage of fleets indicates their capability for sensing. Based on GPS data from three commercial fleets (i.e., taxi, bus and freight truck), we show the percentage of 160 thousand road segments with at least one vehicle from a particular fleet in 1-minute slots in Figure 1. We found that these fleets, if combined, cover a high percentage of road segments under one-minute slots, e.g., 61% of road segments on average, which indicates commercial fleets have a fine-spatiotemporal coverage. Thus they have a high potential to capture regular vehicles at urban scale.

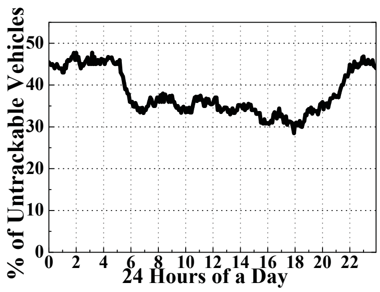

Fig. 2. Omissive Vehicles

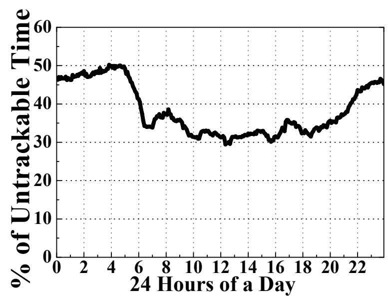

Fig. 3. Gap Duration

Challenges for Fleet-oriented Sensing: Along with GPS data from the above commercial fleets, we utilize GPS data from 10 thousand regular vehicles to show the challenges of fleet-oriented sensing based on their real-time mobility patterns. We envision that if a regular vehicle is within 100 meters of a commercial vehicle for more than 10 seconds, it can be captured by this commercial vehicle (e.g., based on local broadcasting through DSRC [40]). We will discuss these parameter settings along with an onboard-camera-based solution (i.e., without requirements for communication-based sensing) and details of regular vehicles' data in Section 3. We show the percentage of regular vehicles that do not have any commercial vehicle within 100 meters for more than 10 seconds (i.e., omissive vehicles) during one slot on 24 hours of a day in Figure 2. We found that in the daytime from 6 AM to 8 PM, more than 30% of the regular vehicles in our dataset are omissive, i.e., they do not have any commercial vehicles in its proximity of 100 meters longer than 10 seconds in one slot. During the early morning and late night, this percentage increases to 45%. This is because commercial vehicles cannot cover all spatiotemporal combination of a city due to the random mobility and limited quantity, leading to spatiotemporal sensing gaps.

It indicates utilizing commercial fleets to sense regular vehicles, though promising as shown in Figure 1, still has a key challenge to address, i.e., inferring detailed traces from all these regular vehicles with spatiotemporal sensing gaps.

To understand these spatiotemporal sensing gaps, we calculate the percentages of the time for all vehicles' trips without any commercial vehicle in their 100-meter radius longer than 10 seconds and cluster all these trips by their start time in Figure 3. We found that for vehicles' trips starting from daytime, they cannot be captured by any commercial vehicles during more than 30% of their total time. During early morning and late night, it increases to 45%, which is a large gap for sensing these vehicles in real time.

Summary: We explore the opportunities for vehicle sensing based on urban commercial fleets given their spatiotemporal road coverage. Further, we identify major spatiotemporal sensing gaps for fleet-oriented sensing in terms of omissive vehicles and omissive time. These gaps have to be addressed in order to enable urban-scale vehicle sensing by commercial fleets in real time.

3 MOBILITY DATASETS

We have been collaborating with several service providers and the Shenzhen Committee of Transportation (SCT) for real-time fleet access. As in Figure 4, we consider three instrumented commercial fleets, i.e., taxi, bus, and truck fleets, in this version of implementation, which sense regular vehicles from complementary perspectives.

Taxi Fleet		Bus Fleet		Truck Fleet		Regular Vehicles	
Beginning	1/1/2012	Beginning	1/1/2013	Beginning	12/1/2013	Beginning	2/1/2016
# of Taxis	14,453	# of Buses	13,032	# of Trucks	45,356	# of Vehicles	10,849
Size	3.5 TB	Size	1.2 TB	Size	1.1 TB	Size	20 GB
# of Records	29 billion	# of Records	11 billion	# of Records	9 billion	# of Records	27 million
Format		Format		Format		Format	
Plate ID	Date&Time	Plate ID	Date&Time	Plate ID	Date&Time	Device ID	Date&Time
Status	GPS&Speed	Stop ID	GPS&Speed	GPS	Speed	GPS	Speed

Fig. 4. Fleets and Their Data

Taxi Fleet: We access the Shenzhen taxi fleet and their data through SCT to which all taxi companies upload their taxi status in real time through a centralized connection with monthly fees. The taxi fleet in Shenzhen has 14 thousand taxis generating one status record per 30 seconds including GPS locations, time, speed, etc. The taxi fleet has a random mobility pattern to cover most of the road segments in Shenzhen as we show in Figure 1.

Bus Fleet: We access the Shenzhen bus fleet including 976 bus lines and their bus data through SCT to which all buses upload their status in real time by cellular networks. The Shenzhen bus fleet has 13 thousand buses, and their status records are generated every 30 seconds when buses are operating. Compared to the taxi fleet, the bus fleet has a regular pattern due to their operating routes. As a result, their spatial patterns are fixed, while their temporal patterns are varied because of real-time traffics even with a fixed timetable.

Truck Fleet: We access a truck fleet with 45 thousand trucks, among which 13 thousand trucks are operating in Shenzhen, by working with a large logistics company. In general, every truck uploads its status records including GPS locations and travel speeds back to a company server every 15 seconds on average for real-time monitoring, which then are routed to our server in real time. Most of these trucks are for delivery, and a truck typically has an urban area to cover, but its daily delivery schedule changes based on actual demand, leading to a semi-random mobility pattern. The above urban fleet access enables urban-scale regular vehicle sensing in real time.

Fig. 5. Taxi and Bus Onboard Devices

Fig. 6. OBD Data Collection Hardware

Distributed Regular Vehicle Detection: Currently, in our fleet platform in Shenzhen, all commercial vehicles are equipped with an onboard device with front-view cameras and centralized cellular communication devices shown in Figure 5. However, they are not equipped with peer-to-peer communication devices, e.g., DSRC. Further, the image data captured by front-view cameras are stored in vehicles and can only be obtained offline, because online raw image data uploading is not practical based on current cellular plans. As a result, we cannot have an urban-scale prototype system for a large-scale sensing. In this project, given the recent trend of connected vehicles and autonomous vehicles [26], we envision that a regular vehicle can be potentially detected by a commercial vehicle in the following two scenarios: (i) they are closer to each other than a certain threshold (i.e., simulating a peer-to-peer communication-based detection); (ii) a regular vehicle is within a certain degree (e.g., 90 degrees) front view of a commercial vehicle and closer than a certain threshold (i.e., simulating camera-based detection). The impacts of different variations of these thresholds are evaluated in the evaluation section. Based on the GPS data of all vehicles studied, we utilize all trace-driven detection for our design and evaluation.

Real-world Regular Vehicle Data Collection: For the evaluation data only, we have access to a vehicular network with more than 293 thousand regular vehicles, among which 10 thousand regular vehicles are registered in Shenzhen, as part of our evaluation platform. The data of these regular vehicles are collected by plugin OBD (On-Board Diagnostics) devices and together with a smartphone app (providing GPS locations and uploading OBD data to the insurance company) from drivers of a large insurance company in Shenzhen. Even though these data can be directly used for traffic-related applications, e.g., travel time estimation, in the real world, the auto insurance companies do not have the incentive in general to contribute data for these applications. Thus, to position our work in a more realistic setting, we use regular vehicle data for evaluation only. We will discuss the details of data collection in the Discussion Section. As shown in Figure 6, the main components of OBD are (i) a control board STM32F103CBT6, (ii) an adxl345 3-axis digital tilt sensor, (iii) a Bluetooth 3.0 BM57SPP05, (iv) a DC-DC chip MP24971, (v) a W25Q256, 8192 sectors (4kb each sector), 32MB total storage, (vi) 3.4mAh SEIKO Battery. For the performance, the control board is an M3 microcontroller with 128 K Flash, 72 MHz, and 20K RAM, which seems to be limited. For the sensing, it only has ADXL345 without a gyroscope, which makes it fine to detect acceleration and deceleration but cannot accurately detect turns. Without the cost consideration, MPU-6050 is a better choice. For the storage, if a regular driver has 4 trips per day, and each trip uses a sector for storage, then the storage can hold data for five and half years, which seems more than enough. For the data transmission, it takes around 2350 seconds (i.e., around 40 mins) to move all data from a full FLASH chip to a smartphone based on Bluetooth with 115200bps. Finally, the smartphone app will obtain GPS locations and timestamps while uploading OBD data to the insurance company. Based on our discussion with the insurance company, the main design objective of this OBD device is to enable large-scale low-cost data collection (less than 10 USD per OBD) since the insurance company covers all the cost of OBD devices and the participating drivers obtain devices for free.

Coexistence between Commercial and Regular Vehicles: We define an encounter between regular vehicles with commercial vehicles as coexistence, which is utilized for our vehicle sensing. In our setting, an encounter indicates two vehicles are within a given threshold, e.g., 100 meters. Based on their one-day data, Figure 7 gives a heatmap visualization of the coexistence between three commercial fleets with regular vehicles, i.e., a lighter region indicates higher coexistence between regular vehicles and one particular commercial fleet. In addition, we also provide a quantitative study of the coexistence, i.e., the percentage of regular vehicles encountering with commercial vehicles. We found that each commercial fleet has its own unique coexistence pattern with regular vehicles, e.g., (i) the taxi fleet has the highest coexistence in general across urban areas; (ii) the truck fleet has higher coexistence on highways and a few industrial areas and the highest coexistence during the early morning; (iii) the bus fleet has higher coexistence on residential areas and commercial areas and the highest coexistence during the daytime.

4 VEHICLE TRACE INFERENCE MODEL

We first present the key design challenge and then discuss mobility patterns to justify our design, and finally present our model design. Note that as follows, we use detection based on peer-to-peer communications as an

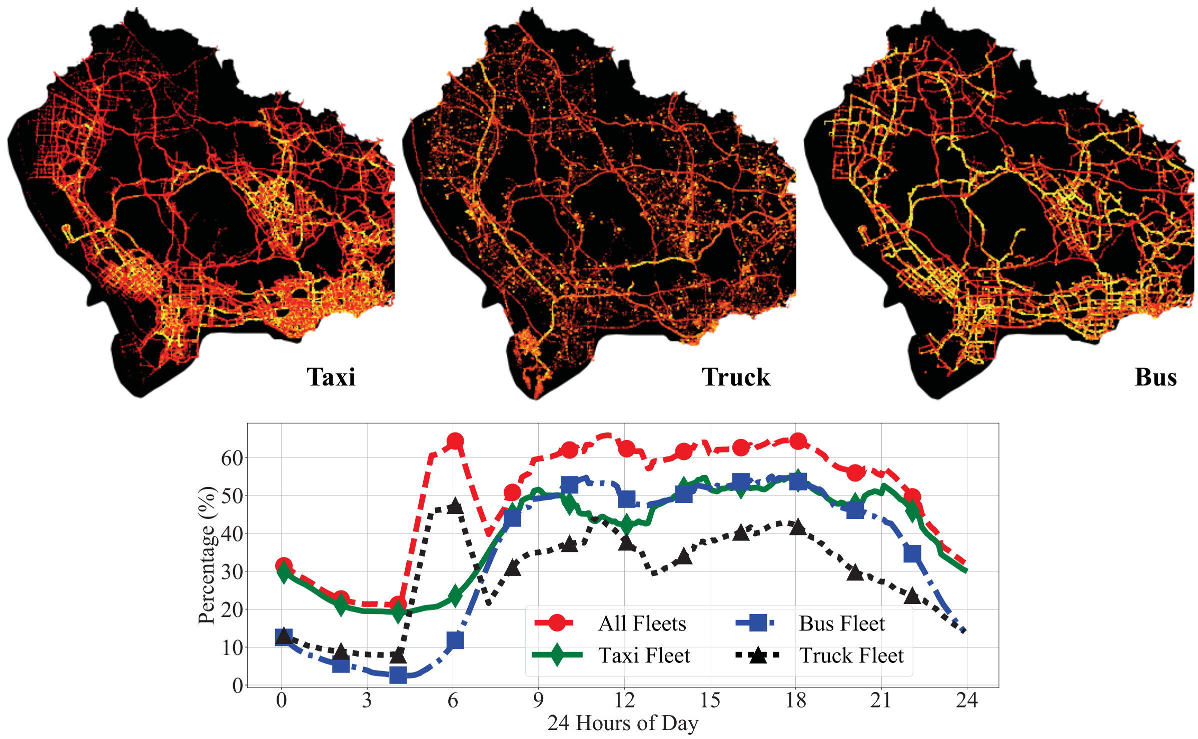

Fig. 7. Fleet Correlation Visualization

example to show our design, but in the evaluation, we also evaluate an alternative detection approach based on cameras.

4.1 Challenge: Spatiotemporal Gaps

Based on our assumption, since commercial vehicles can detect regular vehicles in terms of real-time locations, a naive solution would be to simply use these locations to sense regular vehicles. However, such a solution does not work because a regular vehicle does not always have commercial vehicles in its communication range for a period of time. To support our claim, we study a joint mobility pattern of commercial and regular vehicles based on our real-world fleet data in Shenzhen. Our detection model is under the assumption that a commercial vehicle captures a regular vehicle if they are closer than 100 meters for more than 10 seconds. Here we generate the intermediate position of vehicles between two records to obtain the position of vehicles in each second. Note that we set 100 meters as a peer-to-peer communication range and 10 seconds as a contact duration given urban environments with interferences, although in practice some peer-to-peer communication devices, e.g., DSRC, have a range of 300-500 meters in open spaces and shorter contact duration [14]. We evaluate these parameters in Section 6. Besides, we define a time period during which a regular vehicle does not have any commercial vehicle in its communication range as a spatiotemporal gap. A longer spatiotemporal gap potentially leads to a lower accuracy of our model. We perform a discretization process where we divide time into 10-second slots to study mobile interactions between regular and commercial vehicles.

As follows, we study the frequency and duration of these gaps using 10-second slots in Figures 8 and 9.

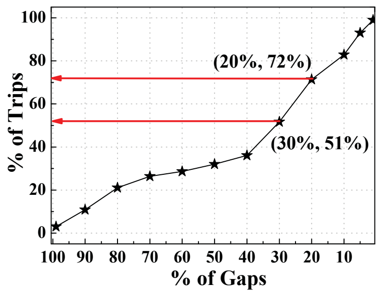

Fig. 8. Gap Frequency

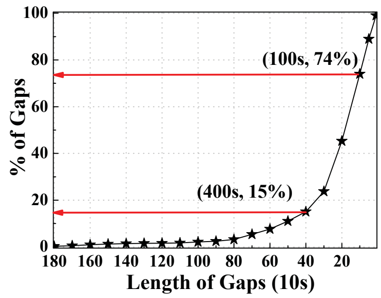

Fig. 9. Gap Duration

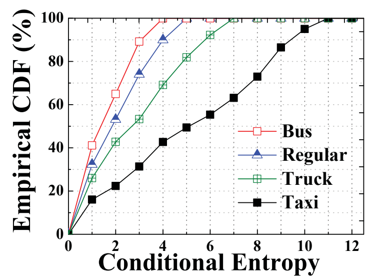

Fig. 10. Mobility Patterns

Figure 8 gives a distribution of spatiotemporal gaps based on one-week GPS data of regular and commercial vehicles including taxi, bus, and truck fleets. 51% of the regular vehicle trips have at least 30% of travel time during which there are no commercial vehicles in its communication range; 72% of the regular vehicle trips have at least 20% of travel time without any commercial vehicles nearby. Thus, a naive solution where we directly use regular vehicles' locations from commercial vehicle data cannot work given these spatiotemporal gaps. Further, Figure 9 gives the duration distribution of these spatiotemporal gaps. We found that 15% of the gaps have a duration at least 400s; 74% of the gaps have a duration at least 100s. The above results indicate there are a few gaps with high frequencies and long duration, which have to be addressed to improve the accuracy of sensing.

4.2 Opportunity: Mobility Patterns

The key reason for the above spatiotemporal gaps is the difference of mobility patterns between commercial and regular vehicles. Hence, in this paper, we aim to explore and understand the mobility patterns of these vehicle networks to address spatiotemporal gaps for better sensing.

For a vehicle k starting from a segment s_k at the time t_k, we analyze the regularity of its destination d_k based on its historical data. Our analysis is based on the previous observation that human mobility patterns (in terms of origins, destinations, and starting time) are highly regular and can be learned based on historical data [23]. We rigorously examine this observation by investigating the entropy of destination $E(d_k)$ for a vehicle k and its conditional entropy $E(d_k|s_k, t_k)$ given start location s_k and time t_k for a trip.

$$E(d_k|s_k, t_k) = \sum_{d_k, s_k \in \Psi, t_k \in \chi} p(s_k, d_k, t_k) \log \frac{p(s_k, t_k)}{p(s_k, d_k, t_k)}, \tag{1}$$

where Ψ, the set of all segments, is the support of s_k and d_k, which are random variables; χ, the set of minutes in a day, is the support of the random variable t_k. A lower entropy indicates higher predictability and regularity. We plot the CDF of Conditional Entropy in Figure 10.

We found that the conditional entropies of mobility patterns for buses and regular vehicles are lower than others. For example, given the start location and time, 53% of regular vehicles trips have fewer than $2^2 = 4$ possible segments as destinations; 51% of taxi trips have fewer than $2^5 = 32$ possible segments as destinations, among all segments in Shenzhen. Based on the above analysis, we introduce our fleet-specific mobility model given their patterns. **(i) Bus Pattern**: The bus fleet has the most regular pattern since their mobility patterns are prefixed by their routes as shown in Figure 10. However, due to traffic conditions in the urban area, buses are mostly late compared to their timetables. **(ii) Truck Pattern**: The truck fleet has a different operating feature with a semi-regular pattern as in Figure 10. Since the truck fleet we studied is from a set of logistics companies, every truck basically has a region to cover, and the detailed daily route varies based on different daily requests informed by the truck companies. **(iii) Taxi Pattern**: Compared to bus and truck fleets as shown in Figure 10, taxi fleet has the most random pattern since their routes are decided by passengers and cruising patterns of drivers. However, it has been shown by previous studies, given pickup locations and time along with the route passed, an occupied taxi's future route can be predicted with a high probability given trip regularity [55]. **(iv) Regular Vehicle Pattern**: It has been shown that human daily mobility pattern is highly regular [23]. Given the origin, start time, and the route passed for a trip, the final destination of this trip can be predicted with a high probability as in Figure 10. Based on the above analysis, we found that the spatial patterns of these vehicle fleets (especially our sensing objective, i.e., the regular vehicles) are predictable and regular on the individual level given some contexts.

4.3 Our Solution: Vehicle Sensing

The mobility patterns of the regular vehicle fleet motivate us to address spatiotemporal gaps by predicting the mobility patterns, instead of isolated locations.

4.3.1 Key Idea. Our key idea is that based on historical direct or indirect location observations of a regular vehicle, we utilize the offline training to understand its mobility pattern, and then utilize the online inference to infer its real-time destination, and thus the current locations.

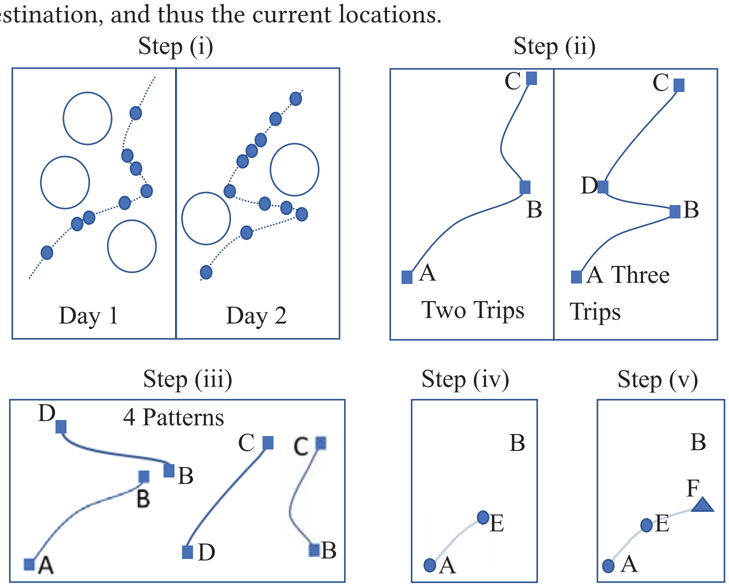

Fig. 11. Key Idea

Figure 11 gives an example of our key idea for vehicular sensing with two components. **(i) Offline Trace Inference**: Given historical partial observations of a regular vehicle (i.e., the 8 dots for Day 1 and 10 dots for Day 2), we infer the complete traces for each day (i.e., the two dash lines). In particular, in addition to the direct observation, we also utilize the indirect observation, i.e., the hollow circle indicating the area without this vehicle since a commercial vehicle in this area did not detect this regular vehicle. **(ii) Offline Trace Segmentation**: Based on the inferred traces, we divide them into different trips (i.e., trips AB and BC for Day 1, and trips AB, BD, and DC for Day 2). **(iii) Offline Trip Clustering**: We cluster these trips into different patterns (i.e., AB, BC, BD, and DC). **(iv) Online Trip Classification**: In real time setting, i.e., Day 3, when we experience a sensing gap after location E, we classify the observed partial trace from A to E into one of the patterns, i.e., AB, and infer the potential destination for this pattern, i.e., B. **(v) Online Trace Inference**: Based on inferred destination B, we infer the current location F of this vehicle.

In these 5 components, we utilize existing techniques for Offline Trace Segmentation, Offline Trip Clustering, and Online Trip Classification. (i) Trace segmentation is to divide a continuous trace into trips where each trip has an origin and a destination, and we utilize a stay point detection algorithm based on stretching factor [17] to divide a trace into several segments, and each segment is considered as a trip. (ii) Trip clustering is to combine a set of trips with similar features into a cluster to indicate their similarity. In this work, we utilize a density-based clustering [60] to obtain a set of clusters as mobility patterns. (iii) Given that the classification is a classic topic for deep learning, we perform our online trip classification based on a state-of-the-art deep learning classification technique [62].

Our key contribution is in Offline Trace Inference and Online Trace Inference. The objective of our offline trace inference is given a few direct or indirect observations of a vehicle, we want to infer its complete traces; whereas the objective of our online trace inference is given a partial trace (which can be treated as very dense historical observations) and the potential destination (which can be treated as an observation), we infer the current location (which can be obtained by a predicted complete trace). Given their similar objectives, we design a common deep learning framework for these two components together in this work.

4.3.2 Deep Learning for Trace Inference. We adopt deep learning for trace inference since deep learning utilizes long-term and large-scale historical data better than other methods, e.g., HMM and CRF. For example, [47] and [48] utilize HMM for trace inference rarely considering current spatial-temporal information of vehicles and road networks. We formulate this trace inference as a classic state prediction problem in deep learning [41] where we refer the next unknown location as the next state for prediction. We define a spatial-temporal record as one state, which is represented as a vector that contains the spatial information and the temporal information. Here we convert the map of Shenzhen into a grid with cells whose size is 100m × 100m and then convert the GPS of vehicles into this grid as the spatial information of the state. Given a sequence of state $x_{1:t}$ from time 1 to t, we utilize the previous t states as the input and the state at time $t + 1$ as the output. With this setting, we explore this sequence to train a Convolutional Neural Networks (CNN) to predict the next state at time $t + 1$. Given a partial trace of a vehicle $x_{1:t}$, coSense will use this network to infer its next location at x_{t+1}. In the offline setting, we use this method to recover the complete historical trace; whereas in the online setting, we use this method to predict the next location for sensing.

Fig. 12 shows the architecture of our deep learning model with three phases. **(i) Encoding Phase**: we have one normalization layer and four convolution layers with 64(8 × 8), 128(6 × 6), 128(6 × 6), and 128(4 × 4) filters with stride of 2. In our model, we convert the spatial-temporal features of a GPS record of one vehicle into a vector (lon, lat, time of day, day of week), representing the location of this vehicle in a time slot on a day of week. In each state, we have such a vector of one vehicle. **(ii) Transformation Phase**: we have three conditional transformation layers that transform the encoded features into a prediction of the next state in a high-level feature space by introducing road networks and indirect observation from commercial vehicles (i.e., the area without this regular vehicle) as extra inputs. In particular, we have three fully-connected layers with 128 hidden units that 1) magnifies the direction vector, 2) weights the high-level features, 3) obtains the high-level feature of next state, and an element-wise multiplication by using the weighted high-level features and the magnified direction vector as inputs. **(iii) Decoding Phase**: we have two fully-connected layers with 32 and 2 hidden units, respectively, to map predicted high-level features to a spatial-temporal vector.

4.3.3 Training Objective. In the encoding phase, our model takes a set of previous states as an input and applies 4 convolution layers to extract spatiotemporal features. In this phase, an encoded feature vector $h^{enc}_{t+1} \in \mathbb{R}^n$ at time $t + 1$ is given by:

$$h^{enc}_{t+1} = \text{Encoding}(x_{t-m:t}) \quad (2)$$

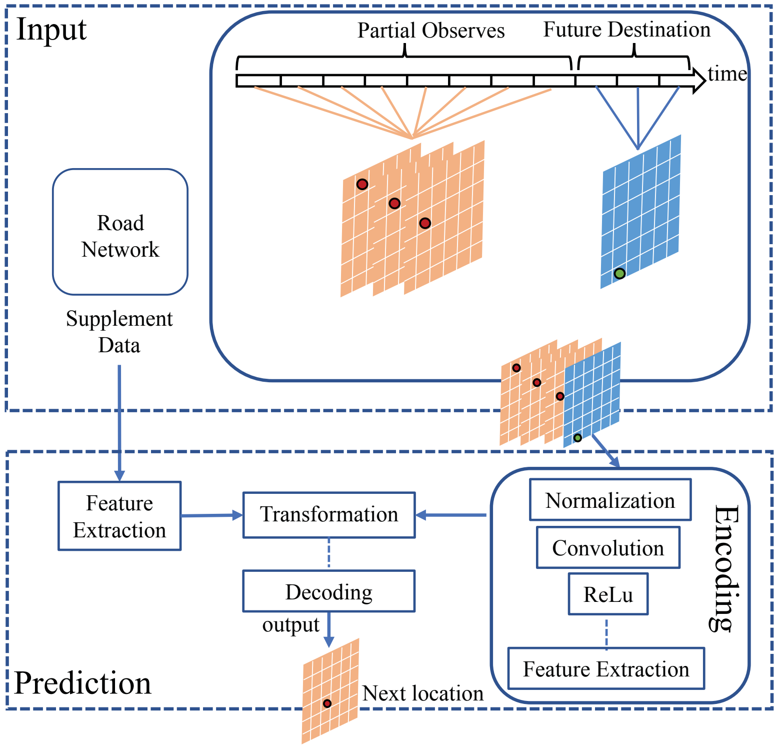

Fig. 12. Deep Learning Framework

where $x_{t-m:t}$ is m states of a vehicle from time $t-m$ to t. Further, when a vehicle is driven on a road segment, its next state should be also under the constraints of (i) the road network, (ii) indirect observations of commercial vehicles, (iii) either the future states in the offline trace inference or the predicted destination in the online trace inference, in addition to the previous states. Based on the three constraints, we create a direction vector d_{t+1} as the direction vector in the state $t+1$ to indicate the possibility of 9 heading directions of this vehicle.

In the transformation phase, we first multiple the encoded feature vector h_{t+1}^{enc} with the direction vector d_{t+1} with two layer weights given by W^{enc} and W_{t+1}^{d}, and obtain the transformed feature vector h_{t+1}^{dec} by:

$$h_{t+1}^{dec} = W^{dec}(W^{enc}h_{t+1}^{enc} \odot W_{t+1}^{d}d_{t+1}) + b \tag{3}$$

where W^{dec} is the weight of the fully-connected layer with 128 hidden units for decoding, and b is bias.

In the decoding phase, the transformed feature vector h_{t+1}^{dec} is decoded into a vector as

$$\hat{x}_{t+1} = \text{Decoding}(h_{t+1}^{dec}) \tag{4}$$

where Decoding is the two fully-connected layers with 32 and 2 hidden units.

In practice, our predictive model has some noisy predictions since it is trained on a one-step prediction objective and the small prediction errors will accumulate through time. To address this issue, we use a K-step prediction objective to enable the network to be repeatedly unrolled by K time steps via using its prediction as the input for the next time-step. Thus, our model is trained to infer the short-term future locations and fine-tuned to predict the longer-term future frames after the previous phase converges. It has been shown by the previous research that this curriculum learning approach is effective to stabilize the training [62]. Another important issue is the Cold Start problem that how to predict a regular vehicle location if we do not have any historical pattern detected by commercial vehicles. In this case, we utilize historical patterns of taxi passengers similar to the partially observed traces of regular vehicles as the initial patterns. The accuracy is low at first but increases after more partial observed traces are collected.

5 FIELD STUDY

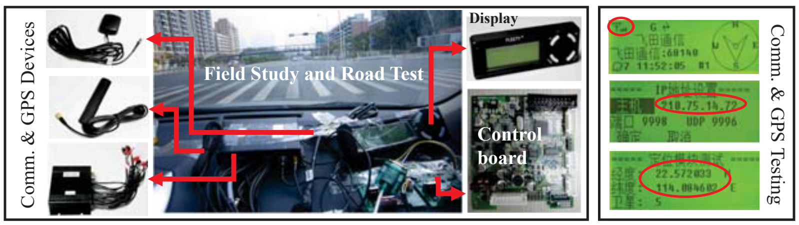

Fig. 13. Initial Deployment and Road Test for sensing Record Uploading

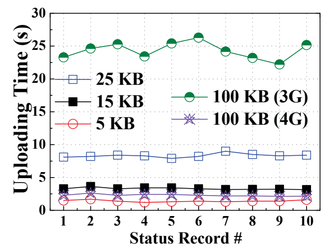

Fig. 14. Sensing Record Upload

We introduce our preliminary deployment for a road test to evaluate the performance of the current cellular and onboard infrastructures with a 3G service plan in term of data uploading for real-time sensing.

Although most of the commercial fleets in Shenzhen have communication devices already for data uploading (this is how the data are collected), they are focused on centralized communications mostly for accounting. Therefore, only a small amount of data about a vehicle itself, e.g., its GPS records, is uploaded with a 3G plan for a monthly fee. But in coSense, a vehicle has to upload location data (potentially large image data) of both itself and nearby vehicles. Thus, it is unclear if current infrastructures in urban mobile environments can support our real-time sensing, without upgrading to expensive 4G services.

Based on our collaboration with Shenzhen Transportation Committee, we instrumented 106 taxis as a small portion of 14 thousand taxis in Shenzhen for our vehicular sensing project. The instrumentation includes a GPS module, a communication module for both peer-to-peer and centralized communications, a central control system (STM32F103), and a display. A preliminarily instrumented taxi for our road test is given in Figure 13, along with GPS and communication configuration screenshots. We use this road test to evaluate real-time data uploading speeds for our centralized vehicle sensing. During the road test, different sizes of status records for a commercial vehicle were generated to simulate different densities of regular vehicles. A large size of status records of a commercial vehicle simulates a higher density of nearby regular vehicles. We repeatedly uploaded 10 sensing records with different sizes and the average uploading speeds are given in Figure 14.

The results show that for a 5KB record, it takes less than 2 seconds to upload, which can contain information for more than 100 vehicles in its range. For a denser scenario where 300 vehicles in a 100-meter radius, we can receive a 15KB record within 4 seconds. Even for an extremely dense scenario, i.e., 500 vehicles in a 100-meter radius, a record of 25KB can be uploaded within 9 seconds. If we further increase the size of status records to 100KB (potentially for image data), the delay for 3G uploading is more than 20 seconds, which seems unacceptable for vehicle sensing. However, we found that with a 4G data plan, the delay for a 100KB package is significantly reduced to 2 seconds. These results validate the commercial fleets and cellular networks in Shenzhen are ready for communication-based sensing data uploading in the current 3G data plans. But for large status record uploading (more than 100KB), an upgraded 4G data plan is a better choice without considering the increased monthly fee.

As for an urban-scale sensing with thousands of vehicles, we currently do not have a dense network of instrumented vehicles for a field test. The main reason is that we are having some difficulties to instrument regular vehicles with peer-to-peer communication (mainly due to lacking incentives) without the mandatory order from the government. Instead, in the next section, we perform a trace-driven evaluation for coSense.

6 EVALUATION

6.1 Evaluation Methodology

Evaluation Data Management: We utilize one month of data introduced in Figure 4 for evaluation. Since we concentrate on the evaluation of the coSense system, we briefly introduce issues related to fleet data management for analyses and evaluations given the space limitation. For data management and processing, we employ a high-performance cluster with two open-source data processing frameworks, i.e., Hadoop and Spark. In particular, the details of the cluster for the coSense implementation are given as follows: (i) 12 Hewlett-Packard machines with 2 Tesla K80c each; (ii) 10 Dell machines with 4 Tesla K80c each; (iii) 4 Xeon E5-2650 with a half TB memory each; (iv) A series of 800GB SSD and 15TB of spinning-disk spaces; (v) 2 PB additional disk space. Due to the extremely large size of our data and their streaming nature, we have been finding some duplicated and missing data along with logical errors. So we have been conducting a detailed cleaning process to filter out errant data on a daily basis.

Ground Truths: We obtain the ground truth of regular vehicle trajectories with their uploaded GPS data obtained by onboard devices as introduced in Section 3. This set of 10 thousand regular vehicles accounts for 0.5% of all 2 million vehicles in Shenzhen. As shown in Figure 7, this regular vehicle fleet covers major road segments in Shenzhen and can be used as a representative set of all regular vehicles in Shenzhen.

Evaluation Metrics: We utilize accuracy between inferred results and the ground truth of vehicle locations to evaluate coSense. We quantify the accuracy between two trajectories (one is the predicted result, and the other is the ground truth) by measuring the percentage of predicted locations matching the ground truth within 50 meters due to GPS errors every 10 seconds (this is the data uploading time interval of regular vehicles). The average accuracy values for all vehicles are reported. As follows, we use Accuracy to represent the average accuracy in the different spatial-temporal partition.

Fig. 15. Regular Vehicle Density on Cell Tower Levels

Baseline Approaches: (i) STrack: We compared coSense with an approach based on Stationary urban infrastructures to sense regular vehicles in major intersections. This stationary baseline system STrack represents a wide range of infrastructure-based systems, e.g., cameras or RFID, to sense urban-scale regular vehicles. In STrack, we envision fixed devices, e.g., camera, RFID, or road-side unit for DSRC devices, are deployed in major intersections of Shenzhen road networks. We implement STrack by assuming that a given percentage (40% in default) of intersections have been installed a device that can sense regular vehicles. We also evaluate the impact of percentages of intersections with infrastructures on STrack. **(ii) CTrack:** It is a state-of-the-art system [47] to sense regular vehicles based on cellular networks by periodical communications between onboard cellphones and cell towers, and then to use the locations of observed cell towers to infer locations of cellphones and thus vehicles.

We implement CTrack based on cell tower locations of a cellular network in Shenzhen by assuming every regular vehicle has a cellphone to capture a vehicle's location on cell tower levels. Figure 15 gives a visualization of regular vehicles density on cell tower levels where we assign regular vehicles to the closest cell towers based on their GPS locations. To make CTrack more competitive, we use regular vehicle GPS to obtain locations and time of regular vehicles making turns, which is to simulate an optional function of CTrack where various smartphone sensors are used to infer the turns of the vehicles to increase inferred accuracy. Note that even though detailed sensor data are collected by the insurance companies through OBD devices, we do not have access to these detailed data due to privacy issues.
Impacts of Factors: We evaluate three real-world factors and their impacts on the performance of coSense. **(i) Fleet Types:** To investigate the impact of different fleets on our systems, we feed coSense with four different commercial fleets, i.e., bus, taxi, logistic truck, and a combination of them. **(ii) Locations and Day of Week:** We evaluate the performance of coSense in different regions of the city, i.e., downtown and suburban districts, and the day of the week, i.e., weekend and weekday, where the mobility patterns of both commercial and regular vehicles are different. **(iii) Sensing Parameters:** Finally, four parameters have significant impacts on vehicle sensing based on peer-to-peer communications: the sensing device density (i.e., how many commercial vehicles are used in terms of percentage), the sensing range, the sensing duration, and the sensing probability, i.e., the probability of sensing a regular vehicle by a commercial vehicle if they are in the sensing range longer than the sensing duration. The default settings for them are 100%, 100m, 10s, and 100%. Also, to evaluate the possibility of using on-board cameras to detect nearby vehicles, we also implement a model called **coSense-Camera** where a regular vehicle can be detected if it is within a 90-degree front view of a commercial vehicle and closer than 20 meters [30]. We evaluate how severely this detection condition limits the inferred results.

6.2 Evaluation Results

Baseline Comparison: We evaluate coSense by comparing it to STrack and CTrack, and Figure 16 gives the results of these systems.

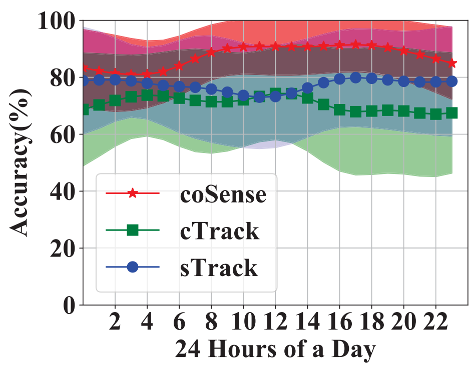

Fig. 16. Baselines

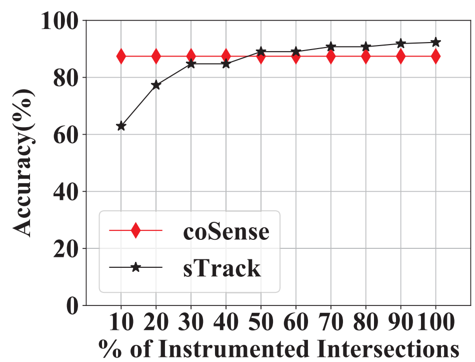

Fig. 17. % of Infra.

In Figure 16, the X-axis is the time of day, and Y-axis is Accuracy for every hour. The filled regions indicate the upper bounds and lower bounds of these three methods respectively. Based on the results, we found that coSense has better performance than STrack during 24 hours of a day with an average performance gain of 10.1%. In addition, the distance between the upper bound and lower bound of coSense is less than that of CTrack and STrack, indicating that coSense is more stable. This is due to the fact that coSense utilizes a virtual mobile infrastructure, which has a more flexible coverage. Moreover, coSense outperforms CTrack by 16.6% on average because of its mobile nature, and CTrack only utilized fixed cell towers to sense regular vehicles, which limits its sensing range. Finally, coSense utilizes the existing infrastructure in commercial vehicles, while implementing both STrack and CTrack needs significant new infrastructure investment, e.g., asking every driver to install an app to detect nearby cell towers or deploying a large number of cameras or road side units. To show the impacts of STrack infrastructure scales on the results, we vary the percentage of intersections with sensing devices and results are given in Figure 17. We found with the increase of sensing devices in intersections, the performance of STrack becomes better. But even with a 40% deployment rate, STrack still cannot outperform coSense due to its static nature. Compared to the high cost of the sensing cameras in the intersections, e.g., the automated red light camera systems of USDOT [39], our method has a major advantage in the cost of installation of infrastructure.

Impacts of Fleet Types: We apply coSense with 4 different fleet types, which are bus, taxi, truck, and a combination of them. We call them coSense-B, coSense-T, coSense-L and coSense. Figure 18 plots the Accuracy of these 4 versions of coSense. In general, the Accuracy of coSense is better than the other three models. This is because the performance of coSense is dependent on the encounter frequency between regular vehicles and commercial vehicles. In particular, comparing three individual fleet-driven versions, i.e., coSense-T, coSense-B and coSense-L, the accuracy of coSense-T is better than coSense-B and coSense-L. This is because the mobility pattern of the taxi fleet is more diverse compared to the bus fleet and logistics truck fleet, which leads to a larger coverage. It was validated by Figures 7 and 10.

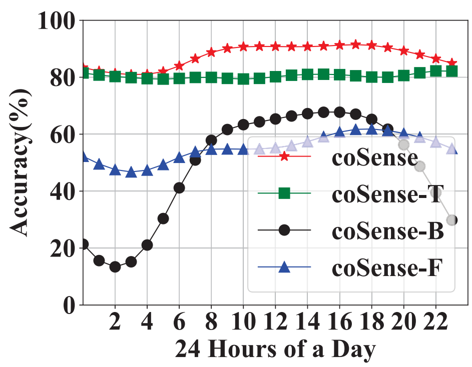

Fig. 18. Impacts of Fleets

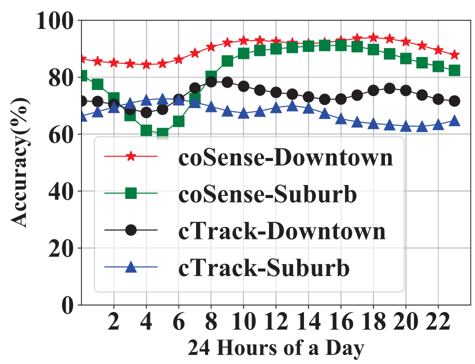

Fig. 19. Impacts of Location

Impacts of Locations and Time: We investigate the performance of coSense in a downtown and a suburb district, among 11 districts in Shenzhen. These two districts also indicate different combinations of geographic and demographic features. Figure 19 shows its performance in the downtown and suburb. The X-axis is the time of day, and the Y-axis is our metric Accuracy. We found that coSense performs betters in downtown compared to its performances in the suburb. This is because for coSense the downtown has more commercial vehicles to sense regular vehicles compared to the suburb district. We also investigate the performance of CTrack in downtown and suburb. We found coSense is still better than CTrack in suburb district since the density of cell towers in the suburb district is sparse too. As a result, more commercial vehicles lead to better spatiotemporal coverage, thus a better sensing performance. Further, to investigate the impacts of the day of the week, we plot the performances of coSense on the weekday and weekend in Figure 20.

We found that in general coSense-Weekday has a better performance than coSense-Weekend. This is because, on weekdays, we have more commercial vehicles to sense regular vehicles mainly used for daily commutes. However,

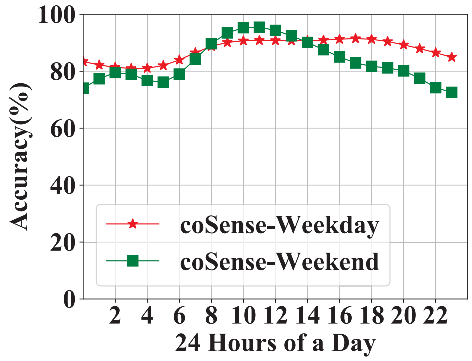

Fig. 20. Impacts of Week

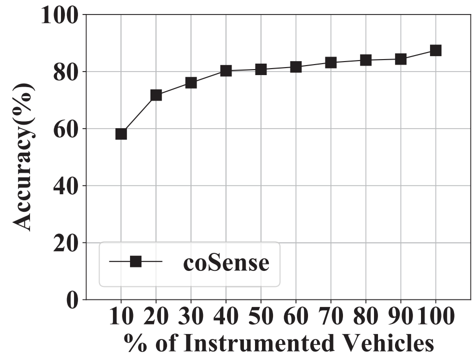

Fig. 21. Impacts of Density

on weekends, coSense has the better performance from 8 AM to 2 PM and then becomes worse, compared with that on the weekday. This is because from 8 AM to 2 PM people on weekends also have a lot of activities in the city.

Impacts of Sensing Device Density: We change the number of commercial vehicles used for sensing from 10% (e.g., 4K) to 100% (e.g., 40K) to evaluate the performance of coSense. Figure 21 plots the results where we found that even with 10% of the commercial vehicles, the performance of coSense is around 59%. The inferred Accuracy is more than 80% if only 40% of commercial vehicles are used. These results suggest that an even smaller number of commercial vehicles can be used to detect regular vehicles based on their random mobility patterns.

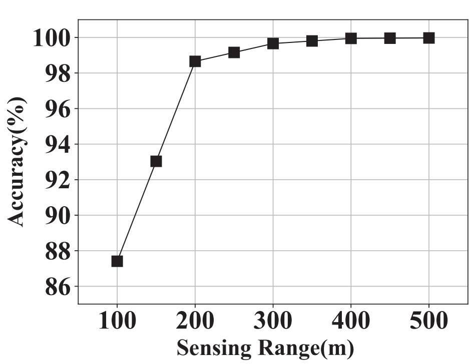

Fig. 22. Impact of Ranges

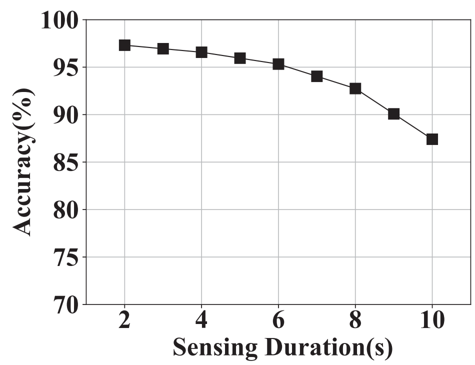

Fig. 23. Impacts of Duration

Impacts of Sensing Ranges: We change the sensing range from 100 meters to 500 meters to see the performance of coSense. Figure 22 plots the Accuracy of coSense under different sensing ranges. We found that in general, the longer the sensing ranges, the better the performance, e.g., coSense has the best performance with the sensing range of 500 meters. The reason is that when the sensing range is longer, there are more regular vehicles that can be tracked by commercial vehicles given the same patterns.

Impacts of Sensing Duration: The sensing duration indicates the period during which a commercial vehicle can sense a regular vehicle if the distance between them is shorter than the predefined sensing range. If they are in the sensing range shorter than the predefined sensing duration, the commercial vehicle cannot sense the regular vehicle in our setting. Since the time interval of data uploading is 10 seconds, we obtain the location of vehicles in every second by the assumption that the speed in this 10 seconds is fixed. We change the sensing duration from 2 seconds to 10 seconds, and then plot the results in Figure 23. We found that in general, the shorter the sensing duration, the better the performance, e.g., coSense has the best performance with the sensing duration of 2 seconds. However, in the real world, this parameter is varied based on many factors, e.g., interferences, radio types, etc.

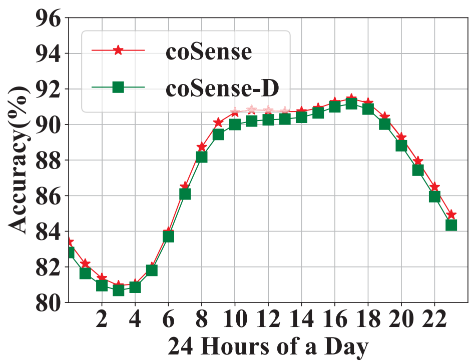

Fig. 24. Impacts of Probability

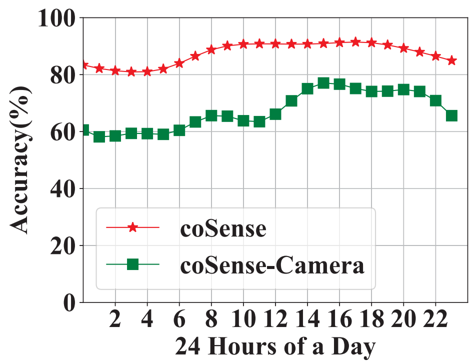

Fig. 25. Using Camera

Impacts of Sensing Probability: We change the sensing probability according to the distance of two vehicles by a Gaussian function. Figure 24 shows the comparison between coSense with fixed sensing probability value 1 and a version of coSense with a dynamic probability, called coSense-D. We found that although in coSense-D, a vehicle has a lower chance to be captured by commercial vehicles due to a dynamic probability of their distance, the performances of coSense and coSense-D are similar. A possible explanation is that for the most of the time, we have enough commercial vehicles to capture regular vehicle redundantly. So a dynamic probability does not have a major impact on the performance of our model.

Impacts of Sensing Approaches: In this setting, a regular vehicle can be detected if it is within a 90-degree front view of a commercial vehicle and closer than 20 meters [30]. We evaluate how severely this detection condition limits the sensing results in Figure 25. It shows regular coSense (i.e., detecting based on communication) is better than the coSense-camera 24 hours a day. The performance gain of regular coSense is higher during the early morning since fewer commercial vehicles are operated during this period. But the performance gain becomes smaller during the daily time, especially the evening rush hour, given the high density of both commercial and regular vehicles. It suggested that coSense-camera is more suitable for high-density scenarios given limited detection range of cameras. Please note our evaluation is based on the GPS information of vehicles. In our setting, a regular vehicle is sensed if its location is in the front of a commercial vehicle within 20 meters without any obstacles. We did not present a detailed technique for the plate recognition since it is a mature pervasive technology [50][42][24] and is used in many real-world applications[21]. In addition, we did not consider the night vision and occlusion problems in vehicle detection based on cameras due to a lack of real-world data.

7 AN APPLICATION OF COSENSE

7.1 Application Background

Inferring urban scale travel time in a real-time fashion is essential to navigation services, e.g., Google Maps [34] and Apple Maps [33]. Currently, these map services mainly use large-scale historical data and small-scale real-time data from a limited number of vehicles to infer travel time given origins and destinations. This is because they do not have enough data to infer real-time traffic speeds at urban scale [48]. In this section, we focus on using our regular vehicle sensing results for travel time between different regions for interregional mobility detection. Inferring inter-regional mobility is important for transportation since it reflects traffic congestion and human mobility on region levels. The inter-regional travel time typically varies based on real-world contexts, e.g., day of week, social events, accidents, etc., as shown by Figure 27 where we found that the travel time between the two most crowded regions in the Shenzhen varies significantly given the different time of day. To detect such travel time in real time, the existing work is based on devices and infrastructures including smartphones [47] or call detail records [27]. However, these approaches may not work at urban scale because of the limited penetration rates. Some services have also been proposed to estimate travel time based on commercial vehicles, e.g., taxis [5]. But due to the limited number of taxis, real-time travel time between large-scale regions is still inferred based on historical data. In contrast, we detect travel time using regular vehicles, which have better coverage due to their large volumes.

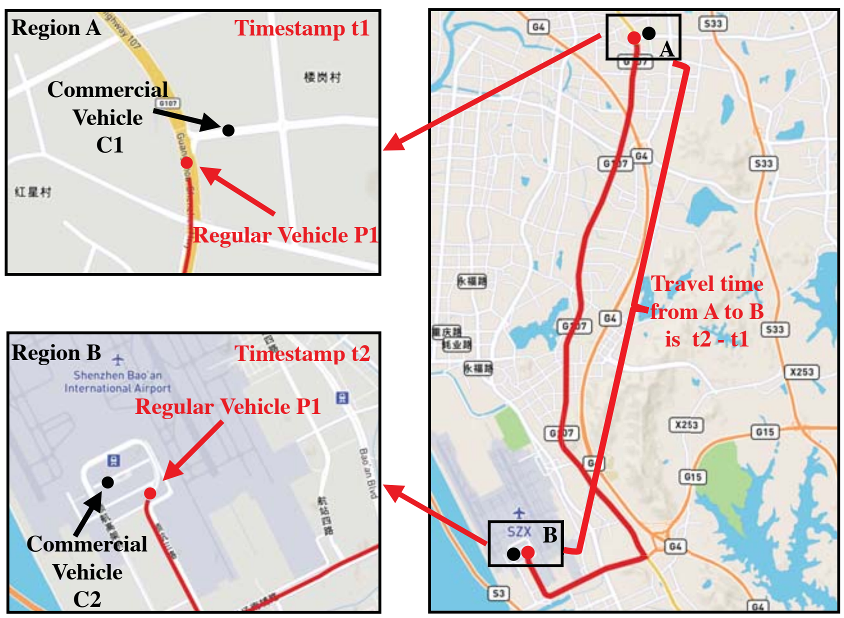

Fig. 26. Interregional Mobility

In Figure 26, if (i) a regular vehicle P_1 is detected by a commercial vehicle C_1 at region A at time t_1, and (ii) the same regular vehicle P_1 is detected by another commercial vehicle C_2 at region B at time t_2, then we use $t_2 - t_1$ to infer the real-time travel time between these region A and B even if we do not have other commercial vehicles detecting P_1 from t_1 to t_2 or a commercial vehicle actually traveling from A to B. In addition, the travel time between the regions P_1 pass through could also be able to be obtained based on the inferred trace from coSense even the commercial fleets only detect P_1 in region A and region B. This result can then be improved based on a large number of regular vehicles traveling between these two regions and captured by commercial vehicles.

7.2 Service Evaluation

We divide Shenzhen into 491 urban regions according to a given administrative partition based on geographical and demographical features.

In k time slots, e.g. $\frac{24\times 60}{5}$ five-minute slots of a day, there are $491 \times 491 \times k$ sets of travel time that needs to be detected based on regular vehicles. By using the trace inference model, we map the inferred traces of regular vehicles into 491 regions and then calculate the average travel time between different regions of each slot. At each time slot, we compute how long it takes for a vehicle to travel from one region to another region. Since different vehicles may start from different locations and select different paths, we choose the mean value to represent the expected travel time from one region to another region. Finally, we group the results based on the travel start time.

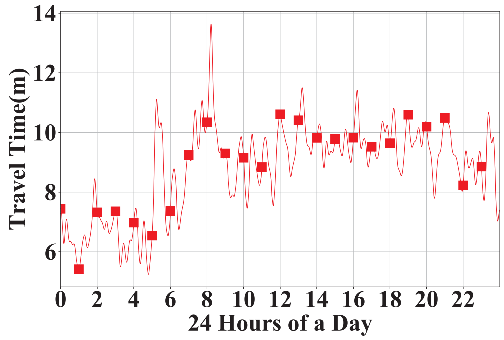

Fig. 27. Travel Time

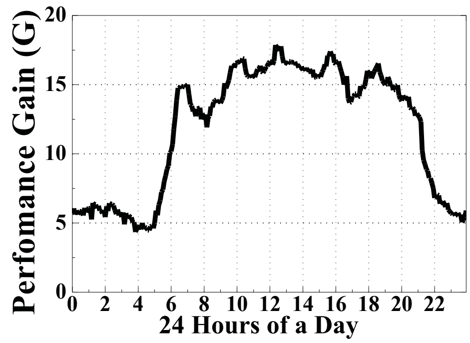

Fig. 28. Performance Gain

7.2.1 Spatial Performance Gain. One of our objective in this application is to obtain real-time travel time estimations between region pairs as many as possible. The more region pairs we have, the higher coverage that we can obtain at urban scale. We compare our service to a taxi-based approach [5] as a baseline. We utilize a spatial performance gain $G = \frac{RP_{RegularOnly}}{RP_{Total}}$ to quantify our performance where $RP_{RegularOnly}$ is the number of region pairs detected by regular vehicles only; where RP_{Total} is the number of region pairs detected by both regular vehicles and taxis. Figure 28 gives the distribution of G by 24 hours in five-minute slots.

With diversity patterns of regular vehicles, we infer more than 15% of region pairs during the regular daytime from 7AM to 8PM where more regular vehicles are traveling between regions providing diverse travel patterns. Even during the night time or early morning, e.g., from 8PM to 6AM, we still infer 5% more region pairs when using regular vehicle traces. This is because Shenzhen is a large city of 2,000 km^2, and some suburban regions are often without any taxis during certain hours, and our regular vehicle sensing help to provide travel time estimations starting or ending within these regions. Given the limited number of regular vehicles we studied, i.e., 0.5%, it suggests the performance gain would be higher if more vehicles involved. But we cannot verify it based on the current scale of our regular vehicles.

7.2.2 Travel Time Estimation. Our another objective in this application is to accurately estimate the travel time between region pairs in real-time. We quantify the accuracy between the estimated travel time and ground truth by Root Mean Square Error (RMSE) in a time slot as

$$RMSE = \sqrt{\frac{\sum_{k=1}^{n} T_{k,e} - T_{k,g}}{n}} \tag{5}$$

where $T_{k,e}$ is the estimated travel time of region pair k whereas $T_{k,g}$ is the ground truth, and n is the number of region pairs.

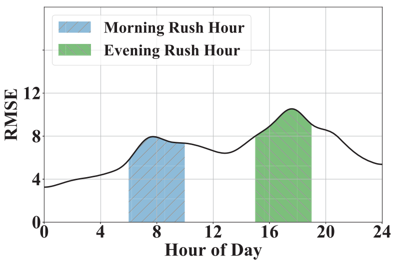

Fig. 29. Travel Time Estimation

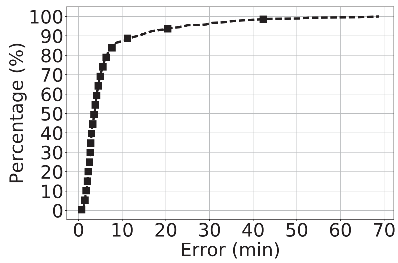

Fig. 30. Travel Time Distribution

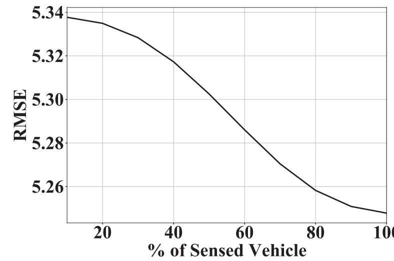

Fig. 31. Penetration Rate

Based on the inferred traces of regular vehicles, we estimate the travel time between two regions every 5 minutes and show it in Figure 29. As in Figure 29, the X-axis is the hour of the day and the Y-axis is RMSE in minute level, e.g., 1 in Y-axis means the error between the estimated travel time and the real travel time is 1 minute. We found during the rush hour in the morning, the RMSE is 8 while that of the rush hour in the evening is about 10. In addition, we also provide the CDF of the average estimated error of each region pairs in Figure 30. As in Figure 30, there are 70% of region pairs whose estimated error is less than 5 minutes. Therefore, based on the inferred traces of regular vehicles, without an elaborated designed model, it is still able to estimate the travel time between two regions with a small error.
In the evaluation of our method, we assume all the regular vehicles could be sensed if they encounter with commercial vehicles. However, in the real world scenario, the number of regular vehicles who are sensed by the commercial vehicle might be much smaller compared to our assumption, since 1) some regular vehicles might not install the device for V2V communication, 2) some drivers of the regular vehicles might be unwilling to be sensed by the commercial vehicles, 3) for the coSense-camera, some vehicles might be obscure due to the NLOS (Non-Line-of-Sight). To study the impact of this, we vary the percentage of the regular vehicles to be sensed from 10% to 100%, given in Figure 31. We found the RMSE will increase 0.08 when the percentage of the sensed regular vehicle changes from 100% to 10%. This might be because there are many regular vehicles traveling between the same region pairs.

7.3 Summary

In this section, we provide an example of the potential applications based on the inferred traces of regular vehicles, i.e., the estimation of travel time. We believe there exists more applications based on the inferred traces, e.g., the traffic volume estimation, the vehicle routes planning, and the traffic flow analyses, etc.

8 DISCUSSION AND LESSONS LEARNED

Limitations: Even with a trend of connected and autonomous vehicles [26] [4], a major limitation of our coSense is to require either (i) regular vehicles to broadcast their status or (ii) a large number of commercial vehicles to be installed with cameras. However, we believe that the design philosophy of coSense, i.e., utilizing a small number of well-equipped vehicles to sense a large number of minimally-equipped vehicles, can be generalized to other scenarios. For example, in the autonomous driving scenario, a vehicle will be equipped with at least one camera, and capture real-time images of their nearby environment. These images from a commercial vehicle can also be used to achieve a similar goal.

Based on our evaluation, it works well for Shenzhen, and we expect the similar performance for other big cities with high vehicle densities, e.g., NYC, Beijing, and Dehli. But coSense's performance is unclear for small cities/towns with low vehicle densities.

Regular Vehicle Sensing Implication: coSense has the potential to sense large-scale regular vehicles, which may lead to potential privacy issues [53]. A possible solution is to sense locations of vehicles under drivers' consent by providing sufficient incentives. Since many applications of coSense are to provide convenience for drivers, e.g., providing urban-scale real-time travel time or speeds for drivers themselves, instead of focusing on real-time locations. As a result, we envision the drivers are OK to passively participate in this sensing activity with a transparent fashion since their locations on the streets may be considered as public information. Having said that, if some drivers are more sensitive to their locations, we envision they can actively opt out from some transportation services provided by city governments based on coSense. Further, although this paper is written from a constructive standpoint, it can also be used as a challenge paper from a different perspective to validate to what degree a vehicle can be sensed by commercial vehicles in the setting of connected vehicles and autonomous vehicles. It may appeal for immediate attention from academy, industry and government agencies.

Privacy Protections: While vehicle sensing has the potential for great social benefits, we have to protect the privacy of drivers involved. We took the following active steps for privacy protections. (i) De-identification: all data analyzed are anonymized by service providers, and all identifiable IDs are replaced by a serial identifier during the analyses. (ii) Relative Locations: we utilize relative locations instead of GPS coordinates when conducting the evaluation, so the results cannot be traced back to individuals. (iii) Minimal Exposure: we only process data that are useful for our vehicle tracking project, and drop other information for the minimal exposure. We process these data in a secured facility, and the raw data never leave the promise. (iv) Aggregation: the sensing results obtained by coSense are given at aggregated results at segment levels in a time duration, instead of street addresses for a specific timestamp.

More Applications: Urban-scale vehicular sensing has many more interesting applications including public safety, e.g., tracking a vehicle with criminals onboard, or stolen vehicles recovered. Though criminals can disable or jam DSRC transmissions, the camera data from commercial vehicles can still enable sensing based on coSense. New York City has been deploying toll collecting device readers across city, not just toll stations, to sense vehicles, which is part of Midtown in Motion initiative [45], and improved travel time by 10% in a 110-block area. We expect coSense has the potential for the similar purpose.

Data Collection: All the data used in this project are legally collected by the service providers. The data from commercial vehicles are collected by the transportation committee who owns and manages all the fleets. The regular vehicle data are collected by an insurance company by an onboard device, and all drivers are informed about data collection and agreed (by assigning a contract and installing the device) to provide their real-time data for the company and its partners to analyze their travel behaviors, and in turn, they received monthly premium reductions. Further, our application of coSense is to provide an essential service, e.g., driving time estimation, for drivers themselves.

Vehicle Instrumentation: As shown by our data description, vehicle instrumentation is ubiquitous in both the commercial fleets (customized devices as in Figure 5) and regular cars (off-the-shelf OBD devices as in Figure 6). At least in Shenzhen, all 40 thousand commercial vehicles and 10 thousand cars are already instrumented although for different purposes. The low cost (less than 10 USD per OBD) and high performance of these onboard devices enable a mobile sensing infrastructure for various applications. However, how to utilize such valuable testbeds for practical applications is complicated. We believe some trace-driven study as proof-of-concept, e.g., coSense, is a good start to convince fleet owners to implement real services.

Fleet Mobility Patterns for Implicit Interaction. The commercial vehicles provide high spatiotemporal urban coverage due to their operating features, leading to high potential for urban sensing as in Figure 1. However, each of fleets has its own mobility patterns as shown by coexistence analyses in Figure 7 and by entropy analyses in Figure 10. Thus, considering these fleet-specific mobility patterns may improve potential applications, e.g., implicit travel time inference as given in Figure 26.

Human Subjects & IRB: Based on the policy of authors' institutions, the research activities in this paper are exempt from human subject protection under U.S. Department of Health and Human Services Title 45 CFR Part 46. This is because research activities include no greater than minimal risk, and de-identified relative location data used cannot be directly or through identifiers linked to the subjects.

9 RELATED WORK

The real-time vehicle sensing at urban scale is crucial for real-world applications, e.g., navigation, traffic control, and location-based services [60]. Recently, due to the ubiquity of GPS devices and the upgrades of urban infrastructures, real-time vehicle sensing has made significant improvements [48] [58] [25] [31] [36] [12]. Basically, all the existing related work can be divided into four categories.

- **Static City Infrastructures:** Public infrastructures, e.g., cameras and RFID-based toll stations, are widely used in cities for traffic monitoring, which can be used potentially for urban-scale vehicle sensing [28]. In New York City, there are 643 cameras for real-time traffic cameras [10], and 88 thruways supporting E-ZPass, which is an RFID-based sensing system [16]. Wireless access points as a part of urban infrastructures have also been used to detect positions of vehicles [32], which thus can be potentially used for location sensing. However, all these systems suffer from the static nature of infrastructures. They can only cover some specific locations, instead of the entire urban area. In contrast, coSense is based on urban fleets to sense vehicles in a mobile fashion.
- **Manufacturer Services:** Many vehicle manufacturers have services to sense their own vehicles for different applications, e.g., navigation, OnStar [11], Ford Sync [44] and BMW Assist [3]. Typically, these systems obtain locations of vehicles through built-in GPS and communicate with servers through cellular connections. However, these systems are only available in specific brands and cannot be used for urban-scale vehicle sensing.
- **Smartphone-based Systems:** They are the mainstream approach for vehicle sensing because of the ubiquity of smartphones. Several systems are proposed to sense vehicles in real time [57] [47], to monitor traffic conditions [48] [63] [64] [49], to find real-time parking spots [37], to infer transportation modality [46], to predict bus arrival times [7], to infer traffic signal [59], and to facilitate traffic safety [51]. Generally, these systems rely on smartphones and utilize one or multiple sensors to detect locations and movements of vehicles. However, these systems are limited by low penetration rates of apps and are hard to use for urban-scale vehicle tracking [22] [63] [13] [25].
- **Fleet-based Systems:** Several systems based on large-scale urban fleet data have also been proposed, e.g., dispatching for-hire vehicles (e.g., Uber and Lyft) [52]; modeling highway vehicle travel time [54]; inferring real-world road maps [6]; estimating city traffic volumes for drivers [2] [5]; predicting passenger demand for taxi drivers [20]; recommending optimal pickup locations [19] [35]; modeling the urban transit [61]; detecting the taxi anomaly [43]. However, these systems are typically used for one particular fleet, e.g., taxis or buses, and are not focused on the regular vehicle sensing.

Summary: Based on our analyses, almost all the above approaches are limited for urban-scale vehicle sensing by infrastructure coverage, penetration rates, or deployment scale. These limitations motivate us to design coSense by taking advantages of commercial fleets that are already deployed in the city with data collection capability to sense vehicles at urban scale. coSense utilizes a mobile infrastructure combining fleets with a mobility-driven approach, which makes our work significantly different from the state-of-the-art approaches.

10 CONCLUSION

In this work, we design and implement a vehicle sensing system coSense.

We evaluate it with a preliminary road test and a large-scale trace-driven evaluation based on vehicular fleets in the Chinese city Shenzhen, including 50 thousand vehicles. We compare coSense to infrastructure and cellphone-based approaches, and the results show that we increase the sensing accuracy by 10.1% and 16.6% on average. We demonstrate a practical application of coSense to infer real-time travel time of 15% more region pairs than a state-of-the-art solution. We expect that the design and evaluation of coSense will provide technical insights for various future vehicular applications where a small-scale yet well-equipped vehicles can be used to transparently understand large-scale yet minimally-equipped vehicles.

ACKNOWLEDGMENTS

This work is partially supported by Rutgers Global Center, Rutgers Research Council, China 973 Program (2015CB352400), National Natural Science Foundation of China (NSFC) 41401470, 61772026 and 61332018, and National Key R&D Program of China 2018YFB0803400.

REFERENCES

[1] Connected Ann Arbor. 2018. (2018). https://www.its.dot.gov/factsheets/dsrcfactsheet.htm
[2] Javed Aslam, Sejoon Lim, Xinghao Pan, and Daniela Rus. 2012. City-scale traffic estimation from a roving sensor network. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 141–154.
[3] BMW Assist. 2018. (2018). https://www.bmwusa.com/privacy-policy/bmw-assist.html
[4] Autonews. 2018. Cadillac to soon roll out V2V safety tech in Canada. (2018). http://canada.autonews.com/article/20170309/CANADA01/303099941/cadillac-to-soon-roll-out-v2v-safety-tech-in-cts-in-canada-u-s
[5] Rajesh Krishna Balan, Khoa Xuan Nguyen, and Lingxiao Jiang. 2011. Real-time trip information service for a large taxi fleet. In *Proceedings of the international conference on Mobile systems, applications, and services (MobiSys '11)*.
[6] James Biagioni and Jakob Eriksson. 2012. Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '12)*.
[7] James Biagioni, Tomas Gerlich, Timothy Merrifield, and Jakob Eriksson. 2011. EasyTracker: Automatic Transit Tracking, Mapping, and Arrival Time Prediction Using Smartphones *(SenSys '11)*.
[8] Two billion vehicles projected to be on roads by 2035. 2018. (2018). http://www.csmonitor.com/Business/In-Gear/2014/0729/Two-billion-vehicles-projected-to-be-on-roads-by-2035
[9] 9 Delightfully Geeky Stats About NYC Bridges and Tunnels. 2018. (2018). https://www.buzzfeed.com/johntemplon/9-delightfully-geeky-stats-nyc-bridges-tunnels
[10] New York City Camera. 2018. (2018). http://dotsignals.org
[11] Nicholas Capurso, Eric Elsken, Donnell Payne, and Liran Ma. 2014. Poster: A robust vehicular accident detection system using inexpensive portable devices. In *MobiSys*.
[12] Lien-Wu Chen, Kun-Ze Syue, and Yu-Chee Tseng. 2010. A Vehicular Surveillance and Sensing System for Car Security and Tracking Applications. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*. ACM, New York, NY, USA, 426–427. https://doi.org/10.1145/1791212.1791288
[13] Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Hähnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian E. Smith, and Alex Varshavsky. 2006. Practical Metropolitan-Scale Positioning for GSM Phones. In *UbiComp*.
[14] Bin Cheng, Ali Rostami, and Marco Gruteser. 2016. Experience: Accurate Simulation of Dense Scenarios with Hundreds of Vehicular Transmitters. In *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York, NY, USA, 271–279. https://doi.org/10.1145/2973750.2973779

[15] Cnsnews. 2018. DOT Proposes Mandating Cars Broadcast Location, Direction and Speed. (2018). http://www.cnsnews.com/commentary/terence-p-jeffrey/dot-proposes-mandating-cars-broadcast-location-direction-and-speed

[16] EZPass. 2018. (2018). https://www.e-zpassny.com/en/about/plazas.shtml

[17] Raghu Ganti, Iqbal Mohomed, Ramya Raghavendra, and Anand Ranganathan. 2012. Analysis of Data from a Taxi Cab Participatory Sensor Network *(Mobile and Ubiquitous Systems: Computing, Networking, and Services)*.

[18] Raghu K. Ganti, Nam Pham, Hossein Ahmadi, Saurabh Nangia, and Tarek F. Abdelzaher. 2010. GreenGPS: A Participatory Sensing Fuel-efficient Maps Application. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. ACM, New York, NY, USA, 151–164. https://doi.org/10.1145/1814433.1814450

[19] Yong Ge, Chuanren Liu, Hui Xiong, and Jian Chen. 2011. A taxi business intelligence system. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '11)*.

[20] Yong Ge, Hui Xiong, Alexander Tuzhilin, Keli Xiao, Marco Gruteser, and Michael Pazzani. 2010. An energy-efficient mobile recommender system *(KDD '10)*.

[21] Genetec. 2018. (2018). https://www.genetec.com/solutions/industries/parking-enforcement

[22] Hector Gonzalez, Jiawei Han, Xiaolei Li, Margaret Myslinska, and John Paul Sondag. 2007. Adaptive fastest path computation on a road network: a traffic mining approach. In *Proceedings of the 33rd international conference on Very large data bases (VLDB '07)*.

[23] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196 (2008), 779–782.

[24] Chao Gou, Kunfeng Wang, Yanjie Yao, and Zhengxi Li. 2016. Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines. *IEEE Transactions on Intelligent Transportation Systems* 17, 4 (2016), 1096–1107.

[25] Bo-Jhang Ho, Paul D. Martin, Prashanth Swaminathan, and Mani B. Srivastava. 2015. From Pressure to Path: Barometer-based Vehicle Tracking. In *BuildSys@SenSys*.

[26] V2V Communication in New Cars Starting in 2021. 2018. (2018). http://blog.caranddriver.com/feds-want-v2v-communication-in-new-cars-starting-in-2021

[27] Sibren Isaacman, Richard A. Becker, Ramón Cáceres, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. 2012. Human mobility modeling at metropolitan scales. In *MobiSys*.

[28] Vipin Jain, Ashlesh Sharma, and Lakshminarayanan Subramanian. 2012. Road traffic congestion in the developing world. In *Proceedings of the 2nd ACM Symposium on Computing for Development*. ACM, 11.

[29] Lei Kang, Bozhao Qi, Dan Janecek, and Suman Banerjee. 2015. EcoDrive: A mobile sensing and control system for fuel efficient driving. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 358–371.

[30] Gorkem Kar, Shubham Jain, Marco Gruteser, Fan Bai, and Ramesh Govindan. 2017. Real-time Traffic Estimation at Vehicular Edge Nodes. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing (SEC '17)*.

[31] Gorkem Kar, Hossen Mustafa, Yan Wang, Yingying Chen, Wenyuan Xu, Marco Gruteser, and Tam Vu. 2014. Detection of On-Road Vehicles Emanating GPS Interference. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. ACM, New York, NY, USA, 621–632. https://doi.org/10.1145/2660267.2660336

[32] Wenping Liu, Jiangchuan Liu, Hongbo Jiang, Bicheng Xu, Hongzhi Lin, Guoyin Jiang, and Jing Xing. 2016. WiLocator: WiFi-Sensing Based Real-Time Bus Tracking and Arrival Time Prediction in Urban Environments. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*.

[33] Apple Map. 2018. (2018). http://www.apple.com/ios/maps

[34] Google Map. 2018. (2018). https://www.google.com/maps

[35] F. Miao, S. Lin, S. Munir, J. Stankovic, H. Huang, Desheng Zhang, T. He, and G. J. Pappas. 2015. Taxi Dispatch with Real-Time Data in Metropolitan Areas *(ACM ICCPS 2015)*.

[36] Shashika Ranga Muramudalige and H.M.N. Dilum Bandara. 2016. Demo: Cloud-Based Vehicular Data Analytics Platform. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion (MobiSys '16 Companion)*. ACM, New York, NY, USA, 1–1. https://doi.org/10.1145/2938559.2948849

[37] Anandatirtha Nandugudi, Taeyeon Ki, Carl Nuessle, and Geoffrey Challen. 2014. PocketParker: Pocketsourcing Parking Lot Availability *(UBICOMP '14)*.

[38] Garmin Navigator. 2018. (2018). https://explore.garmin.com/en-US/nuvi

[39] Assessment of Red Light Running Camera Enforcement Technologies. 2018. (2018). https://www.itscosts.its.dot.gov/ITS/benecost.nsf/ID/2B209AD2C5AD2AB985256DB10045892B?OpenDocument&Query=CApp

[40] DSRC: The Future of Safer Driving. 2018. (2018). https://www.its.dot.gov/factsheets/dsrcfactsheet.htm

[41] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. 2015. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems*. 2863–2871.

[42] Rahim Panahi and Iman Gholampour. 2017. Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications. *IEEE Transactions on intelligent transportation systems* 18, 4 (2017), 767–779.

[43] Rijurekha Sen and Rajesh Krishna Balan. 2013. Challenges and Opportunities in Taxi Fleet Anomaly Detection *(SENSEMINE'13)*.

[44] Alexander V. Smirnov, Alexey Kashevnik, Andrew Ponomarev, and Nikolay Shilov. 2014. Location-based on-board system for e-tourism: Implementation for Ford SYNC. In *ICCVE*.

[45] NYC Midtown Congestion Management System. 2018. (2018). http://www.nyc.gov/html/dot/html/pr2012/pr12_25.shtml

[46] Arvind Thiagarajan, James Biagioni, Tomas Gerlich, and Jakob Eriksson. 2010. Cooperative transit tracking using smart-phones. In *SenSys*.

[47] Arvind Thiagarajan, Lenin Ravindranath, Hari Balakrishnan, Samuel Madden, and Lewis Girod. 2011. Accurate, Low-energy Trajectory Mapping for Mobile Devices *(NSDI'11)*.

[48] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. 2009. VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys*.

[49] Xiong Wang, Jinbei Zhang, Xiaohua Tian, Xiaoying Gan, Yunfeng Guan, and Xinbing Wang. 2017. Crowdsensing-Based Consensus Incident Report for Road Traffic Acquisition. *IEEE Transactions on Intelligent Transportation Systems* (2017).

[50] Yang Wang, Wuji Chen, Wei Zheng, He Huang, Wen Zhang, and Hengchang Liu. 2017. Tracking Hit-and-Run Vehicle with Sparse Video Surveillance Cameras and Mobile Taxicabs. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 495–504.

[51] Yan Wang, Jie Yang, Hongbo Liu, Yingying Chen, Marco Gruteser, and Richard P Martin. 2013. Sensing vehicle dynamics for determining driver phone use. In *MobiSys*. ACM, 41–54.

[52] Xiaoyang Xie, Fan Zhang, and Desheng Zhang. 2018. PrivateHunt: Multi-Source Data-Driven Dispatching in For-Hire Vehicle Systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 45.

[53] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. 2017. Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*.

[54] Yu Yang, Fan Zhang, and Desheng Zhang. 2018. SharedEdge: GPS-Free Fine-Grained Travel Time Estimation in State-Level Highway Systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 48.

[55] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories. In *GIS*.

[56] Y. Liu C. Tan W. Chang D. Zhang Z. Qin, Z. Fang. 2018. EXIMIUS: A Measurement Framework for Explicit and Implicit Urban Traffic Sensing *(Sensys'18)*.

[57] Mingmin Zhao, Tao Ye, Ruipeng Gao, Fan Ye, Yizhou Wang, and Guojie Luo. 2015. VeTrack: Real Time Vehicle Tracking in Uninstrumented Indoor Environments. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15)*.

[58] Mingmin Zhao, Tao Ye, Ruipeng Gao, Fan Ye, Yizhou Wang, and Guojie Luo. 2015. VeTrack: Real Time Vehicle Tracking in Uninstrumented Indoor Environments. In *SenSys*.

[59] Yiran Zhao, Shen Li, Shaohan Hu, Lu Su, Shuochao Yao, Huajie Shao, Hongwei Wang, and Tarek Abdelzaher. 2017. Greendrive: A smartphone-based intelligent speed adaptation system with real-time traffic signal prediction. In *(ICCPS*.

[60] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 38.

[61] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei-Ying Ma. 2010. Understanding transportation modes based on GPS data for web applications. *ACM Trans. Web* 4, 1 (Jan. 2010).

[62] Sheng-hua Zhong, Yan Liu, and Yang Liu. 2011. Bilinear Deep Learning for Image Classification. In *Proceedings of the 19th ACM International Conference on Multimedia (MM '11)*. ACM, New York, NY, USA, 343–352. https://doi.org/10.1145/2072298.2072344

[63] Pengfei Zhou, Zhiyuan Chen, and Mo Li. 2013. Smart traffic monitoring with participatory sensing. In *SenSys*.

[64] Pengfei Zhou, Shiqi Jiang, and Mo Li. 2015. Urban traffic monitoring with the help of bus riders. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*. IEEE, 21–30.

Received August 2018; revised October 2018; accepted October 2018