Driver Behavior-aware Parking Availability Crowdsensing System Using Truth Discovery

YI ZHU and ABHISHEK GUPTA, SUNY Buffalo
SHAOHAN HU, IBM T. J. Watson Research Center
WEIDA ZHONG, SUNY Buffalo
LU SU, Purdue University
CHUNMING QIAO, SUNY Buffalo

Spot-level parking availability information (the availability of each spot in a parking lot) is in great demand, as it can help reduce time and energy waste while searching for a parking spot. In this article, we propose a crowdsensing system called SpotE that can provide spot-level availability in a parking lot using drivers' smartphone sensors. SpotE only requires the sensor data from drivers' smartphones, which avoids the high cost of installing additional sensors and enables large-scale outdoor deployment. We propose a new model that can use the parking search trajectory and final destination (e.g., an exit of the parking lot) of a single driver in a parking lot to generate the probability profile that contains the probability of each spot being occupied in a parking lot. To deal with conflicting estimation results generated from different drivers, due to the variance in different drivers' parking behaviors, a novel aggregation approach SpotE-TD is proposed. The proposed aggregation method is based on truth discovery techniques and can handle the variety in Quality of Information of different vehicles. We evaluate our proposed method through a real-life deployment study. Results show that SpotE-TD can efficiently provide spot-level parking availability information with a 20% higher accuracy than thestate-of-the-art.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools;**

Additional Key Words and Phrases: Mobile sensing, parking availability, crowdsourcing, truth discovery

ACM Reference format:
Yi Zhu, Abhishek Gupta, Shaohan Hu, Weida Zhong, Lu Su, and Chunming Qiao. 2021. Driver Behavior-aware Parking Availability Crowdsensing System Using Truth Discovery. *ACM Trans. Sen. Netw.* 17, 4, Article 41 (June 2021), 26 pages.
https://doi.org/10.1145/3460200

This work is supported in part by the US National Science Foundation under Grants CNS-1652503 and CNS-1737590. The views and conclusions contained in this documents are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF.
Authors' addresses: Y. Zhu, A. Gupta, W. Zhong, and C. Qiao, Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260; emails: {yzhu39, agupta33, weidazho, qiao}@buffalo.edu; S. Hu, JPMorgan Chase - FLARE, New York, NY, US; email: hushaohan@gmail.com; L. Su, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907; email: lusu@purdue.edu.

1550-4859/2021/06-ART41 $15.00
https://doi.org/10.1145/3460200

1 INTRODUCTION

Nowadays, as the traffic volume in urban area becomes increasingly large, the demand of finding parking spaces grows rapidly. Searching for available spots oftentimes leads to time and energy wastage and adds to additional harmful emissions. A recent survey [Shoup 2006] shows that finding an available parking space takes drivers 3.5 to 14 minutes on average in downtown areas, and 30% of street traffic is caused as a result of such searches. Obviously all these problems could be greatly mitigated if real-time parking availability information becomes available. Additionally, such information can also be utilized by transportation authorities to adjust the parking price policies. Especially, spot-level availability information (i.e., the availability of each spot in a parking lot) can greatly reduce drivers' searching time in the parking lot.

Many systems [Gandhi and Rao 2016; LADoT 2017; Lu et al. 2009; Mathur et al. 2010; Ng and Chua 2012; SFMTA 2017; Suhr and Jung 2014] that provide spot-level parking availability information have been developed and used in recent years. Most of such systems rely on specialized devices, such as sensors embedded in each parking spot to detect whether the spot is occupied [SFMTA 2017]. Although those conventional systems can be deployed in some small parking lots or street-side parking places with few parking spots, the high cost of installation and maintenance makes these systems unsuitable for large-scale deployment. For example, SFPark costs around $2,500 for a single parking spot. In large-scale campus parking lots with a thousand parking spots, deploying the SFPark system to cover all the spots could take around $2,500,000.

To alleviate the high-cost problem, a significant body of works use smartphone as a sensing platform to provide information on parking availability [Cherian et al. 2016; Ma et al. 2014; Nandugudi et al. 2014; Nawaz et al. 2013; Soubam et al. 2016; Xu et al. 2013]. These methods estimate the number of available spots in a parking lot by detecting vehicles' parking and departure events using drivers' smartphones. However, these methods are unable to provide spot-level parking availability information. To infer spot-level parking availability, ParkScan [Liu et al. 2017] utilizes drivers' search trajectories to estimate the availability of the parking spots that are passed by the vehicles. However, ParkScan has two main drawbacks: First, a large number of spots may not be visited by any vehicle especially when the parking lot is large, which causes the low coverage rate of the parking spots. Second, ParkScan does not account for the nature of varying **Quality of Information (QoI)** from different vehicles in their aggregation framework. Considering each vehicle in ParkScan as an information source that can provide parking availability information, the varying QoI arises due to different searching and parking behaviors from different drivers. For example, some drivers try to park as close to their destination as possible, while some prefer to park at the first spot they find. This varying QoI may result in conflict estimation results from different vehicles and affect the estimation performances.

In this article, we propose **SpotEstimation (SpotE)** system that can provide spot-level parking availability information using data from drivers' smartphone sensors including GPS and IMU sensors. Compared with conventional parking sensing systems that are based on additional dedicated sensors, our smartphone-based system can improve the parking sensing coverage with very little cost, which is especially suitable for large-scale outdoor parking areas. We first propose three key observations, which demonstrate the relationship between the spot-level availability of the parking lot and drivers' parking search process. Based on these observations, we propose a probability profile generation model, which provides the probability of each parking spot being occupied in a parking lot. The generated probability profile can cover all the spots in the parking lot, which solves the low coverage rate problem.

To aggregate the probability profiles from different vehicles and capture the variety in the QoI, we utilize truth discovery methods [Galland et al. 2010; Li et al. 2014a, b; Wang et al. 2012; Yin et al.

2008; Yin and Tan 2011; Zhao and Han 2012; Zhao et al. 2012; Zhong et al. 2019] that have been used in crowd-sensing applications to aggregate information from different sources. Truth discovery captures the variety in the QoI from various sources by estimating the reliability of the sources and weighing correspondingly. In this article, we formulate the parking availability problem into a Truth Discovery problem, where each vehicle is a **source** and provides information about the parking availability. Each parking spot in a parking lot is an **object** to be sensed and the **truth** is the parking availability information.

However, the traditional truth discovery methods only consider static sources. In real-world parking scenarios, vehicles enter and leave the parking lot constantly so the sources are constantly evolving. The parking availability (or "truth") is also changing with time because of the constant parking and departure events. The evolving sources together with the evolving truth are not considered by existing truth discovery methods. Besides, the sources might not observe all the objects equally, so the QoI from the same vehicle regarding different objects may be different. For example, when an estimation is generated from a vehicle, the estimated availability of some spots might not be reliable, because these spots are wrongly observed or not observed by the driver. To deal with the evolving sources problem and the observation difference problem, we propose a novel truth discovery algorithm, **SpotE-TruthDiscovery (SpotE-TD)**, that can accurately capture the dynamic reliability information. The applicability and robustness of our proposed system is demonstrated via evaluations on multiple real-life scenarios. To recap, our work makes the following contributions:

- We design SpotE system that can estimate the availability of each parking spot using drivers' search trajectories and destinations. The driver's search trajectory and destination are extracted from his smartphone sensor data.
- We present three key observations on the relationship between the parking availability and drivers' parking behaviors. Based on these observations, we propose an availability probability profile generation model to estimate the spot-level availability.
- To deal with the evolving sources problem, we propose a new truth discovery algorithm, SpotE-TD, that can utilize information from the previous sources and generate current truth estimation. In addition, we take observation difference into consideration and introduce fine-grained source reliability estimation.
- We collect real-world data and conduct experiments under various scenarios. Results show that our proposed method can achieve better performances than the state-of-the-art methods under various scenarios. Our new truth discovery method can capture source reliability better than existing truth discovery approaches as well.

The article is organized as follows: Section 2 provides the key idea and the architecture overview of the system. Section 3 discusses our observations about parking lot availability and the probability profile generation model. Section 4 describes the challenges of aggregation and the truth discovery algorithm we proposed. Section 5 shows the evaluation experiments and results under different scenarios. In Section 6, we discuss some practical concerns and privacy issues. Section 7 discusses related works and Section 8 concludes the article.

2 OVERVIEW

Given a destination of a driver, the parking availability of each parking spot in a parking lot affects the driver's parking search trajectory. Thus, the search trajectory of the driver reflects the parking availability of each parking spot. The key idea of our proposed system is to utilize drivers' search trajectories and destinations to estimate the availability of each parking spot. To extract trajectories and destinations of drivers, the sensors in commodity smartphones are used.

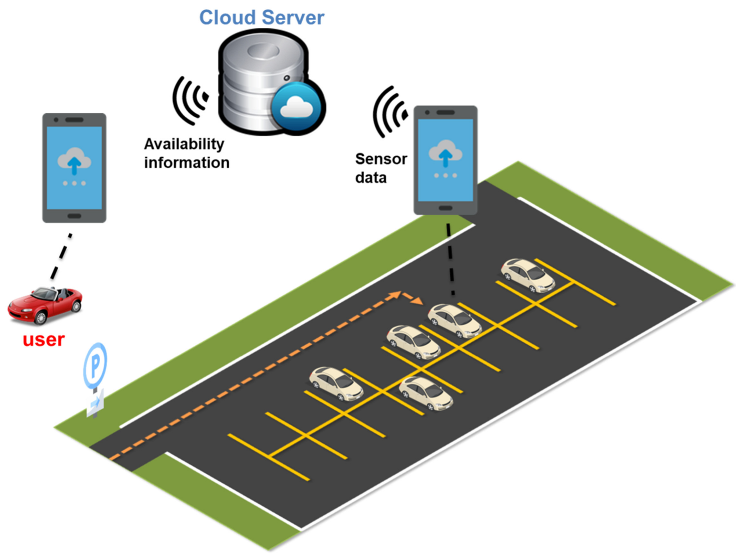


Fig. 1. System overview.

Figure 1 shows an example of our system design. Every time when a monitored vehicle (its driver installs our App) enters the parking lot, the sensors present in his smartphone collect the data and upload the data to a cloud server. The server receives the sensor data and extracts the search trajectory and final destination of the driver. The server aggregates the trajectory and destination information from all the monitored vehicles that have parked in this parking lot before and generates real time spot-level parking availability of the parking lot. When other drivers try to park at the same parking lot, they receive the spot-level availability information from the server, which can be used to select optimal parking spots.

2.1 System Architecture

Our system architecture, illustrated in Figure 2, comprises three main steps: data collection and pre-processing, probability profile estimation, and information aggregation.

In the data collection step, the driver's smartphone uploads sensor data to the server. In addition to the sensor data, information such as the layout and historical availability of the parking lot is also needed. The collected sensor data is used to extract the driver's search trajectory and destination. The sensor data is segregated into driving and walking phases leveraging acceleration data from the smartphone. The driver's search trajectory in the parking lot and the final destination are extracted using GPS data from segregated driving and walking phases. The server then estimates the spot-level probability profile of the entire parking lot from each vehicle's searching trajectory and final destination information. The probability profile records the probability of each spot being occupied in the parking lot. In the aggregation step, the server aggregates all the individual probability profiles from different vehicles and generates the final spot-level availability of a parking lot. Also, the final availability information is stored as Historical profile of the parking lot in the cloud.

2.2 Data Collection

We leverage smartphone as a sensing platform for gathering data to estimate drivers' search trajectory and final destination information. Due to their low costs and the presence of a wide

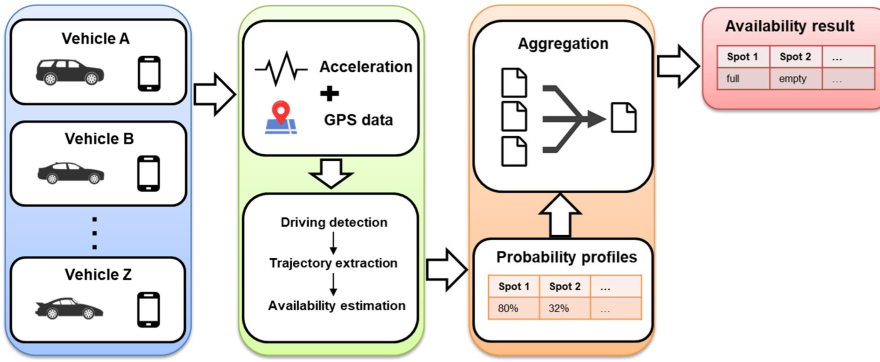


Fig. 2. System architecture.

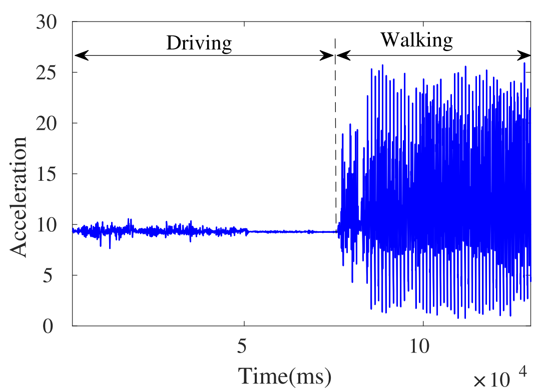


Fig. 3. Acceleration.

array of sensors, smartphones have been widely used in crowd-sensing systems. We develop an Android app that collects integrated sensor data from accelerometer, gyroscope, and GPS of a smartphone.

Sampled at 100 Hz, accelerometer data is used to detect the driver's driving and walking activities. The trajectories are extracted from GPS traces sampled at 1 Hz, which is sufficient, because vehicles normally move slowly when searching for spots in parking lots. Then the smartphone data is uploaded to the cloud server. The data upload process is triggered once the driver is in the parking lot, which can be easily achieved by GPS.

2.3 Trajectory and Destination Extraction

To extract a driver's search trajectory in the parking lot and determine the location of his destination, we look at both the accelerometer and GPS traces. In a nutshell, we first use an activity classifier [Nandugudi et al. 2014] on the accelerometer data to distinguish the driver's driving and walking activities, as they display much different acceleration characteristics. To illustrate this, an example trace is shown in Figure 3, from which we can clearly see that the data trace during driving displays much less variance compared to that of walking. With the activity information handy, we then extract the driver's parking search trajectory as well as the walking trajectory towards their final destination from the accompanying GPS trace.

Fig. 4. An example of trajectory and destination extraction.

2.3.1 Trajectory Extraction. As previously discussed, the parking search trajectories are extracted from GPS data with the help of driving/walking activity information. However, due to the low localization accuracy of smartphone GPS (around 4 m, which is larger than the size of a parking spot), the extracted trajectory might be inaccurate. To improve accuracy, the extracted trajectory is rectified against the static map of the parking lot using map-matching [Lou et al. 2009], as illustrated in Figure 4. The static map is built using information from Google Maps and contains the GPS locations of parking spots, driving paths, entrances, and exits.

2.3.2 Destination Estimation. A driver's final destination is estimated from the trajectory of walking towards his destination after parking the car. The final location where the driver leaves the parking lot after parking, called the "leaving location," is extracted from the walking trajectory. Due to error-prone data from GPS, the exit of the parking lot that is closest to the "leaving location" is treated as the final destination of the driver. As shown in Figure 4, the leaving location is close to Exit 2, so Exit 2 is chosen as the final destination.

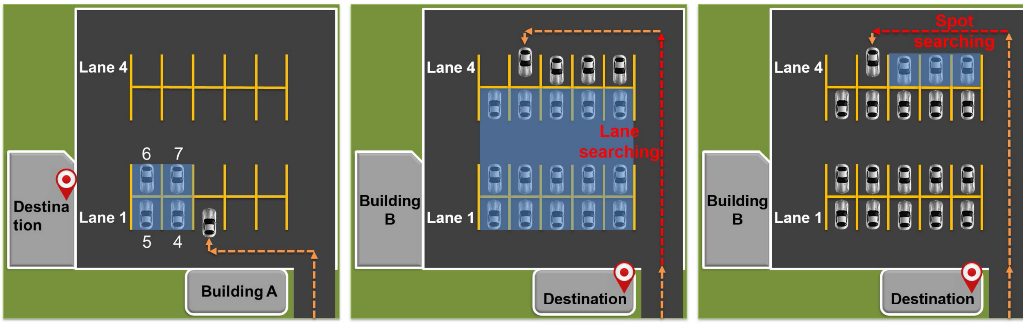
3 PROBABILITY PROFILE GENERATION

Although a driver's parking search trajectory and destination reflect the availability of each parking spot in a parking lot, accurately estimating each parking spot's availability from that information is still a challenging task. Since the vehicle may only pass by a few parking spots, the coverage rate of a vehicle can be very small. Estimating the availability of all the parking spots with low coverage rate brings extra challenges, because some spots may not be visited by any vehicle in a given time period. To deal with these challenges, we first analyze the parking search process and propose three key observations on the relationship between the parking availability and the driver's parking search process. Then, we propose a **probability profile** generation model, which generates probability profile from each driver's search trajectory and destination. The probability profile contains the probability of each spot in the parking lot being occupied (occupancy probability). Note that each parking event in the parking lot from the monitored vehicle generates one probability profile.

3.1 Search Process in Parking Lot

In most outdoor parking lots, spots are arranged as parallel parking lanes. Normally, a driver's search is carried out in two phases:

- **Lane Searching:** The driver first searches for a parking lane that might have available spots when he/she enters the parking lot.



(a) observation 1 (b) observation 2 (c) observation 3

Fig. 5. Three observations and spot sets S_1, S_2, and S_3.

- **Spot Searching:** After entering the selected parking lane, the driver searches for an available parking spot in this parking lane.

Note that, if the driver cannot find any available spot or choose not to park in a lane during spot searching, then he will go back to the lane-searching phase. Thus, the two phases alternate during the entire searching process.

We also define two states of driving direction during searching: driving away from the destination and driving towards the destination.

3.2 Three Key Observations

As previously discussed, a driver's search trajectory and destination reflect the availability of parking spots. To extract the availability information, we make the following observations about the relationship between the parking availability and the driver's parking search process.

- Assuming a driver parks at a Spot A, the spots that are closer to the destination than Spot A (denoted as spot set S_1) are likely to be occupied.
- During the lane-searching process, if a driver is driving away from his destination, the spots in the passed-by parking lanes (denoted as set S_2) are likely to be occupied.
- During the spot-searching process, if a driver is driving away from his destination, then the passed-by spots in this lane (denoted as set S_3) are likely to be occupied.

Examples of the three aforementioned observations are illustrated in Figure 5. As can be seen from Figure 5(a), the driver selects parking Lane 1 and parks at Spot 3. The spots that are closer to the destination than Spot 3 belong to the spot set S_1, which are highlighted by shaded blue. In Figure 5(b), the driver drives away from the destination during lane searching and selects parking Lane 4, as shown in red dashed line. The parking lanes (Lane 1, 2, and 3) passed by the driver during lane searching belong to the spot set S_2, as shown in the shaded blue in Figure 5(b). In Figure 5(c), the driver drives away from the destination when searching for available spots in parking Lane 4 during spot searching. The passed-by spots during spot searching belong to the spot set S_3, as shown in the shaded blue in Figure 5(c).

Previous studies [Axhausen and Polak 1991; Feeney 1989] show that drivers tend to minimize their walking distance to the destination rather than minimizing driving distance. This also corresponds to the intuition that the driver tends to find a parking spot as close to his destination as possible to reduce walking distance. This explains our first observation. We can further extend this finding that: If a driver passes by and does not park at the spots/lines that are closer to destination, it is more likely that these spots/lines are occupied. This explains our second and third observations. The three spot sets generated from the three key observations can cover a large portion of spots even when the driver only passes by a few spots, as shown in Figure 5, which solves the low coverage rate problem. Obviously, there could be overlaps between spot sets S_1, S_2, and S_3.

3.3 Probability Profile

We have introduced the three key observations on the relationship between parking availability and the drivers' search process. Based on these observations, we propose a probability profile generation model to generate the occupancy probability of each spot. In the proposed model, each spot has only two status, occupied or empty, and its probability is treated as a Bernoulli distribution. We calculate the probability of spot i being occupied is x_i (the probability of it being available is $1 - x_i$) by Equation (1):

$$x_i = \min\{\alpha(C_i^1 + C_i^2 + C_i^3) + x_i^{\text{default}}, 1\}, \tag{1}$$

where,

$$C_i^1 = \begin{cases} 1, & \text{if } i \in S_1 \\ 0, & \text{o.w.} \end{cases}, C_i^2 = \begin{cases} 1, & \text{if } i \in S_2 \\ 0, & \text{o.w.} \end{cases}, C_i^3 = \begin{cases} 1, & \text{if } i \in S_3 \\ 0, & \text{o.w.} \end{cases}. \tag{2}$$

x_i^{default} is the historical occupancy probability of spot i, which is the frequency of the spot being occupied in each time period from historical records. The historical occupancy probability is only required in the initialization phase, and the estimated results thereafter can be used as historical occupancy. The historical occupancy probability of the entire parking lot can be used as x_i^{default} during initialization. Detailed discussions about historical probability will be given in Section 6. It is intuitive that a spot has higher probability of being occupied if it belongs to more than one spot sets. In Equation (1), if spot i belongs to all three sets S_1, S_2, and S_3, the occupancy probability will be high. If spot i does not belong to any of the three spot sets, the occupancy probability is the same as the historical probability. The parameter α balances the probability introduced by the three observations and the historical probability. If α is very large, then the spots in $S_1 \cup S_2 \cup S_3$ will have occupancy probability of 100%; if α is very small, then the occupancy probability is almost the same as historical availability. With the estimated occupancy probability of each spot i, a probability profile of the entire parking lot is created from an individual vehicle. Further experiments to validate the proposed model are presented in Section 5.

In some parking lots, there are some spots that are assigned to specific vehicles or people such as green energy vehicles. However, for most drivers, those spots are not allowed to park and are not taken into consideration. In addition, only a few numbers of spots in a parking lot are assigned to specific vehicle/people. To handle those special spots, our system first eliminates those special spots from the static map of the parking lot before generating the probability profile. For drivers that are not allowed to park at those special spots, they can ignore the special spots and our system can still provide parking availability information of other spots. For drivers who are allowed to park at the special spots, they can easily check the availability of the special spots, because there are only a few special spots, or they can still park at other spots based on the parking availability information provided by our system.

3.4 Variety in the Quality of Information

The estimated probability profile from a single vehicle is prone to errors introduced due to varying parking behaviors and inaccuracy in observations made by drivers. For example, in Figure 6, the driver (denoted as d_1) took a look for available spots when passing the first and second lanes. However, as can be seen from the snapshot shown on the right-hand side of Figure 6, the visibility

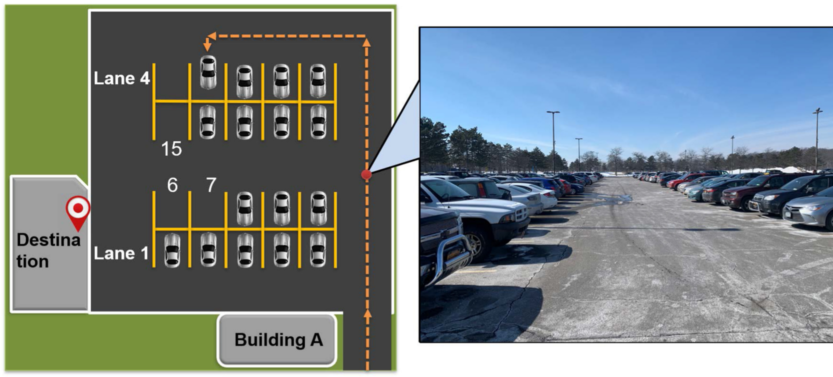


Fig. 6. Observation at parking lot.

of the spots far away from the trajectory is poor and sometimes even blocked. The driver cannot decide whether all the spots on each line are taken or not. The driver d_1 does not want to take any risk wasting time and directly drives to the most likely available spots (i.e., the spots that are far from any of the buildings). Thus, the driver ignores the available spots closer to destination (Spots 6, 7, and 15) and parks at Spot 17, which is conflicted with our observations. Based on our probability profile generation model, the closer Spots 6, 7, and 15 should have high probability of being occupied, which obviously deviate from the truth. In this example, we can say that the information provided by this driver is of low quality.

Thus, the differences in drivers' parking behaviors result in the variety in the **Quality of Information (QoI)** of different vehicles. When multiple vehicles (or sources) park at the parking lot and generate multiple probability profiles in a given time period, we wish to trust more on the sources that provide high-quality information (more accurate probability profile) and ignore the sources that provide low-quality information. The QoI of sources are not considered by existing smart parking systems.

4 PROBABILITY PROFILES AGGREGATION

In a given time period, there can be multiple vehicles entering and parking in a parking lot, and each vehicle generates availability information based on the proposed probability profile generation model. To resolve conflicts of probability profiles from different vehicles, an intuitive solution is to perform aggregation of probability profiles from different vehicles to improve the overall accuracy of the system.

4.1 Problem Definition

We first formally define the aggregation problem. Each spot in the parking lot is denoted by $n \in \{1, 2, 3, \dots\}$. The timestamp is denoted as $t \in \{1, 2, 3, \dots\}$. At time t, the set of participating vehicles (i.e., vehicles that provide sensor data) that park at a parking lot is denoted as K_t. Let $x_{i,t}^k$ represent the probability profile generated from vehicle k on spot i at time t, where $k \in K_t$. In this case, the problem is to aggregate the individual profiles on spot i at time t, $\{x_{i,t}^k\}_{k \in K_t}$ and calculate the final result $\{x_{i,t}^*\}$. The aggregated result of all the spots at time t is X_t^*. Based on the aggregated probability profile, we can find out whether a specific spot i is available by setting the probability threshold to 0.5. Table 1 summarizes all notations we use.

Table 1. Notations

Symbol	Definition
i	index of spots
N	number of spots
t	index of timestamps
k	index of vehicles
K_t	the set contains all vehicles that park at time t
w_k	weight of vehicle k
$x_{i,t}^k$	probability profile from vehicle k on spot i at time t
$x_{i,t}^*$	aggregated results on spot i at time t
X_t^*	the set of aggregated results of all spots at time t

4.2 Truth Discovery

One way of aggregating the estimations from multiple sources is to simply take the average values or the majority of the estimations for categorical data. However, these methods treat all sources equally and ignore the differences in sources' reliability. Thus, majority voting and averaging are not suitable for our problem due to the varying QoI, as we discussed before. Truth discovery methods [Cherian et al. 2016; Ma et al. 2014; Nandugudi et al. 2014; Nawaz et al. 2013; Soubam et al. 2016; Xu et al. 2013] take the varying QoI into consideration in the data aggregation framework by jointly estimating source reliability and aggregating multi-source data weighted by the estimated reliability. The general formulation is as follows:

$$\min_{X^*, W} f(X^*, W) = \sum_{i=1}^{N} \sum_{k=1}^{K} w_k ||x_i^k - x_i^*||^2, \tag{3}$$

$$s.t. \sum_{k=1}^{K} exp(-w_k) = 1, \tag{4}$$

where w_k is the weight (i.e., source reliability) for source k, and x_i^k is the report of object i from source k. x_i^* is the estimated truth for object i. The expected outputs are the estimated truths of all the objects X^* and sources' weights W. Equation (3) shows that the weighted sum of the distance between the report on object i from source k and the estimated truth needs to be minimized. Equation (4) constrains the weights w_k that avoids the weights w_k to be non-positive.

In our problem, each participating vehicle can be treated as a source, and each parking spot can be treated as an object. The estimated truth X^* in Equation (3) is the occupancy probability of all the parking spots. Our goal is to make estimated truth close to the ground truth occupancy probability.

4.3 Proposed Algorithm

Two major challenges prevent us from directly applying existing truth discovery frameworks to our problem of parking spot availability estimation. First, along with the truth (i.e., occupancy probability of spots), the sources (i.e., participating vehicles) are constantly evolving. In real parking scenarios, new vehicles keep entering and parking in the parking lot. For example, there are N participating vehicles at timestamp t and there might be $N+K$ participating vehicles at next timestamp $t+1$, which means K new vehicles park at the parking lot from time t to $t+1$. Equation (3) can not be applied to our problem directly, because the sources $\{k\}_{k \in K}$ is evolving. Second, due to limited view of the drivers, they are incapable of making accurate observation on all parking spots. Therefore, in a real-world parking scenario, a vehicle may need to be treated as an unreliable source when estimating the availability of certain spots. This results in the problem of observation difference, where a source (vehicle) may have different levels of reliability on different objects (parking spots). To overcome these two challenges, we propose a novel truth discovery algorithm, called **SpotE-TD**.

4.3.1 Evolving Sources. To deal with the evolving sources, a naive solution would be to execute the truth discovery algorithm at each timestamp t. This can be achieved by considering the vehicles that enter the parking lot at time t as sources, assuming that the parking availability does not change during this period of time. However, this solution suffers from information loss, because historical estimations from previous sources are not considered. For example, the parking availability at time t can be very similar to the availability at time $t-1$. Besides, when the penetration rate (i.e., the percentage of participating vehicles) is small, the vehicles entering at each time slot are sparse, which might result in poor accuracy.

To handle this problem, our proposed algorithm utilizes observations from all previous vehicles. We introduce a decay factor F to control the weight of a source. The new weight is updated to $w_k F(|t_k - t|)$. t_k is the time when vehicle k enters and parks in the parking lot, and t is the current time. The decay function $F(*)$ assigns lower weights to older sources.

Based on the above description, at time t, all the sources before time t need to be taken into consideration and fed into the equation. This can be inefficient during busy hours when the number of participating vehicles entering the parking is large. To improve the efficiency, we treat the sources before the current time slot as an integrated source. The estimation at time $t-1$ is treated as a new source at time t. So the algorithm runs recursively at each timestamp, formulated as:

$$\min_{X_t^*, W} f(X_t^*, W) = \sum_{i=1}^{N} \sum_{k \in K_t} w_k ||x_{i,t}^k - x_{i,t}^*||^2 + \sum_{i=1}^{N} \eta w_{t-1} ||x_{i,t-1}^* - x_{i,t}^*||^2, \tag{5}$$

$$s.t. \sum_{k \in K_t} exp(-w_k) + exp(-w_{t-1}) = 1, \tag{6}$$

where w_{t-1} is the weight of previously estimated truth, and $x_{i,t-1}^*$ is the estimated truth of object i at previous timestamp $t-1$. The first part in the right-hand side of Equation (5) is the weighted distance between sources' reports and estimated truth of object i at time t, and the second part in the right-hand side of Equation (5) is the weighted distance between the estimated truth at time $t-1$ and the estimated truth at time t. The decay factor η is a predefined parameter that controls how much we discount previously estimated truth. The range of η is between $(0, 1)$. Smaller η means lower reliability value on previously estimated truth. When the truth is evolving very fast, the decay factor should be small.

By introducing the previously estimated truth and its weight, our algorithm not only considers all the previous sources, but also reduces the computation load, because it does not need to recalculate all the previous sources' weights.

4.3.2 Observation Difference. When a vehicle enters the parking lot, some of the spots are not observed well due to the observation range of the driver being limited. Thus, the observation differences affect the source's reliability for some parking spots. As the example shown in Section 3.4, the occupancy probability estimations on Spots 6, 7, and 15 deviate from truth due to the poor observation.

To capture the observation difference, we adjust the source reliability on different objects. We define an observation function $O(d_i^k)$. If a spot has larger distance from the trajectory d_i^k, then the value of the function is smaller, which corresponds to the fact that drivers have poor observations on the far-away spots. The source weight becomes $\tilde{w}_k = w_k O(d_i^k)$. Thus, when a vehicle is searching in the parking lot, the spots that are far away from its searching trajectory have low observation values, and the reliability score of the vehicle on those spots is low. We let $O(d_i^k) = exp(-\beta d_i^k)$, where β is a predefined positive parameter. If β is small enough, then the observation value becomes 1, which means the source has equal reliability on all the spots. If β is very large, then the observation value will decrease very fast with respect to the distance and the number of spots with reliable observation might be too small to enable accurate availability information for the whole parking lot. We will discuss the effect of β in our experiment.

The new weight becomes:

$$\begin{aligned} \tilde{w}_k &= w_k O(d_i^k), \\ &= w_k exp(-\beta d_i^k). \end{aligned} \tag{7}$$

4.3.3 Optimization. After introducing the evolving sources and observation difference, the objective function is:

$$\min_{X_t^*, W} f(X_t^*, W) = \sum_{i=1}^{N} \sum_{k \in K_t} w_k O(d_i^k) ||x_{i,t}^k - x_{i,t}^*||^2 + \sum_{i=1}^{N} \eta w_{t-1} ||x_{i,t-1}^* - x_{i,t}^*||^2, \tag{8}$$

$$s.t. \sum_{k \in K_t} exp(-w_k) + exp(-w_{t-1}) = 1. \tag{9}$$

The weights and estimated truths need to be learned from this equation. Block coordinate descent is adopted. The idea is to fix other parameters and optimize one parameter at each time. After several iterations, the estimated truth will converge. In this problem, the optimization process contains two steps: weights update and truths update. In the weights update step, we fix the truths X_t^* and optimize the weights $\{w_k\}_{k \in K_t}$ and w_{t-1}. In the truths update step, we fix the weights $\{w_k\}_{k \in K_t}$ and w_{t-1}, and optimize the truths X_t^*.

Weights update. After applying Lagrange multiplier, the equation becomes:

$$\begin{aligned} \min_{X_t^*, W} f(X_t^*, W) = & \sum_{i=1}^{N} \sum_{k \in K_t} w_k O(d_i^k) ||x_{i,t}^k - x_{i,t}^*||^2 + \sum_{i=1}^{N} \eta w_{t-1} ||x_{i,t-1}^* - x_{i,t}^*||^2 \\ & + \lambda \left(\sum_{k \in K_t} exp(-w_k) + exp(-w_{t-1}) - 1 \right). \end{aligned} \tag{10}$$

Set the derivative with respect to $\{w_k\}_{k \in K_t}$ to be 0, then:

$$\lambda exp(-w_k) = \sum_{i=1}^{N} O(d_i^k) ||x_{i,t}^k - x_{i,t}^*||^2. \tag{11}$$

Set the derivative with respect to w_{t-1} to be 0, then:

$$\lambda exp(-w_{t-1}) = \sum_{i=1}^{N} \eta ||x_{i,t-1}^* - x_{i,t}^*||^2. \tag{12}$$

According to the Equation (9) we can get:

$$\lambda = \sum_{k \in K_t} \sum_{i=1}^{N} O(d_i^k)||x_{i,t}^k - x_{i,t}^*||^2 + \sum_{i=1}^{N} \eta ||x_{i,t-1}^* - x_{i,t}^*||^2. \tag{13}$$

Then, the updated weights $\{w_k\}_{k \in K_t}$ and w_{t-1} can be calculated as:

$$w_k = -\log\left(\frac{\sum_{i=1}^{N} O(d_i^k)||x_{i,t}^k - x_{i,t}^*||^2}{\sum_{k \in K_t} \sum_{i=1}^{N} O(d_i^k)||x_{i,t}^k - x_{i,t}^*||^2 + \sum_{i=1}^{N} \eta ||x_{i,t-1}^* - x_{i,t}^*||^2}\right), \tag{14}$$

$$w_{t-1} = -\log\left(\frac{\sum_{i=1}^{N} \eta ||x_{i,t-1}^* - x_{i,t}^*||^2}{\sum_{k \in K_t} \sum_{i=1}^{N} O(d_i^k)||x_{i,t}^k - x_{i,t}^*||^2 + \sum_{i=1}^{N} \eta ||x_{i,t-1}^* - x_{i,t}^*||^2}\right), \tag{15}$$

where,

$$O(d_i^k) = exp(-\beta d_i^k). \tag{16}$$

Truths update. Similar to weights update step, we assume the weights are known and apply the Lagrange multiplier. Then, we let the derivative with respect to X_t^* be 0, then:

$$x_{i,t}^* = \frac{\sum_{k \in K_t} w_k O(d_i^k) x_{i,t}^k + \eta w_{t-1} x_{i,t-1}^*}{\sum_{k \in K_t} w_k O(d_i^k) + \eta}, \tag{17}$$

where,

$$O(d_i^k) = exp(-\beta d_i^k). \tag{18}$$

The algorithm is summarized in Algorithm 1.

ALGORITHM 1: SpotE-TD

Input: η and x_i^k and previous truth X_{t-1}^*
Output: weights $\{w_k\}_{k \in K_t}$, w_{t-1} and truth X_t^*
1 Initialize $x_{i,t}^*$
2 **Repeat:**
3 update sources' weights $\{w_k\}_{k \in K_t}$, w_{t-1} according to Equation (14) and Equation (15)
4 update truths $x_{i,t}^*$ according to Equation (17)
5 **Until** converge

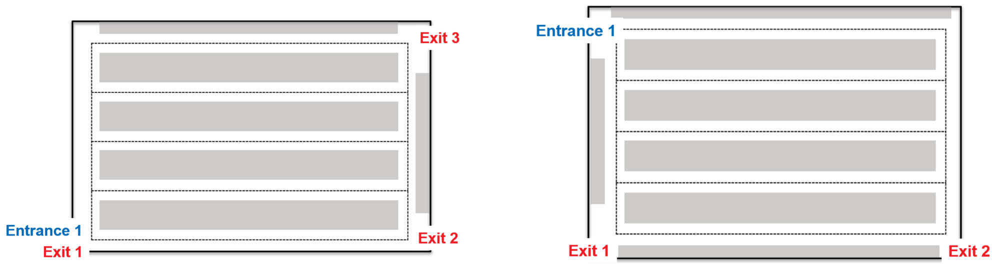
5 EVALUATION

5.1 Evaluation Setup

We ask volunteers to install our App and collect sensor data from their smartphones. In Section 2, we gave an introduction on how our system collects and processes sensor data from users. However, the sensor data is not enough to evaluate our system, due to the limited number of volunteers and smartphones. To obtain enough data for evaluation, we set up a camera in each parking lot and collect video data, as shown in Figure 7. Video processing techniques [Kovvali et al. 2007; Wei et al. 2005] are used to capture the vehicles' trajectories and identify their destinations. Note that the trajectories and destinations generated from videos might also be inaccurate due to video processing algorithms.

The ground truth availability is also collected by cameras that can cover the whole parking lot. Image processing techniques are used to infer the availability of each spot at each time slot. Many existing imaging processing algorithms [Lee and Seo 2016; Yusnita et al. 2012] can be used to detect whether a parking spot is occupied or not.

Fig. 7. A snapshot from video.



(a) Layout of parking lot 1 (b) Layout of parking lot 2

Fig. 8. Layouts of parking lots.

To study the generalization of our system, we collect data under different scenarios, including work days and holidays, in different parking lots and at different time periods in a day. We first select two parking lots on campus, named as parking lot 1 and lot 2. The layouts of parking lots are shown in Figure 8. Parking lot 1 has 237 parking spots with three exits and parking lot 2 has 354 parking spots with two exits. We combine the results from the two parking lots in the following evaluations. We collect 12 days of video data by setting up cameras on parking lot and smartphone sensor data from five volunteers. The data is collected from 8:00 to 18:00 in each day, which covers both busy hours and non-peak hours. In total, we collect around 120 hours data, which contains 1,691 valid parking events. We estimate the parking lot availability every 10 minutes, so the data size for classification is 177,720 in total.

The spot-level parking availability estimation at each time slot is a two-class classification problem of multiple objects. The state of each spot is either "occupied" or "empty." **Confusion Matrix** can be used to evaluate this classification problem. For drivers who are searching for available spots, the empty spot is more important than the occupied spot. The performance of estimating/detecting the empty spot is an important aspect when evaluating a parking monitoring system. So, in the confusion matrix, we define "empty" spot as positive class and "occupied" spot as negative class. In our evaluations, we calculate **Accuracy**, i.e., $\frac{True\ positive+True\ negative}{Total\ number\ of\ spots}$ and **F-score**, i.e., $2 * \frac{Precision*Recall}{Precision+Recall}$ at each time slot and take the average values of all time slots as final measurement. Accuracy is used to evaluate the overall performances of estimating both empty spot and occupied spot. F-score evaluates both the Precision and Recall. The Precision reflects the percentage of correctly estimated empty spots among all the spots that are estimated as "empty," while the Recall reflects the percentage of correctly estimated empty spots among all the spots that are actually empty, both of which can be used to evaluate the performances of detecting empty spots. In addition, when calculating the Accuracy and F-score, we repeat the experiments for 20 times and report the averaged results.

5.2 Performances Analysis

To validate our three key observations and probability profile generation model, we calculate the availability results from individual probability profile without applying aggregation method, named as **SpotE-Individual**. We compare SpotE-Individual with existing methods including ParkScan and historical probability. To evaluate our proposed truth discovery method, we aggregate the individual probability profiles using the proposed method **SpotE-TD** and compare it with other aggregation baselines. We summarize all the methods below.

- **ParkScan [Liu et al. 2017]:** ParkScan proposes a parking decision model and infers the availability of the spots along the vehicles' trajectories. Then it models the state of each parking spot via a hidden Markov process. Each spot updates its availability based on the pass-by events (a vehicle passes by the spot). If the spot is not passed by any vehicle at time t, then its availability is updated according to the historical transition probability, which is calculated by the number of parking events divided by length of time interval.
- **Historical:** It calculates the frequency of each spot being occupied in the given time period based on historical records, and uses the frequency as the availability probability of this spot. The probability threshold is set to 0.5: If the probability of being occupied is larger than 0.5, then the spot is occupied.
- **SpotE-Individual**: Based on the three key observations and probability profile generation model, it generates the occupancy probability of each spot (probability profiles). The availability results are directly generated from the probability profile without aggregation. The probability threshold is 0.5.
- **SpotE-Mean:** Similar to SpotE-Individual, it first generates the probability profiles from each vehicle in a given time period. It introduces an aggregation step to aggregate the profiles from different vehicles. The final availability results are generated by calculating the mean values of the probability profiles in the given time period. The probability threshold is 0.5.
- **SpotE-GTM [Zhao and Han 2012]:** GTM is a truth discovery method that deals with numerical data. It is a Bayesian probabilistic model and tries to maximize a posterior probability of each object. After generating probability profiles, GTM is performed at each time slot to aggregate the probability profiles and generate the final availability results. The probability threshold is 0.5.
- **SpotE-CRH [Li et al. 2014b]:** CRH is also an existing truth discovery method. It treats the problem as an optimization problem and minimizes weighted sum of distance between the individual claims and estimated truth. In our problem, CRH is performed at each time slot to aggregate the probability profiles and generate the final availability results. The probability threshold is 0.5.

Table 2 summarizes the performances of **SpotE-Individual**, **SpotE-TD**, and the baseline methods. We did not compare our system with other smartphone-based parking availability monitoring system, because ParkScan is the only smartphone-based method that can provide spot-level availability information. We can find that even without aggregation, the performance of SpotE-Individual is better than the performances of ParkScan and Historical, with an accuracy of 75.87%. The results indicate that we can accurately estimate the parking availability from the three key observations even without aggregation, and our three key observations proposed in Section 3.2 can capture the relationship between parking availability and drivers' parking search process. The poor performance of ParkScan is because of its low coverage rate problem in large parking lots, as described before. In our experiments, the parking lots are large and most spots are not visited by any vehicle in some time periods. For ParkScan, the availability of those spots is updated

Table 2. Performances Comparison

Methods	Accuracy	F-score
Historical	0.6852	0.6501
ParkScan	0.7353	0.6731
SpotE-Individual	**0.7587**	**0.7273**
SpotE-GTM	0.8586	0.8258
SpotE-CRH	0.8607	0.8226
SpotE-Mean	0.8600	0.8228
SpotE-TD	**0.8842**	**0.8505**

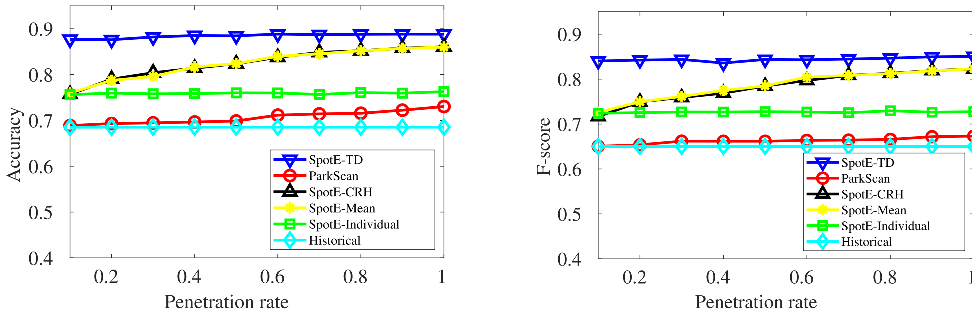
according to the historical transition probability, which can be very inaccurate. Compared with ParkScan, the proposed probability profile generation model can handle the low coverage rate problem and achieves better performances than existing methods.

The performances improvement after aggregation is also illustrated in Table 2. The performances of SpotE-CRH, SpotE-GTM, and SpotE-Mean are better than SpotE-Individual, which means aggregation can improve the estimation performances. Theoretically, SpotE-CRH and SpotE-GTM should perform better than SpotE-Mean, because truth discovery methods consider source reliability. However, as shown in Table 2, SpotE-CRH and SpotE-Mean perform equally, and SpotE-GTM performs even worse than SpotE-Mean. This is because SpotE-CRH and SpotE-GTM can not handle the problem of observation difference and evolving sources, as described in Section 4. SpotE-GTM assumes that the claims from sources follow the Gaussian distribution and performs well when the number of sources are very large. However, the problem of estimating parking availability is characterized by sparse sources during a time slot and the estimations might not follow the Gaussian distribution. Careful consideration of evolving sources and observation difference in our aggregation framework (Section 4) results in better performance than any of the baselines, with the accuracy of 88.42%.

5.3 Performance w.r.t. Penetration Rate

In practice, it is almost impossible to make sure every vehicle that parks in the parking lot is monitored by our App. Otherwise, we could simply record every vehicle's parking location and generate the availability. We introduce the penetration rate, which is defined as the percentage of participating vehicles among all the vehicles that enter the parking lot. High penetration rate means that a large percentage of vehicles are monitored by our system. The penetration rate can sometimes be very small, which might influence the performance. We evaluate the performances of our system and other baselines with respect to penetration rate in this experiment. To obtain data for different penetration rates, we randomly sample the vehicles from all the vehicles in the video data. Figure 9 shows how penetration rate affects the performances of SpotE-Individual, SpotE-TD, and other baseline methods.

As shown in Figure 9, SpotE-Individual performs better than ParkScan and Historical. This shows that our probability profile generation model can accurately estimate the parking availability under different penetration rate. Compared with the performance of Historical, the performance of ParkScan is not improved significantly when the penetration rate increases, which is different from the evaluation results in Liu et al. [2017]. This is because the performance of ParkScan depends on the number of pass-by events of the vehicles at each time slot. For the spots that are not visited by any vehicles, their occupancy probability depends on the historical availability. In our experiments, the number of parking spots in the parking lots are much larger (three



(a) Accuracy (b) F-score

Fig. 9. Performance with respect to penetration rate.

times larger) than that in Liu et al. [2017], so vehicles only pass by a few spots when searching for available spots at each time slot. Even with 100% penetration rate, the pass-by events for individual spots are still very sparse. As a result, the estimated availability of each spot is dominated by historical availability.

After aggregating probability profiles, SpotE-TD outperforms other baselines under different penetration rate. As the penetration rate decreases, the performance of SpotE-TD maintained well. The performances of SpotE-CRH and SpotE-Mean are very similar as mentioned above. Their performances decrease when the penetration rate decreases. When the penetration rate is 0.1, the accuracy of SpotE-CRH and SpotE-Mean is around 75%, which is similar to that of SpotE-Individual. This means SpotE-CRH and SpotE-Mean can not generate accurate results with low penetration rate. Compared with these baselines, SpotE-TD can still achieve high accuracy even when the penetration rate is small, demonstrating the robustness of SpotE-TD with respect to the percentage of participating vehicles. This is because SpotE-CRH and SpotE-Mean only aggregate the sources at each given time slot and do not consider the evolving sources. When the penetration rate is small, the number of sources at each time slot is small, which affects the estimation results. Our method can provide accurate results with small penetration rate by considering the previous sources and previous truths.

5.4 Performance w.r.t. Different Overall Occupancy

To illustrate the validity of the three key observations and our proposed probability profile generation model with respect to the percentage of occupied spots (i.e., overall occupancy), we compare the performances of SpotE-Individual with other baselines including Historical and ParkScan. We calculate the Accuracy and F-score with the overall occupancy changing from 0.3 to 0.9. The penetration rate is set to 0.7.

As shown in Figure 10, SpotE-Individual performs better than Historical under different overall occupancy. This proves that our key observations can accurately capture the occupancy probability of the parking spots no matter how many spots are occupied. As the overall occupancy increases, the accuracy of SpotE-Individual increases. However, when the overall occupancy is too large (over 90%), the accuracy drops. This means driver's parking behavior does not follow our observations well when the overall occupancy is too small or too large. This can be explained by the following: When few spots are occupied in the parking lot, the drivers have many parking options, and some drivers might park their vehicles on their preferred spots such as the spots in a tree's shadow; when there are many occupied spots in the parking lot, drivers tend to minimize their walking distance and park in the spots that are closer to their destinations; when most spots are occupied

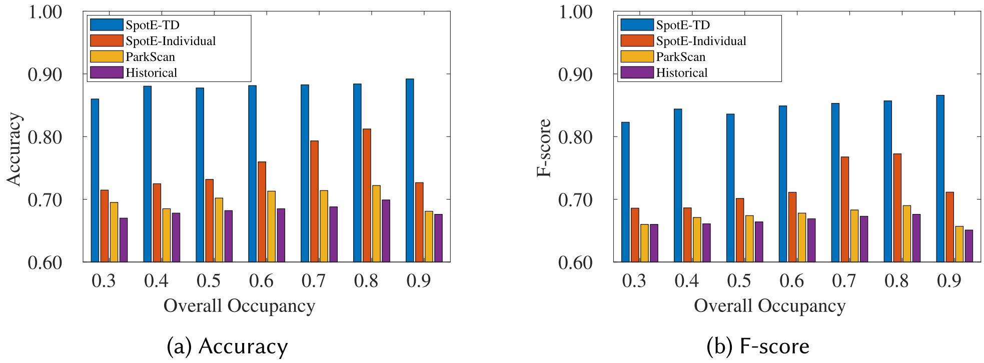


Fig. 10. Performance with respect to the overall occupancy.

in the parking lot, drivers might not want to take risk and park their vehicles in the first available spots they see.

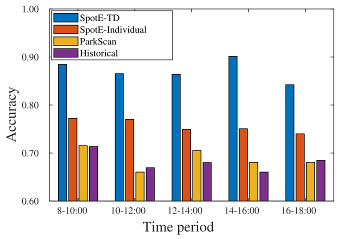
The performance of SpotE-TD is not influenced by the overall occupancy, as shown in Figure 10. This is because our truth discovery algorithm can capture the variety in QoI and assign different weights to different sources when aggregating the results.

As we discussed, correctly estimating the empty spots is more important for drivers, especially when most of the spots are occupied. To measure the performances of finding empty spots, F-score is calculated. As shown in Figure 10, the F-score value of SpotE-Individual is higher than the F-score values of Historical and ParkScan. We also calculate the Recall values of finding empty spots. SpotE-Individual achieves 73.66% Recall when the overall occupancy is 0.9, while the Recall of ParkScan is 65.4% and the Recall of Historical is 64.6%. With our proposed truth discovery method, SpotE-TD achieves 83.35% Recall when the percentage of occupied spots is 0.9. This shows that SpotE system based on the proposed probability profile generation model and the proposed truth discovery method can provide meaningful and accurate parking availability information for drivers.

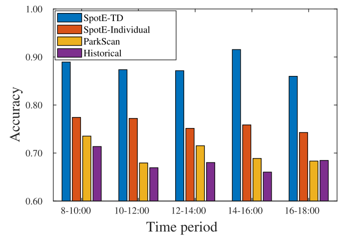
5.5 Performance at Different Time in a Day

Drivers are more concerned about the parking availability during busy hours, because finding an available parking spot might be competitive during that time. To evaluate the performances of our system in different time periods of a day, we collect data from 8:00 to 18:00, when most of the parking events happen. The entire duration is divided into five time periods, and the penetration rate is set to 20%, 60%, and 100%. Figure 11 shows the performances of SpotE-TD, SpotE-Individual, ParkScan, and Historical in different time periods with different penetration rates.

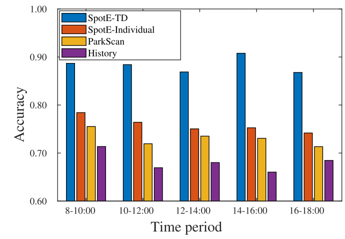
The performances of SpotE-Individual in different time periods are constantly better than the performances of ParkScan and Historical in different time periods and penetration rates. After aggregating with the proposed truth discovery method, SpotE-TD has the best performances. Especially, SpotE-TD has better performances in time period 8:00–10:00 and 14:00–16:00 compared with the performances in time period 12:00–14:00 and 16:00–18:00. This is because the time period 8:00–10:00 and 14:00–16:00 are busy hours and a large number of vehicles enter and park in the parking lot. Our proposed method estimates the availability based on the parking events instead of the leaving events. More parking events provide more availability information and help improve the performance. The accuracy values of all the methods are larger when the penetration rate is larger.



(a) Penetration rate is 0.2



(b) Penetration rate is 0.6



(c) Penetration rate is 1.0

Fig. 11. Accuracy in different time periods.

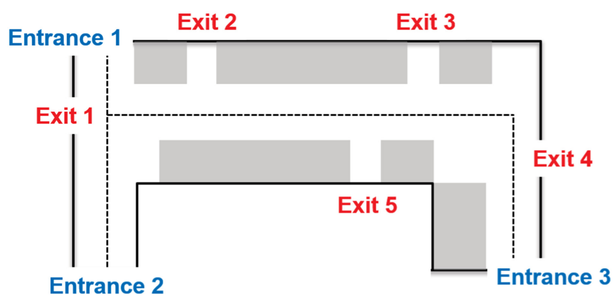


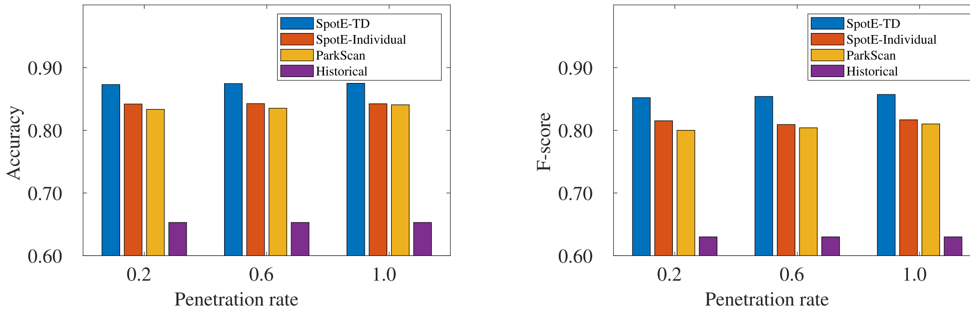
Fig. 12. Layout of parking lot 3.

5.6 Performance in Different Types of Parking Lots

To show the generalization of the proposed three key observations and the proposed truth discovery method, we evaluate the performances in a different parking lot located in an apartment, named as parking lot 3. Figure 12 shows the layout of parking lot 3. The layout of this parking lot is different from parking lots 1 and 2, and is similar to on-street parking places. Compared to parking lots 1 and 2, parking lot 3 is smaller in size and less busy, which means the frequency of parking events is lower than parking lots 1 and 2. Moreover, it has more exits/destinations than parking lots 1 and 2. Figure 13 illustrates the performances of SpotE-Individual, SpotE-TD, Historical, and ParkScan on parking lot 3 under three different penetration rates, including 0.2, 0.6, and 1.0. It can be seen that ParkScan performs better in parking lot 3 than in parking lots 1 and 2, with an accuracy of 84.17%. This is because parking lot 3 is smaller than the other two parking lots, and each spot can be passed by many times in each time slot. The high coverage rate in parking lot 3 helps ParkScan achieve accurate estimations. However, SpotE-Individual still performs better than ParkScan and Historical in the new parking lot, with an average accuracy of 84.2% when the penetration rate is 1. With our proposed truth discovery method SpotE-TD, the performances can be further improved. These results show that our proposed system still performs well in different types of parking lots.

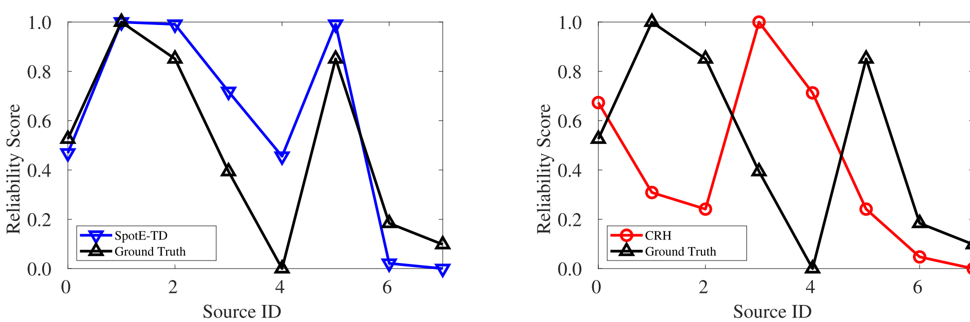
5.7 Reliability Scores

Our method as well as other truth discovery methods can learn the reliability scores of the sources. To show the accuracy of source reliability scores learned by our SpotE-TD, we compare the reliability scores learned by SpotE-TD and CRH with ground truth reliability. The ground truth reliability score is calculated by the similarity between each source's report and the ground truth values. In our evaluation, the similarity is measured by the distance between source's report and the ground truth. The distance is defined as $D = \sum_{i=1}^{N} ||x_i^k - x_i^G||^2$, where x_i^k is the occupancy probability of



(a) Accuracy (b) F-score

Fig. 13. Performance in parking lot 3.



(a) SpotE-TD (b) CRH

Fig. 14. Comparison of reliability scores.

spot i generated by vehicle k and x_i^G is the ground truth occupancy probability of spot i. If spot i is occupied, then x_i^G is 100%, otherwise it is 0. We then normalize the distance D into the range of [0,1], and the ground truth reliability score is $S = 1 - D$. The generated reliability score of each source in traditional truth discovery methods is the weight of the source. In SpotE-TD, the generated reliability score is the sum of fine-grained weights of all the objects. We also normalize the generated reliability score into the range of [0,1]. The larger the reliability score is, the more reliable the source is. Note that the reliability of a source can be 0.

We select eight sources at a selected time slot and generate their reliability scores using SpotE-TD and CRH. Figure 14(a) and Figure 14(b) show the comparison of reliability scores generated by SpotE-TD and CRH with the ground truth. It can be seen that SpotE-TD learns the source reliability scores more precisely than CRH. Take Source 4 and Source 1, for example: The true reliability of Source 4 is low while the true reliability of Source 1 is high, but CRH assigns Source 4 with a high reliability score and assigns Source 1 with a low score.

The reason for CRH's poor performance is that it does not consider the observation difference of each vehicle. As we discussed in Section 4, some spots are far away from the trajectory, so the occupancy probability of these spots in the profile can sometimes be wrong. CRH does not consider the observation difference and treats all the spots equally, which results in inaccurate reliability scores.

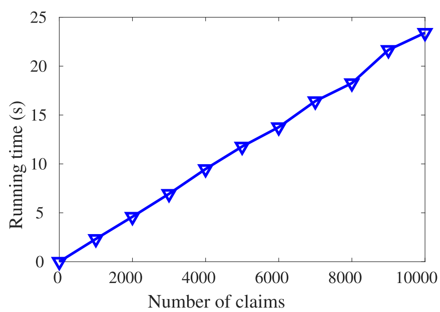


Fig. 15. Running time.

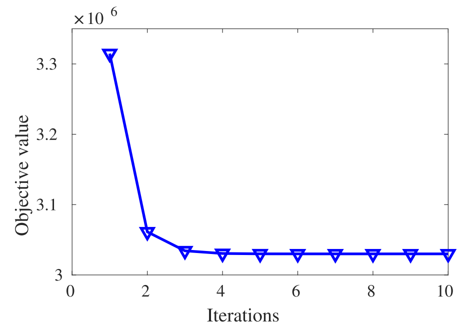


Fig. 16. Convergence.

5.8 Running Time

To demonstrate the runtime efficiency of our truth discovery method, we calculate the running time with respect to different number of claims/sources. To achieve this, we simulate the parking process at one time slot and generate the probability profiles from the simulated data. Then, we aggregate the probability profiles and generate the final results, as well as the running time. As shown in Figure 15, the running time increases linearly as the number of claims at this time slot increases. Our method only considers the sources in each time slot and the previous truth instead of aggregating all the previous sources, which avoids the repeating computation of the same source weight. The number of vehicles that park in the parking lot at each time slot is normally less than 100. In Figure 15, when the number of claims is 100, the running time is less than 0.3 second. For users who want to find a parking spot, the time delay within 0.3 second is acceptable. This demonstrates the SpotE's capability of providing real-time parking availability information.

5.9 Converge

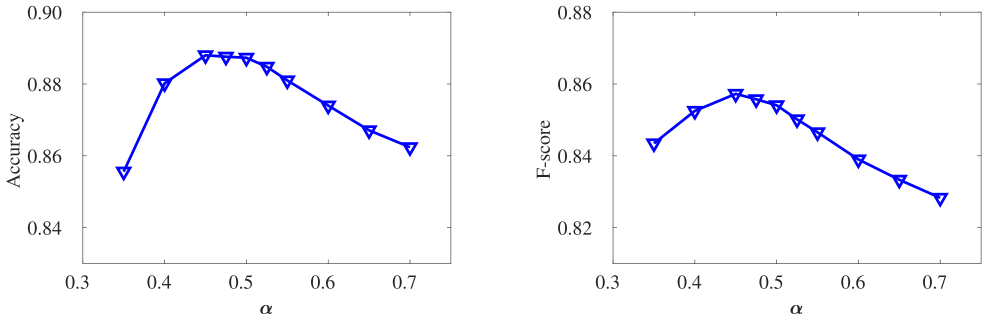
To show the convergence of the proposed algorithm, we record the objective function value at each iteration. As shown in Figure 16, the x-axis denotes the number of iterations, and the y-axis shows the objective value of formula. The objective function value decreases quickly after the first two iterations and becomes stable after the third iteration. The fast convergence shows that our algorithm can capture the reliability of each source. It also shows the efficiency of the algorithm, since it does not require many iterations to converge.

5.10 Performance w.r.t. Hyper-parameters

In this section, we compare the performance with respect to two hyper-parameters, α and β. In the following experiments, the penetration rate is set to 70%.

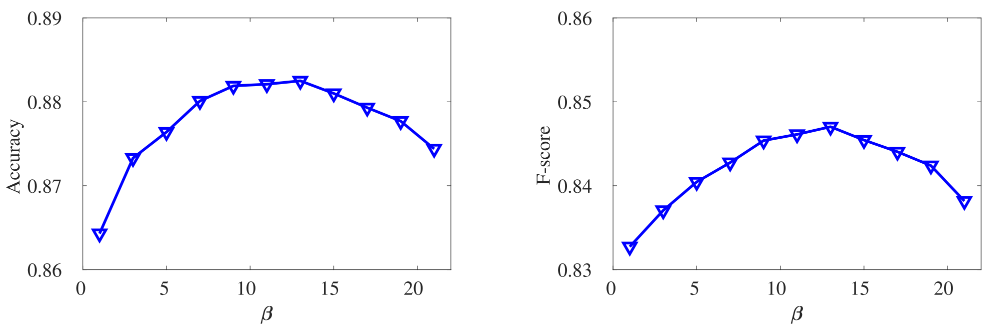
In Figure 17, we show the performances under different value of α. As the value of α increases, the accuracy and F-score increase at first and then decrease. As we mention in Section 3, α controls how much occupancy probability we assign to the spots in spot sets S_1, S_2, and S_3. According to Equation (1), when α is too small, the occupancy probability is almost the same as the historical probability. When α is too large, all the spots in S_1, S_2, and S_3 have 100% occupancy probability. We can notice that if α is in the range (0.4, 0.6), then its accuracy is over 87%. The best performance is reached when α is 0.45, with the accuracy of 88.8% and F-score of 85.73%.

In Figure 18, we show the results under different value of β. β controls how much we trust the occupancy probability estimation of spots that are far away from trajectory. The performance drops when β is very small or too large. If β is too small, then the observation difference will not



(a) Accuracy (b) F-score

Fig. 17. Performance with respect to α.



(a) Accuracy (b) F-score

Fig. 18. Performance with respect to β.

be considered into the source's reliability. And if β is too large, then the reliability on most spots will be too small, which can not reflect the true reliability accurately. Our system estimates the parking availability with an accuracy over 87% when β is between 5 and 20, which means it can achieve good performance in a large range.

In the previous evaluations, we assign α with 0.55 and β with 8, which are not the optimal values. But our method still outperforms the baseline methods. This shows that our method performs well in a large range of parameter selections.

6 DISCUSSION

In this section, we discuss about the practicability and privacy issues of our system.

Privacy: Privacy is a major concern for users, especially in a crowdsensing system. Information such as GPS trajectories and acceleration data can be used to extract drivers' personal information such as driving routes, destinations, and so on, which makes the system prone to malicious activities. In our system, users' privacy is handled in the following ways: On one hand, since the smartphone uploads raw sensor data to servers only when users are inside a parking lot, sensitive data exposed to the attackers is limited. On the other hand, GPS data and acceleration data are anonymous and will not be shared to any other third-party organizations.

GPS location: In our system, the searching trajectories and destinations are extracted using GPS data. In urban areas, the accuracy of smartphone GPS is around 10 m, which might cause errors in the trajectory extraction. However, the algorithm proposed in Section 4 can eliminate the impact of GPS errors by aggregating estimations from different vehicles. Although some parking lots are located inside buildings where GPS signal is too poor to be used, indoor localization techniques can be used to extract the drivers' trajectories in these indoor parking lots. There are many existing techniques, which utilize other information such as WiFi signal [Luo et al. 2016; Mariakakis et al. 2014] to locate the vehicle. Since WiFi signal is not always available in most indoor parking lots, Cherian et al. [2018] propose a vehicular localization system that only relies on IMU sensors. Once the vehicle enters an indoor parking lot, the search trajectory and destination can be obtained by those indoor localization systems instead of GPS.

Historical probability: Our algorithm requires historical occupancy probability information of the parking lot as the default occupancy probability in Equation (1). Obtaining the historical probability can be a problem. However, accurate historical probability of each spot is necessary in our system. Instead, average historical probability of all the parking spots is enough in our proposed system. For those parking lots that are equipped with meters at their entrances, the average historical probability can be easily obtained by counting the number of vehicles entering the parking lot. However, there are some parking lots that are not equipped with meters. Luckily, various parking Apps [ParkMobile, LLC 2008; ParkWhiz, Inc. 2006] can estimate the average availability of a parking lot. The obtained historical data can be used to initialize our proposed system. The estimated availability results can be used as historical profiles from then on.

Generalization: In our experiments, we evaluate the performances in different types of parking lots and different times in a day. The results show that the system performs well under those various scenarios. Unfortunately, we could not collect enough sensor data from smartphones due to the limited number of participants. In the future, more experiments could be done if more drivers use the App.

7 RELATED WORK

There is a variety of parking lot availability estimation systems developed in recent years. These systems can be divided into two main categories: dedicated sensor-based solutions and smartphone-based solutions.

Dedicated sensor-based solutions require sensors deployed either in the parking lot or in the vehicles. LAPark [LADoT 2017] and SFPark [SFMTA 2017] count the number of parked vehicles in parking lots using sensors installed in each parking spot. SPARK [Lu et al. 2009] provides parking availability information using **On Board Unit (OBU)** communication device and **Road-side Units (RSUs)**. ParkNet [Mathur et al. 2010] focuses on detecting the availability of road-side parking places by utilizing vehicles equipped with ultrasonic sensors. **Around View Monitor (AVM)** [Suhr and Jung 2014] and RFIDs [Gandhi and Rao 2016] can also be used in parking lot monitoring systems. Due to significant development in computer vision technology in recent years, many studies focus on detecting the occupancy of each spot using cameras [Masmoudi et al. 2014; Mateus et al. 2015; Ng and Chua 2012]. These dedicated sensor-based solutions suffer from high installation and maintenance cost.

Recent studies are making use of the sensors present in smartphones to detect the parking and unparking activities and then infer parking availability. PhonePark [Xu et al. 2013] makes use of smartphone sensors (e.g., GPS, accelerometer, and gyroscope) to detect parking and unparking activities. To mitigate the errors in detection, it also constructs a historical availability profile. ParkSense [Nawaz et al. 2013] detects and localizes parking events by capturing the wireless signature when users pay parking fee via smartphone apps and then estimates the parking availability. Updetector [Ma et al. 2014] introduces several parking and unparking indicators using smartphone sensors and proposes a method to fuse the results from different indicators. PocketParker [Nandugudi et al. 2014] also detects drivers' parking and unparking activities using smartphone's accelerometer. It assumes that some vehicles are not monitored and proposes a probability model to estimate the number of hidden drivers as well as parking availability. BluePark [Soubam et al. 2016] utilizes both accelerometer and WiFi signature to detect parking events and fuses the results together. ParkGauge [Cherian et al. 2016] proposes several parking characteristics (e.g., time-to-park and time-in-queuing/cruising) and a learning model to estimate the parking availability. Google OpenSpot [Developers 2010] aggregates drivers' manual reports on parking spot availability using crowd-sourcing techniques. Most of the smartphone-based solutions can not provide accurate spot-level availability information. ParkScan [Liu et al. 2017] builds a parking-decision model that estimates the availability of parking spots using vehicle's parking searching trajectory. It merges the estimations from different vehicles using hidden Markov model, which provides spot-level availability estimation. However, ParkScan suffers from low coverage rate and requires large training data.

Truth discovery was first proposed in Yin et al. [2008]. It is a Bayesian-based method that estimates source reliability and truth repeatedly. SSTF [Yin and Tan 2011] estimates the truths and finds reliable sources with the help of a small set of labeled ground truth data. In 3-estimate [Galland et al. 2010], the authors incorporate the single truth assumption and take the difficulty of getting truth for every object into consideration. GTM [Zhao and Han 2012] is a Bayesian probabilistic model that can deal with continuous data. Regular EM [Wang et al. 2012] formulates the truth discovery problem as a maximum likelihood estimation problem. LTM [Zhao et al. 2012] considers two types of errors (false positive and false negative) when estimating source reliability. It can also merge multi-valued attribute types. CRH [Li et al. 2014b] converts the truth discovery problem into an optimization problem. It consists of weight update and truth update steps, which minimize the weighted sum of the distance between claims and truth. CATD [Li et al. 2014a] was introduced to deal with the long-tail phenomenon, which is, most sources provide few claims and only few sources provide many claims by considering the confidence interval of the estimation. Unlike most of the above methods, our algorithm considers the evolving source problem. Besides, our algorithm considers the observation difference by using fine-grained reliability, as discussed in Section 4.

8 CONCLUSIONS

In this article, we develop a parking availability estimation system that can provide spot-level parking availability information using sensor data from smartphones. Upon three key observations on the relationship between the parking availability and drivers' parking search process, we build a probability profile generation model to generate each spot's occupancy probability from a driver's search trajectory and destination. We propose a new truth discovery method to aggregate the probability profiles from different vehicles. Our method takes evolving sources and evolving truths into consideration and introduces the source observation difference on different parking spots. We collect real-world data and evaluate the system under different scenarios. The results show that our system can provide accurate availability information under various scenarios, outperforming existing methods.

REFERENCES

Kay W. Axhausen and John W. Polak. 1991. Choice of parking: Stated preference approach. *Transport.* 18, 1 (1991), 59–81.

Jim Cherian, Jun Luo, Hongliang Guo, Shen-Shyang Ho, and Richard Wisbrun. 2016. ParkGauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics. In *17th IEEE International Conference on Mobile Data Management (MDM)*, Vol. 1. IEEE, 92–101.

Jim Cherian, Jun Luo, and Shen-Shyang Ho. 2018. ParkLoc: Light-weight graph-based vehicular localization in parking garages. *Proc. ACM Interact., Mob., Wear. Ubiq. Technol.* 2, 3 (2018), 1–23.

Google Developers. 2010. Google Open Spot Parking. Retrieved from http://techcrunch.com/2010/07/09/google-parking-open-spot/.

Bernard P. Feeney. 1989. A review of the impact of parking policy measures on travel demand. *Transport. Plan. Technol.* 13, 4 (1989), 229–244.

Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. 2010. Corroborating information from disagreeing views. In *3rd ACM International Conference on Web Search and Data Mining*. 131–140.

B. M. Kumar Gandhi and M. Kameswara Rao. 2016. A prototype for IoT based car parking management system for smart cities. *Indian J. Sci. Technol.* 9, 17 (2016).

Vijay Gopal Kovvali, Vassili Alexiadis, P. E. Zhang, et al. 2007. Video-based vehicle trajectory data collection. In *Proceedings of the Transportation Research Board 86th Annual Meeting*. 1–18.

LADoT. 2017. LA Express Park. Retrieved from http://www.laexpresspark.org/.

Soomok Lee and Seung-Woo Seo. 2016. Available parking slot recognition based on slot context analysis. *IET Intell. Transport Syst.* 10, 9 (2016), 594–604.

Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. 2014a. A confidence-aware approach for truth discovery on long-tail data. *Proc. VLDB Endow.* 8, 4 (2014), 425–436.

Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014b. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *ACM SIGMOD International Conference on Management of Data*. 1187–1198.

Ruilin Liu, Yu Yang, Daehan Kwak, Desheng Zhang, Liviu Iftode, and Badri Nath. 2017. Your search path tells others where to park: Towards fine-grained parking availability crowdsourcing using parking decision models. *Proc. ACM Interact., Mob., Wear. Ubiq. Technol.* 1, 3 (2017), 78.

Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. 2009. Map-matching for low-sampling-rate GPS trajectories. In *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 352–361.

Rongxing Lu, Xiaodong Lin, Haojin Zhu, and Xuemin Shen. 2009. SPARK: A new VANET-based smart parking scheme for large parking lots. In *IEEE International Conference on Computer Communications*. IEEE, 1413–1421.

Chengwen Luo, Long Cheng, Mun Choon Chan, Yu Gu, Jianqiang Li, and Zhong Ming. 2016. Pallas: Self-bootstrapping fine-grained passive indoor localization using WiFi monitors. *IEEE Trans. Mob. Comput.* 16, 2 (2016), 466–481.

Shuo Ma, Ouri Wolfson, and Bo Xu. 2014. UPDetector: Sensing parking/unparking activities using smartphones. In *7th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. ACM, 76–85.

Alex T. Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. 2014. Sail: Single access point-based indoor localization. In *12th International Conference on Mobile Systems, Applications, and Services*. 315–328.

Imen Masmoudi, Ali Wali, and Adel M. Alimi. 2014. Parking spaces modelling for inter spaces occlusion handling. In *22nd Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*. 119–124.

Paola A. Mateus, Edisson O. Maldonado, and Cesar L. Nino. 2015. Surveillance and management of parking spaces using computer vision. In *20th Symposium on Signal Processing, Images and Computer Vision (STSIVA)*. IEEE, 1–7.

Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. 2010. ParkNet: Drive-by sensing of road-side parking statistics. In *8th International Conference on Mobile Systems, Applications, and Services*. ACM, 123–136.

Anandatirtha Nandugudi, Taeyeon Ki, Carl Nuessle, and Geoffrey Challen. 2014. Pocketparker: Pocketsourcing parking lot availability. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 963–973.

Sarfraz Nawaz, Christos Efstratiou, and Cecilia Mascolo. 2013. ParkSense: A smartphone based sensing system for on-street parking. In *19th International Conference on Mobile Computing & Cetworking*. ACM, 75–86.

Lih Lin Ng and Hong Siang Chua. 2012. Vision-based activities recognition by trajectory analysis for parking lot surveillance. In *IEEE International Conference on Circuits and Systems (ICCAS'12)*. IEEE, 137–142.

ParkMobile, LLC. 2008. ParkMoble. http://parkmobile.io/.

ParkWhiz, Inc. 2006. ParkWhiz. Retrieved from http://www.parkwhiz.com/.

SFMTA. 2017. SFpark. Retrieved from http://www.sfpark.org/.

Donald C. Shoup. 2006. Cruising for parking. *Transport Policy* 13, 6 (2006), 479–486.

Sonia Soubam, Dipyaman Banerjee, Vinayak Naik, and Dipanjan Chakraborty. 2016. BluePark: Tracking parking and unparking events in indoor garages. In *17th International Conference on Distributed Computing and Networking*. ACM, 33.

Jae Kyu Suhr and Ho Gi Jung. 2014. Sensor fusion-based vacant parking slot detection and tracking. *IEEE Trans. Intell. Transport. Syst.* 15, 1 (2014), 21–36.

Dong Wang, Lance Kaplan, Hieu Le, and Tarek Abdelzaher. 2012. On truth discovery in social sensing: A maximum likelihood estimation approach. In *11th International Conference on Information Processing in Sensor Networks*. 233–244.

Heng Wei, Chuen Feng, Eric Meyer, and Joe Lee. 2005. Video-capture-based approach to extract multiple vehicular trajectory data for traffic modeling. *J. Transport. Eng.* 131, 7 (2005), 496–505.

Bo Xu, Ouri Wolfson, Jie Yang, Leon Stenneth, S. Yu Philip, and Peter C. Nelson. 2013. Real-time street parking availability estimation. In *IEEE 14th International Conference on Mobile Data Management*, Vol. 1. IEEE, 16–25.

Xiaoxin Yin, Jiawei Han, and S. Yu Philip. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Trans. Knowl. Data Eng.* 20, 6 (2008), 796–808.

Xiaoxin Yin and Wenzhao Tan. 2011. Semi-supervised truth discovery. In *20th International Conference on World Wide Web.* 217–226.

Rahayu Yusnita, Fariza Norbaya, and Norazwinawati Basharuddin. 2012. Intelligent parking space detection system based on image processing. *Int. J. Innov. Manag. Technol.* 3, 3 (2012), 232–235.

Bo Zhao and Jiawei Han. 2012. A probabilistic model for estimating real-valued truth from conflicting sources. In *Proc. of QDB.*

Bo Zhao, Benjamin I. P. Rubinstein, Jim Gemmell, and Jiawei Han. 2012. A Bayesian approach to discovering truth from conflicting sources for data integration. *arXiv preprint arXiv:1203.0058* (2012).

Weida Zhong, Qiuling Suo, Fenglong Ma, Yunfei Hou, Abhishek Gupta, Chunming Qiao, and Lu Su. 2019. A reliability-aware vehicular crowdsensing system for pothole profiling. *Proc. ACM Interact., Mob., Wear. Ubiq. Technol.* 3, 4 (2019), 1–26.

Received July 2020; revised January 2021; accepted April 2021