Real-time Street Parking Availability Estimation[*]

Bo Xu[⊥§] Ouri Wolfson[§] Jie Yang[§] Leon Stenneth[§] Philip S. Yu[§] Peter C. Nelson[§]

[§] University of Illinois at Chicago, USA
[⊥] Argonne National Laboratory, USA
{boxu,wolfson,jyang06,lstenn2,psyu,nelson}@uic.edu

Abstract— **Real-time parking availability information is important in urban areas, and if available could reduce congestion, pollution, and gas consumption. In this paper, we present a software solution called PhonePark for detecting the availability of on-street parking spaces. The solution uses the GPS and/or accelerometer sensors in a traveler's mobile phone to automatically detect when and where the traveler parked her car, and when she released a parking slot. PhonePark can also utilize the mobile phone's Bluetooth sensor or piggyback on street parking payment transactions for parking activity detection. Thus, the solution considers only mobile phones and does not rely on any external sensors such as cameras, wireless sensors embedded in the pavements, or ultrasonic sensors on vehicles. Further contributions include an algorithm to compute the historical parking availability profile for an arbitrary street block and algorithms to estimate the parking availability in real-time for a given street block. The algorithms are evaluated using real-time and real world street parking data.**

Keywords-transportation mode, bluetooth, activity recognition, GPS, Kalman Filter, mobile phones, parking

I. INTRODUCTION

Finding on-street parking spaces in crowded urban areas is a painful challenge to drivers and costly to society. In one business district of Los Angeles, researchers found that vehicles searching for parking traveled a distance equivalent to 38 trips around world, produced 730 tons of carbon dioxide, and burned 47,000 gallons of gasoline [1] in one year. To deal with the parking problem, cities such as San Francisco have invested millions of dollars on smart parking infrastructures. These infrastructures detect the availability of parking spaces using fixed sensors installed in the asphalt. The SFPark project in San Francisco [2] has sensors covering 8,000 parking spaces which are about 25 percent of the available street parking spaces. The total cost of the project together with smart parking management is over USD $23M [1, 2]. Another project [1] uses ultrasonic sensors that are externally mounted on car side-doors to detect parking spaces. The cost of the system for each car is approximately US $400, which includes an ultrasonic sensor, GPS, and a light weight PC with WIFI card.

In this paper, we propose a solution, called PhonePark, which detects on-street parking availability using mobile phones carried by the drivers. The solution does not rely on any dedicated infrastructure or external sensors, and therefore is much more economical than existing solutions. For any driver carrying a mobile device with GPS and/or accelerometer, PhonePark can automatically determine when and where she parked her car. The proposed technology does not rely only on GPS/accelerometer for parking activity detection, instead Bluetooth enabled mobile devices can also be utilized. Parking activities can also be detected by taking advantage of pay-by-phone parking services.

In the PhonePark solution, for parking detection using the GPS/accelerometer, we follow the general principle of sensor fusion and classification that is used for context detection or activity recognition. First, from training data, we build a transportation mode classification model in terms of mobility patterns [3]. Then, when parking is to be determined, sensor inputs are fed to the already trained classification model. Finally, certain transportation mode state transitions are detected. For example, for street parking, the following transportation mode transitions are expected to be observed: *car* → *stationary* → *walking*. Based on this transition pattern (i.e., *car* → *stationary* → *walking*), we may infer that the driver parked at the stationary point.

For parking detection via Bluetooth, the pairing connection between the driver's mobile phone and the car's on-board Bluetooth is considered. This relies on the fact that nowadays more and more cars have in-vehicle Bluetooth functionalities. In this work, parking activities can also be detected by pay-by-phone piggyback. The motivation for this technology is that over 30 cities such as Washington DC, Philadelphia, and San Francisco are using pay-by-phone parking systems whereby drivers enter their street parking slot number, duration, in addition to other payment information [4, 5]. By piggybacking on this payment technology, one can further determine when street parking slots are occupied and released.

Now consider how PhonePark would use observations of mobile phones to estimate the number of available parking spaces in a street block. There are two challenges for this

[*] This research was supported, in part, by the U.S. Department of Transportation National University Rail Center (NURAIL), National Science Foundation grants IIS-1213013, CCF-1216096, DGE-0549489.

problem. First, PhonePark may produce false observations of parking activities due to GPS errors, transportation mode detection errors, and Bluetooth pairing errors. Second, not all drivers are equipped with PhonePark and therefore the observations of PhonePark represent only a portion of the full picture. In other words, the penetration ratio of the PhonePark enabled mobile phones may be low, so an estimate is needed for the determining parking availability on a street block.

In this paper, we introduce an algorithm to construct the *historical availability profile* (HAP) for a street block. This algorithm computes estimated values for the mean and the variance of parking availability for an arbitrary street block.

In addition to the HAP construction algorithm, we also introduce other algorithms that can estimate the real-time parking availability on a street block. These real-time estimation algorithms are called *parking availability estimation* (PAE) algorithms and they utilize different combinations of the historical parking availability and real-time observations to estimate the current street parking availability.

In summary, this paper makes the following scientific contributions: (1) Algorithms to detect the status of street parking slots. These include parking/deparking detection by transportation mode transitions, Bluetooth pairing, and parking pay-by-phone piggyback, (2) An algorithm that can construct the historical availability profile (i.e., mean and variance of parking) for a given street block, (3) Algorithms for estimating in real-time the current parking availability on a street block, (4) Experimental evaluation of the algorithms using actual real-time street parking data from SFPark.org, and (5) Theoretical underpinnings of the proposed techniques.

II. OVERVIEW OF PHONEPARK SYSTEM

The PhonePark software system consists of two components, namely *parking status detectors* (PSDs) and the *parking availability estimator* (PAE) (see Figure 1). A PSD runs on a mobile device. It detects when and where a driver parks/unparks and whether she pays for parking. When the PSD recognizes a parking or deparking activity, it submits a report to the PAE component indicating that a parking space is occupied or released[1]. The PAE component aggregates the reports of individual PSDs to estimate the number of available parking slots in each street block. For estimation, PAE considers the facts that not all drivers are equipped with the PSD component (i.e., the market penetration ratio of PhonePark is not 100%) and the mobile device can have false positive and false negative possibilities for parking detection. Furthermore, to compensate for the inaccuracy of PSD observations, it combines PSD observations with historical parking statistics to get the final estimate. Historical parking statistics is obtained from the proposed HAP construction algorithm while PSD observations are obtained from the drivers with the PSD devices.

At a high level, the PSD works as follows. It employs a transportation mode detection method to track the transportation mode of its user. It watches the transitions of transportation mode via GPS and/or accelerometer and infers parking status from a sequence of transitions. For example, if the transition sequence *car* → *stationary* → *walking* is observed, then the PSD infers that the driver parked at the stationary point. Similarly, if the transition sequence *walking* → *stationary* → *car* is observed, then the PSD infers that the driver unparked at the stationary point. Additionally, apart from the transportation mode transition module for parking detection, the PSD also considers Bluetooth. Bluetooth is more common than both GPS and accelerometer sensors on mobile phones. Thus, further considering Bluetooth as an option to GPS/accelerometer improves the market penetration ratio of the proposed technology. Further, the PSD can also consider pay-by-phone piggyback for parking detection [8]. The use of these PSD's is described in the next section.

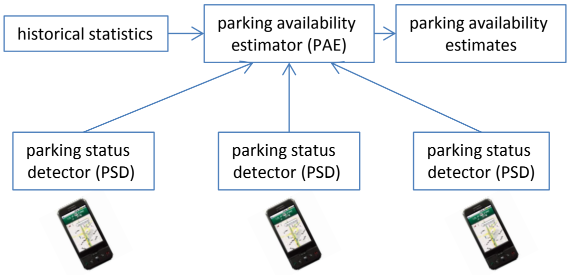
historical statistics
parking availability estimator (PAE)
parking availability estimates
parking status detector (PSD)
parking status detector (PSD)
parking status detector (PSD)

Figure 1. Components of the PhonePark system

An observation of the PSD may be incorrect due to many reasons. First of all, the classification of the transportation mode may be wrong due to sensor errors and ambiguity between different transportation modes. For example, the mobility pattern of walking is similar to that of driving in a heavy traffic situation. Second, even the transportation modes are correctly tracked, the inference from the transition sequence may be wrong. For example, the transition sequence *car* → *stationary* → *walking* may be the result of a passenger being dropped off (as opposed to a driver parking). Likewise, Bluetooth pairing or pay-by-phone piggyback can have errors.

All the errors of PSD observations propagate to the PAE component when PAE aggregates these observations to estimate the number of available parking spaces. Furthermore, when PAE scales PSD observations to compensate for the market penetration ratio, there are sampling errors. PAE takes all these errors into account when combining the real-time observations with historical statistics. Recall, historic statistics are computed by the *historical availability profile* (HAP) algorithm.

Apart from studying the effectiveness on real-time parking estimation of using the historical statistics and the scaled real-time observations solely, we also consider the combination of both the historic statistics and the real-time observation. The combination strategies are: (1) Using a fitted weight scheme and (2) Using an adaptive Kalman

[1] The submission of a parking/deparking activity can be anonymized and incentivized. In general, privacy and incentives are orthogonal issues that are not addressed in this paper.

Filter scheme. These four schemes are compared with the true real-time parking availability in the evaluation section.

For the fitted weight scheme, based on the market penetration ratio of the PhonePark enabled mobile phones, false positive, and false negative probabilities of the PhonePark mobile devices most effective weight can be determined beforehand. Using this predetermine weight, we now have the most effective combination of historic statistics and real-time observations.

For the Kalman Filter scheme, the combination is essentially a weighted average between the parking availability detected by PSDs and the historical statistics as obtained from HAP where the weights are proportional to the accuracies of the two availability sources. The variances represent the two accuracies. In this paper we use a Kalman filter, called *adaptive, limited memory filter* (ALMF) [6], to estimate the accuracy of the parking availability detected by PSDs. ALMF estimates the accuracy based on the gap between the historical statistics and the recent PSD observations. It uses this gap to capture the collective effect of all kinds of errors in the PhonePark system. Despite its simplicity, ALMF has been shown effective in situations where error statistics are unknown a priori, changing over time, or hard to analyze (see e.g., [6, 7]). Since errors are handled by PAE, the process of the PSD is deterministic.

The Kalman Filter based weighting between PSD-detected parking availability and historical statistics captures the following intuitions: (i) PSD-detected parking availability is more useful when the market penetration ratio is high and when parking availability is unpredictable from historical statistics; and (ii) historical statistics are more useful when PhonePark is inaccurate.

III. PARKING STATUS DETECTOR

We propose three methods for a PSD to detect parking status:

1. *Connection to in-vehicle Bluetooth*. This method utilizes the Bluetooth connection between the driver's mobile phone and the car to detect parking/deparking activities. Nowadays more and more cars have in-vehicle Bluetooth capability, which allows the owner of a car to register her mobile phone to the in-vehicle Bluetooth system. Once the mobile phone is registered, whenever the owner is inside the car and the engine is started, the mobile phone is automatically connected to the in-vehicle Bluetooth system. On the other hand, when the engine is stopped or the owner goes away from the car for a short distance (~10 meters), the Bluetooth connection between the mobile phone and the car is broken. Thus, if the Bluetooth connection between the driver's phone and the car is broken, then it can be inferred that the driver parks the car. If the Bluetooth connection is established, then it can be inferred that the driver deparks. This concept is illustrated by Figure 2. The Bluetooth ID of the car can be used by the PSD to distinguish between the Bluetooth connection with car and that with other devices such as a Bluetooth headset.

2. *Transportation mode monitoring*. This method employs a transportation mode detection algorithm to track the transportation mode of the driver. It monitors the transitions of transportation mode and infers parking status from a sequence of transitions. For example, if the transition sequence *driving* → *stationary* → *walking* is observed, then the PSD infers that the driver parked at the stationary point. Similarly, if the transition sequence *walking* → *stationary* → *driving* is observed, then the PSD infers that the driver deparked at the stationary point. This concept is illustrated by Figure 3. In our prior work we explored this method (see [8]). The reliability of this method is enhanced by detecting the activity of going to a nearby pay-box (as in Chicago Parking see [18]), paying, and returning to the car to place the ticket on the dashboard (see our prior work in this direction in [8]). Using only transportation mode transitions, we achieved over 85% accuracy for parking/deparking detection on the PSD. Combining this with Bluetooth and pay-by-phone piggyback can improve the accuracy further.

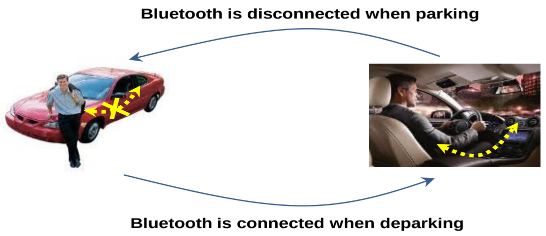
Bluetooth is disconnected when parking
Bluetooth is connected when deparking

Figure 2. Parking detection based on Bluetooth connection between mobile phone and car

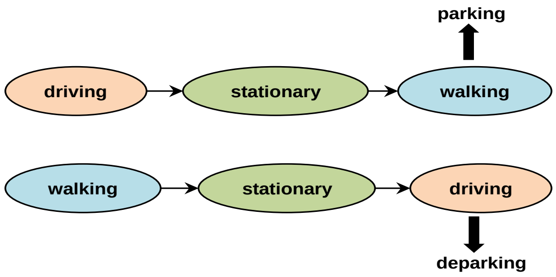

Figure 3. Parking detection based on transportation mode transitions

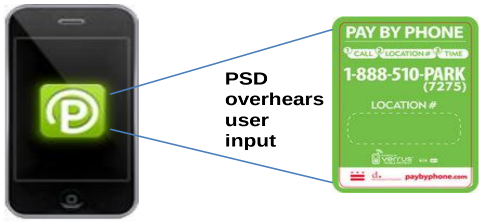

Figure 4. Parking detection using pay-by-phone piggyback

3. *Pay-by-phone Piggyback*. This method takes the advantage of the pay-by-phone parking payment service. This service allows a driver to pay for parking by entering to her mobile phone the parking slot number (posted on parking meters) and parking duration [4, 5]. The information is then submitted from the mobile phone to a payment service system. The pay-by-phone may help parking detection in the

following way. With the mobile phone user's agreement, the PSD (which resides on the same phone as the pay-by-phone application) monitors the user's usage of pay-by-phone service. When the user submits the payment, the PSD infers that the user has parked (see Figure 4). The PSD may possibly overhear the parking duration input and thus predict the time of deparking. The pay-by-phone service has been available in over 30 cities in US including San Francisco, Chicago, Washington DC, etc. (see [4, 5]).

IV. Constructing Historical Profiles

The historical profile of a street block contains the mean and variance of the block's parking availability as a function of time. In this section, we discuss how to construct a historical profile from historical PSD observations.

A. *Historical Availability Profile*

Periodically a street block *S* experiences a status such that all the parking slots in S are available. For example, there may be a limitation that parking is prohibited from 8am to 9am of each day, or there is street cleaning every Tuesday from 1pm to 3pm during which parking is prohibited. When parking is prohibited, all parking slots are available. The time period during which parking is permitted is referred to as a *permitted period.* For example, if parking is prohibited from 8am to 9am of each day, the permitted period is from 9am of a day to 8am of the next day.

Time is discrete and proceeds by an atomic unit (e.g., second). Define the *parking availability* of *S* to be the number of available parking slots in *S*. We assume that the time unit is small (e.g., second) such that the parking availability can only change at the beginning or end of a time unit but not during it. A *historical availability profile* (HAP) for *S* includes the average and variance of parking availability of *S* at each time point of the permitted period.

We distinguish between two types of PSD observation errors. One is called *false negative* which occurs when a parking or deparking activity is not detected by a PSD the owner of which performs the activity. Another type is called *false positive* which occurs when a PSD detects an activity that does not actually happen. The probability that a false negative error occurs is called the *false negative probability* and denoted by *fn*. *fn* is a fixed parameter. The probability that a false positive error occurs is called the *false positive probability* and denoted by *fp*. *fp* may vary depending on how the observation is generated. *fp* is attached to the PSD report.

B. *HAP Construction Algorithm*

Denote by *N* the total number of parking slots on the street block *S*. Denote by *b* the penetration ratio of PhonePark. The HAP construction algorithm starts with the beginning of a permitted period. At this time point the parking availability is equal to *N*. Then when a parking report is received, the parking availability is decreased by $\frac{1-fp}{b\cdot(1-fn)}$. This is called *scaled decrease*. The intuition behind scaled decrease is as follows. Out of all vehicles that park, only fraction *b* of them are equipped with PhonePark and out of these PhonePark equipped vehicles only fraction ($1-f_n$) detect the parking activity and send parking reports. Thus, each received parking report represents $\frac{1}{b\cdot(1-fn)}$ actual parking activities. On the other hand, due to false positive errors, only fraction $1-fp$ of received parking reports represent actual parking activities. Similarly, when a deparking report is received, the parking availability is increased by $\frac{1-fp}{b\cdot(1-fn)}$. This is called *scaled increase*. In this way we obtain an estimated function that describes how the parking availability changes over time for the considered permitted period. We repeat this procedure for a number of permitted periods and average among all the obtained functions. Specifically, denote by $\hat{a}_i(t)$ the parking availability estimated by scaled decrease or increase at time *t* for the *i*-th permitted period. Denote by $\hat{q}(t)$ the estimated average and by $\hat{Q}(t)$ the estimated variance of $\hat{a}_i(t)$, respectively.

$$\hat{q}(t)=\frac{\sum_{i=1}^{m}\hat{a}_i(t)}{m} \qquad (4.1)$$

$$\hat{Q}(t)=\frac{\sum_{i=1}^{m}(\hat{a}_i(t)-\hat{q}(t))^2}{m} \qquad (4.2)$$

where *m* is the number of permitted periods for which parking availability is collected. The HAP algorithm stops when a certain level of confidence for the accuracy of $\hat{q}(t)$ is reached, e.g., the difference between $\hat{q}(t)$ and the true average parking availability at the time *t* is smaller than 2 with 95% confidence. In the rest of this section we discuss how the confidence is computed.

1) *Computation of Confidence*

Let {PP$_1$, PP$_2$,…, PP$_m$} be a sequence of permitted periods for a block *S* for which parking availability has been collected. We make the following denotations:

$a_i(t)$: the true parking availability at the time *t* of the PP$_i$.

$P_i(t)$: the true number of parking activities starting from the beginning of PP$_i$ until time *t* of PP$_i$.

$D_i(t)$: the true number of deparking activities starting from the beginning of PP$_i$ until time *t* of PP$_i$.

Clearly, $a_i(t)=N-P_i(t)+D_i(t)$.

$p_i(t)$: the number of parking activities detected by PhonePark starting from the beginning of PP$_i$ until time *t* of PP$_i$.

$d_i(t)$: the number of deparking activities detected by PhonePark starting from the beginning of PP$_i$ until time *t* of PP$_i$.

Throughout the analysis we make the following three reasonable assumptions:

1. PhonePark-equipped vehicles are uniformly distributed among all vehicles, thus a parking (or deparking) activity is performed by a PhonePark-equipped vehicle with probability *b*.
2. Parking activities, if detected, are detected independently of each other. In other words, given any two parking activities A and B such that each is performed by a PhonePark-equipped vehicle, whether A is detected or not is independent of whether B is detected or not.
3. $a_1(t), a_2(t), ..., a_m(t)$ are independently and identically distributed. Particularly, they have a common mean which is denoted by *q*(*t*).

Furthermore, we consider the situation in which only the Bluetooth connectivity method is used for parking availability. In this situation, false positive errors are negligible. Thus, we assume that *fp*=0. Now we show that $\hat{a}_i(t)$ is an unbiased estimate of $a_i(t)$. We treat $\hat{a}_i(t)$ as a random variable here since it depends on the parking and deparking detections which involve random errors.

Lemma 1: The conditional expectation of $\hat{a}_i(t)$ given $a_i(t)$ is equal to $a_i(t)$, i.e., $E(\hat{a}_i(t) | a_i(t)) = a_i(t)$.

Proof: Observe that since each parking activity is detected by PhonePark with probability *b*·(1–*fn*) and parking activities are detected independently of each other, $p_i(t)$ follows a Binomial distribution with $n=P_i(t)$ and $p=b\cdot(1-fn)$. That is,

$$p_i(t)|P_i(t) \sim \text{Binomial}(P_i(t), b\cdot(1-fn)) \qquad (4.3)$$

Similarly,

$$d_i(t)|D_i(t) \sim \text{Binomial}(D_i(t), b\cdot(1-fn)) \qquad (4.4)$$

When *fp*=0, according to the HAP algorithm, it follows that

$$\hat{a}_i(t) | a_i(t) = N - (p_i(t) | P_i(t)) \cdot \frac{1}{b\cdot(1-fn)} + (d_i(t) | D_i(t)) \cdot \frac{1}{b\cdot(1-fn)} \qquad (4.5)$$

Thus,

$$E(\hat{a}_i(t) | a_i(t)) = N - E(p_i(t) | P_i(t)) \cdot \frac{1}{b\cdot(1-fn)} + E(d_i(t) | D_i(t)) \cdot \frac{1}{b\cdot(1-fn)} \qquad (4.6)$$

According to Eq. 4.3 and Eq. 4.4, it follows that

$$E(p_i(t) | P_i(t)) = P_i(t) \cdot b \cdot (1-fn) \qquad (4.7)$$

$$E(d_i(t) | D_i(t)) = D_i(t) \cdot b \cdot (1-fn) \qquad (4.8)$$

Thus,

$$E(\hat{a}_i(t) | a_i(t)) = N - P_i(t) + D_i(t) = a_i(t) \qquad (4.9)$$

Q.E.D.

Lemma 2. $E(\hat{a}_i(t)) = q(t)$ for *i*=1,2,…, *m*.

Proof:

$$E(\hat{a}_i(t)) = \sum_{k=0}^{N} (prob\{a_i(t)=k\} \cdot E(\hat{a}_i(t) | a_i(t)=k)$$

$$= \sum_{k=0}^{N} (prob\{a_i(t)=k\} \cdot k)$$

$$= E(a_i(t))$$

$$= q(t) \qquad (4.10)$$

Q.E.D.

Now we study the relationship between $\hat{q}(t)$ computed by Equation 4.1 and the true mean *q*(*t*). Since $a_1(t), a_2(t), ..., a_m(t)$ are independently and identically distributed, $\hat{a}_1(t), \hat{a}_2(t), ..., \hat{a}_m(t)$ are also independently and identically distributed. Recall that

$$\hat{q}(t) = \frac{\sum_{i=1}^{m} \hat{a}_i(t)}{m} \qquad (4.11)$$

According to the central limit theorem, $\hat{q}(t)$ approximately follows a normal distribution with mean equal to $E(\hat{a}_i(t))$ and variance equal to $\frac{\text{var}(\hat{a}_i(t))}{m}$. According to Lemma 2, $E(\hat{a}_i(t)) = q(t)$. $\text{var}(\hat{a}_i(t))$ is set to be $\hat{Q}(t)$ computed by Equation 4.2.

Now we can talk about confidence towards estimating *q*(*t*). That is, given an error tolerance *δ*, with what probability the difference between $\hat{q}(t)$ and *q*(*t*) is smaller than *δ*. Let $\lambda = \delta \cdot \sqrt{\frac{m}{\text{var}(\hat{a}_i(t))}}$. According to the property of normal distribution,

$$\text{Prob}\{|\hat{q}(t) - q(t)| < \lambda \cdot \sqrt{\frac{\text{var}(\hat{a}_i(t))}{m}}\}$$

$$= 2\cdot\Phi(\lambda) - 1$$

$$= 2\cdot\Phi(\delta \cdot \sqrt{\frac{m}{\text{var}(\hat{a}_i(t))}}) - 1$$

$$= 2\cdot\Phi(\delta \cdot \sqrt{\frac{m}{\hat{Q}(t)}}) - 1 \qquad (4.12)$$

Equation 4.12 quantifies the following intuitions: i) the more samples are collected (i.e., the bigger *m*), the higher confidence; ii) the lower requirement on reliability (i.e., the bigger *δ*, the higher the confidence; iii) the higher fluctuation of the estimated availability (i.e., the bigger $\hat{Q}(t)$), the lower the confidence.

Example: Suppose that the standard deviation of $\hat{a}_i(8:00am)$ is 10 (i.e., the average fluctuation of *estimated* availability at the 8:00am is 10). Now if we want the error of the estimated mean $\hat{q}(t)$ and the true mean *q*(t) to be sm*a*ller than *2* with 90% confidence, then we need *to* collect 68 permitted periods. This amounts to about two months if each day is a permitted period.

V. Parking Availability Estimation Algorithms

In this section we present four algorithms for estimating parking availability in real-time. Denote by *x*(*t*) a random variable representing the true parking availability in *S* at time

t of a permitted period. Denote by $\hat{x}(t)$ the parking availability estimated for time t of the same permitted period.

Historical Statistics (HS). In HS, the real-time parking availability at time t of a permitted period is taken to be the estimated historical mean $q(t)$, i.e.,

$$\hat{x}(t) = \hat{q}(t)$$

Scaled PhonePark (SPP). In SPP, PSD-observed parking availability is scaled to the entire set of parking spaces in a street block based on the penetration ratio and detection accuracy. Specifically, in SPP, the estimation of parking availability is initialized to be N at the beginning of a permitted period. Then when a parking report is received, the parking availability is decreased by $\frac{1-fp}{b\cdot(1-fn)}$ and set to 0 if it becomes negative after the decrease; when a deparking report is received, the parking availability is increased by $\frac{1-fp}{b\cdot(1-fn)}$ and set to N if it exceeds N.

Weighted Average (WA). This algorithm computes a weighted average between the historical mean and the scaled PSD-observed parking availability. Specifically, let $\hat{a}(t)$ be the parking availability estimated by SPP. Then

$$\hat{x}(t) = w_{HS}\hat{q}(t) + (1-w_{HS})\hat{a}(t) \qquad (5.1)$$

The weight w_{HS} is determined by experiments as will be discussed in section VI.C.

Kalman Filter (KF). We make the following two hypotheses:

Hypothesis 1. The true parking availability is equal to the historical mean plus a Gaussian noise with mean zero and variance equal to the historical variance, i.e.,

$$x(t) = q(t) + w(t) \qquad (5.2)$$

where w(t) is a random variable normally distributed with mean zero and variance Q(t). w(t) is called the state noise.

Hypothesis 2. The scaled PSD-observed parking availability (i.e., the availability estimated by SPP) is equal to the true parking availability plus a Gaussian noise, i.e.,

$$\hat{a}(t) = x(t) + v(t) \qquad (5.3)$$

where $v(t)$ is a random variable normally distributed with mean zero and variance $R(t)$; $R(t)$ are to be estimated. The mean of v(t) is zero because the expectation of $\hat{a}(t)$ is equal to x(t) according to Lemma 1. $v(t)$ is called the observation noise.

Using Kalman filter, and replacing q(t) by $\hat{q}(t)$, $x(t)$ is estimated as follows:

observed residual: $$y(t) = \hat{a}(t) - \hat{q}(t) \qquad (5.4)$$

Kalman gain: $$K(t) = \frac{Q(t)}{Q(t)+R(t)} \qquad (5.5)$$

state estimate: $$\hat{x}(t) = \hat{q}(t) + K(t)\cdot y(t) \qquad (5.6)$$

Incorporating (5.4) and (5.5), Eq. (5.6) can be rewritten as

$$\hat{x}(t) = \frac{R(t)}{Q(t)+R(t)}\cdot\hat{q}(t) + \frac{Q(t)}{Q(t)+R(t)}\cdot\hat{a}(t) \qquad (5.7)$$

From Eq. 5.7 it can be seen that the estimated parking availability is a weighted average between the PSD-observed parking availability (i.e., $\hat{a}(t)$) and the historical mean (i.e., $\hat{q}(t)$). The weights are inversely proportional to the expected errors. Specifically, Eq. 5.7 has the following properties:

1. The bigger the historical variance (i.e., Q(t)), the heavier is a(t) weighted towards the final estimate. This property quantifies the intuition that PhonePark is more useful when the true availability fluctuates in a wide range from the historical mean. In this case the parking availability is highly unpredictable and thus real-time observations are more important.

2. The bigger the value of $R(t)$, the heavier $\hat{q}(t)$ is weighted. This property quantifies the intuition that the historical statistics are more useful when real-time observations are inaccurate.

The observation noise parameter R(t) is estimated by the method introduced in [6] called ALMF (adaptive, limited memory filter).

VI. Evaluation by Simulations

A. Simulation Method

Our simulations used real world parking availability data retrieved from SFPark.org. SFPark.org provides APIs for access to real-time parking availability for street blocks in the city of San Francisco. Using these APIs we built a program that retrieves parking availability of street blocks every second. The simulations use SFPark data collected in a four month period from April 10 to August 11, 2012.

The experiments were conducted on two street blocks in San Francisco, namely 300-398 Polk Street and 2050-2098 Chestnut Street (see Figure 5). The Polk block has 12 parking slots in total and 4.6 available slots on average. The Chestnut block has 4 parking slots in total and 1.5 available slots on average.

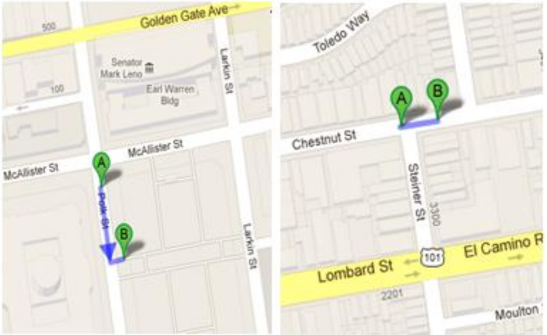

Figure 5. Street blocks considered for evaluation

For a considered street block, we derive a sequence of PSD reports from its SFPark data, using the following procedure. We sequentially visit each change of parking availability recorded in the SFPark data. Let t_{last} be the time of the last visited change and t_{cur} be the time of the currently visited change. If the change indicates that some parking slots have been released since t_{last}, then each of these parking slots is released by a PhonePark driver with probability b. Furthermore, a release executed by a PhonePark driver is detected by PhonePark with probability 1–fn. Thus, each release since t_{last} is detected by PhonePark with probability b·(1–fn). Thus, for each release since t_{last}, a deparking report is generated with probability b·(1–fn), at time t_{cur}. Let r be the number of generated reports for the currently visited change. In addition, we generate a random number of extra deparking reports which simulate false positive errors. The random number follows a negative Binomial distribution NB(r, fp)[2]. The justification is that, by its definition, NB(r, fp) represents the number of false positive reports that are supposed to be generated until r correct reports are generated, given that the probability for a report to be false positive is fp. A similar procedure follows for the case that the change at t_{cur} indicates that some parking slots have been occupied since t_{last}. Now we are ready to evaluate various parking availability estimation algorithms. We use 10-fold cross validation. For this we split the SFPark data of a street block into pieces such that each piece covers exactly one permitted period. These pieces are then used for sampling of cross validation. All the system parameters and their values are shown in Table I.

For the evaluation of the HAP algorithm the performance measure is the root mean square error (RMSE) between the estimated mean computed by HAP and the true mean. For the evaluation of the PAE algorithms, we consider two performance measures:

1. The *RMSE* between the estimated average availability and the true average availability;

2. The *boolean availability accuracy* which is defined as follows. We say that a PAE algorithm is *boolean correct* at time t if (i) the algorithm indicates that there is at least one slot available and there indeed is, or (ii) the algorithm indicates that there is not any slot available and there indeed is not. The boolean availability accuracy is defined to be the percentage of the number of time units at which a PAE algorithm is boolean correct.

TABLE I. SYSTEM PARAMETERS AND THEIR VALUES.

Parameter	Symbol	Value
Penetration ratio	b	1%, 50%
False negative probability	fn	0, 0.05, 0.1, 0.15, 0.2, 0.25
False positive probability	fp	0, 0.05, 0.1, 0.15, 0.2, 0.25
Length of collection period		4 months
Length of permitted period		1 day
Length of time unit		minute

B. *Evaluation of HAP*

In this subsection, we evaluate the performance of HAP on computing historic parking availability profiles for a street block. Each experiment is conducted as follows. For each permitted period, both the historic mean and HAP's estimation is recorded. For each time unit within the permitted period, the mean of both HAP's estimation and the historic mean are computed across the permitted periods. The RMSE is then computed across all the time units. Each experiment is repeated several times and the average RMSE taken.

Figures 6-9 show the RMSE of HAP From Figure 6 it can be seen that for the Polk block, the RMSE of HAP is below 2.3 when the penetration ratio is 1%. On the other hand, the average true availability is 4.6. A higher penetration ratio or a longer collection of historical observations can increase accuracy. For example, Figure 7 shows that when the penetration ratio increases to 50%, the RMSE of HAP is below 0.72, which is 16% of the average true availability of the Polk block. For the Chestnut block, the RMSE is below 0.57 when the penetration ratio is 1% and below 0.47 when the penetration ratio is 50%, while the average true availability is 1.5 (see Figures 8 and 9). Also, observe that as the false positive or false negative probability increases, the algorithm generally becomes less effective.

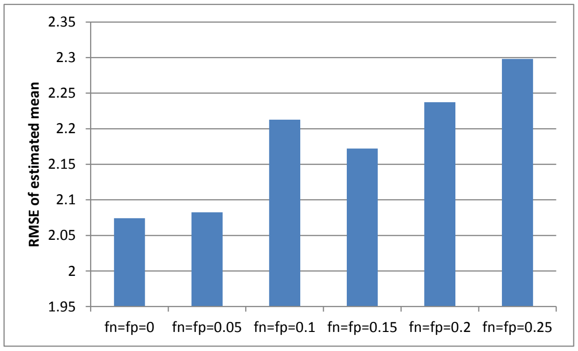

Figure 6. RMSE of HAP, b=1%, Polk St.

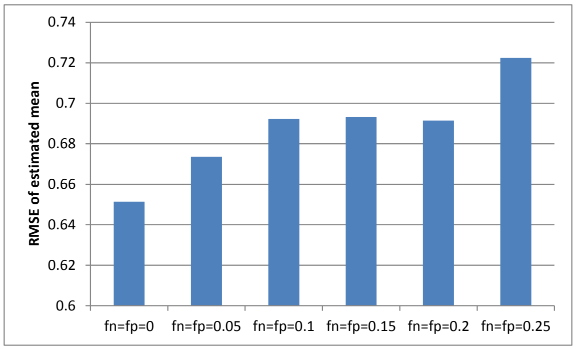

Figure 7. RMSE of HAP, b=50%, Polk St.

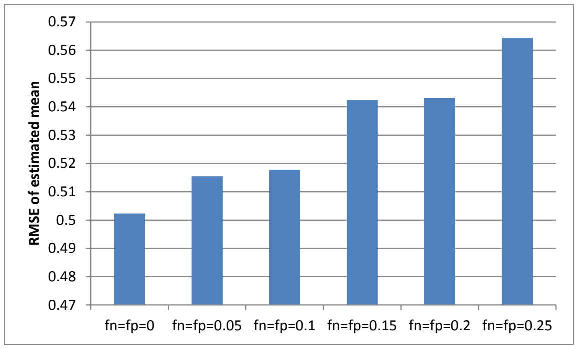

Figure 8. RMSE of HAP, b=1%, Chestnut St.

[2] In simulations fp is fixed for each experiment.

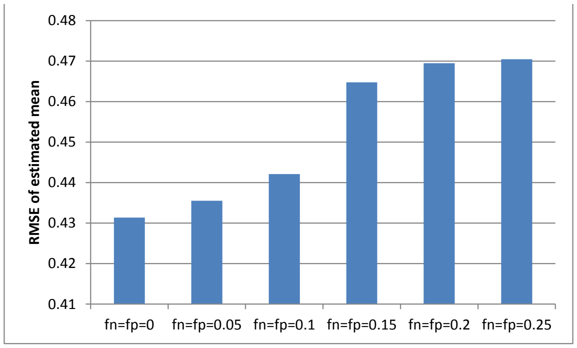

Figure 9. RMSE of HAP, b=50%, Chestnut St.

C. *Tuning-up Weighted-Average (WA)*

We conduct experiments to determine the optimal weight w_{HS} for the WA method. We vary w_{HS} between 0 and 1 with a step size of 0.1. Figure 10 shows that different weights produce . HAP's estimation of $q(t)$ is done using the same penetration ratio as in computing $\hat{a}(t)$ for all the configurations in Figure 10. For example, if 5% penetration ratio is considered for estimating $q(t)$ by HAP, then 5% is also utilized for $\hat{a}(t)$ estimation.

From Figure 10, it can be seen that the optimal weight, i.e., the weight that leads to minimum RMSE, depends on parameters including the penetration ratio *b*, the false positive probability *fp*, the false negative probability *fn*. For example, the optimal weight is 0 when *b*=1%, *fn*=*fp*=0.1 for the Chestnut block whereas it is 0.6 when *b*=50%, *fn*=*fn*=0.25. Figure 10 demonstrates that the optimal weight for the WA method can be calibrated.

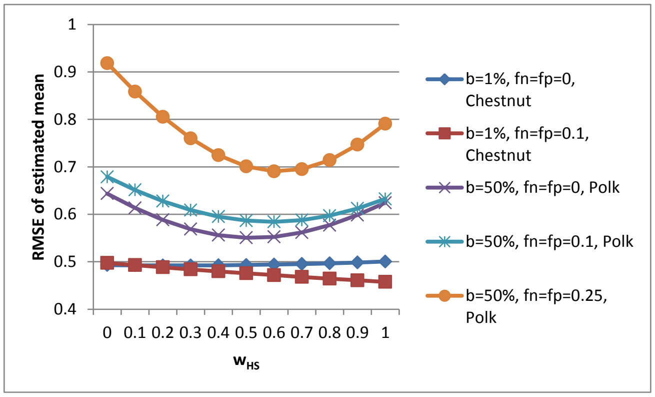

Figure 10. Determining weight (w_{HS}) for WA

D. *Comparison of HS, SPP, WA, and KF*

In this section, we present evaluation results for the four algorithms for estimating street parking availability in real time. The four parking availability estimation algorithms are (1) Scaled PhonePark (SPP), (2) weighted average (WA), historical statistics (HS), and (4) Kalman Filter (KF). For the WA approach, the optimal weight is determined as discussed in section VI.C.

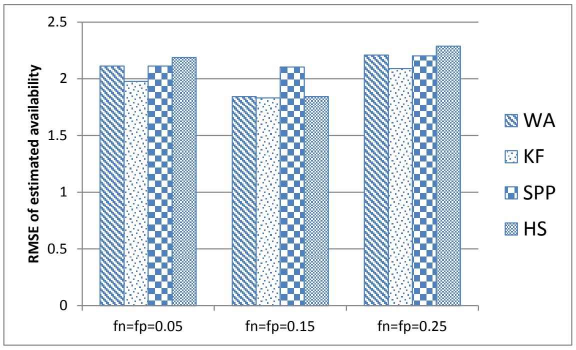

Figure 11. RMSE of PAE algorithms, b=1%, Polk St.

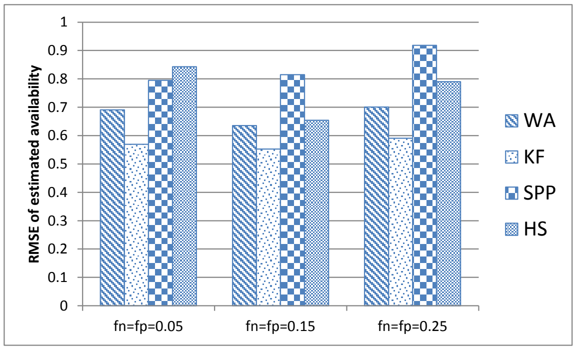

Figure 12. RMSE of PAE algorithms, b=50%, Polk St.

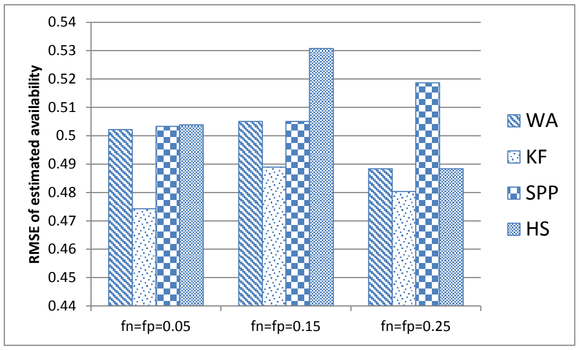

Figure 13. RMSE of PAE algorithms, b=1%, Chestnut St.

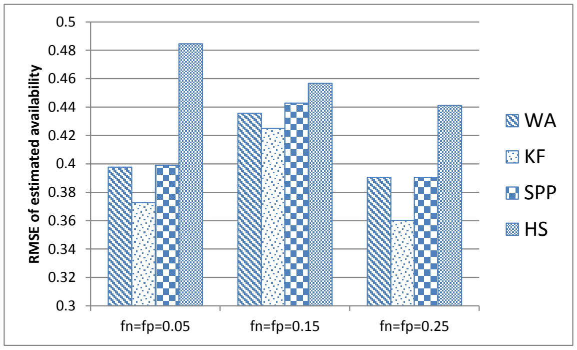

Figure 14. RMSE of PAE algorithms, b=50%, Chestnut St.

Comparison on RMSE. Figures 11-14 show the RMSE performance of the PAE algorithms for different penetration ratios and different street blocks. From the figures it can be seen that in most cases KF has the lowest RMSE among all the algorithms. Furthermore, for all PAE algorithms, the RMSE decreases as the penetration ratio increases. Particularly, when the penetration ratio is 1%, the RMSE of KF for Polk St. is about 2 (see Figure 11), which is 40% of the average parking availability of the street block. When the penetration ratio increases to 50%, the RMSE of KF reduces to about 0.6 (see Figure 12), which is 13% of the average parking availability. For the Chestnut St., the RMSE of KF is 30% of the average parking availability when the penetration ratio is 1% (see Figure 13).

In most cases, WA is second to KF in terms of the RMSE performance. This indicates that dynamic weighting between PSD-observed parking availability and the historical mean, as done in KF, works better than static weighting as done in WA. In addition, the accuracy of WA is higher than that of SPP and HS. This shows the benefit tuning up w_{HS} for WA.

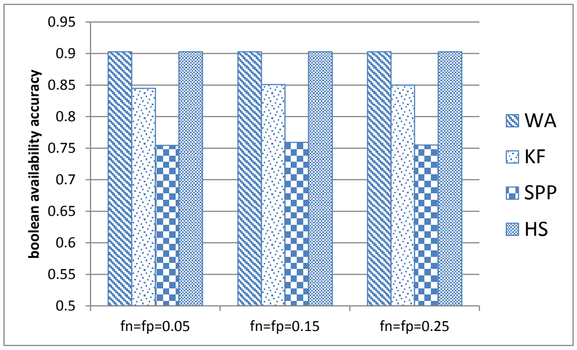

Figure 15. Boolean availability accuracy, b=1%, Polk St.

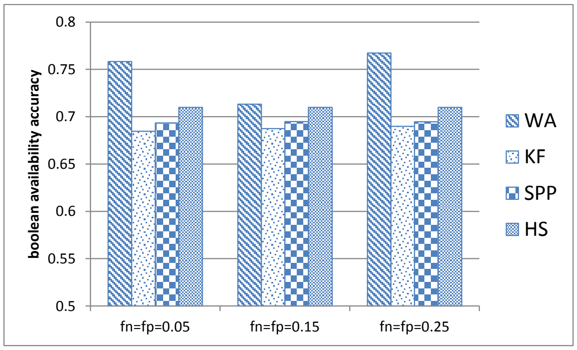

Figure 16. Boolean availability accuracy, b=1%, Chestnut St.

Comparison on boolean availability accuracy. Figures 14-16 show the boolean availability accuracy of the algorithms. From the figures it can be seen that WA has the highest accuracy among all the algorithms. For the Polk St., the accuracy of WA reaches 90% even when the penetration ratio is only 1% (see Figure 15). For the Chestnut St., the accuracy of WA reaches 70-75% when the penetration ratio is 1% (see Figure 16). For both blocks, HS is second to WA. Notice that even though KF is better than WA on RMSE, it is not effective as WA on boolean availability accuracy.

VII. RELATED WORK

A number of approaches have been considered for monitoring parking spaces. Some approaches monitor parking spaces by color histogram classification and car feature point detection from still images captured by externally mounted cameras [9, 10]. Our solution only needs sensors on a mobile phone and not external cameras.

Parking garages use in-out counters at exit and entry points to compute the number of additional vehicles they can accommodate [10]. There exist other systems that allow travelers to access parking information and make prior reservations for parking in areas such as airports and rail stations [12, 13]. PhonePark is different and focuses on street parking.

The GPS sensor has been used in the past for parking guidance. For example, GPS guided parking is used in [14]. PhonePark solves a different problem, namely automatically detecting parking and deparking activities. The available parking spaces detected by PhonePark may serve as an input to the parking guidance system.

The availability of the vacant parking spaces can be calculated by external sensors such as those installed in the parking areas, which count the number of cars which enter and exit from the parking space [2]. For example, in San Francisco, several parking projects have been initiated [2, 15] that utilize externally implanted sensors. However, this strategy is expensive to deploy and maintain. For example, in one project that covers 8000 parking spaces the cost was over USD $23 million [1, 2]. The proposed work in this paper is not dependent on external sensors.

Ultrasonic sensors at the top of each parking space or on vehicle side-doors [1] can be used to sense the availability or unavailability of each parking space. Our solution uses only sensors available on mobile phone to infer when and where a traveler had parked their car. Sensors implanted under road surfaces or attached to the car side-doors are expensive to deploy and maintain (e.g., [2] cost USD $500 per system for each parking space, and [1] cost USD $400 per system for each car). These sensors may underperform in extreme weather. For example, in heavy snow these sensors may be covered. Using mobile phones is cheaper, more convenient, and more flexible.

The works [11, 16, 17] are based on parking but the focus is on how to choose parking slots ideally in competitive parking settings. Thus, in these other works routing to the most optimal parking slot is the focus. This paper is different and focuses on estimating the current street parking availability on a street block in real time.

The work in [19, 22] analyze the capacity of an opportunistic system to assist the search for parking space. Vehicles searching for parking space are equipped with sensors that allow them to sense the location and status of parking spots as they drive across the city. This information is subsequently shared upon encounters with other vehicles or submitted to a central server. The proposed work in this paper is different and focuses on detecting and estimating the availability of street parking spaces. On the other hand, [19, 22] focus on optimizing the effectiveness of the parking search process through user oriented performance metrics, such as the parking search time, route length, and the proximity of the found or assigned parking spot to the user travel destination.

Other areas of parking research includes pursuing parking spaces and responding to pricing policies about public and private parking facilities that is studied in [20, 21]. The drivers are modeled as strategic agents who make rational decisions while minimizing the cost of the acquiring parking spots, under deterministic or probabilistic information for the overall parking demand.

Google's approach to parking status detection is called OpenSpot [23, 24] and relies on drivers to manually report an empty parking spot when they depark or see one available. This system is cumbersome and drivers don't feel comfortable to use their mobile phones for manual reporting while driving. The proposed approach in this paper automates parking status detection and manual entry of available street parking spaces is not required.

VIII. Conclusions

In this paper, parking status detection, historical parking profile construction, and parking estimation algorithms are proposed. The algorithms are validated using real-time and real world parking availability data. For parking status detection, we propose a cost effective solution that utilizes mobile phone sensors such as GPS, accelerometer, and Bluetooth sensors. Furthermore, the parking status detection algorithm may piggy back on pay-by-phone for parking transactions.

Given the fact that not all drivers carry a mobile phone and the fact that not all drivers with a mobile phone have the PhonePark system installed, the penetration ratio of the PhonePark system may not be 100%. Additionally, PhonePark may produce false observations of parking activities due to GPS errors, transportation mode detection errors, and Bluetooth pairing errors. The parking availability estimation algorithms take these limitations into consideration and then provide a final estimate on the availability of street parking spaces.

Specifically, parking availability estimation (PAE) algorithms estimate the street parking availability in real-time by combining historical parking statistics with real-time parking observations. Several approaches on combining historical parking statistics with real-time parking observations are considered. Experimental and theoretical analysis demonstrate the effectiveness of the parking availability estimation algorithms. Specifically, for penetration ratios of 1%, the average error of street parking availability estimation is below 2. More studies show that the combination of historical parking statistics and real-time parking observations as done in weighted average (WA) and Kalman Filter (KF) is more effective than using solely historical parking statistics (HS) or solely real time observations (SPP).

In general, the proposed approach is economical, convenient, and flexible. It produces good estimation in real-time on the availability of street parking spaces. The estimation of parking availability has potential to improve drivers' parking experience by saving parking search time, reducing gas consumption and CO_2 emission.

IX. REFERENCES

[1] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. ParkNet: drive-by sensing of road-side parking statistics. *ACM MobiSys*, 2010.

[2] San Francisco Park website http://sfpark.org/

[3] L. Stenneth, O. Wolfson, P. Yu, and B. Xu. Transportation mode detection from mobile phones and GIS Information. *ACM SIGSPATIAL GIS*, 2011.

[4] Parkmobile's website. http://us.parkmobile.com/

[5] Parkmobile's Pay by Phone website. http://us.parkmobile.com/en/everyone/how-it-works/pay-by-phone-parking

[6] K.A. Myers and B.D. Tapley. Adaptive Sequential Estimation with Unknown Noise Statistics. *IEEE Transactions on Automatic Control*, Vol. 21, No. 4, pp.520-523, 1976.

[7] L. Chu, S. Oh, and W. Recker. Adaptive Kalman filter based freeway travel time estimation. *TRB Annual Meeting*, Jan., 2005.

[8] L. Stenneth, O. Wolfson, B. Xu, P. S. Yu. Street parking using mobile phones. *IEEE MDM*, 2012.

[9] S. Lin , Y. Chen, and S. Liu. A Vision-Based Parking Lot Management System. *IEEE International Conference on Systems, Man and Cybernetics*, 2006.

[10] H. Al-Absi, J. Devaraj, P. Sebastian, and Y.V. Voon .Vision-based automated parking system. *ISSPA*, 2010.

[11] D. Ayala, O. Wolfson, B. Xu, B. DasGupta and J. Lin. Stability of Marriage and Vehicular Parking. *2nd International Workshop on Matching Under Preferences*, 2012.

[12] Smart-parking at rockridge BART station, http://www.path.berkeley.edu/path/research/featured/120804/smart-park.html.

[13] About Airport Parking website, http://www.aboutairportparking.com/

[14] H. Kurogo, K. Takada, and H. Akiyama. Concept of a parking guidance system and its effects in the Shinjuku area-configuration, performance and future improvement of system. *Proceedings of Vehicle Navigation and Information Systems Conference*, 1995.

[15] Parking Carma website, http://www.parkingcarma.com.

[16] D. Ayala, O. Wolfson, B. Xu, B. DasGupta and J. Lin. Parking in Competitive Settings: A Gravitational Approach. *IEEE MDM*, 2012.

[17] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta and J. Lin. Parking Slot Assignment Games. *19th ACM SIGSPATIAL GIS*, 2011.

[18] Chicago Parking Meters http://chicagometers.com/find-parking.aspx

[19] E. Kokolaki, M. Karaliopoulos, I. Stavrakakis. Opportunistically-assisted parking service discovery: now it helps, now it does not. *Pervasive and Mobile Computing*, Elsevier, vol. 8(2), 2012.

[20] E. Kokolaki, M. Karaliopoulos, I. Stavrakakis. Trading public parking space. *Technical Report*. Department of Informatics and Telecommunications. University of Athens, 2012.

[21] E. Kokolaki, M. Karaliopoulos, I. Stavrakakis. Leveraging information in vehicular parking games. *Technical Report*, Department of Informatics and Telecommunications. University of Athens, 2012.

[22] E. Kokolaki, M. Karaliopoulos, I. Stavrakakis. Value of information exposed: wireless networking solutions to the parking search problem. *IFIP/IEEE WONS*, 2011.

[23] Google Open Spot Parking http://techcrunch.com/2010/07/09/google-parking-open-spot/

[24] Google's Open Spot for street parking availability. http://www.androidauthority.com/google-labs-open-spot-a-useful-application-that-no-one-uses-15186/