ParkSense: A Smartphone Based Sensing System For On-Street Parking

Sarfraz Nawaz, Christos Efstratiou, Cecilia Mascolo
Computer Laboratory, University of Cambridge, UK
{firstname.lastname}@cl.cam.ac.uk

ABSTRACT

Studies of automotive traffic have shown that on average 30% of traffic in congested urban areas is due to cruising drivers looking for parking. While we have witnessed a push towards sensing technologies to monitor real-time parking availability, instrumenting on-street parking throughout a city is a considerable investment.

In this paper, we present ParkSense, a smartphone based sensing system that detects if a driver has vacated a parking spot. ParkSense leverages the ubiquitous Wi-Fi beacons in urban areas for sensing unparking events. It utilizes a robust Wi-Fi signature matching approach to detect driver's return to the parked vehicle. Moreover, it uses a novel approach based on the rate of change of Wi-Fi beacons to sense if the user has started driving. We show that the rate of change of the observed beacons is highly correlated with actual user speed and is a good indicator of whether a user is in a vehicle. Through empirical evaluation, we demonstrate that our approach has a significantly smaller energy footprint than traditional location sensors like GPS and Wi-Fi based positioning while still maintaining sufficient accuracy.

Categories and Subject Descriptors

C.3 [**Special-Purpose And Application-Based Systems**]

General Terms

Design; Experimentation; Measurement

Keywords

Smartphone Sensing; On-street Parking; Wi-Fi Fingerprinting

1. INTRODUCTION

In metropolitan city centers and dense urban areas around the world, parking space is an expensive and often scarce resource. This scarcity leads to high demand for parking space and during busy periods of the day, it is common for drivers to spend a significant amount of time searching for available spots. While off-street parking garages are usually equipped with necessary infrastructure to estimate the number of unoccupied spaces, the situation with on-street parking is less favorable. This lack of information on the availability of on-street spaces is usually frustrating for users and leads to cruising drivers looking for vacant spots.

A study [21] of major downtown areas has shown that 30% of the traffic in these congested areas is in fact due to cruising vehicles. This situation can be alleviated by providing real-time occupancy information to cruising drivers. In [19] the authors show, through a simulated model of an urban city center, that when real-time occupancy of on-street parking is disseminated to drivers, it results in a reduction of 15% in total travel time. In a survey reported in [18], it is shown that 30% of the 483 surveyed drivers changed their intended parking destination in response to road-side guidance signs indicating space availability at car parks. It is reasonable to expect similar behaviour change when users are presented with on-street parking availability information. If this information is accessed before the start of the journey, it can also potentially result in a change of transport mode, for example, a user might decide not to drive if there are no spaces available in the vicinity of his/her intended destination.

A number of on-going research efforts are aiming at systems that can offer real-time parking availability estimations. These efforts rely on specialized infrastructure, either embedded in roads [4] or on vehicles [17] to capture real-time space occupancy. However, there are significant costs involved in instrumenting large cities. On the other hand, smartphone applications such as ParkMobile[1] and PaybyPhone[2] that allow users to view where parking spaces are located and also pay for these, have seen a wider adoption. Considering the ease of deployment and maintenance costs, it is reasonable to assume that soon the majority of urban parking will be managed and paid for through smartphones.

Motivated by this trend, we develop ParkSense, a smartphone based sensing system that can detect if a driver has vacated a previously occupied parking space. The primary design goal of ParkSense is to minimize its impact on the battery life of the user's device while still maintaining an accuracy level that could allow estimations on parking availability. The requirement for low energy prevents us from using traditional location sensors available on smartphones which typically impose a significant energy cost. ParkSense,

MobiCom'13, September 30–October 4, Miami, FL, USA.

http://dx.doi.org/10.1145/2500423.2500438.

[1] `http://www.parkmobile.co.uk/`
[2] `http://paybyphone.co.uk/`

therefore, uses the wireless interface to sense user mobility. It takes advantage of the widely deployed Wi-Fi access points in urban environments that periodically transmit their SSIDs as beacons. ParkSense uses robust Wi-Fi signature matching based on the beacon reception ratio to detect a user's return to the parked vehicle. We show that, in our outdoor scenario, this beacon reception ratio based signature matching performs significantly better than signal strength based functions used by previous location tracking approaches [14, 15]. ParkSense also utilizes a novel approach to detect if the user has started driving by observing the rate of change of visible access points. The primary contributions of this work are as follows:

- We develop a smartphone based sensing system for on-street parking space occupancy that can be readily integrated into systems in use today.
- We show through micro-benchmark energy measurements and experiments from actual devices that our approach is much more energy efficient as compared to traditional location sensors. Our approach consumes less than 30% of the energy of possible alternatives based on traditional location sensors on smartphones.
- We empirically demonstrate through a four month user trial, the feasibility of using smartphone based sensing to detect unparking events in order to estimate parking availability.

The rest of this paper is organized as follows. In the following section, we discuss the motivation and the design guidelines that led us to ParkSense in more detail followed by the analysis of energy consumption of traditional location sensors available on current smartphones. We then present our approach and the results from our four month long user study. We conclude with a discussion of some of the limitations of our approach and how these can be mitigated.

2. MOTIVATION

The majority of on-street parking spaces are managed through legacy parking meters where the driver has to pay a fixed amount for a predefined duration, or pre-pay for the expected time that they intend to leave their vehicle. The side-effect of this model is that the majority of the drivers tend to over-pay for their parking time to avoid being penalized with large fines for overstaying. This makes information collected through such systems unreliable for estimating parking availability. Furthermore, as the majority of the legacy parking meters are offline, there are no mechanisms to capture accurate real-time availability of parking spaces which can be very precious to route drivers to appropriate parking areas. While the situation is rapidly changing with electronic and smartphone based payments for parking, the legacy model of pre-payment is still prevalent.

The proliferation of mobile phone based parking payment applications offers a unique opportunity to capture parking availability information through the users' mobile devices. In the central London borough of Westminster, over $9,000$ on-street parking spaces managed by the local city council, accept only electronic payments for parking, with a smartphone application as the predominant means of payment. This smartphone based system is being gradually rolled out to other parts of London and also in other UK cities. In this work, we explore the feasibility of augmenting such parking payment mobile applications with sensing capabilities that would allow the accurate detection of available parking space for on-street parking. Through a series of consultations with domain experts and parking management companies, we identified that the primary objective is to design a mobile phone sensing service capable of detecting when a driver vacates a parking spot, with minimal impact on the driver's mobile device. Indeed, considering that phone-based parking payment applications can accurately capture the time when a driver occupies a space, the main challenge is to discover when that space is vacated — in most cases long before the payment has expired.

We established a close collaboration with a major company offering phone-based parking payment systems in the UK in order to explore the requirements for a parking availability estimation system. During this process we discussed the possiblity of relying on user feedback about parking duration, expecting users to report through their mobile phone when they unpark their car. However, such solution was considered unreliable. Previous attempts to rely on user feedback about parking availability, such as Google OpenSpot [1] have failed to gain traction, with users often forgetting to report when they depart from a parking spot, thus not allowing a reliable estimation of city-wide parking availability. The possiblity of using financial rewards to incentivise unparking reporting was dismissed as well. In most on-street parking areas in the UK, there are a number of stakeholders involved in managing them. Parking prices are planned and managed by local city authorities, often based on a pricing strategy that is intented to incentivize drivers to avoid certain locations at different times of the day. Payment technologies, such as smartphone based payments, are typically outsourced to specialised companies. For such companies to accept the incorporation of a parking availability detection system on their mobile applications, it was important to avoid any interference with the pricing scheme managed by the local authorities. Offering financial rewards to drivers related to their parking behaviour was considered such interference. Furthermore, domain experts raised concerns that linking the use of such application with financial rewards would carry the risk of motivating users to try to deceive the system in order to maximise their fiancial gains. Based on these observations we, therefore, aimed for a system that can detect unparking without user intervention.

Unparking detection through mobile phone sensing can have a significant impact on the battery life of users' mobile device. Therefore, for any such system to be accepted, the primary concern is to minimize the battery consumption required for detecting an unparking event. On the other hand, using such information to estimate parking availability means that the accuracy requirements can be relatively low. The SFpark [4] system, an infrastructure based parking detection system deployed in the city of San Francisco, considers 70% detection accuracy as acceptable [3] for estimating parking occupancy. This assessment is justified by the fact that a parking availability information system estimates availability based on aggregated data from multiple users in a given area. Inaccuracies in the sensing data are therefore less important as long as the number of participating users is high. Based on these observations, we attempt to design a detection mechanism with minimal energy requirements for the user's device (in order to facilitate wider adoption), while relaxing the need for high accuracy that perhaps could be achieved through more expensive sensing.

The key requirements for the design of such system are: (i)

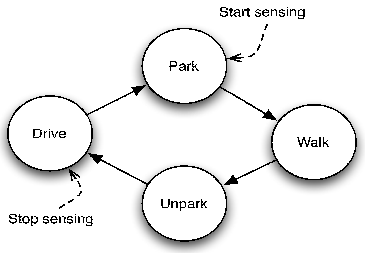


Figure 1: In a *pay-by-phone* parking application, sensing is initiated when the driver has parked and paid. Detection of unparking requires tracking the driver until they return to their vehicle and drive away.

considering that parking is initiated by payment, the system should be able to detect the time at which the driver returns to the vehicle and unparks, (ii) the system should be able to detect unpacking incidents without user intervention with minimal energy overhead, (iii) the system should not collect user location information beyond the parking location.

3. ALTERNATIVE SENSORS

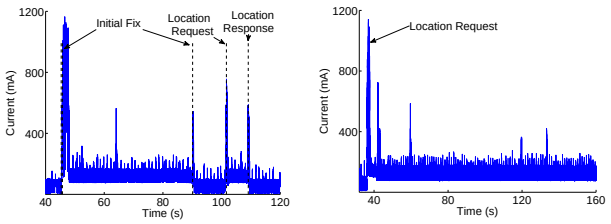
Using mobile phone sensing to detect when a driver vacates a parking space can be considered a special case of an activity detection problem. It can be defined as a sequence of specific actions that are expected by the driver: (i) initiate a parking incident through mobile phone payment, (ii) walk away from the vehicle, (iii) return to the vehicle, (iv) drive away (Figure 1). This sequence of actions is primarily related to the relative location of the driver and the vehicle. A typical approach for detecting these actions could be to rely on localization technologies readily available on current smartphones. In this section, we explore the feasibility of detecting these actions using location sensors.

3.1 GPS Based Location

GPS based location tracking could be considered a natural choice for our application. Continuous tracking of users, after they have parked their vehicles, could allow us to determine when they return to their vehicles. Furthermore, estimation of the user's speed from the GPS sensor can be used to discover if the user is driving again after returning to the parked vehicle. However, there are several issues with this approach as we further discuss.

It is well known that GPS receivers require a significant amount of time to acquire an initial fix and to produce a location estimate after being switched on. The time required to acquire an initial fix is also unpredictable because it depends on several factors such as the time since last fix and surrounding environment affecting the signal multipath [13]. While Assisted GPS aims to shorten this delay by acquiring ephemeris data through cellular networks, the time required to search and acquire satellite signals is unavoidable. This delay is typically translated to additional energy consumption by the GPS chip.

While new GPS chips have significantly improved energy consumption, GPS sensors are still very power hungry. Figure (2a) shows the instantaneous current drawn by a Samsung Galaxy S2 running a location-based application that requests a GPS reading every 10s. The measurement was collected outdoors where the phone had a clear view of the



(a) Outdoors (b) Indoors

Figure 2: Energy consumption of GPS sensor

sky. The "baseline" current drain of the device, when the GPS is switched off is 36.3 mA. The mean current consumption jumps to 115mA when the API requests location from the GPS sensor. In this particular case, it takes about 45s for the GPS to acquire an initial fix and subsequent requests take on average 7s. The energy consumed for the initial fix is 23,447mJ and the mean energy consumption for subsequent requests is 2,831mJ.

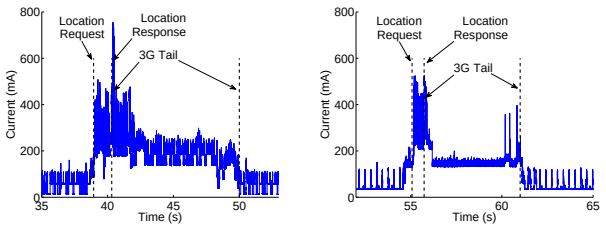
This measurement represents a best case scenario as the phone is outdoors with a clear view of the sky. In our parking application, it is highly likely that the user will move indoors after parking his or her vehicle. While it is well known that GPS receivers struggle to estimate location without a clear line of sight to satellites [9, 14], the energy consumption is also significantly higher in these scenarios. Figure (2b) shows the current consumption on the same smartphone as our previous experiment when it is moved indoors. In this case, not only is the sensor unable to estimate location, the device current consumption also stays constantly high at 120mA.

3.2 Network Based Location

An alternative location API available on current smartphones is network based location. This requires the mobile device to perform a scan for surrounding Wi-Fi access points. The captured access point SSIDs are transmitted over the cellular network to a remote server which translates this SSID set into a location coordinate using a wardriving database. Although this enables the mobile device to quickly acquire location coordinates, this approach is more suited for one off queries rather than periodic location updates due to the high energy consumption of the cellular radio interface. Figure (3) shows the current consumption for a single location query using the network based location API on both Samsung Galaxy S2 and Samsung Galaxy S1 devices. It shows that the device continues to draw high current even after the location response is received due to the 3G tail. The total energy cost of performing a single network based location request is 8,229mJ and 3,920mJ for Samsung Galaxy S2 and S1 respectively.

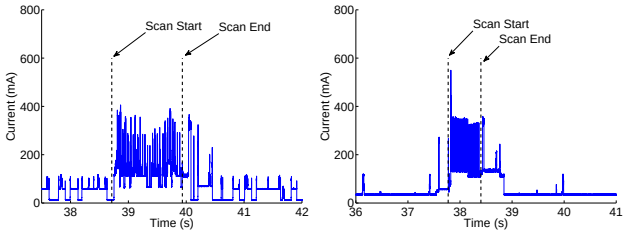
In addition to its energy characteristics there are other issues with this approach. As network based localization relies on wardriving data, it offers relatively coarse grained location. Due to the high energy costs of network based location queries, smartphone operating systems also aggressively cache the location responses thus making this approach less suitable for continuous tracking.

We now briefly analyse the energy consumption of Wi-Fi scanning which will form the basis of our approach for detecting unparking events. Figure (4) shows the current



(a) Samsung S2 (b) Samsung S1

Figure 3: Energy consumption of network location



(a) Samsung S2 (b) Samsung S1

Figure 4: Energy consumption of Wi-Fi scanning

consumption of both of our test devices while performing a passive scan for Wi-Fi access points. The mean current consumption for Samsung Galaxy S2 and S1 is 114mA and 140mA, respectively. A passive scan usually takes a fixed amount of time. It takes 1.2s and 0.65s to complete the scan resulting in an energy cost of 512mJ and 335mJ on Galaxy S2 and S1, respectively.

3.3 Accelerometer

Although an accelerometer is not an alternative sensor for tracking the location of a user, in our specific scenario it can be used to detect the actual activity of the driver. The accelerometer is an attractive option because of its low energy consumption. It has been exploited in a number of activity detection systems [8] to detect physical activities, such as "walking", "cycling", "driving", etc. However, achieving high accuracy through accelerometer data alone can be quite challenging. Challenges include the variability of the readings depending on the phone placement — traces from a phone attached to the vehicle's dash board will be very different to those from a phone in the driver's pocket — and whether the classifier is trained over a dataset collected from the actual user.

In the parking scenario, the accelerometer can assist in detecting the transition between a driver walking back to their vehicle and driving. However, if used without any additional information regarding the relative location between the driver and the vehicle, it can lead to erroneous results. For example, simply detecting the change between walking and riding a car could lead to wrong inferences for users who park their car in order to use public transportation, such as busses or trains — a common scenario in metropolitan centers. A more practical use of the accelerometer in this scenario could be to act as a low power sensor that can trigger more accurate detection through location based sensing [23].

This shows that while current smartphones are equipped with a range of sensors and location technologies, these are not suitable for our application scenario either due to high energy consumption or dependence on user specific factors. Following the analysis of possible sensing modalities, we now embark on the design of a novel approach motivated by the specific requirements of our application.

4. WIRELESS BASED SENSING

The primary motivation for the design of ParkSense is to build a mobile phone application that can detect the evacuation of parking spaces, while minimizing the impact on the user's mobile device. We saw in the previous section that performing passive wireless scans is the lowest energy operation as compared to other location sensors. This motivates us to explore the wireless interface as a sensor to sense driver mobility.

Current smartphone based parking payment systems require the user to make a payment for the amount of time that they intend to stay in on-street parking before they leave the vehicle. These users generally use a smartphone application provided by the parking management company where they enter a unique location code and their payment details to pay for parking while they are still at the parking location. Each set of parking slots has a unique location code that is clearly marked on a sign on the pavement beside the parking space. When the payment transaction completes, the user can leave the vehicle. Leaving the vehicle without making the payment leaves them open to parking infringement fines and tickets. ParkSense captures a wireless signature during this payment phase by running multiple Wi-Fi scans when the user is at the parking location waiting for the transaction to complete. This wireless signature consists of the SSIDs of the Wi-Fi access points and the number of times a beacon frame was received from each access point. Let us represent this Wi-Fi signature with sets

$$\mathbf{S_p} = \{s_p(1), s_p(2), \ldots s_p(n)\} \quad \text{and}$$
$$\mathbf{W_p} = \{w_p(1), w_p(2), \ldots w_p(n)\}$$

where $s_p(i)$ is the SSID and $w_p(i)$ is the beacon reception ratio for access point i and n is the total number of access points observed in all the scans. The beacon reception ratio $w_p(i)$ is the fraction of scans for which the access point i was visible i.e. $w_p(i) = v_p(i)/m$ where $v_p(i)$ is the number of scans in which a beacon frame was received from access point i and m is the total number of scans. ParkSense saves this signature and then periodically performs new wireless scans as the user moves away from the vehicle. From each successive window of size m, it creates a new signature with $\mathbf{S}_t$ and $\mathbf{W}_t$ in the same manner as described above. It uses these periodically generated signatures to decide if the user has returned to the parked vehicle.

This is similar to Wi-Fi fingerprint based localization [6]. However, as opposed to traditional localization where each fingerprint is assigned a coordinate in the geographic space, we do not perform any translation from the fingerprint to the real coordinates. This allows us to do away with any other external input (either from the user or any other sensor like GPS) and thus the sensing relies only on performing passive Wi-Fi scans. This makes this approach significantly more energy efficient. Detecting that the user has vacated a parking spot can be decomposed into two sub-problems,

1) sensing that the user has returned to the parked vehicle location 2) detecting that the user is in a moving vehicle after returning to the parking location. In the following subsections, we describe how ParkSense functions to sense these scenarios.

4.1 User Returning to Vehicle

ParkSense detects the return of the user to the parked vehicle by comparing the periodically generated signature sets $\mathbf{S}_t$ and $\mathbf{W}_t$ with the saved signature $\mathbf{S}_p$ and $\mathbf{W}_p$. However, this is a challenging problem because wireless signals are affected by multi-path fading, device placement close to the body and the changing environment in urban areas. The access points are usually located within the buildings and other infrastructure that attenuate the radio signals significantly. Therefore, the signal strength of radio signals received by the mobile device is usually just above the radio transceiver sensitivity making it difficult for the radio to receive the beacon frames. This manifests in a continuously changing wireless environment even if the user or the mobile device is stationary. To overcome these issues, we define the Wi-Fi signatures over a set of scans rather than a single scan. We also use the beacon reception ratio as opposed to the signal strength in our Wi-Fi signatures. It has been shown that the beacon reception ration can be more robust and a better indicator of distance in outdoor urban environments [7] as compared to received signal strength. From $\mathbf{W}_p$, we define a set of normalized reception ratios $\hat{\mathbf{W}}_p$ as,

$$\hat{\mathbf{W}}_p = \{\hat{w}_p(1), \hat{w}_p(2), \ldots, \hat{w}_p(n)\} \tag{1}$$

where $\hat{w}_p(i) = w_p(i)/\sum_i^n w_p(i)$. These normalized beacon reception ratios can be thought of as weights assigned to access points. An access point that was visible in all the scans gets a higher weight whereas the ones that appear in only a few scans get lower weights assigned to them. We then define the matching between saved signature and the current signature as,

$$M = \sum_{i=1}^{l} \hat{w}_p(i) \quad for \quad s_p(i) \in \mathbf{S}_p \cap \mathbf{S}_t \tag{2}$$

where $0 \leq M \leq 1$ and $l = |\mathbf{S}_p \cap \mathbf{S}_t|$. Therefore, the matching function is just the sum of normalized weights of access points that are common to both signatures. When no beacons are received from access points that were seen at the parking location, the matching $M = 0$ and it increases as more and more of those access points re-appear. Access points that were seen more frequently at the parking location result in a larger increase in M as opposed to those that were observed infrequently. This matching function takes into account all of the access points initially observed at the parking location and does not discard the ones with low reception ratios. It merely trusts them less than those that were frequently seen.

Here we must point out that Eq. (2) relies on the normalized weights from $\hat{\mathbf{W}}_p$ only and does not use $\mathbf{W}_t$. To observe if this has any effect on the signature matching performance, we define two more matching functions. Eq. (3) defines a matching function that penalizes the access points according to the difference between the saved reception ratios and currently observed values. Eq. (4), on the other hand, defines a function that removes the reception ratio completely and uses the percentage of saved access points as the matching function. We will refer to Eq. (2) as Weighted, Eq. (3) as Weighted Difference and Eq. (4) as Percentage. In Section 6 we will compare the performance of these to signal strength based signature matching functions.

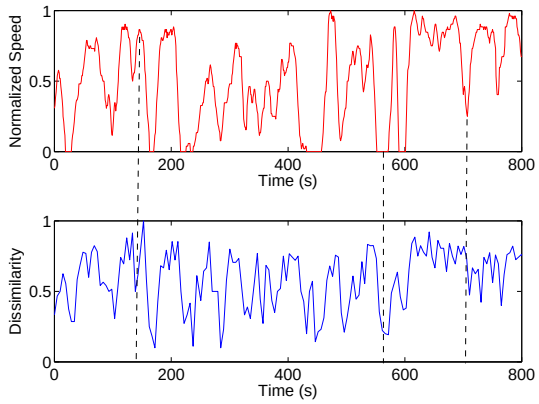


Figure 5: Comparison of normalized GPS speed and Jaccard dissimilarity between successive Wi-Fi signatures.

$$M = \sum_{i=1}^{l} \hat{w}_p(i)(1 - |w_p(i) - w_t(i)|) \tag{3}$$

$$M = \frac{|\mathbf{S}_t \cap \mathbf{S}_p|}{|\mathbf{S}_p|} \tag{4}$$

4.2 User Driving Away

When the user is in a vehicle that is moving at a typical urban road speed, the successive wireless signatures observed by the mobile device bear very little similarity as the vehicle comes within range of access points that may be located in buildings beside the road and then moves away from them quickly. A vehicle traveling in urban traffic tends to follow a certain pattern. It accelerates quickly, travels at a certain speed, decelerates and then comes to a stop due to queues, traffic lights or road junctions. In fact, the entire journey of a vehicle in an urban area is composed of these start stops known as driving cycle in transportation research [11]. In order to demonstrate this, we collect a GPS trace from a mobile device in a vehicle traveling in urban traffic. Figure (5) shows the speed of the vehicle for this particular trip in the top plot. It shows that in urban traffic, a vehicle continuously speeds up and then stops either due to other vehicles or traffic lights.

But what effect do these drive cycles have on the wireless beacons observed by the mobile device inside the vehicle? In order to answer this question, we take a trace of Wi-Fi scans collected by the same mobile device. These scans were divided into windows of size m=4 (We will show in Section 6 that other window sizes are possible and that the window size has little impact) and for each window, we create the signature sets $\mathbf{S}_t$ and $\mathbf{W}_t$. Then we compute the Jaccard similarity [12] between each pair of successive windows as,

$$J = \frac{|\mathbf{S}_t \cap \mathbf{S}_{t-1}|}{|\mathbf{S}_t \cup \mathbf{S}_{t-1}|} \tag{5}$$

Jaccard similarity is merely the ratio of the number of access points observed in both windows to the total number

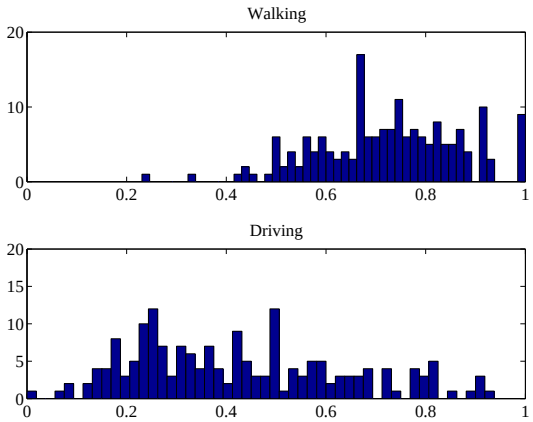


Figure 6: Comparison of the distribution of Jaccard Index for walking and driving in an city center.

of access points in the two windows. As the vehicle speeds up, we expect the successive windows to be different (low similarity) and as the vehicle slows down, these successive windows to be more and more similar. In order to make the visual comparison easy, we plot the dissimilarity $1 - J$ (also known as the Jaccard distance) between each pair of successive windows and the normalized GPS speed in Figure (5). This shows that there is a strong correspondence between the wireless beacons observed by the smartphone and the speed at which the user carrying the phone is traveling. Therefore, the wireless signature can be used as a proxy to estimate user mobility.

However, one might ask, what does the wireless signature look like when the user is not in the vehicle but continuously moving. Could this be mistaken for vehicular movement? In order to answer this, we collected another trace where a user carrying the same mobile device walked in the same urban area where the vehicle was traveling. Figure (6) shows the distribution of Jaccard Index for a user walking in the same urban area. It also compares it with the one obtained from the vehicle trace and shows that the Jaccard Index for the vehicular trace varies much more than pedestrian type mobility. In order to detect that a user is moving away in a vehicle, we need a mechanism that can detect a change from ordinary pedestrian mobility to vehicular mobility by inspecting the distribution of the Jaccard Index. In the following paragraphs, we briefly describe an algorithm for this.

Page-Hinckley Test

The Page-Hinckley Test (PHT) also known as Cumulative Sum (CUSUM) test is a widely used statistical algorithm for detecting a changes in processes [10]. It uses sequential observations of the process and detects if the process mean has shifted by more than the specified amount. The sequential nature of the algorithm is particularly attractive for our application where the results of Wi-Fi scans arrive periodically. As the name suggests, the algorithm sequentially computes cumulative sums using the following two equations,

$$S_{hi}(i) = \max(0, S_{hi}(i-1) + x_i - T - k) \quad (6)$$

$$S_{lo}(i) = \max(0, S_{lo}(i-1) + T - k - x_i) \quad (7)$$

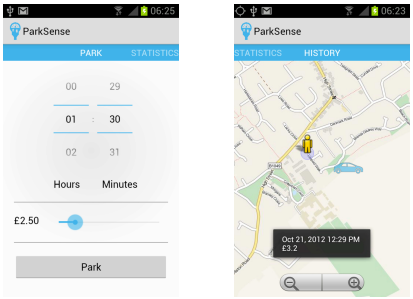


Figure 7: Android App used for collecting realistic data during user study.

where $S_{hi}(0) = S_{lo}(0) = 0$, x_i is the process mean usually calculated from a sample of q observations, T is the target value of mean for which the process is considered to be under control and h and k are design values. When the value of $S_{hi}(i)$ or $S_{lo}(i)$ exceeds h, the process is considered to have undergone a change. The value of k is calculated as,

$$\bar{\sigma} = \sigma/\sqrt{q} \quad (8)$$

$$k = \delta\bar{\sigma}/2 \quad (9)$$

where σ is the process standard deviation and δ is the amount of shift in the process mean that we want to detect expressed as a multiple of standard deviation of sample means. We use this approach to detect when the mean of the Jaccard Index J computed over successive windows drops below a threshold which indicates that the user is moving in a vehicle in urban traffic.

5. EXPERIMENTAL STUDY

We developed a smartphone application for Android OS that was released through Google Play (Figure 7). The application was launched as a parking logger allowing users to record where they parked their vehicle and collect statistics about parking payments. Essentially the application emulates the behaviour of a pay-by-phone parking application where users enter the amount of time that they intend to park and the amount of money that they had paid when they parked their vehicles. When a parking event is triggered by the user, the application records the wireless signature of the parking location and then continues to perform Wi-Fi scans and saves this data in a log file. Users can then mark the unparking event by pressing a button. This triggers the GPS sensor which is used to collect the data during driving. The smartphone application communicates with a back-end service hosted on Google App Engine to upload the logged data using a RESTful API. The application displays all the previous parking locations, durations and payments on a map. The same information is also available from a webpage where the users can login to view these details on a larger map in a browser window.

The data collection period lasted four months. During this period, 59 traces in 4 different cities were collected. Out of these 59 traces, the GPS data was available for 45 parking sessions. On inspection, we observed 41 of these sessions are complete sessions where the users eventually unpark and drive away. However, the remaining 4 traces were incomplete where the users appear to have triggered unparking but did not return and drive away. Nevertheless,

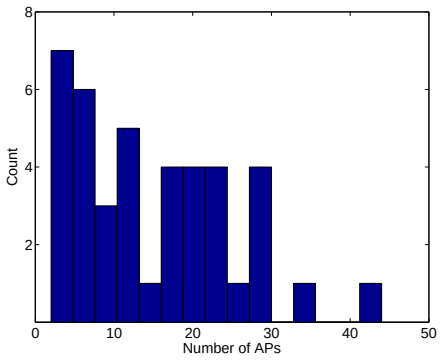


Figure 8: Distribution of APs at parking locations

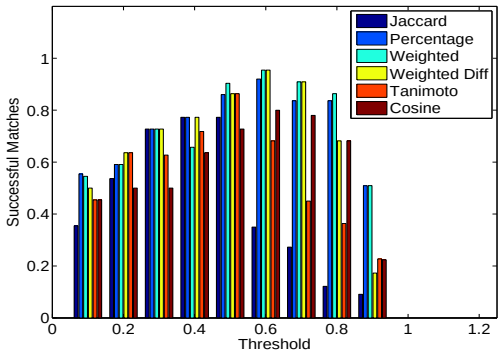


Figure 9: Comparison of different signature matching techniques.

we include these traces in our evaluation to test for false positives for driving detection. In our traces, the minimum parking duration is 15 minutes, longest one is 3.5 hours and the mean difference between the paid and actual parking time is 30 minutes. Figure (8) shows the distribution of access points observed at these parking locations. All the traces were collected from actual on-street parking at various locations.

6. EVALUATION

In this section, we evaluate the performance of our Wi-Fi based sensing approach to detect unparking events. First we evaluate the accuracy of the signature matching functions proposed in Section 4.1, then we look at the ability of the Wi-Fi signals to predict the mobility of the user, as described in Section 4.2, and finally we investigate the overall accuracy of the system in detecting unparking events.

6.1 Signature Matching

Previous research on Place Learning [14] and Proximity estimation [15] use a signal strength based similarity metric known as Tanimoto Coefficient to calculate similarity between Wi-Fi signatures. These applications are focused on indoor environments where the received signal strength from different access points located in the environment is usually quite high. Our outdoor application scenario, on the other hand, requires us to work with Wi-Fi signals that are usually severely attenuated. Under these conditions, the beacon reception ratio has previously been shown [7] to be a more robust choice. There are also several other general similarity metrics, including Jaccard Index and Cosine Similarity. We compare the three signature matching functions, Weighted, Weighted Difference and Percentage Intersection, proposed in Section 4.1 with the signal strength based Tanimoto Coefficient, signal strength based Cosine similarity and the Jaccard Index for comparing periodically captured signatures to the one saved during parking.

All of these matching functions produce a match value $0 \leq M \leq 1$ with zero for no match and one for a perfect match. We also have to choose a threshold τ such that $M \geq \tau$ is considered a match. If τ is set too high, it will result in a lower ratio of successful detections of the user returning to the parked location. This can be caused by irregular changes of the wireless signature due to multi-path fading, changing environment, missing or low reception ratio access points and thus rarely matching completely with the saved signature. If τ is set too low, it can result in false positives. For example, if the user parks the vehicle and walks to a nearby location, some of the access points will keep appearing in periodic scans and can cause a false match, indicating that the user has returned to the parking location when in reality he/she has not done so. Figure (9) shows the percentage of correctly identified user returns in 41 of our driving away traces as we vary the matching threshold τ. It shows that our reception ratio based matching functions are able to correctly identify more than 90% of user returns with a threshold $\tau = 0.6$. The cosine similarity captures 80% of returns with Tanimoto just above 60%. As expected, the performance of all matching functions drops as the value of the threshold τ is reduced (resulting in false positives) or increased (resulting in missing user return).

In our traces, the scanning interval is 2s i.e. after every 2s the application performed and logged a scan. However, while the user is away, the scanning rate can be reduced significantly to save energy. We adopt an approach for adaptive scanning where the scanning frequency is throttled down when it is estimated that the user is far from the parked vehicle. Specifically, the approach that we adopt is to perform a single scan infrequently. If none of the access points $s_p(i)$ that were captured during the parking event appear in this scan, we can continue to perform infrequent scans. If, however, any of the captured access points appear again, it triggers frequent (2s) scanning and thus signature matching. This frequent scanning times out if the match value M does not appear to increase or exceed the threshold τ indicating that the user is staying in the vicinity of parking location but not approaching it. If this happens, the next trigger to fast scanning is done on access points $s_p(i)$ that have not been observed so far to avoid continuous triggering. We explored the potential duration of infrequent scanning interval, by subsampling the collected traces captured by our mobile application. We found that in our traces an interval of up to 60s can be maintained with this approach, without affecting the accuracy of detection.

6.2 User Driving

In Figure (5), we showed that there is a strong correspondence between the Jaccard Index calculated over successive windows and user speed. Now we quantify this correspondence. We split the traces where the drivers unparked and

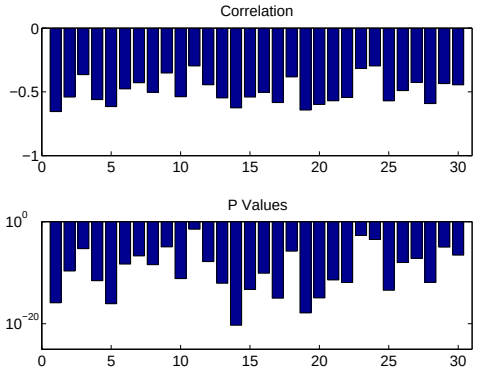


Figure 10: Correlation between normalized GPS speed and Jaccard Index computed over successive scan windows.

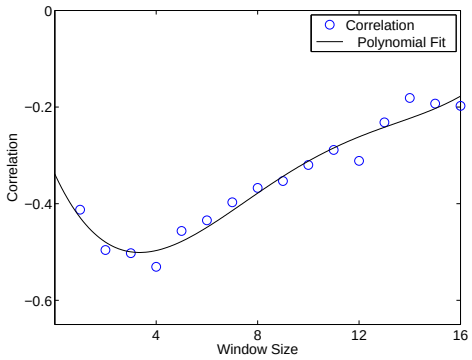


Figure 11: Optimal window size.

drove away from parked locations into parking and driving periods using the GPS ground truth. Then we calculate the Jaccard Index over successive windows of Wi-Fi scans (with window size m=4) during the driving portion of the trace. And finally, we calculate the correlation between the user speed obtained from GPS and the Jaccard similarity over adjacent scan windows. Figure (10) shows the correlation coefficients for a random sample of 30 traces: a strong correlation between vehicle speed and the Jaccard similarity calculated from the Wi-Fi scans as the user moves in an urban space is evident. The coefficients are negative because, as the user speed increases, the similarity between scan windows becomes small and vice versa. Figure (10) also plots the P-values for these coefficients. These values are extremely small (on the order of 10^{-15} to 10^{-20}), which suggests that the associated correlation values are indeed significant. To determine the best window size for sensing, we vary it and observe its effect on the correlation coefficients. Figure (11) shows that as the window size increases, the median correlation between user speed and the Jaccard similarity goes down. It shows that the maximum (negative) correlation is achieved with window size of 4.

6.3 Sensing Accuracy

We now look at the sensing accuracy of the complete system i.e. the ability of the system to detect unparking events. An unparking event consists of a user return (indicated by the wireless signature match) followed by the system sensing (using the PHT test) that the user is moving at high speed. We first look into how the parameters of the PHT are set. In Figure (6), we showed that the distribution of the Jaccard Index calculated over successive scan windows is quite different for a user that is walking along a road in an urban space as opposed to the one traveling in a vehicle in typical urban traffic. We take a random sample of 10 traces from our 41 driving traces and use these as a training set for our detection algorithm. The sample is split into portions of when the user is away from the vehicle and when the user is traveling in the vehicle. Figure (12) shows the distribution of the standard deviation of Jaccard similarity for the two portions of these traces. It shows that the standard deviation of Jaccard Index is about 0.16 before driving and rises above 0.22 when the users drive away. We, therefore, choose $\sigma = 0.22$, $\delta = 1$ (to detect a change as small as one standard deviation) and sample size $q = 5$ to calculate k. Figure (6) shows that the mean of Jaccard Index is around 0.5 when the user is traveling in the vehicle. Thus, we choose a target mean value of $T = 0.55$ and use PHT (Equations (6) and (7)) to detect when the mean value of Jaccard Index calculated over successive windows falls below T indicating that the user is driving. Furthermore, for the driving away detection we used a time threshold for the maximum duration of the detection run. This was set to 15min after the user has returned to the vehicle, so as to keep the duration of any background sensing within reasonable bounds.

From the 41 traces where the users actually unparked and drove away, ParkSense was able to detect user returns in 38 instances. The misses are due to the fact that the urban radio environment is dynamic and at times can lead to beacons with high reception ratio to disappear during the scan window. Out of these 38 instances, ParkSense was able to sense that the user is driving in 34 cases within the 15min time window. On average, PHT took 5.3min to detect driving. The remaining instances where ParkSense was not able to detect driving was due to users driving slowly or stopping for extended periods of time. It is clear that as the drivers eventually speed up and the mean Jaccard Index falls, driving will eventually be detected. This, however, requires the sensing process to run longer in order to capture such special cases. For the 4 non driving cases where users accidentally marked return to vehicle but did not drive, it was able to correctly detect non driving behaviour in all 4 instances.

The accuracy of the detection mechanism depends on the selected parameters for configuring PHT (values of σ, δ, T and maximum sensing duration). These parameters can be fine tuned in the deployed system by collecting data from probe vehicles. It is possible to use some of the users as probes by triggering the GPS sensor when a return to vehicle is detected and using the GPS trace as ground truth to fine-tune the specific parameters. These fine tuned parameters can then be passed to the mobile devices of new users during the payment transaction. Finally, as discussed in Section 2 since the purpose of the ParkSense app is to feed data into a parking availability information system, we anticipate that possible detection inaccuracies will be smoothed out when data from multiple users is aggregated for a given location.

6.4 Special cases

In all the traces we collected through the ParkSense application, the behavior of the users followed the pattern shown

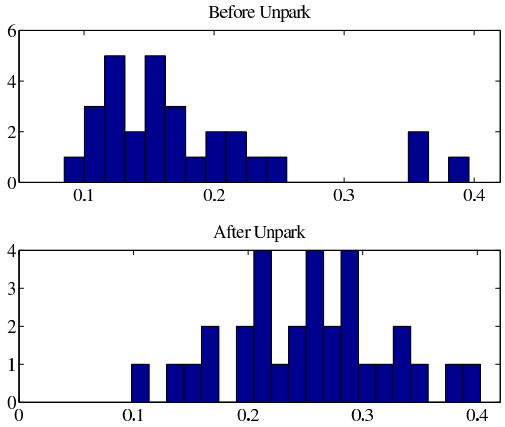


Figure 12: Standard deviation of the Jaccard Index calculated over successive windows before and during driving.

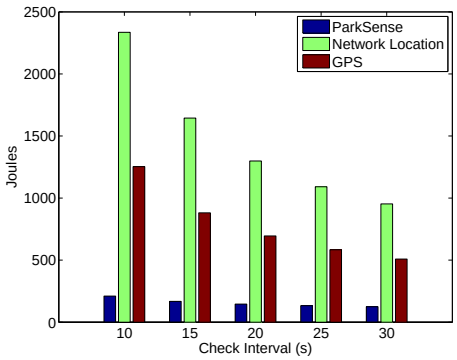


Figure 13: Energy cost of detecting an unparking event.

in Figure (1). However, there are situations where the user may not follow this particular pattern. Example cases are:

- After parking, the driver returns to the parking location and drives away in a friend's car. If the two cars are close to each other, the ParkSense application will falsely detect such an action as "unparking".
- A driver parks a vehicle and makes a payment that triggers sensing. However, the driver then gives the keys to someone else, who unparks the vehicle. ParkSense will not be able to detect this unparking event.
- The driver returns and unparks as expected. However, immediately after vacating the space, he/she gets stuck in the traffic. In this scenario, ParkSense may not accurately detect the release of the space, considering that the car has not yet driven away. In this particular case ParkSense will eventually detect the evacuation when the vehicle moves at a speed higher than the walking speed.

Although these are rare scenarios, and we have not witnessed any such cases during our deployment, nevertheless they can have an impact on the estimated parking availability. Such errors can be mitigated through a more conservative recommendation engine for parking availability, where users are directed to areas where more than one space might be available. Current infrastructure based systems already take this approach by reporting only coarse-grained availability of parking spaces. These systems label streets with high, moderate or low probability of finding a space as opposed to reporting the exact number of available spaces.

6.5 Energy Consumption

The energy consumption of sensing an unparking event depends on the number of wireless scans performed during each phase of our approach i.e. detecting a return to the parked vehicle and sensing that the user is driving. Thus the total energy consumption for a parking session can be given as,

$$E_{ParkSense} = E_{scan}\left(\frac{t_{away}}{t_{check}}\right) + E_{scan}\left(\frac{t_{drive}}{t_{scan}}\right)$$

where the two parts of above equation represent the energy consumed during the two phases of our approach. During the first phase it performs infrequent scans to check if a user is returning to the parked vehicle by observing if any of the access points captured during parking payment has reappeared. It then switches to frequent scans to perform signature matching and driving detection. In the above equation, t_{check} is the interval with which infrequent scans are performed, t_{away} is the time the user spends away from the vehicle, t_{drive} is the time required to detect that the user has started driving, t_{scan} is the wireless scanning interval during this second phase and E_{scan} is the energy required to perform one wireless scan. The mean value of E_{scan} for Samsung Galaxy S2 (from Section 3) is 512mJ, the mean value of t_{away} in our traces is 42 minutes, the mean value of t_{drive} (from Section 6.3) is 5.3 minutes and t_{scan} is 2s. Based on these values, Figure (13) shows the energy consumption of our approach for different check intervals. The energy consumption of a possible alternative approach that relies on GPS can be given as,

$$E_{GPS} = 2\ E_{Fix} + E_{indoor}\left(\frac{t_{away}}{t_{check}}\right) + E_{outdoor}\left(\frac{t_{drive}}{t_{sample}}\right)$$

This assumes the typical scenario in which the user moves indoors during t_{away}. The GPS sensor looses its initially acquired fix and consumes E_{indoor} amount of energy for each location request during this period. As the user moves out and starts driving, it acquires a fix again with the energy cost of E_{Fix} and then starts consuming $E_{outdoor}$ energy for each location estimate generated every t_{sample} seconds. We use $E_{Fix} = 23,447$mJ, $E_{outdoor} = 2,831$mJ and $E_{indoor} = 4,428$mJ (assuming that it is kept on for at least 10s) from Section 3 and $t_{sample} = 10$s for these calculations.

Another possible alternative approach would be to use the network based location. The energy consumption of this approach can be given as follows,

$$E_{Network} = E_{netloc}\left(\frac{t_{away}}{t_{check}}\right) + E_{netloc}\left(\frac{t_{drive}}{t_{sample}}\right)$$

where E_{netloc} is the energy required to perform one network based location query. We use $E_{netloc} = 8,229$mJ calculated in Section 3 for these calculations. Figure 13 compares the energy consumption of both of these possible alternatives to our approach. It shows that the energy consumption of our approach is significantly smaller than both of these alternatives. The cost of our approach is only 11% of the energy

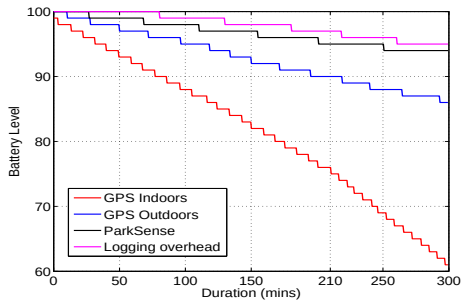


Figure 14: Battery level for various sensing approaches.

consumption of network based location and 21% of the energy consumption of GPS based approach.

In our traces, the longest parking session is 3.5 hours. With a typical 1650mAh phone battery and a 60s check interval that can trigger frequent scanning, the predicted energy consumption of our approach is around 1% of the battery life. In order to verify this, we run a series of experiments on a Samsung Galaxy S2 phone. For each experiment, the battery is fully charged and an application is run that samples the selected sensor at 60s intervals. It also logs the remaining battery level reported by the operating system. Figure (14) shows the results of these experiments. It shows that in the standby mode the phone consumes 3% of the battery due to the battery logging overhead. With our approach of Wi-Fi scanning at 60s intervals, it consumes only 5% of the battery, a small increase of 2% over the standby mode. On the other hand, battery usage for periodic sampling from the GPS sensor depends on where the phone is located during sensing. If the phone is outdoors with a clear view of the sky, GPS is able to acquire periodic location fix and consumes about 10% of the battery. However, it consumes 24% of the battery when it is indoors where it is difficult to acquire a fix. This presents the upper and lower bounds for the energy consumption of GPS sensor. In our parking application, we expect that most of the users will move indoors after parking. This not only increases the energy consumption but also makes it difficult to duty cycle the GPS sensor based on user distance from the parking location as the GPS sensor typically fails to produce a location estimate in this case. Another issue that makes it difficult to duty cycle GPS in an energy efficient manner is the unpredictable time required to acquire a fix [16]. On the other hand, our Wi-Fi based approach is free from these issues due to the fixed response time and energy consumption of the wireless transceiver.

Energy consumption for location sensing may be further reduced in some cases through the use of a low energy sensor such as acceleromter that can trigger more expensive sensing when needed (as described in Section 3.3). Such techniques are orthogonal to any of the aforementioned methods. Therefore, the difference in energy consumption will still follow the same relative trend.

7. RELATED WORK

7.1 Location Tracking Systems

There are previous pieces of work that use Wi-Fi signature matching for tracking user location. In [9] authors use Wi-Fi signature matching to learn places that can be of significance to a user. They use signature matching to sense user dwell time and mark places with large dwell times as significant. In [14], the authors present another approach that can detect more fine-grained places using signature matching. However, these approaches are focused on sensing in indoor environments and therefore use signal strength based matching functions to discriminate between nearby places in buildings, offices etc. These matching functions perform well in these environments where signal strength is high due to proximity to access points. But, as we have shown, these signal strength based functions do not perform well in our outdoor scenario where the RSSI is usually low. In [7], authors show that beacon reception ratio is a more robust indicator of distance in outdoor environments. They use it to estimate distances to access points. These distances are then used for performing localization. There is a large body of work on Wi-Fi fingerprint based localization [6]. These approaches use parameters of the signal strength distribution to form a fingerprint for a location. This makes them susceptible to changes in signal strength which is common in outdoor environments.

7.2 Parking Detection Systems

A number of smart street parking aids systems have been proposed. The SFPark project [4] is trialling the use of wireless sensor nodes embedded into asphalt at each parking location for real-time occupancy detection in down-town San Francisco. The occupancy information is collected in a central server and then provided to users looking for parking spots. A similar trial is under way in central London [2] and commercial ventures like StreetLine [5] are commercializing this technology. However, the main issue with this approach is the extremely high cost of the complete system that hinders large scale deployment. For example, the SFPark project proposes to cover only 25% of the parking spots in San Francisco at the total cost of 19.8 million US dollars, and this does not account for the system maintenance once in place. Similarly, the trial installation in London covers only 137 parking spots. Such systems have not been widely adopted often because they require large investments and long term commitments from local governments.

Mathur et. al. [17] propose a mobile sensor network for collecting road side parking spot occupancy. They use a vehicle equipped with a GPS receiver and an ultrasound transceiver attached to the passenger side as a mobile sensor. As the vehicle drives forward, the ultrasound transceiver measures its distance to the road side objects and thus used to estimate if the space is occupied or vacant. The authors propose to equip all the vehicles that routinely travel across the urban space, such as taxis, buses, police vehicles etc. with this sensing technology.

Google OpenSpot [1] smartphone application is a crowdsourcing alternative and allows drivers to mark a parking spot as *open* if they were vacating the parking location. Other drivers looking for parking could view these empty spots on their smartphones. The main drawback of this approach is the lack of incentive for the drivers to submit a report. Yan et. al. [24] try to address this problem by designing an online auctioning system around the availability of a parking spot, offering financial incentives to participating users. The drawback of this approach is that it still requires manual user intervention for the system to be use-

ful. This often leads to users either forgetting or not caring enough to send availability reports. An approach that automatically detects user departure from a parking location can overcome such problems.

7.3 Transport Modality Detection

Significant research efforts have concentrated on the detection of transportation modes using smartphone sensing. Typical approaches rely on accelerometer or a combination of accelerometer and location sensing (e.g. GPS). In [20] authors use a combination of accelerometer and GPS data and achieve a 93.6% accuracy in detecting user's mode of transportation. Although such mechanisms are valuable for a general purpose activity detection framework, they come at the expense of power hungry modalities, such as GPS. Other pieces of work have explored the use of accelerometer as a low power sensor that can trigger the use of expensive sensing such as GPS. In [22] authors perform a light-weight activity classification on continuous accelerometer data in order to trigger GPS tracking when the user is within a public mode of transportation such as bus or underground. We consider this approach complementary to the Wi-Fi sensing presented in the paper. Although not the focus of this paper, suspending sensing when the user is not moving can certainly allow further reductions in energy consumption.

8. CONCLUSIONS

We have presented an approach to on-street parking real-time information management through mobile phone wireless sensing. Our approach takes advantage of the low power consumption of the Wi-Fi radio to effectively report when drivers leave parking spots. In contrast to other approaches that either use fixed sensing infrastructure or user originated information, our aim was to devise an automatic framework to capture real time parking occupancy. Our evaluation reports on accuracy and power efficiency of the approach with respect to both a GPS and network location based solution. We present robust signature matching based on beacon reception ratio and a novel approach to driving detection.

In terms of future work, our plan is to negotiate with the developers of one of the exiting parking payment applications to incorporate this technique in their framework.

9. ACKNOWLEDGMENTS

This research has been funded by the EPSRC Innovation and Knowledge Centre for Smart Infrastructure and Construction project (EP/K000314).

10. REFERENCES

[1] Google open spot: A useful application that no one uses. http://www.androidauthority.com/google-labs-open-spot-a-useful-application-that-no-one-uses-15186/. [Online; accessed 4-December-2012].

[2] London parking technology trial. http://www.westminster.gov.uk/services/transportandstreets/parking/bay-sensor-technology/. [Online; accessed 4-December-2012].

[3] Parking sensor performance standards and measurement. http://sfpark.org/wp-content/uploads/2011/09/SFpark_SensorPerformance_v01.pdf. [Online; accessed 10-March-2013].

[4] SFPark. http://sfpark.org. [Online; accessed 4-December-2012].

[5] Street Line. http://www.streetline.com. [Online; accessed 4-December-2012].

[6] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM*, pages 775–784, 2000.

[7] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, MobiSys '05, pages 233–245, New York, NY, USA, 2005. ACM.

[8] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1797–1806, New York, NY, USA, 2008. ACM.

[9] Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian Smith, and Jeff Hughes. Learning and recognizing the places we go. In *Proceedings of the 7th international conference on Ubiquitous Computing*, UbiComp'05, pages 159–176, Berlin, Heidelberg, 2005. Springer-Verlag.

[10] D. V. Hinkley. Inference about the change-point from cumulative sum tests. *Biometrika*, 58(3):509–523, 1971.

[11] WT Hung, HY Tong, CP Lee, K. Ha, and LY Pao. Development of a practical driving cycle construction methodology: A case study in hong kong. *Transportation Research Part D: Transport and Environment*, 12(2):115–128, 2007.

[12] P. Jaccard. Distribution de la flore alpine dans le bassin des drouces et dans quelques regions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37(140):241–272, 1901.

[13] Elliott D. Kaplan and Christopher Hegarty. *Understanding GPS: Principles and Applications*. Artech House Publishers, 2 edition, November 2005.

[14] Donnie H. Kim, Younghun Kim, Deborah Estrin, and Mani B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In Jan Beutel, Deepak Ganesan, and Jack A. Stankovic, editors, *SenSys*, pages 43–56. ACM, 2010.

[15] M.B. Kjærgaard, M. Wirz, D. Roggen, and G. Troster. Mobile sensing of pedestrian flocks in indoor environments using wifi signals. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 95–102. IEEE, 2012.

[16] Mikkel Baun Kjærgaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjær. Entracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys 2009)*, pages 221–234. ACM, 2009.

[17] Suhas Mathur, Tong Jin, Nikhil Kasturirangan,

Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. Parknet: drive-by sensing of road-side parking statistics. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 123–136, New York, NY, USA, 2010. ACM.

[18] Mike McDonald and Kiron Chatterjee. VMS in urban areas - Results of cross-project collaborative study. Technical Report TR1101 D3.3.1-a, Transport Sector of the Telematics Applications Programme (T-TAP), March 2000.

[19] Nadereh Moini, David Hill, and Rooholamin Shabihkhani. Impact assessments of on-street parking guidance system on mobility and environment. In *Transportation Research Board 92nd Annual Meeting*. Transportation Research Board, 2013.

[20] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27, March 2010.

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

[22] Arvind Thiagarajan, James Biagioni, Tomas Gerlich, and Jakob Eriksson. Cooperative transit tracking using smart-phones. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 85–98, New York, NY, USA, 2010. ACM.

[23] Yi Wang, Jialiu Lin, Murali Annavaram, Quinn A. Jacobson, Jason Hong, Bhaskar Krishnamachari, and Norman Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys 2009)*, pages 179–192, New York, NY, USA, 2009. ACM.

[24] Tingxin Yan, Baik Hoh, Deepak Ganesan, Ken Tracton, Toch Iwuchukwu, and Juong-Sik Lee. A crowdsourcing-based parking reservation system for mobile phones. Technical Report UM-CS-2011-001, Department of Computer Science, University of Massachusetts, Amherst MA, 2011.