19th International Computer Science and Engineering Conference (ICSEC)
Chiang Mai, Thailand, 23-26 November, 2015

Mobile Crowdsourcing Platform for Intelligent Car Park Systems

Wantanee Viriyasitavat, Pattaraporn Sangaroonsilp, Jiranda Sumritkij and Natnaree Tarananopas
Faculty of Information and Communication Technology, Mahidol University, Thailand
E-mail: wantanee.vir@mahidol.ac.th, [pattaraporn.san,jiranda.sum,natnaree.tar]@student.mahidol.ac.th

***Abstract*—Mobile crowdsourcing concept has attracted attentions of researchers, developers and businesses in the last decade. This exact concept has also been used in the area of Intelligent Transportation System (ITS) mainly for collecting and distributing information about the current traffic condition. In this paper, we extend the mobile crowdsourcing concept to be used for an intelligent car park system. Among several existing applications, the system proposed in this paper has been designed so that it is suitable for the car parking behavior exhibited in asian countries. To be specific, the Pick N Park system was proposed and implemented to gather and deliver real-time information about a number of car parks in crowded urban areas. The proposed system not only assists drivers in planning their journeys; but it also helps alleviate the traffic congestion problem especially in front of the shopping malls (where a line of cars waiting to enter the car park blocks other traffic). In addition to the system, we also propose a car park model that can be used to estimate the congestion level and the search time of each car park. This can be useful especially during the initial phase that the technology is introduced. The model has been evaluated by monte-carlo simulations and the results show that the proposed model can provide excellent search time estimates.**

I. Introduction

Crowdsourcing concept has gained a lot of attention over the recent years. With the ubiquity of smartphones, the shift from www-based crowdsourcing platforms (such as Amazon's Mechanical Turk [1], Scribe [2], etc.) to mobile platforms has been an obvious transition. In this emerging platform, mobile phones are used as the main tool both for gathering data and delivering relevant information/results. Examples of mobile crowdsourcing platforms include *txteagle* [3] and *AskUs* [4] which are mobile platforms that allow users to request for services and sell their services through mobile devices; *placeMeter* pays users to video record their surroundings [5]. Data from videos are then processed and used for businesses such as marketing. Waze is an Intelligent Transportation System (ITS)-related crowdsourcing application in which users can report real-time data about traffic condition and incidents [6]. In Waze, users are able to search and view the incidents reported by other users. Nevertheless, most of these mobile crowdsourcing platforms such as Waze require manual input from the users which can sometimes be quite cumbersome. As a result, in this paper, we propose a mobile crowdsourcing platform that is able to gather data both explicitly (i.e., from the user input) and implicitly (i.e., transparent to the user).

While the mobile crowdsourcing has been shown to be able to provide benefits in a number of domain, the concept (with an exception of Waze) has not yet been well explored in the urban transportation problem. In this paper, we thus propose an intelligent car park system, namely Pick N Park application, that uses the mobile crowdsourcing concept to gather and deliver real-time information of a number of car parks in crowded urban areas.

Key contributions of this paper are summarized below:

- A mobile crowdsourcing platform is proposed and implemented. The proposed platform enables the data collection process to be implicit and transparent to the users while still allows user-generated content.
- We illustrate how one can use the proposed crowdsourcing platform to provide real-time information for location-specific applications. A mobile crowdsourcing application, namely Pick N Park, was proposed and implemented to provide an intelligent car parking service.
- Since the crowdsourcing platform requires a certain number of users before its benefits can be realized, we propose a mechanism to be used during the transition period that can provide a near-real-time information for the application. The performance of the proposed mechanism has been evaluated by monte carlo simulations.

The remainder of this paper is organized as follows. In Section II, existing studies that relate to our work are presented. Section III describes the system architecture of the proposed mobile crowdsourcing platform. The search time (i.e., circling time) estimation model, simulation setting and results are presented in Sections IV and V, respectively. Finally, conclusions are drawn in Section VI.

II. Related Work

In this section, studies and applications that are related to the intelligent car park systems are presented. The most widely used system for the car park systems today is the variable sign message which shows the current availability of a car park as shown in Figure 1. While the information displayed on these signs are very accurate, the information only assists the drivers in finding a parking spot at a particular parking structure and the drivers can obtain this information only when they are physically at the parking structure. Due to these disadvantages, several mobile applications have been developed in order to deliver the real-time parking information to drivers on their smartphone devices so that they can plan their journeys ahead of time. Examples of these applications are ParkMe [7], Parker^{TM} [8] and VoicePark [9].

978-1-4673-7825-3/15/$31.00 ©2015 IEEE

Fig. 1. Parking sign that shows the current availability of a parking structure [10].

Among all mobile parking applications, ParkMe [7] has the largest coverage and it can provide parking information of more than 500 cities in 19 countries (mostly in Europe and USA). ParkMe supports both on- and off-street parking. The application navigates the drivers to the closest and/or cheapest parking nearby and also allows the drivers to reserve a parking spot in advance. However, this application might not be applicable to countries in Asia, especially in Thailand whereby the parking reservation may not be available. Parker$^{\text{TM}}$ [8] is another parking application that provides real-time information about vacant spots on the street and selected local garages based on information obtained from the low-cost sensors [11] installed on the streets and in parking structure. Similarly, VoicePark application uses a real-time information gathered from the parking meters in San Francisco in guiding drivers to the closest vacant on-street parking spot [9]. This sustainable parking solution has been shown to be able to significantly decrease the circling time by more than 80% and reduce the carbon footprint by $25,000$ pounds each year.

Despite several parking applications available in the market, these applications might not be well applicable to the parking problem in Asia due to the difference in road networks and parking behavior. For instance, almost all of the parking structures are owned and maintained by a company (or a shopping mall) to provide parking for people who visit the company (or the shopping mall). As a result, there is a tendency for these companies not to share the status of their parking structure. As a result, we propose a parking application based on the mobile crowdsourcing concept in which the information is gathered from drivers to infer the current status of a parking structure. While automatic data collection is implemented in this application in order to avoid having the drivers manually input the data, other types of information, if available, can also be integrated with the proposed system.

Furthermore, the parking problem in Asian countries including Thailand is quite different from those observed in western societies. The parking problem usually causes a queue of cars lining in front of a shopping mall. This often blocks the traffic flow on the street outside the shopping mall and thus worsens the traffic congestion condition in the nearby area. As a result, the proposed system aims to convey the real-time information about the parking status to the drivers before their journey, estimate the circling time a driver will take before he/she finds a vacant spot, and navigate the driver to the selected spots. This system could help the drivers make a better decision to select the least crowded car park in the area and thus might alleviate the traffic congestion in the crowded urban areas.

III. Pick N Park

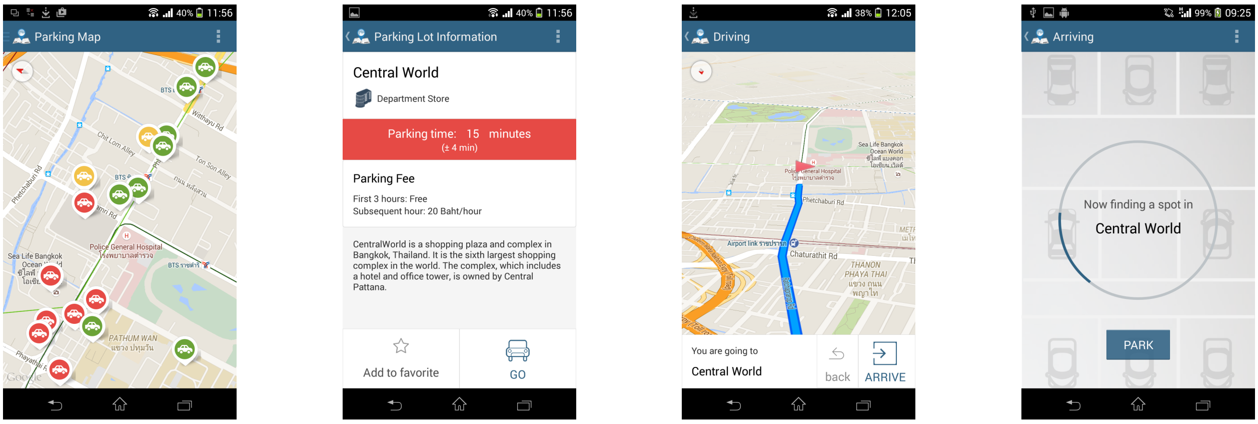
Pick N Park system is a mobile crowdsourcing platform which implicitly gather information from the drivers, processes information from the crowd, and then displays location-specific information to the drivers. Architecture of the Pick N Park system is shown in Figure 2. As depicted in the figure, the Pick N Park system mainly consists of three components: Pick N Park Android-based mobile application, Pick N Park cloud computing platform (using Amazon Web Service), and cloud storage. The mobile application is implemented on Android mobile platform and makes use of the Global Positioning System (GPS) and the Google Map API.

Fig. 2. System architecture of the Pick N Park system.

A. Overview

When a user accesses the Pick N Park application on an Android device, the application will fetch the users current location from GPS satellite. The retrieved user's current location is sent to the Pick N Park cloud computing platform which will retrieve information about nearby parking and display the results to the users as shown in Figure 3(a). Different color of each parking area indicates the congestion level; for example, the parking area with red pin indicates that the estimated circling time (i.e., the time taken before finding a vacant spot) is more than 15 minutes. Details about the circling time estimation will be provided in the subsequent section.

After the driver selects the parking area, the information about the selected area is displayed along with the estimated circling time as shown in Figure 3(b) and the application navigates the driver to the selected parking area (see Figure 3(c)). Parking areas in the Pathumwan district, Bangkok are used to illustrate the functionalities of the system as shown in Figure 3(a).



(a) Available parking areas (b) Parking area information (c) Navigation to the parking area (d) Circling timer started

Fig. 3. Snapshots of the Pick N Park mobile application on Android smartphones.

B. Data collection

In the proposed mobile crowdsourcing platform, the information of interest is the circling time a driver spends in a particular parking structure. In other words, we neglect the time a driver takes to reach a selected parking area but only consider the time a driver spends inside the parking area. In order to have a data collection mechanism that is transparent to the users, we employ the geofencing mechanism and activity recognition API provided by Google. Geofencing technique allows us to draw a polygon around each parking area [12] and is able to automatically detect when the user drives into the parking area as identified by a polygon.

Once detected, the circling timer starts to keep track of how much time a user takes to find a vacant spot (see Figure 3(d)). Google's activity recognition is used to identify the user's current activity (e.g., walking, stay still, biking, driving) [13]. When the Pick N Park mobile application detects that the user is walking, it stops the timer, records and sends the recorded circling time to the Pick N Park cloud computing platform for further processing.

C. Data processing

Once the actual circling time is received by the Pick N Park cloud computing module, the system records the received value and re-computes the estimated circling time. Note that the circling time estimation is computed for each 15-minute interval. Both the average circling time and its variation are reported to the user (see Figure 3(b)). In addition to these two pieces of information, the system also provides the number of users that are heading toward the same destination. It is worth pointing out that in the proposed system, the circling time is the main metric that determines the congestion level of a parking spot. This is because the circling time is a representative indicator regardless of the size of a parking area, its layout, the current utilization, etc. In the perspective of drivers (i.e., users), the circling time is a critical measure that is used to select a parking destination.

Furthermore, since the accuracy of the circling time estimate depends significantly on the number of users, an alternative to estimate the circling time is employed. Description and evaluation of this method are given in the next sections.

IV. Circling Time Estimation Model

As mentioned previously, the accuracy of the circling time estimation depends mainly on the number of users. External network effect of the proposed Pick N Park system has thus been a challenge for the adoption of the proposed technology. As a result, we propose another technique that could provide an estimate for the circling time in the initial stage of deployment.

In this section, we propose a variant of the model proposed by Balijepalli, et al [14]. In this model, the authors formulate a model for a origin-destination demand and various commodities. Multiple user classes and parking facilities were considered. Adapted from this model, our proposed work models only the circling time, S_t (i.e., the time taken to search for a vacant spot in a car park) given the current utilization (in percentage) of the car park, U_t:

$$S_t = s_t + \beta \left[U_t\right]^{\gamma} \tag{1}$$

where s_t is the minimum circling time of a car park and β and γ are the model parameters which are specific to the car park. It is important to note that according to Eqn 1, the circling time does not increase linearly with utilization level of the car park. In other words, the relationship between the circling time and utilization depends on the values of β and γ. As will be shown later in this paper, the values of β and γ depend on the capacity per level and the number of parking levels available at a car park, respectively.

V. Simulation Setting and Results

In order to validate the model proposed in the previous section, a car park simulator was implemented and monte-carlo simulations were performed to determine the accuracy of the circling time estimated by the model (see Equation 1).

A. Setting

In the simulator, four parameters are required as listed below:

- Capacity of the car park, C
- The number of levels in the car park, L
- Number of rows per floor, R
- Number of entrance to the building from the car park, Ent_i

The algorithm for a driver to select a parking spot is implemented based on the actual behavior of a driver. As shown in Algorithm 1, each driver always starts from the first floor and may decide to go up to another floor if the current utilization of the current floor is less than the total utilization of the car park. Note that this information is already available in most car parks as shown in Figure 1. As the driver has decided on which floor to park, the circling time is computed based on the location of the parking spot. In other words, the circling time is larger if the driver parks in a spot that is further away from the parking entrances.

Furthermore, for each spot in the same floor, the probability that a driver decides to park (when the spot is available) also differs. This is because we have observed that the driver tends to park at a vacant spot that is closer to the building entrances. As a result, the parking probability P_i^l decreases linearly as the distance between the parking location and the location of the closest building entrances increases (but the probability is consistent across different floors; see Figure 4)[1]. In addition, in the situation where the car park is almost full (i.e., the utilization is more than 95%), the driver does not have much preference in terms of parking location and parks at the first vacant spot he/she finds.

In the simulations, we assume a big car park with capacity of more than 100 vehicles and an average vehicle speed of 15 km/h inside a car park. The building entrance is located at one of the corner of the car park. In the case of multiple entrances, all entrances are on the same side and distance between two entrances is uniform as shown in Figure 4.

B. Simulation Results

In this subsection, we perform a number of simulations with various settings in order to verify the accuracy of the model presented in Section IV. Linear regression with the least squares approach is used to estimate the fitting parameters, β and γ (see Equation 2) and a total of 10 simulation runs were performed for each data point.

$$\begin{aligned} \widetilde{\log\beta} &= \frac{\sum\left[\log U_t - \overline{\log U_t}\right]\left[\log(S_t - s_t) - \overline{\log(S_t - s_t)}\right]}{\sum\left[\log U_t - \overline{\log U_t}\right]^2} \\ \widetilde{\beta} &= e^{\widetilde{\log\beta}} \\ \widetilde{\gamma} &= \overline{\log(S_t - s_t)} - \widetilde{\beta}\cdot\overline{\log U_t} \end{aligned} \tag{2}$$

Figure 5 depicts the simulation results (solid lines) and the estimated circling time (dotted lines) of a car park with capacity of 120 cars and a single parking level. Observe that the model can approximate the circling time quite well

[1]Since each vehicle starts from the first floor, the preference to park in lower floors is already taken into account.

Algorithm 1 Mechanism used to simulate driving and parking behavior.

$C \Leftarrow$ capacity of a car park
$v \Leftarrow$ average driving speed in a car park
$L \Leftarrow$ number of levels in the car park
$U^t \Leftarrow$ current utilization of the car park
$U_i^t \Leftarrow$ current utilization of the i^{th} of the car park
$V \Leftarrow$ set of all vehicles that enter the car park
$d_v \Leftarrow$ distance travelled before finding a parking spot of vehicle v
$P_i^l \Leftarrow$ parking probability of the i^{th} spot on l^{th} floor
$t_v \Leftarrow$ circling time of vehicle v

for all $v \in V$ **do**
  for all $l < L$ **do**
    $d_v \Leftarrow d_v + 30$ #distance in meters between each floor
    #starts from the first floor
    if $U_i^t < U_i$ **then**
      park in the current floor
    else
      park in the current floor with probability $(1 - U_i^t)$
    end if
    #vehicle v decides to park in the current floor
    $d_v \Leftarrow d_v + 20$ #distance in meters from the ramp to the first parking spot
    for all spots k in the current floor **do**
      if k is not occupied **then**
        if $U^t > 0.95$ **then**
          Park at this spot
        else
          Park with probability P_i^l
        end if
      else
        $d_v \Leftarrow d_v + 2.4$ #distance in meters between spots
      end if
    end for
  end for
end for
$t_v \Leftarrow d_v/v$

```
Floor Plan

---|ENTRANCE|---------------------|ENTRANCE|------------------------

   |_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_|
   |_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_|

   |_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_|
   |_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_|

   |_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_|
   |_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_||_|

-------------------------------------------------------------------

Probability of each spot

0.95 0.95 0.95 0.95 0.93 0.91 0.89 0.87 0.85 0.83 0.95 0.95 0.95 0.95 0.93 0.91
0.9 0.9 0.9 0.9 0.88 0.86 0.84 0.82 0.8 0.78 0.9 0.9 0.9 0.9 0.88 0.86 0.84 0.82
0.85 0.85 0.85 0.85 0.83 0.81 0.79 0.77 0.75 0.73 0.85 0.85 0.85 0.85 0.83 0.81
0.8 0.8 0.8 0.8 0.78 0.76 0.74 0.72 0.7 0.68 0.8 0.8 0.8 0.8 0.78 0.76 0.74 0.72
0.75 0.75 0.75 0.75 0.73 0.71 0.69 0.67 0.65 0.63 0.75 0.75 0.75 0.75 0.73 0.71
0.7 0.7 0.7 0.7 0.68 0.66 0.64 0.62 0.6 0.58 0.7 0.7 0.7 0.7 0.68 0.66 0.64 0.62
```

Fig. 4. Layout of a car park that is generated from the simulator and the parking probability P_i^l. In this layout, a car park has a capacity of 120 cars, with 1 level, 3 parking rows, and two building entrances.

especially when the utilization is low. This is mainly because of the simulation setting which forces the driver to park at the first vacant spot if the utilization is larger than 95%. In general, a car park that has many parking rows causes longer circling time (i.e., time taken to search for a parking spot) while the number of building entrances do not have significant effect on the circling time.

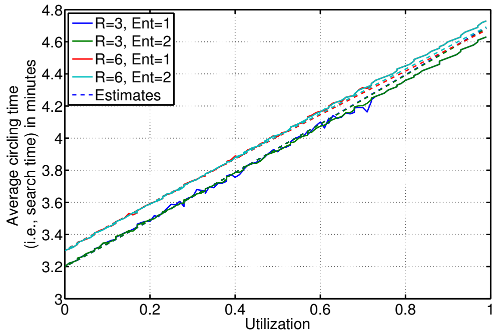


Fig. 5. Results in terms of circling time obtained from the simulations (solid lines) and from the estimation model (dotted lines). A car park is assumed to have a capacity of 120 cars and has a single parking level.

In addition, we also investigate the effect of parking levels on the average circling time. As shown in Figure 6, when the capacity per floor is fixed, the average circling time increases significantly with the increase in the number of parking levels even when the utilization level is quite low. This is mainly because as the utilization level of the first floor is slightly lower than the average utilization, the drivers tend to choose to drive up and park on upper floors. The driving time between the floors leads to the substantial increase in the circling time.

In addition, observe from the Figure that the actual circling time and the estimates predicted by the model also differ. This is not surprising because the model shown in Section IV only considers the utilization but not the structure or layout of the car park. It is important however to note that the estimation was done based on the least squares approach. Because of the logarithmic scales, the discrepancy between the actual and the estimated values is quite large when the utilization is close to 100%.

In addition to the effect of the number of parking levels, we investigate the effect of the car park layout. In other words, in this simulation, the total capacity of a car park is fixed while the layout (in terms of number of floors and number of parking rows) varies. Figure 7 shows the average circling time when a car park with a total capacity of 600 cars is assumed. Three different structure layouts are used: i) with a single level, ii) two parking levels, and iii) five parking levels. All layouts assume 3 parking rows and 2 building entrances in each floor.

Unlike results shown in Figure 6, the average circling time decreases with the increase in the number of car park levels. This is because when the car park capacity is fixed, the higher number of car park levels implies that there are fewer parking spots per floor. As a result, with the same floor utilization, the driver spends shorter time circling each floor before finding a parking spot. The time taken to drive between floors is negligible as compared to the circling time in each floor. Observe from Table I that the value of β increases as the capacity per parking level increases while the value of γ decreases as the number of parking levels increases.

Nevertheless, for all simulation setting, it is clear that the model proposed in Section IV is able to provide a good estimate for the circling time. The estimation error is within 10% from the actual time. Since the current utilization of a car park can be obtained from the historical data or estimated in the case of special events, it is therefore suitable to be used as the main method to estimate the circling time in the Pick N Park system. Once the number of users reach a certain threshold, the mobile crowdsourcing platform can then be used in conjunction with the proposed estimation model.

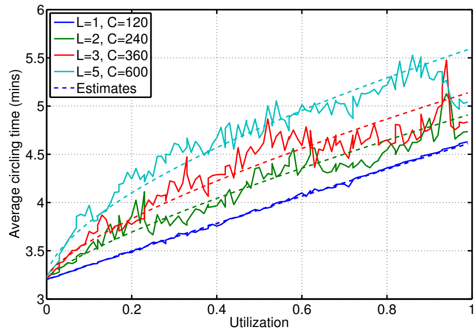


Fig. 6. Results in terms of circling time obtained from the simulations (solid lines) and from the estimation model (dotted lines). A car park is assumed to have a fixed capacity of 120 cars per floor.

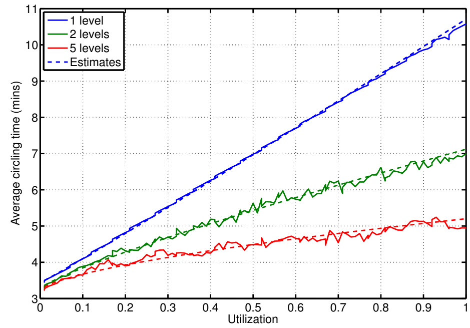


Fig. 7. Results in terms of circling time obtained from the simulations (solid lines) and from the estimation model (dotted lines). A car park is assumed to have a total capacity of 600 cars with different layouts.

TABLE I
ESTIMATED PARAMETER VALUES FOR A SCENARIO WITH A FIXED TOTAL CAPACITY OF 600 CARS WITH DIFFERENT LAYOUTS

Layout	β	γ
1 level	7.2742	1.0413
2 levels	3.8346	0.8370
5 levels	1.9868	0.6394

VI. Conclusion

In this paper, we propose a mobile crowdsourcing platform for an intelligent car park system, namely Pick N Park system. The proposed system consists of several components: mobile application, Pick N Park cloud computing module and the car park model. With the Pick N Park mobile application which uses the location-based APIs, the data about parking search time is implicitly collected and sent to the Pick N Park cloud computing module (which is implemented on Amazon's Web Service). In addition, since crowdsourcing requires a certain number of users before the application can be useful, we propose a car park estimation model which can provide a good approximation in terms of parking search time based on the car park utilization. This approximation model can be used during the initial phase and later can be integrated with the crowdsourcing technique. Monte carlo simulations were used to determine the accuracy of the model and the results show that the estimation error is less than 10%.

References

[1] Amazon Mechanical Turk, "https://www.mturk.com/mturk/welcome," [Accessed on 12 June, 2015].

[2] J. Condliffe, "Crowdsourcing Could Help Deaf People Subtitle Their Everyday Life," http://gizmodo.com/5928856/crowdsourcing-could-help-deaf-people-subtitle-their-everyday-life, 25 July 2012, [Accessed on 12 June 2015].

[3] N. Eagle, "Txteagle: Mobile Crowdsourcing," in *Proceedings of the 3rd International Conference on Internationalization, Design and Global Development: Held As Part of HCI International 2009*, IDGD '09, (Berlin, Heidelberg), pp. 447–456, Springer-Verlag, 2009.

[4] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: Extending crowdsourcing to the real world," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, (New York, NY, USA), pp. 13–22, ACM, 2010.

[5] PlaceMeter, "http://www.placemeter.com/," [Accessed on 12 June, 2015].

[6] Waze, "https://www.waze.com/," [Accessed on 12 June, 2015].

[7] ParkMe Inc., "http://www.parkme.com/how-it-works/," [Accessed on 12 June, 2015].

[8] Parker By Streetline, "http://www.streetline.com/find-parking/parker-mobile/," [Accessed on 12 June, 2015].

[9] VoicePark, LLC, "http://www.voicepark.org/," [Accessed on 12 June, 2015].

[10] INFINITE Parking Guidance System, "http://www.innetweb.com/products/v/25/parking-guidance-system," [Accessed on 12 June, 2015].

[11] V. Tang, Y. Zheng, and J. Cao, "An intelligent car park management system based on wireless sensor networks," in *Pervasive Computing and Applications, 2006 1st International Symposium on*, pp. 65–70, Aug 2006.

[12] Geofence, "https://developers.google.com/android/reference/com/google/android/gms/location/geofence," [Accessed on 12 June, 2015].

[13] Activity Recognition, "https://developers.google.com/android/reference/com/google/android/gms/location/activityrecognition," [Accessed on 12 June, 2015].

[14] N. C. Balijepalli, S. Shepherd, and A. May, "Modelling the choice of car parks in urban areas and managing the demand for parking," in *87th Annual Meeting of Transportation Research Board*, 2008.