Proceedings of the Twenty-Ninth AAAI Conference on Innovative Applications (IAAI-17))

ParkUs: A Novel Vehicle Parking Detection System

Pietro Carnelli,* Joy Yeh, Mahesh Sooriyabandara, Aftab Khan
Toshiba Research Europe Ltd., 32 Queen Square, Bristol, UK
pietro.carnelli@toshiba-trel.com

Abstract

Finding on-street parking in congested urban areas is a challenging chore that most drivers worldwide dislike. Previous vehicle traffic studies have estimated that around thirty percent of vehicles travelling in inner city areas are made up of drivers searching for a vacant parking space. While there are hardware sensor based solutions to monitor on-street parking occupancy in real-time, instrumenting and maintaining such a city wide system is a substantial investment. In this paper, a novel vehicle parking activity detection method, called ParkUs, is introduced and tested with the aim to eventually reduce vacant car parking space search times. The system utilises accelerometer and magnetometer sensors found in all smartphones in order to detect parking activity within a city environment. Moreover, it uses a novel sensor fusion feature called the Orthogonality Error Estimate (OEE). We show that the OEE is an excellent indicator as it's capable of detecting parking activities with high accuracy and low energy consumption. One of the envisioned applications of the ParkUs system will be to provide all drivers with guidelines on where they are most likely to find vacant parking spaces within a city. Therefore, reducing the time required to find a vacant parking space and subsequently vehicle congestion and emissions within the city.

Introduction

Since 1950 rural populations across the world have been declining as a greater proportion of the world's population live in urban areas or cities. A recent UN urbanisation report notes that 54% of the world's population now live in urban areas, this is expected to rise to 66% by 2050 (United Nations 2014) . As cities grow in population, further demand and pressure is placed on transportation systems. Vehicle journeys accounted for 83% of the distance travelled by the UK population in 2012 (a huge increase from just 27% in 1952) (Department for Transport 2015). Recent studies have highlighted that approximately 30% of inner city congestion is made up of drivers searching for a vacant parking space, with the average parking search time being approximately 8 minutes. Although this may initially seem small, research has shown that even a search time of just 3.3 minutes (as experienced in Westwood Village, LA) per driver per parking activity can have adverse consequences on congestion and pollution (Shoup 2007). Shoup concluded that over the course of a year, vehicles searching for a parking space in Westwood Village contributed over 600 tonnes of CO_2 emissions by driving over 1.5 million km.

One solution is to provide drivers with better, real-time parking occupancy information, as this would reduce the time spent searching or cruising for a vacant parking space. Often this is done by using thousands of sensors mounted on short pillars in private off-street parking facilities to monitor parking occupancy levels. However, these systems are expensive, especially when implemented on a city wide scale for all on-street parking. For example, San Francisco, US, spent over US\$20 million to equip $6,000$ on-street vehicle parking bays (or 25% of on-street parking) with sensors connected to the internet as part of their SFPark system. As a result, recent research has focused on finding ways to monitor in real-time parking occupancy without incurring huge physical infrastructure costs. Potential solutions have included: cameras overlooking parking spaces (Jermsurawong et al. 2012), ultrasonic distance sensors mounted on vehicles (Mathur et al. 2010) and individual user reporting (Sherwin 2011). However, they are often either difficult to implement and scale, relatively expensive and or require users to collaborate by tirelessly logging all their parking activities.

Motivated by the strong trend of smartphone ownership and usage in paying for on-street parking, we develop a novel parking activity detection system called ParkUs for a smartphone. The primary design goal of the ParkUs system is to maximise detection accuracy whilst minimising energy consumption. The proposed system can eventually be scaled to a city wide smart parking solution; whereby parking activity detected by the ParkUs application can be collated and plotted onto a road map of the city to high-light to other users where it's best to search for parking (given a destination address) and whether or not if it would be simpler or faster to take public transport to their desired destination, Figure 1 shows the envisaged ParkUs system diagram. The main contributions of this paper are the following:

1. A robust and energy efficient vehicle parking activity detection method;
2. Implementation as part of a generic system called ParkUs, that uses a novel feature (also presented in this paper) to

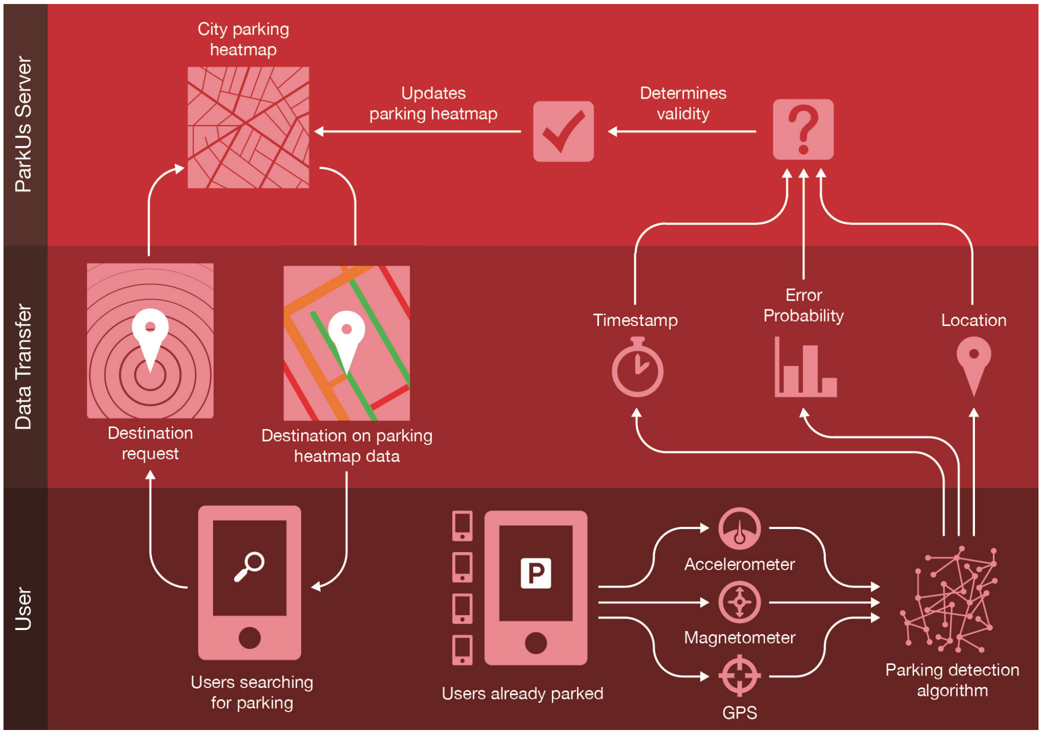

Figure 1: Overall envisaged ParkUs city smart parking system.

detect vehicle parking activities;

3. We demonstrate improved accuracy and reduced energy consumption with respect to existing systems without requiring any user input or sensor calibration.

Related Work

There have been several attempts to develop parking activity detection systems with similar aims of reducing congestion and pollution within urban areas. Most of these efforts have focused on using energy consuming wireless radio modules such as GPS and Wi-Fi to detect parking activity.

PhonePark (Stenneth et al. 2012), to our knowledge, was the earliest attempt at detecting parking activities with a driver owned smartphone. Stenneth et al. were able to sense whether a user was walking, stationary or driving. They were able to detect parking by virtue of identifying state changes. Their detection algorithm relied on GPS, accelerometer and Bluetooth connectivity data collected on the user's smartphones. PhonePark was able to detect parking and unparking (when a driver vacates a parking space) activities with 80% and 85% accuracy respectively.

In an attempt to reduce the reliance on GPS sensors, ParkSense (Nawaz, Efstratiou, and Mascolo 2013) chose to use the Wi-Fi radio module of smartphones to approximate speed and location in order to detect only unparking events in a city. Each location was associated with a Wi-Fi access point ID. Using the change in Wi-Fi ID's (as the user drove past multiple access points) they were able to infer speed at which the user was travelling and as a result when the user unparked their vehicle. ParkSense was able to achieve 83% true positive rate on unparking detections. However, ParkSense was not able to run in the background of smartphones, since it relied on the user to manually geo-tag where they had parked their vehicles (as ParkSense only focused on detecting unparking events). ParkSense also required a 60s data collection window and had a 5 minute detection delay.

Park Here! (Salpietro et al. 2015) utilised an accelerometer as a well as a gyroscope sensor in order to detect parking activity. Park Here!'s classification was binary; driving or not driving. Similar to PhonePark's system, it was able to detect parking activities by recording changes between the two states. Park Here! was able to achieve perfect detection (with a true positive rate of 100%) when the user had a Bluetooth system installed in their vehicles.

PhonePark and Park Here! rely on energy intensive radio modules and sensors such as GPS and gyroscopes as well as Bluetooth connections to make parking detections. Whereas our system, ParkUs, uses modality detection that matches specific sequences detected using low-power sensors such as the magnetometer and accelerometer before briefly triggering location sensors (such as GPS) to automatically geo-tag the parking or unparking event. Using this method, ParkUs was able to incur lower energy costs but still achieve a similarly high parking activity detection and location accuracy.

ParkUs: Design and Algorithms

The ParkUs vehicle parking detection system is shown in Figure 2 with four key components: an initial median data filter, a feature processor, a modality detector (Figure 3) and a finite state machine (FSM) model (Figure 4).

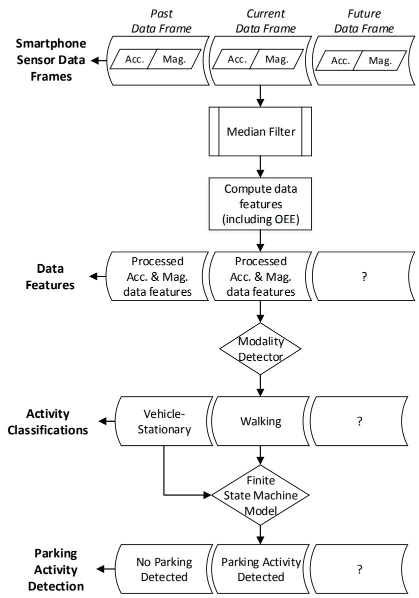

Figure 2: ParkUs detection architecture.

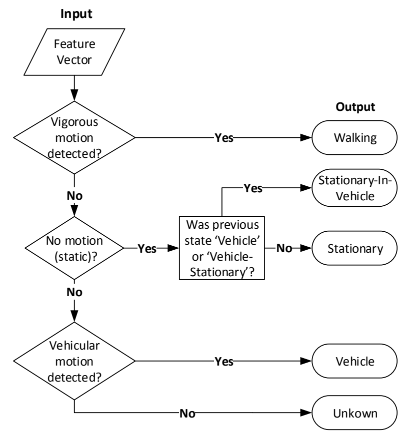

Figure 3: ParkUs cascaded modality classifier.

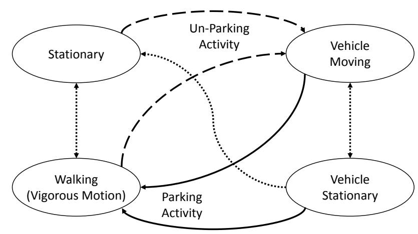

Figure 4: ParkUs FSM model. Solid and dashed lines are parking and unparking state changes respectively.

Data Pre-Processing

Due to the openness and resultant fragmentation of the Android platform, no sensor calibration was conducted. To counter potentially low quality sensors, a median filter and hard thresholds were used to remove any outlier data. A sliding time window was carefully chosen to ensure that finer changes in activities were captured and detected.

Feature Extraction

Previous research suggested that better detection between stationary, walking and in-vehicle modes can be achieved by estimating the gravity vector to generate an accurate decomposition of the tri-axial data (Hemminki, Nurmi, and Tarkoma 2013). Using this algorithm, the gravity and the North vectors were estimated from the accelerometer and magnetometer data respectively. The algorithm opportunistically searched for moments where the accelerometer readings were stable (low variance) in order to estimate the gravity vector. The variance threshold would increase gradually in order to obtain the optimal number of stable moments within the accelerometer data. The North vector was estimated using the same algorithm for the magnetometer data.

When resting, the smartphone is always supported by some upward force (that opposes gravity). This constant force forms the vertical vector referred to as G. On the other hand, the magnetometer tends to point towards the magnetic North pole, and is estimated as vector N. Although heavily skewed by the Earth's magnetic field, the horizontal North vector, N_f, can be compensated by subtracting N's projection onto G from itself. Using these two vectors with respect to the smartphone, any acceleration can be rotated (Figure 5) and decomposed into vertical and horizontal components (Tundo, Lemaire, and Baddour 2013) (Figure 6). Once the magnetometer and accelerometer data were filtered, transformed and decomposed, we derived per window; gravity eliminated acceleration, and estimations of G and N.

Feature Family Descriptions

Over 300 features were considered in the development of the ParkUs detection algorithm, many inspired by previous research in human activity detection. The features fall roughly into the following main feature families:

Statistical Features: Standard statistical metrics such as minimum, maximum, mean, median, interquartile range, variance, overall range and root mean square were used to analyse and capture changes in the data. In addition, the distribution of signal values (within the data window) were represented by its empirical cumulative distribution function (ECDF), with a resolution of 10 bins (Hammerla et al. 2013).

Discrete Fourier Transform (DFT): In the frequency domain, the 1-3Hz DFT coefficients were shown to be good indicators of cyclical walking motion (Wang, Chen, and Ma 2010). Peak frequencies were also identified as features in addition to the peak coefficient (inspired by 'peak frequency power' in (Thiagarajan et al. 2010)). Similar statistics from Welch's power spectrum were also included.

Peak Statistics: A 'peak' in this context refers to a period of sustained acceleration in one direction (see the horizontal peak in Figure 6). Inspired by Jermsurawong et al., peak characteristics were captured using area under curve, cumulative sum, kurtosis, skewness, length between peaks, and length of peak.

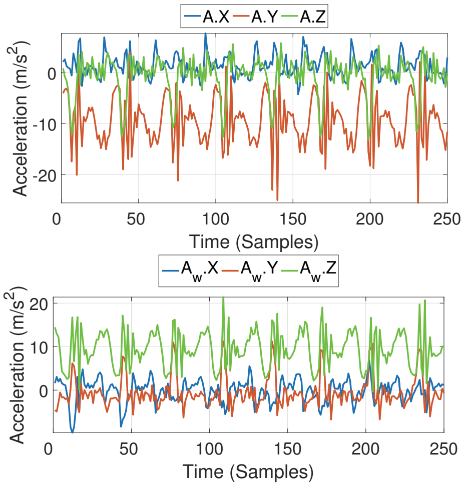

Figure 5: Rotation of walking motion accelerations from smartphone (top) to world axes.

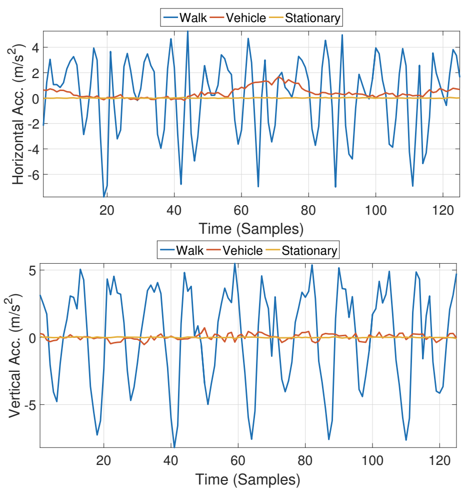

Figure 6: Acceleration components for each motions class.

Wavelet Entropy: The information contained in the wavelet coefficients help analyse transient features and non-stationary signals. A chaotic signal contains more information than one that doesn't vary significantly (Langley 2015).
Orthogonality Estimation Error (OEE): A novel feature representation was developed for the ParkUs system which works by estimating the error between the gravity and north vectors. The OEE was a good indicator of vehicular motion, as it had a correlation of -0.53 with the user's travelling speed. This level of correlation is similar to the Jaccard index used by ParkSense. The OEE is calculated as follows; for an arbitrary window, where G and N are approximations of the vertical 'up' vector (opposite to gravity) and the magnetic North vector respectively, the angle, θ_{GN}, between the two vectors is given by:

$$\theta_{GN} = \cos^{-1}\left(\frac{G \cdot N}{|G| \cdot |N|}\right); \mathrm{OEE} = |\theta_{GN} - 90°|$$

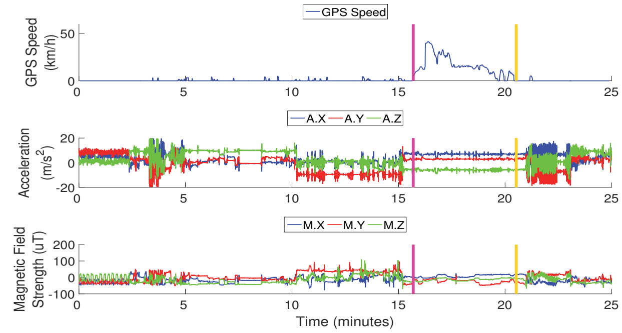

Figure 7: Example of data recorded for a journey.

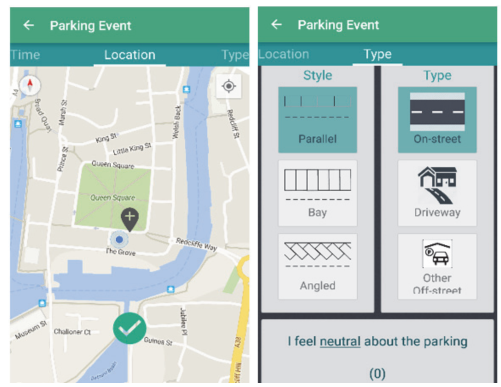

Figure 8: Screenshots from our data collection application.

Ideally, G always points vertically up and N always points to true North without deviation. The angle between them should therefore be perpendicular. Realistically, G and N are almost never orthogonal due to the shape of the Earth's magnetic field. There is a natural inclination of N in most areas on Earth except those very near to the equator. Since the study was conducted in the UK, the inclination effect was very pronounced (around 60 degrees below the horizon). Furthermore, G and N are only approximations. They are both easily corrupted by unstable motion and abrupt changes in orientation, which aides parking detection.

Parking and Unparking Detection

Parking and unparking detection was achieved by correctly matching the modality sequences. For example, a sequence of *walking→walking/stationary→vehicle moving* indicates that the user has *unparked* and vice versa for parking detection (Figure 4). To reduce activity recognition fluctuations the ParkUs modality detector took the mode of the past five modalities. Figure 3 shows the cascaded modality classifier.

To further reduce false detections a modified version of the detection algorithm was also developed and evaluated using the same method (cross user validation). The modified version, known as ParkUs-SA (Speed Assisted), included speed data collected from the GPS sensor or network based provider every 10 seconds until either the speed of the user confirms with the current detection decision, or denies it.

Training Data Collection

To implement the ParkUs system, we developed a custom data logging and tagging Android smartphone application for our volunteers (all located within Bristol, UK) to use during the four-week long data collection period. The volunteers were free to place and carry their smartphones as they normally would everyday. The multiple sensor data (along with the user tags) were stored within the application and sent only over Wi-Fi to our IES Cities server platform (López-de-Ipiña, Aguilera, and Pérez 2015). Figure 7 shows sensor data recorded for a typical trip with our application; the yellow and purple vertical lines respectively denote the unparking and parking events annotated by the volunteer. Screenshots of our application are shown in Figure 8.

By virtue of the Nyquist criterion, the accelerometer was sampled at 25Hz, as it was previously shown that 98% of human physical motions occur below 10Hz (Antonsson and Mann 1985). We also sampled the magnetometer, GPS, ambient noise and light sensors at 5, 1, 10 and 5Hz respectively.

Seven volunteers recorded a combined total of 62 journeys over the trial period. The mean journey length taken was 43 minutes. Overall, we recorded 52 and 57 parking and unparking events. The disparity was caused by volunteers occasionally forgetting to tag their parking activities.

Results and Evaluation

Modality Classification

One-vs-all classifiers were trained for three motions: *walking*, *static* and *vehicular motion*. The following learning algorithms were compared: Decision Tree (J48), k-Nearest Neighbour, Multilayer Perceptron, Support Vector Machine (trained using a standard RBF kernel), Naïve Bayes, AdaBoost and Random Forest. For non-ensemble learning, three different feature selection algorithms were tested: correlation based subset (Hall 1999), information gain and gain ratio (Hall et al. 2009).

Overall, the Random Forest models achieved the highest accuracies: 98.4% in 10-fold cross validation and 95.8% in cross-user validation. The latter was done to mimic the real world application where all the data from one user is excluded from the training set and is instead used as the test set. To gauge the importance of our novel OEE feature presented in this paper, a Random Forest without the OEE feature was also trained, which in turn achieved a lower average accuracy of 87.4%, 9% less than the Random Forest trained with the OEE features.

A grid search of bagging parameters saw optimal accuracies at 100 simple trees with a maximum of 10 splits. Ensembles of any greater size yielded negligible accuracy gains while taking much longer training times, furthermore less splits leads to better generalisation.

Parking Detection

The ParkUs algorithm was evaluated using cross-user validation, whereby in each fold all of the data from one user is held out as the test set. This simulated the realistic situation where a deployed parking activity detection algorithm is not able to train on every users' data beforehand. This avoids the contamination of information in the training data, as highlighted by (Hammerla and Plötz 2015).

Detection Accuracy: Table 1 shows the results of the performance comparison between the different parking detection systems. Using the sequence matching technique with a 27.5s confirmation window (the length in time of prior saved modality classifications), ParkUs correctly detected 57 out of 58 unparking events and 52 out of 53 parking events. Thus achieving a True Positive Rate (TPR) of 0.981 and 0.983 for unparking and parking detection respectively.

In total, ParkUs falsely raised events 22.4% and 16.2% of the time for parking and unparking respectively. Only one false negative detection (i.e. a missed parking activity detection) occurred. Both versions of ParkUs considerably outperformed rivals such as PhonePark and ParkSense in terms of parking and unparking detection. Although ParkUs did not achieve a TPR of 1 as Park Here! claims, in terms of false positives ParkUs is 8.2% more likely to detect a non-existent event than Park Here!. PhonePark and ParkSense did not report false positives. ParkUs-SA was almost able to match Park Here! in terms of false positives; 12% vs 11.1% claimed by Park Here!.

This could have been for several reasons, firstly, ParkUs was evaluated on a larger and more diverse data set than any other system. Secondly, Park Here! produced a large number of false modality detections during their all-negative experiment. By contrast, there were extremely few false modality detections for ParkUs in our all-negative (all driving or all walking) journeys. Thirdly, our modality detection is more accurate; 96% compared to 90%.

Detection Delay: In ParkUs the window length was 5s for each of the motion classifiers in the cascaded modality detector. A ten window 'lookback scope' was used to ensure that fragmentation was low. Since each window overlapped by 50% this meant a minimum of 30 seconds was required for stable motion classifications. Lastly, error corrections were applied to avoid illogical inferences such as two parking events happening next to each other within a couple of seconds. For the modified, ParkUs-SA detection algorithm the speed requests were overlapped with error checking and logical inferences, thus causing no extra delay during simulation. However, realistically GPS sensor calls typically take a few seconds in order to get accurate speed measurements. Nonetheless the ParkUs system was able to detect events on average within 1 minute of them occurring; a great improvement over the only other reported detection delay by ParkSense of 5.3 minutes.

Energy Consumption

We developed a model to approximate energy consumption for each stage of the detection process for each system, thus allowing us to investigate and compare the relative performances of the different parking detection systems.

Since all related work ran on different smartphones and operating systems, their energy consumption has been estimated assuming that all parking detection systems ran on a Nokia N95. Although it's been nearly a decade since its

Table 1: Performance comparison results of different parking detection systems.

Study	Accuracy (%)	Unparking TPR	Parking TPR	FPR	Detection delay (minutes)	Test dataset size (events, users)	Energy Usage (J)
ParkUs	98	0.98	0.98	0.19	0.90	111, 7	1240
ParkUs-SA	98	0.98	0.98	0.12	1.00	111, 7	1880
PhonePark	93	0.85	0.80	N/A	N/A	N/A, 5	10600
ParkSense	93	0.83	N/A	N/A	5.30	41, N/A	3330
Park Here!	90	1.00	1.00	0–0.11	N/A	40, N/A	1840

initial launch, the Nokia N95 was equipped with all the sensors necessary to run the various parking detection systems being compared. Furthermore, multiple studies (Abdesslem, Phillips, and Henderson 2009; Constandache, Choudhury, and Rhee 2010; Wang et al. 2009; Perrucci et al. 2009; Yang and Cho 2014; Kjærgaard et al. 2009; Perrucci, Fitzek, and Widmer 2011) have evaluated and verified its sensor's power consumption. Their data was used in our energy consumption model.

However, several assumptions had to be made to simplify computation and allow for fair comparison. For example, 3G and GPS signals were considered excellent and Wi-Fi was ubiquitous. Initial GPS fixes took no extra time or energy. Each GPS fix and Wi-Fi scan took 3s and 2s respectively to obtain. There was no Bluetooth connection in the vehicle, data packets were assumed small and took 1s to send.

In terms of battery life, Nokia N95 has a 1200mAh (roughly 7992 J, assuming 100% efficiency) that supplies 3.7 V direct current. The test case for each algorithm was as follows: a user carried a Nokia N95 for 3 hours. He took a 30 minute drive to visit town for a dinner with friends. After having the dinner, which took 2 hours, he heads home. Driving home also took 30 minutes. In total, 2 parking and 2 unparking events took place. This test case was chosen as it closely resembled typical travel patterns and would allow for fair estimation of the different parking detection algorithms' energy consumption. The governing equations of the energy model are included in the appendix (Table 2).

Each parking system was subjected to the same test scenario. In order to aid comparison, some specific assumptions were made for certain systems. For example, ParkSense only detected unparking events, therefore, to aid comparison (by virtue of the test scenario), it was assumed that it could also detect parking events with the same energy consumption.

Overall ParkUs had the lowest estimated energy consumption in the test scenario. The modified version, ParkUs-SA, had a slightly higher energy consumption; this was due to the fact that on average ParkUs-SA made four speed data requests per event detection. Although ParkUs-SA consumed more energy per detection, the false positive rate was reduced to 10.3% and 13.6% for parking and unparking respectively (roughly half of the original, detection algorithm).

Conclusions and Future Work

In this paper we presented the ParkUs parking activity detection system: a novel method that leverages a smartphone and its sensors to perform parking activity detection in the background without user involvement. The parking detection system is robust and power efficient through its reliance on a novel feature also presented here called the OEE. The OEE relied on low power sensors such as the accelerometer and magnetometer to achieve parking and unparking detection. We evaluated the system extensively through cross-user validations conducted on over 50 hours of travel data provided by seven volunteers. The ParkUs parking activity detection system was able to achieve detection accuracies comparable to the best of currently published methods whilst utilising less energy (33% less than the next best method).

In our broader vision for the future, the ParkUs system will incorporate the novel detection system presented in this paper as well as features for allowing users to visualise parking occupancy in a city. For this to happen, further research needs to be conducted to understand when users start to search for parking spaces during their journey towards a destination. Statistically, any roads that the driver drives down before eventually parking will have been full (devoid of vacant parking spaces). This knowledge will greatly reduce the number of users required to provide real-time information for other city dwellers also searching for parking.

Acknowledgements

The authors thank Dr S. Vatsikas and Mr M. Faies for their help developing our application. This work received funding from the EU's Horizon 2020 programme (grant agreement No. 691735) and was supported by the EPSRC Industrial Doctorate Centre in Systems (EP/G037353/1).

Appendix

Energy Model Test Case Scenario Assumptions

PhonePark: Bluetooth turned off, GPS sampled every 15s, 3G transmission every 15s, accelerometer briefly turned on for 3 minutes after each parking event and no false positives: $E_{go}(T_{out}, T_{out}/15) + E_{gi}(T_{in}, T_{in}/15) + E_{ut}(T, T/15) + E_{ac}(3 \times 60 \times 2) = 10600J$

ParkSense: Wi-Fi scans every 60s when user is away, Wi-Fi scans every 2s when user driving, 3G transmission with GPS location for 4 of the detected events, no false positives. It is assumed that ParkSense is able to detect parking to aid comparison: $E_{ws}(T_{out}, T_{out}/2) + E_{ws}(T_{in}, T_{in}/60) + 4E_{er} = 3330J$

Park Here!: Accelerometer and gyroscope turned on throughout, 3G transmission with GPS location for 4 of

Table 2: Energy consumption model; governing equations

	Process	Energy Estimate
UTMS	Idle-off	$E_{io}(T, N) = max(min(T + T_{io}, N \cdot T_{io}) \cdot P_i, 0)$
	Active-idle	$E_{ai}(T, N) = max(min(T + T_{iai}, T_{ai} \cdot N) \cdot (P_a, 0)$
	Tail	$E_t(T, N) = E_{ai}(T, N) + E_{io}(T - N \cdot T_{ai}, N)$
	Send	$E_s(N) = N \cdot P_a \cdot T_{tr}$
	Total	$E_u(T, N) = E_s(N) + E_t(T - N \cdot T_{tr}, N)$
GPS	Outdoors	$E_{go}(T, N) = min(T + T_{gpo}, N \cdot T_{gpo}) \cdot (P_{go})$
	Indoors	$E_{gi}(T, N) = min(T + T_{gpo}, N \cdot T_{gpo}) \cdot (P_{gi})$
	Event Report	$E_{er} = E_{go}(1, 1) + E_{ut}(1, 1)$
	Wi-Fi	$E_{ws}(T, N) = N \cdot T_{wtr} \cdot P_{ws} + T \cdot P_{wi}$
	Gyr.	$E_{gy}(T) = T \cdot P_{gy}$
	Acc.	$E_{ac}(T) = T \cdot P_{ac}$
	Compass	$E_{mg}(T) = T \cdot P_{mg}$

the detected events, no false positives: $E_{ac}(T) + E_{gy}(T) + 4E_{er} = 1840J$

ParkUS: Accelerometer and compass turned on throughout, 3G transmission with GPS location for 4 of the detected events, 0.192 false positive probability: $E_{ac}(T)+E_{mg}(T)+(4 + 4 \times 0.192)E_{er} = 1240J$

ParkUs-SA: Accelerometer and compass turned on throughout, 4 GPS samples per detected event, 3G transmission for 4 of the detected events, 0.121 false positive probability: $E_{ac}(T) + E_{mg}(T) + (4 + 4 \times 0.121)(E_{er} + 4E_{go}(30, 4)) = 1880J$

References

Abdesslem, F. B.; Phillips, A.; and Henderson, T. 2009. Less is More: Energy-efficient Mobile Sensing with Senseless. In *Proc. MobiHeld*. ACM.

Antonsson, E. K., and Mann, R. W. 1985. The Frequency Content of Gait. *Journal of Biomechanics* 18(I):39–47.

Constandache, I.; Choudhury, R. R.; and Rhee, I. 2010. Towards Mobile Phone Localization without War-Driving. In *Proc. INFOCOM*. IEEE.

Department for Transport. 2015. Transport Statistics Great Britain. Technical report.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11(1):10–18.

Hall, M. A. 1999. *Correlation-based Feature Selection for Machine Learning*. Ph.D. Dissertation, Univ. of Waikato.

Hammerla, N. Y., and Plötz, T. 2015. Lets (not) Stick Together: Pairwise Similarity Biases Cross-Validation in Activity Recognition. In *Proc. UBICOMP*. ACM.

Hammerla, N. Y.; Kirkham, R.; Andras, P.; and Plötz, T. 2013. On Preserving Statistical Characteristics of Accelerometry Data using their Empirical Cumulative Distribution. In *Proc. ISWC*. ACM.

Hemminki, S.; Nurmi, P.; and Tarkoma, S. 2013. Accelerometer-based Transportation Mode Detection on Smartphones. In *Proc. SenSys*. ACM.

Jermsurawong, J.; Ahsan, M. U.; Haidar, A.; Dong, H.; and Mavridis, N. 2012. Car Parking Vacancy Detection and its Application in 24-Hour Statistical Analysis. In *Proc. Frontiers of Information Technology*. IEEE.

Kjærgaard, M. B.; Langdal, J.; Godsk, T.; and Toftkjær, T. 2009. EnTracked: Energy-efficient Robust Position Tracking for Mobile Devices. In *Proc. MobiSys*. ACM.

Langley, P. 2015. Wavelet Entropy as a Measure of Ventricular Beat Suppression from the Electrocardiogram in Atrial Fibrillation. *Entropy*.

López-de-Ipiña, D.; Aguilera, U.; and Pérez, J. 2015. Collaboration-Centred Cities Through Urban Apps Based on Open and User-Generated Data. In *Proc. UCAmI*. Springer.

Mathur, S.; Jin, T.; Kasturirangan, N.; Chandrashekharan, J.; Xue, W.; Gruteser, M.; and Trappe, W. 2010. ParkNet: Drive-by Sensing of Road-Side Parking Statistics. In *Proc. MobiSys*. ACM.

Nawaz, S.; Efstratiou, C.; and Mascolo, C. 2013. ParkSense: A Smartphone Based Sensing System For On-Street Parking. In *Proc. MobiCom*. Cambridge: ACM.

Perrucci, G. P.; Fitzek, F. H. P.; Sasso, G.; Kellerer, W.; and Widmer, J. 2009. On the Impact of 2G and 3G Network Usage for Mobile Phones' Battery Life. In *Proc. European Wireless Conference*. IEEE.

Perrucci, G. P.; Fitzek, F. H. P.; and Widmer, J. 2011. Survey on Energy Consumption Entities on the Smartphone Platform. In *Proc. Vehicular Technology Conference*. IEEE.

Salpietro, R.; Bedogni, L.; Di Felice, M.; and Bononi, L. 2015. Park Here! A Smart Parking System Based on Smartphones' Embedded Sensors and Short Range Communication Technologies. In *Proc. WF-IoT*. IEEE.

Sherwin, I. 2011. Google Labs Open Spot: A Useful Application That No One Uses — AndroidAuthority.

Shoup, D. C. 2007. Cruising for Parking. *ACCESS Magazine* 30:16–22.

Stenneth, L.; Wolfson, O.; Xu, B.; and Yu, P. S. 2012. PhonePark: Street Parking Using Mobile Phones. In *Proc. Mobile Data Management*. IEEE.

Thiagarajan, A.; Biagioni, J.; Gerlich, T.; and Eriksson, J. 2010. Cooperative Transit Tracking Using Smart-Phones. In *Proc. Embedded Networked Sensor Systems*. ACM.

Tundo, M. D.; Lemaire, E.; and Baddour, N. 2013. Correcting Smartphone Orientation for Accelerometer-based Analysis. In *Proc. MeMeA*. IEEE.

United Nations. 2014. World Urbanization Prospects. Technical report.

Wang, Y.; Lin, J.; Annavaram, M.; Jacobson, Q. a.; Hong, J.; Krishnamachari, B.; and Sadeh, N. 2009. A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition. In *Proc. MobiSys*. ACM.

Wang, S.; Chen, C.; and Ma, J. 2010. Accelerometer Based Transportation Mode Recognition on Mobile Phones. In *Proc. APWCS*.

Yang, K., and Cho, S. 2014. A Non-GPS Low-Power Context-Aware System using Modular Bayesian Networks. In *Proc. MOBILITY*.