LALS: A Low Power Accelerometer Assisted Location Sensing Technique for Smartphones

Thomas Olutoyin Oshin, Stefan Poslad
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, England
Email: {thomas.oshin, stefan}@eecs.qmul.ac.uk

***Abstract*—Advances in location tracking sensors in smartphones have led to the emergence of many location-based services (LBS). Continuous use of these location sensors improves the reliability and accuracy when identifying a user's location, but results in quicker battery depletion due to high energy consumption. In this paper, we present an energy-efficient location determination method for smartphones named Low Power Accelerometer Assisted Location Sensing (LALS). LALS is a high-availability hybrid technique that combines the use of GPS, Wi-Fi Positioning System (WPS), GSM Positioning System (GSMPS) and accelerometer. The novelty of our method is threefold. First, it involves extracting 6 features (5 novel and 1 derived) from the embedded smartphone accelerometer data without need for accelerometer noise filtering. Second, it provides real-time smartphone based user activity classification with a time constraint of 2 seconds avoiding the need to use a remote link to an in-network activity state analyzer. The user activities are stationary, sitting, lying down, standing, walking, jogging, cycling, and motorized movement (travel by bus, overhead train, underground train, taxi, and car). Third, it detects a user's activity transition, promoting a more energy-efficient location sensor selection algorithm. Results show LALS can achieve energy-savings of up to 53% in a typical commuter scenario without compromising on the location accuracy as compared to combinations of GPS, and WPS or GSMPS.**

I. Introduction

The embedded GPS receiver can be used to identify a user's location to enable LBS such as maps and navigation; however the use of GPS alone has limitations such as high energy consumption and unavailability in locations with an obscured view of GPS satellites. Alternative sensors based upon Wi-Fi and GSM can aid in overcoming these limitations, but with an increased average localization error.

Ubiquitous location determination of people during daily activities is challenging because the ability to identify the location is complex, dynamic and variable. Although much travel for work, home, and leisure activities are often pre-planned and habitual requiring little need for location sensing, travel to new destinations and deviations to planned journeys can be facilitated using location sensing. Smartphones that can utilize multiple types of location sensors, rather than dedicated devices, are increasingly being used to enable LBS, but the continuous use of smartphone location determination technologies such as GPS and WPS, lead to rapid mobile device battery depletion.

Excessive energy consumption may become a major obstacle to broader acceptance of location-aware mobile applications or services, no matter how useful the service may be [1]. GPS based location determination can provide a location accuracy to around 8 meters, but the downside is an unacceptable short battery life of less than 7 hours in some mobile devices [4]. Continuous IEEE 802.11 execution consumes even more energy than GPS [5]. These excesses illustrate the need to design energy-efficient location determination (EE-LD) schemes.

Location caching can also play a vital role in reducing energy consumption since the device can reference the cache rather than recalculating the present location. Location caching delivers location updates rapidly. It can work in indoor locations where specific location transmission signals, e.g., GPS, are unavailable, and it aids the use of targeted context based information such as finding a coffee shop or medical clinic nearby. The downside of location caching is the associated ethical and perceived privacy issues of users and their positions being stored and tracked by 3rd parties [2]. In addition, the energy cost of location caching could be more expensive if there is no matching historical cache of a user's position. Due to these disadvantages, location caching isn't considered as a single practical approach to reduce energy consumption needed for location determination.

The issue of deciding which location determination technology to invoke based on the user activity raises several research questions. We present LALS, a high-availability EELD technique that combines the use of GPS, WPS, GSMPS and accelerometer. The primary contributions of this paper are embodied in the following two questions. Under which user location contexts should different combinations of GPS, WPS, or GSMPS technologies be invoked and when should the switch occur? When should location sensors be turned on and off? The novelty of this research as compared to existing smartphone based energy-efficient location sensing techniques is threefold. First, LALS extracts 6 features (5 are novel and 1 is derived) from the smartphone accelerometer data. Second, LALS implements a low-energy light-weight computational model to process in real-time with a 2 second constraint, the user activity accelerometer data without need for accelerometer noise filtering. Third, it detects a user's activity state transition, promoting a more energy-efficient location sensor selection algorithm. To evaluate our framework we used real-world accelerometer data gathered from 15 adult volunteers.

This work has been carried out as part of the ASSET (Adaptive Security for Smart Internet of Things in eHealth) research project funded by The Research Council of Norway VERDIKT program (Grant No: 213131/O70)

II. Related Work

We focus specifically on a review of related EE-LD techniques.

Energy Efficient Mobile Sensing System (EEMSS) [1] uses the embedded mobile phone sensors to recognize user activities and detect state transitions. It uses a combination of sensor readings from the accelerometer, Wi-Fi detector, GPS, and microphone to automatically recognize the user state as described by three real-time conditions; namely motion (such as running and walking), location (such as staying at home or on a freeway) and background environment (such as loud or quiet). Evaluation of EEMSS with 10 users over one week revealed an increase in the mobile device battery life by more than 75% while maintaining both high location accuracy of 92.56% and low latency in identifying transitions between end-user activities. EEMSS powers only a minimum set of sensors and using appropriate sensor duty cycles it significantly improves the device battery life. This localization technique is useful in determining the user state which aids in knowing the appropriate localization technology to invoke.

Escort [6] is a system that guides a user to the vicinity of a desired person in a public place. An audio beacon, accelerometer and compass are used. These sensors can detect when a user is stationary and disable all location sensors. The sensors are then enabled again once movement is detected. On average the accuracy is 8.2 meters without need for war-driving or signal calibration. In this architecture neither GPS nor Wi-Fi is required to locate an individual. It highlights the importance of embedded smartphone sensors such as the accelerometer and compass in determining the user location context.

Azimuth Based Localization for Mobile Phones [8] is a GPS-free localization and traveling route estimation concept based on measured acceleration and compass data of smartphones. The built-in mobile phone compass is used to measure the relative direction changes of the traveling user. The resulting azimuth trajectories are then matched to a vectored street map. Tests reveal this can deliver route estimates, if there is knowledge of the starting point or region of the measurement and if proper azimuth measurements are given. This technique could provide route estimates once the starting location is known. To determine the user location context, this technique relies on the compass and accelerometer which are low energy sensors, but the location accuracy is low and errors accumulate with distance.

These surveyed techniques save battery energy by reducing the location sensor sampling rates. A major side-effect of this is increased average localization error. The energy-efficiency can be further improved without compromising on the location accuracy by implementing a location sensor algorithm managed by erudition of the user activity state. Knowledge of the user activity state aids in the activation and deactivation of location sensors.

III. Architecture and Design

LALS is a hybrid EE-LD technique compatible for use in both indoor and outdoor locations. There are two main human activity states. They are ”in-motion” and ”stationary”. For instance sitting in a moving train or bus is classified as an ”in-motion” state. Asleep at home or sitting at your office desk is classified as a ”stationary” state. Fig. 1 details the flowchart of the LALS model. The LALS model was constructed based

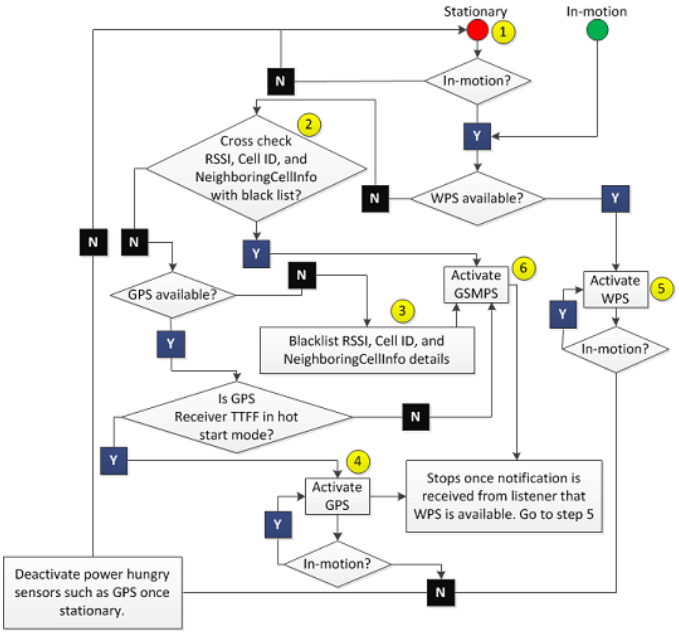


Fig. 1: Flowchart of the LALS location sensing model.

on our earlier research into a method to evaluate the energy-efficiency of wide-area location determination techniques used by smartphones [11]. The EE-LD model is designed such that the average localization error and the energy utilized for location determination are minimized.

The EE-LD model consists of three aspects. The first is a set of location determination technologies $\{ls_1, ls_2, ls_3\}$ where $ls_i \in$ *[GPS, WPS, GSMPS]*. The second is a set of user activity states which are: ”in-motion” and ”stationary”. The user transition states are derived from the four permutations of the user activity states. The transition state is the set $\{ts_1, ts_2, ts_3, ts_4\}$ where $ts_i \in$ *[”in-motion to stationary”, ”in-motion to in-motion”, ”stationary to in-motion”, and ”stationary to stationary”]*. For LALS the transition states used are: ”in-motion to stationary” and ”stationary to in-motion”. LALS use only these two transition states because the model is mainly concerned about whether a user is stationary or in-motion. The in-motion and stationary user states are detected using the LALS algorithm. The third is a set of embedded smartphone sensors $\{ps_1, ps_2, ps_3...\}$ where $ps_i \in$ *[accelerometer, gyroscope, magnetometer...]*. LALS implements only the accelerometer. Three advantages of the accelerometer are: 1) low energy consumption of 60 mW as compared to 330 mW by GPS and 1426 mW for Wi-Fi scans [1]. 2) There is no delay when starting the accelerometer. In contrast receiving location updates from GPS depends on the start mode. In a hot start mode the Termed-Time-to-Subsequent-Fix (TTSF) is about 10 seconds and in a cold start mode the Time-To-First-Fix (TTFF) could take up to 15 minutes. 3) Sensor readings are continuously available with the accelerometer. In contrast to GPS and Wi-Fi which could be obstructed from signals transmitted by GPS satellites and being out of range of Wi-Fi transmitters respectively. The challenge is determining the appropriate conditions to activate and deactivate the location determination technologies sensors which are GPS, WPS, and GSMPS. Once a user becomes stationary then within 2 seconds

GPS and Wi-Fi location sensors can be turned-off. The order of invoking the location technologies was based on the location accuracy vs. energy efficiency trade-off of GPS, WPS and GSMPS. The LALS process is detailed as follows.

1) Once transitioning from a stationary to an in-motion state check for Wi-Fi signals. In the presence of Wi-Fi signals go to step 5 else go to step 2.
2) Cross check the current RSSI, Cell ID, and NeighboringCellInfo details with the GPS blacklist table. If a match is found go to step 6 else check for GPS signals. If the GPS receiver TTFF is not in hot start mode then invoke both steps 4 and 6, else go to only step 4. The reason for also activating GSMPS (step 6) is to reduce the delay when acquiring location updates.
3) Blacklist the RSSI, Cell ID, and NeighboringCellInfo details. Cell IDs could have a range of up to 35 km [9]. With such a large coverage range, Cell IDs and RSSI alone are unreliable in detecting the absence of GPS. Knowledge of the NeighboringCellInfo will aid in pinpointing the area with limited or unavailable GPS signals to prevent activating GPS in such locations. It should be noted that even though the GSM standard permits a mobile phone to receive RSSI information from up to seven cell towers [3], most cellular providers only permit reading signal strength information about the associated cell tower [10].
4) Activate GPS for location determination. Stop GPS and go to step 5 once notification is received that Wi-Fi is available. If GPS becomes unavailable invoke both steps 3 and 6.
5) Activate WPS for location determination. If WPS becomes unavailable then go to step 2.
6) Use GSMPS for location determination. Stop GSMPS and go to step 5 once notification is received that Wi-Fi is available. GSMPS is used only in the absence of GPS and WPS.

LALS has two components. They are the user activity classification and the user personalization.

A. User activity classification

This uses the embedded smartphone accelerometer sensor to detect whether the user is in-motion or stationary. The user activity is detected by extracting features from the magnitude of the accelerometer signal vector, which is calculated using the formula $||v|| = \sqrt{x^2 + y^2 + z^2}$. Based on the study of the accelerometer data gathered for various human activities, we found these features were sensitive enough to classify the human patterns. Our human activity state classification model permits activities such as walking vs. driving to be accurately classified because the accelerometer data is more aligned to the activity. The LALS model requires the smartphone accelerometer to be in continuous execution mode and the accelerometer data is gathered in the Android normal sensing mode at 4 samples per second. To compute whether the user activity is stationary or in-motion the following light-weight computation features are extracted every 2 seconds (8 samples):

1) Peak (P): This is the number of peaks. The Peak is the local maxima if the first and last elements are local minima's. This is detailed in Fig's 2a and b. The acceleration peak is calculated as follows:

$$Q_i = \begin{cases} 1, & \text{if } (x_{i+1} > x_i) \text{ and } (x_{i+2} < x_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$
$$P = \sum_{i=0}^{n-2} (Q_i) \tag{1}$$

2) Trough (T): This is the number of troughs. The trough is the local minima if the first and last elements are local maxima's. This is detailed in Fig's 2c and d. The acceleration trough is calculated as follows:

$$Q_i = \begin{cases} 1, & \text{if } (x_{i+1} < x_i) \text{ and } (x_{i+2} > x_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$
$$T = \sum_{i=0}^{n-2} (Q_i) \tag{2}$$

x_i is the $||v||$ of each accelerometer data point.
n is the total number of data points.
P is the total number of peaks.
T is the total number of troughs.

3) T_{PT}: This is the sum of the total peak (P) and trough (T) acceleration values.

$$T_{PT} = P + T \tag{3}$$

4) mm: This is the difference between the maximum of the peak and trough values; and the corresponding minimum values. The following is the mm equation:

$$mm = max_{\forall i(0<i\leq m)}(max_{\forall j(0<j\leq n)}(G_i^P - G_j^T)) \tag{4}$$

5) P_{mm}: This is the difference between the maximum and minimum peak values given the T_{PT} range every 2 seconds (8 accelerometer samples). The following is the P_{mm} equation:

$$P_{mm} = max_{\forall i(0<i\leq m)}(max_{\forall j(0<j\leq m)}(G_i^P - G_j^P)) \tag{5}$$

6) T_{mm}: This is the difference between the maximum and minimum trough values given the T_{PT} range every 2 seconds (8 accelerometer samples). The following is the T_{mm} equation:

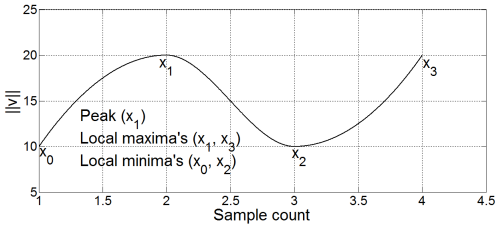
$$T_{mm} = max_{\forall i(0<i\leq n<)}(max_{\forall j(0<j\leq n)}(G_i^T - G_j^T)) \tag{6}$$

where i and j are integers.

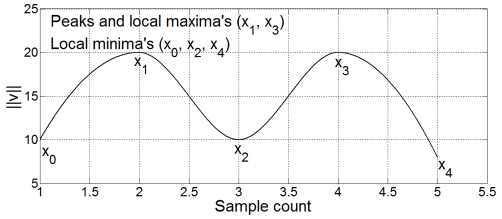
G^P is the group of peak values, which has m elements.
G^T is the group of trough values, which has n elements.

B. User personalization

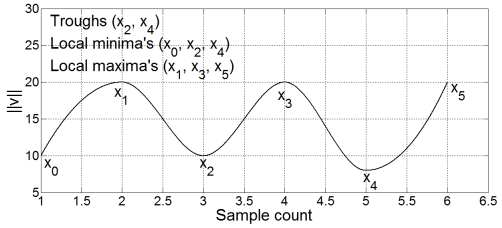
Different user activity patterns tend to be generated by users for similar activities. The algorithm must be able to adapt to variations when a user performs an activity. e.g., what is classified as walking for a certain group might be classified as jogging for another group. The features range thresholds are required to align the algorithm to the user's activity pattern which improves the user activity classification accuracy. This involves personalizing LALS by reconfiguring the algorithm based on the smartphone accelerometer data gathered for the specific activity. The personalization phase is a one-off



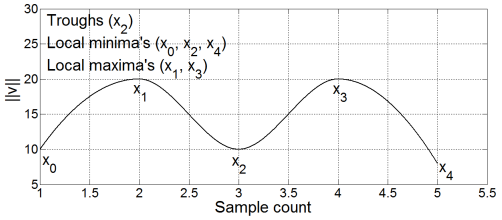
(a) Peak and local maxima count differ because the first and last elements are not local minima's.



(b) Peak and local maxima count are the same because the first and last elements are local minima's.



(c) Trough and local minima count differ because the first element isn't a local maxima.



(d) Trough and local minima count differ because the first and last elements are local minima's.

Fig. 2: Peak and trough vs. local maxima and minima.

process that takes 14 seconds (56 accelerometer samples) per activity. 14 seconds was chosen because a minimum of 56 accelerometer samples are required to cover the T_{PT} range from 0 to 6. We selected the optimal value of 8 accelerometer samples which occurs every 2 seconds after iterations involving 1 second (4 samples), 2 seconds (8 samples), 3 seconds (12 samples), 4 seconds (16 samples), 5 seconds (20 samples), 6 seconds (24 samples), up to 62.5 seconds (250 samples) because it presented the largest differences of T_{PT}, mm, P_{mm}, and T_{mm} within the shortest computation time. It should be noted given 8 samples (2 seconds) the maximum possible T_{PT} value is 6.

We found for stationary activities $mm \leq 1.4$ and $T_{PT} \leq 2$ as compared to in-motion user activities where $mm > 1.4$ and $T_{PT} > 2$. We derived the normal value of mm based on our study of the accelerometer user activity data features. The features studied are: range, mean, standard deviation, and correlation of $||v||$ of accelerometer data. The process involves deriving the following: T_{PT} range estimation, P_{mm} range, and T_{mm} range.

1) T_{PT} range estimation: The gaussian distribution of T_{PT} need to be determined to accurately align the algorithm to the user's activity pattern. Calculate the $||v||$ for each (x, y, z) sample. At intervals of 8 samples extract the peaks and troughs for 7 iterations. Sum the count of peaks and troughs for each iteration and aggregate the T_{PT} value based on the percentage of occurrences within 0 to 6. Given the gaussian distribution of T_{PT}, if the sum of the distribution percentage for 2 or 3 consecutive T_{PT} values is $\geq 75\%$ then the T_{PT} range is between the corresponding minimum and maximum T_{PT} values. E.g., for stationary activities a combination of T_{PT} values 0 and 1 is 98%. For motorized movement the sum of T_{PT} values 3, 4, and 5 is $\geq 75\%$. The T_{PT} range for both activities is $(0, 1)$ and $(3, 5)$ respectively.

2) P_{mm} range: This is the range between the minimum and maximum peak values given the T_{PT} range for the activity. Algorithm 1 details the pseudocode to generate P_{mm} range given the $||v||$ data for the user activity. We found P_{mm} useful in distinguishing between subtle user activity states such as travel by bus vs. car.

Algorithm 1 P_{mm} range pseudocode

Require: $A = \{x_i \dots x_n\}$ // T_{PT} gaussian distribution.
Require: $S_A = size(A)$ // array size of A.
Ensure: $E = \emptyset; i = 0; k = 0$
  for all v in $\{A_0,\ A_1,\ \dots\ A_{(S_A-1)}\}$ **do**
    $E_k = \sum_{i=k}^{k+1} v_i$ // Sum of 2 consecutive v elements.
    $k = i$
  end for
  $M_E = max(E)$ // Maximum element in E
  if $M_E \geq 75$ **then**
    $min_p = min(M_E, M_{E+1})$
    $max_p = max(M_E, M_{E+1})$
    return (min_p, max_p) // p_{mm} range
  else
    reset(E) // reset to an empty set.
    for all v in $\{A_0,\ A_1,\ \dots\ A_{(S_A-2)}\}$ **do**
      $E_k = \sum_{i=k}^{k+2} v_i$ // Sum of 3 consecutive v elements.
      $k = i$
    end for
    $M_E = max(E)$
    if $M_E \geq 75$ **then**
      $min_p = min(M_E, M_{E+1}, M_{E+2})$
      $max_p = max(M_E, M_{E+1}, M_{E+2})$
      return (min_p, max_p) // p_{mm} range
    end if
  end if

3) T_{mm} range: This is the range between the minimum and maximum trough values given the T_{PT} range for the activity. Given the $||v||$ data for the user activity, the T_{mm} range is generated using algorithm 1, but with trough values rather than

peak values. Congruent to P_{mm} we found T_{mm} particularly useful in distinguishing between similar user activities such as walking vs. jogging.

Once the features are extracted from the accelerometer data, the next step involves deriving the user activity state given the personalized range feature thresholds per user. As the user performs the activity the features are recalculated every 2 seconds (8 samples). The features T_{PT}, mm, P_{mm}, and T_{mm} are recalculated using an instantiation of equations 1 to 6. The user activity is determined once the calculated values are within the personalized range feature thresholds. Algorithm 2 shows the pseudocode to determine the user activity given the values of T_{PT}, mm, P_{mm}, and T_{mm}. Where $m^{mod}_{min}, m^{mod}_{max}, k^{mod}_{min}, k^{mod}_{max}, p^{mod}_{min}, p^{mod}_{max}, t^{mod}_{min}$, and t^{mod}_{max} are min mm, max mm, min T_{pt}, max T_{pt}, min p_{mm}, max p_{mm}, min t_{mm}, and max t_{mm} respectively for the user activity.

Algorithm 2 Pseudocode to determine the user activity for selected activities given T_{PT}, mm, P_{mm}, and T_{mm}.

while $(mod \neq null)$ **do**
  if $((mm \geq m^{mod}_{min} \wedge mm < m^{mod}_{max}) \wedge (T_{pt} \geq k^{mod}_{min} \wedge T_{pt} < k^{mod}_{max}) \wedge (p_{mm} \geq p^{mod}_{min} \wedge p_{mm} < p^{mod}_{max}) \wedge (t_{mm} \geq t^{mod}_{min} \wedge t_{mm} < t^{mod}_{max}))$ **then**
    $state = mod$;
    return $state$;
  end if
  $\forall\ mod \in$ {user activity states e.g., walking, etc.}
end while

Fig. 3 shows a screenshot of the LALS Android application. In comparison to other architectures LALS doesn't compromise on the location accuracy by altering the location sensor sampling rate. For energy-efficient location determination, we sample power hungry sensors such as GPS based on the user activity. Our hybrid technique uses the status of the user's activity context to manage the process of turning on and off location-based sensors. The user activity is detected using the embedded smartphone accelerometer. For e.g., if a user becomes stationary then disable power hungry location sensors such as GPS.

IV. Results

The experiments conducted involved the study of accelerometer data gathered from 15 participants for 12 different activities. The user activities are stationary, stationary with slight movements (sitting, lying down, and standing) and in-motion (walking, jogging, cycling, motorized movement including travel by bus, overhead train, underground train, taxi, and car). In order to validate LALS we required a wide range of realistic set of user data to stress test the algorithm. The activities were selected because they offered a wide range of normal urban commuting activities and to potentially detect the variability that could distort the results of our algorithm.

The validation process involved using 10 different models of Android based smart devices. The smart devices include HTC Desire HD running Android version 2.3.5, Samsung Galaxy S smartphone running Android version 2.1-update1, Samsung S II running Android version 4.0.3, Samsung Galaxy Note I running Android version 4.0.4, Hauwe 300C running Android version 2.3, Lynk 3D II running Android version 2.3.3, Samsung Galaxy II running Android version 4.0.4,

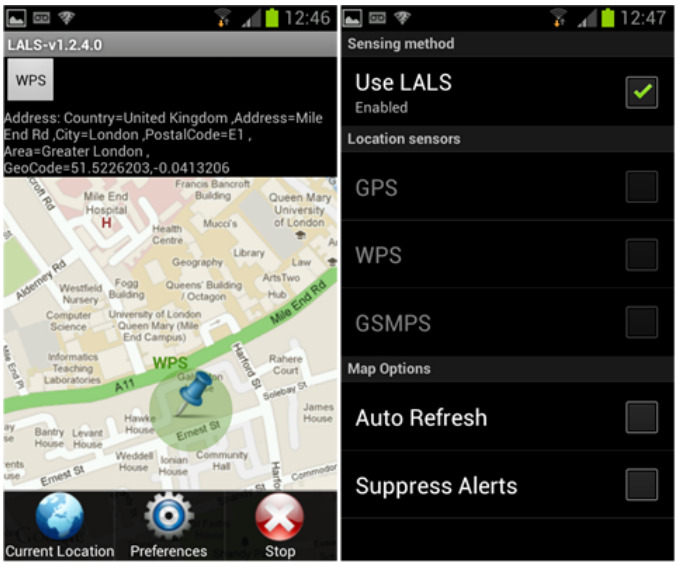


Fig. 3: Screenshot of the LALS Android application.

Samsung Galaxy Tab GT-P5110 running Android version 4.0.4, Sony Xperia U ST25i running Android version 4.0.4, and Samsung Galaxy Tab GT-N8010 running Android version 4.1.1. The devices have a dynamically user-selectable full scale acceleration range of $\pm 2g/\pm 4g$ /$\pm 8g/\pm 16g$. We found no discrepancies in the classification accuracy of the results.

A. Accelerometer noise filtering

We investigated whether or not a discrete Kalman filter algorithm could filter the accelerometer noise thus ameliorating the activity state detection accuracy estimation. The accelerometer readings provide reasonably accurate data for mobility detection, and for this reason the Kalman filter algorithm is well suited for filtering the gaussian process. We chose to use the Kalman filter due the algorithm's ability to efficiently compute accurate estimates of the true value given noisy measurements. Also there is no need to retain historical measurements and estimates, as only the current and confidence estimate levels are required.

1) Discrete Kalman Filter: We estimate the state $x \in R^n$ of a discrete time process using the linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{7}$$

The $n\,x\,n$ matrix A is the state transition model. It elucidates how the state transitions from time $k-1$ to k. The $n\,x\,1$ matrix B is the control signal $u \in R^1$ in relation to the state x. w_k is the process noise which is constant. Using the measurement $z \in R^m$:

$$z_k = Hx_k + v_k \tag{8}$$

The $m\,x\,n$ matrix H relates the state to the measurement z_k which is constant. v_k is the measurement noise with covariance R. There are two distinct equations at each state. They are the time update (prediction) and measurement update (correction).

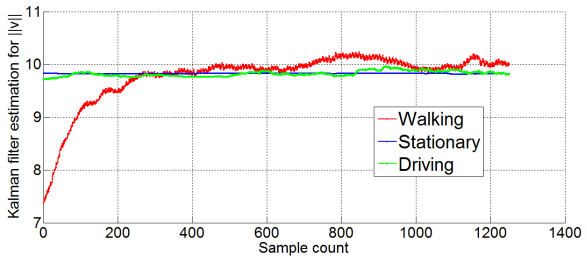


Fig. 4: Graph of magnitude of the accelerometer signal vector after applying the Kalman filter.

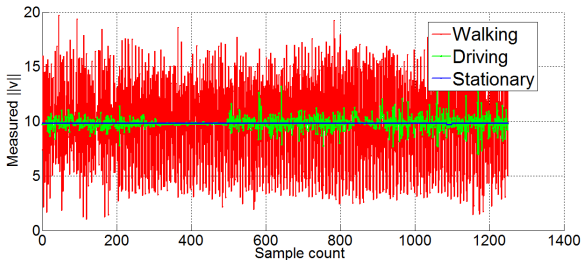


Fig. 5: Shows a graph of the $||v||$ for selected user activities.

2) 3D Accelerometer Model: The 3D accelerometer measurement is modelled as followed [7]:

$$z_k = a_k - g_k + b_k + v_{A,k} \tag{9}$$

z_k is the sensor readings at time k, a_k is the acceleration, g_k is the gravity, b_k is the offset, and $v_{A,k}$ is the observed noise. z_k is a vector in the 3D Cartesian coordinate system as followed:

$$z_k = \sqrt{(w_k^x)^2 + (w_k^y)^2 + (w_k^z)^2} \tag{10}$$

We combine the accelerometer readings by calculating the magnitude of the accelerometer signal vector z_k. We opted to apply Kalman filter directly to $||z_k||$ rather than on vector (z_k^x, z_k^y, z_k^z) because depending on the smartphone placement as the acceleration vector increases in a direction, the associated accelerometer readings grow larger along the affected axis and could be constant along the rest. We applied the Kalman filter on 1250 samples of gathered accelerometer data for the following activities: walking, stationary, and driving. Due to the accelerometer noise the filter caused historical measurements to have adverse effects on estimates. To overcome this issue of a corrupted filter, rather than applying the Kalman filter continuously we reset the filter every accelerometer 8 samples. This ensures in case of errors that only one user activity calculation is affected. Based on different on-body placements we found Kalman filtering not useful in classifying activity states. Fig's. 4 and 5 shows the results with and without applying Kalman filtering to the smartphone accelerometer data for walking, stationary and travel by car user activities. As shown even though the noise was reduced the computation features were stymied in the output required to classify between activities. Using our user activity state algorithm we found noise filtering wasn't beneficial and it introduced an unnecessary computational load.

B. Energy-efficiency

We developed an Android based application named AppResource to study energy-efficiency. AppResource calculates the average consumed resources in terms of CPU and RAM (Mb) usage of active and idle applications over a configurable time period. Table I shows the average consumed resources of LALS vs. standard applications in terms of CPU and RAM usage over a 60 second window. As shown in active usage LALS uses on average 1% CPU and 10.5 Mb.

TABLE I: CPU and RAM utilized per activity.

Activity	CPU% range	Average CPU%	RAM (Mb) range	Average RAM (Mb)
Browsing	(9,90)	45.4	(45.2,76.4)	60.5
Camera	(23,42)	31.2	(13.8,14.6)	14.2
Game	(13,24)	19.6	(13.3,13.9)	13.5
Active call	(2,13)	8.16	(25.4,26)	25.8
Music	(0,2)	2	(12.8,21.4)	19.1
Google maps	(5,67)	22.5	(38.9,40.8)	39.3
LALS	(0,2)	1	(10.1,10.5)	10.5

There are four sensing modes in the Android OS. They are normal, ui, game, and fastest. We conducted accelerometer based experiments to measure the time taken to exhaust the smartphone battery. The results show it took approximately the same time to exhaust the smartphone battery with continuous GPS location sampling (903 minutes) as compared to combined accelerometer in normal Android sensing mode and continuous GPS location sampling (892 minutes). As shown accelerometer sensing in the normal mode is energy-efficient. The energy-efficiency experiments were conducted using a Samsung Galaxy II running Android version 4.0.4 with a 1500 mAh standard battery capacity. LALS is based on the embedded smartphone accelerometer running in normal Android sensing mode.

V. Analysis

The costs for GPS, WPS, GSMPS, and combined GPS, and WPS or GSMPS (GWG) are represented as $C_{gps} = \frac{1}{C_p}, C_{wps} = \frac{1}{C_w}, C_{gsmps} = \frac{1}{C_s}$, and $C_{gwg} = \frac{1}{C_g}$ respectively, where the denominators represent the mean time to exhaust the smartphone battery. The weight factors for GPS, WPS and GSMPS are denoted by α, β, and γ respectively. The Android OS network location provider determines the location based on the availability of GSM and Wi-Fi access points. The combined frequency over time of WPS and GSMPS lookups is 1. The cost computation for LALS as compared to combinations of GPS, and WPS or GSMPS is derived as follows:

$$\begin{cases} \alpha(C_{gps}) + \beta(C_{wps}) + \gamma(C_{gsmps}) = C_{gwg} \\ \alpha \leq \beta + \gamma \\ \beta + \gamma = 1 \end{cases} \tag{11}$$

A. Cost calculation using combined GWG

The following is the total cost calculation for combined GPS, and WPS or GSMPS (GWG):

$$TC_{gwg} = \alpha(C_{gps} * T_t) + \beta(C_{wps} * T_{twps}) + \gamma(C_{gsmps} * T_{tgsmps}) \quad (12)$$

where T_t is the total running time and; T_{twps} and T_{tgsmps} are the running times for WPS and GSMPS respectively.

B. Cost calculation using GPS

The following is the total cost calculation for GPS:

$$TC_{gps} = C_{gps} * T_t \quad (13)$$

where T_t is the total running time.

C. Cost calculation using LALS

The following is the total cost calculation for LALS:

$$TC_{lals} = (C_{gps} * T_{tgps}) + (C_{wps} * T_{twps}) + (C_{gsmps} * T_{tgsmps}) \quad (14)$$

where T_{tgps}, T_{twps}, and T_{tgsmps} are the total running times for GPS, WPS, and GSMPS.

We evaluate LALS based on a real-world typical daily user activity pattern from 08:30 to 18:30 (total of 10 hours). The scenario involves: 1) Leaving home and arriving at the underground train station at 08:30, where GPS is unavailable at home; and GPS, WPS and GSMPS are unavailable at the underground train station. 2) Arriving at 09:30 to the customer site, where GPS is unavailable. 3) At 17:30 commence a return trip back home from the customer site. For continuous location estimates the scenario requires GPS for 3 hours, WPS for 6.5 hours, and GSMPS for 0.5 hours. The underlying issue is increased energy cost due to using combined GWG for continuous location updates. The LALS scheme counters this energy problem by ensuring only a minimum set of location sensors are active.

Our experiment results show the mean time to exhaust the Samsung Galaxy II smartphone with continuous location updates using GPS $\approx$ 12 hours, WPS $\approx$ 46 hours, GSMPS $\approx$ 63 hours, and combined GWG $\approx$ 11 hours. Based on the availability of either Wi-Fi or GSM access points: $\alpha = 1, \beta = .13$, and $\gamma = .87$. It should be noted that β and γ depends on the availability of either Wi-Fi or GSM access points.

Using an instantiation of equations 12, 13, and 14; $TC_{gwg} \approx 0.859$, $TC_{gps} \approx 0.833$, and $TC_{lals} \approx 0.275$ respectively. Inclusive of the accelerometer energy consumption which is negligible, applying LALS utilizes $\approx$ 30% of the smartphone battery. As shown TC_{lals} outperforms TC_{gwg} and TC_{gps} with cost savings of at least 53% in a typical situation.

D. Worst case scenario)

The worst case will occur in the absence of WPS and GSMPS; and when the user activity state is continuously in-motion. In such a case only GPS will be used for location determination. The following is the worst case cost computation for LALS.

$$\begin{aligned} TC_{gps} &\leq TC_{lals} \\ TC_{lals} &\leq TC_{gwg} \end{aligned} \quad (15)$$

VI. Conclusion

An important benchmark for location determination using smartphones is energy-efficiency. Several location based applications require continuous GPS location determination once in execution, e.g., satnav and traffic applications. LALS is particularly suited for such applications as it provides high-availability continuous location updates using smartphones. In this paper we have presented LALS, a hybrid EE-LD that combines GPS, WPS, GSMPS, and accelerometer. LALS delivers continuous location updates while activating a minimum set of smartphone location sensors based on the user activity for optimal energy consumption.

The advantages of LALS, compared to existing methods are: 1) Real-time user activity classification and state transition detection. This knowledge aids LALS in location sensor management with a 2 second delay (8 accelerometer samples). 2) Low-energy consumption due to the light-weight accelerometer data feature extraction and accelerometer sensing mode at 4 samples per second.

The location accuracy is unchanged because our main focus is on the sensor selection algorithm which manages the activation and deactivation of location sensors. In a real case, we have shown battery energy savings of about 53% using LALS as compared to using combinations of GPS and WPS or GSMPS.

Although, we focused mainly on Android based phones, we envisage that LALS can be used with any type of smartphone as the Micro-electro-mechanical systems (MEMS) specification is similar across smartphone devices.

References

[1] Y. Wang, J. Lin, M. Annavaram, Q.A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh *A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition* ACM MobiSys '09, pp. 179-192, 2009.

[2] N. Eagle *Behavioral inference across cultures: Using telephones as a cultural lens* IEEE Intelligent Systems, 23(4):62-64, 2008.

[3] M. Ibrahim and M. Youssef *A Hidden Markov Model for Localization Using Low-End GSM Cell Phones* IEEE ICC '11, pp. 1-5, 2011.

[4] S. Gaonkar, J. Li, R.R. Choudhury, L. Cox, and A. Schmidt *Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation* ACM MobiSys '08, pp. 174-186, 2008.

[5] F.B. Abdesslem, A. Phillips, and T. Henderson *Less is More: Energy-Efficient Mobile Sensing with SenseLess* ACM MobiHeld '09, pp. 61-62, 2009.

[6] I. Constandache, X. Bao, M. Azizyan, and R.R. Choudhury, *Did You See Bob? Human Localization using Mobile Phones* ACM MobiCom '10, pp. 149-160, 2010.

[7] H.J. Luinge and P.H. Veltink *Inclination Measurement of Human Movement Using a 3-D Accelerometer With Autocalibration* IEEE Transactions on Neural Systems and Rehabilitation Engineering, 12(1):112-121, 2004.

[8] A. Thomas, J. Geldmacher, J. Gotze, and E. Coersmeier *Azimuth Based Localization for Mobile Phones* Int. Conf. on Localization and GNSS (ICL-GNSS), pp. 71-76, 2011.

[9] A. Varshavsky, M.Y. Chen, E. de Lara, J. Froehlich, et.al. *Are GSM phones THE solution for localization?* 7th IEEE Workshop on Mobile Computing Systems and Applications, pp. 34-42, 2006.

[10] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao *Energy-Accuracy Trade-off for Continuous Mobile Device Location* ACM MobiSys '10, pp. 285-298, 2010.

[11] T.O. Oshin, S. Poslad, and A. Ma, *A Method to Evaluate the Energy-Efficiency of Wide-Area Location Determination Techniques Used by Smartphones* Proc. of the 15th IEEE Int'l Conf. on Computational Science and Engineering (CSE), pp. 326-333, 2012.