

# Energy-efficient Localization Via Personal Mobility Profiling

Ionut Constandache<sup>1</sup>, Shravan Gaonkar<sup>2</sup>, Matt Saylor<sup>1</sup>,  
Romit Roy Choudhury<sup>1</sup>, and Landon Cox<sup>1</sup>

<sup>1</sup> Duke University, Durham, NC, USA,  
{ionut, sayler, lpcox}@cs.duke.edu, romit@ee.duke.edu,  
<sup>2</sup> University of Illinois at Urbana Champaign, IL, USA  
gaonkar@ieee.org

**Abstract.** Location based services are on the rise, many of which assume GPS based localization. Unfortunately, GPS incurs an unacceptable energy cost that can reduce the phone’s battery life to less than ten hours. Alternate localization technology, based on WiFi or GSM, improve battery life at the expense of localization accuracy. This paper quantifies this important tradeoff that underlies a wide range of emerging applications. To address this tradeoff, we show that humans can be profiled based on their mobility patterns, and such profiles can be effective for location prediction. Prediction reduces the energy consumption due to continuous localization. Driven by measurements from Nokia N95 phones, we develop an energy-efficient localization framework called *EnLoc*. Evaluation on real user traces demonstrates the possibility of achieving good localization accuracy for a realistic energy budget.

## 1 Introduction

Mobile phones are becoming a powerful platform for sensing, sharing, and querying people-centric information. A variety of applications are on the rise, many of which utilize *location* to express the context of information. Examples include geotagging, geocasting, asset tracking, context-aware search, location-specific advertisements, and visualization [1, 2, 3]. Most of these location based applications (LBAs) assume GPS capabilities. While GPS offers good location accuracy of around 10m, it incurs a serious energy cost that can drain a fully charged phone battery in 8.5 hours [1]. We make a few observations in light of this energy-accuracy profile.

1. In real life, the phone battery must be shared with several other needs, including voice calls, emails, pictures, SMS, and an emergency reserve. The energy budget available for localization alone is a small fraction of the battery capacity. If this fraction is assumed to be 25%, *continuous* GPS localization will sustain for less than 2.5 hours.
2. One may argue that continuous GPS localization is not necessary – GPS can be activated only on demand. While this may suffice for some services

(such as geo-tagging a photo), a large number of emerging applications are demanding frequent location updates from users. For example, GeoLife [4] plans to track a phone and display shopping lists when the phone-user is near a grocery store. Micro-Blog [1] proposes to query users in a desired region for participatory sensing applications [3]. TrafficSense [5], Pothole Patrol [6] and Nericell [7] require phones to continuously report accelerometer readings to better estimate traffic/road conditions. All these applications rely on the feasibility of continuous localization over reasonably long time scales.

3. Continuous localization over long time scales results in higher *average* error. Given an energy budget of  $K$  GPS readings, and a duration of  $T > K$  time units,  $(T - K)$  time units cannot be assigned location readings. Locations can only be estimated at these unassigned times, and hence, are likely to be more erroneous than actual GPS readings. When averaged over all actual and estimated readings, the *average localization error* proves to be higher than the instantaneous error (around 10m for GPS). As the energy budget decreases, the difference between average and instantaneous error increases.
4. Alternate WiFi and GSM based localizations are not obvious replacements to GPS. While these schemes are less energy-hungry, they incur higher (instantaneous) localization error (around 40m and 400m respectively). This permits greater number of location readings, each of which is less accurate. Whether this results in lower *average error* than that of a few, but accurate, GPS readings, is an open question. Whether an optimal combination of GPS/WiFi/GSM based localization meets application-demands, is also an unexplored research direction.

This paper investigates the space of energy-efficient localization for mobile phones. We quantify the energy-accuracy tradeoff through measurements on Nokia N95 phones. We formulate a theoretical framework using dynamic programming (DP), that derives the optimal localization accuracy under a given energy budget. When results show that the theoretical optimal with basic GPS/WiFi/GSM technologies may not suffice for high-accuracy applications, we explore usefulness of predictions. Habitual movements of humans, and their similarity to statistical behavior of large populations, are potential opportunities. We incorporate predictions in our offline DP and design online heuristics for use on mobile phones. Performance results, comparing the optimal to the online heuristics, confirm the feasibility of achieving localization in the order of 25m, over a day's energy budget. Certain limitations will have to be addressed before our proposed system, EnLoc, can be deployed for public use – we discuss these limitations, and indicate directions of future work. Our main contributions can be summarized as follows.

**Identifying the space of energy-accuracy tradeoff.** We quantify the tradeoff with measurements on Nokia N95 phones.

**Analysis of the optimal localization accuracy for a given energy budget.** For a given mobility trace, an offline dynamic program computes the maximum accuracy achievable through GPS, WiFi, GSM, or combinations thereof.

**Exploiting habitual activity of individuals, and behavior of populations, to predict location.** Predictions incorporated into the DP offer optimal solutions. Online heuristics permit energy-efficient localization in real time.

**Evaluation of heuristics in real life situations.** A wireless map of the UIUC campus is created through war-driving, and real mobility traces derived from student phones. Performance of heuristics is compared with the optimal using a custom trace-based simulator.

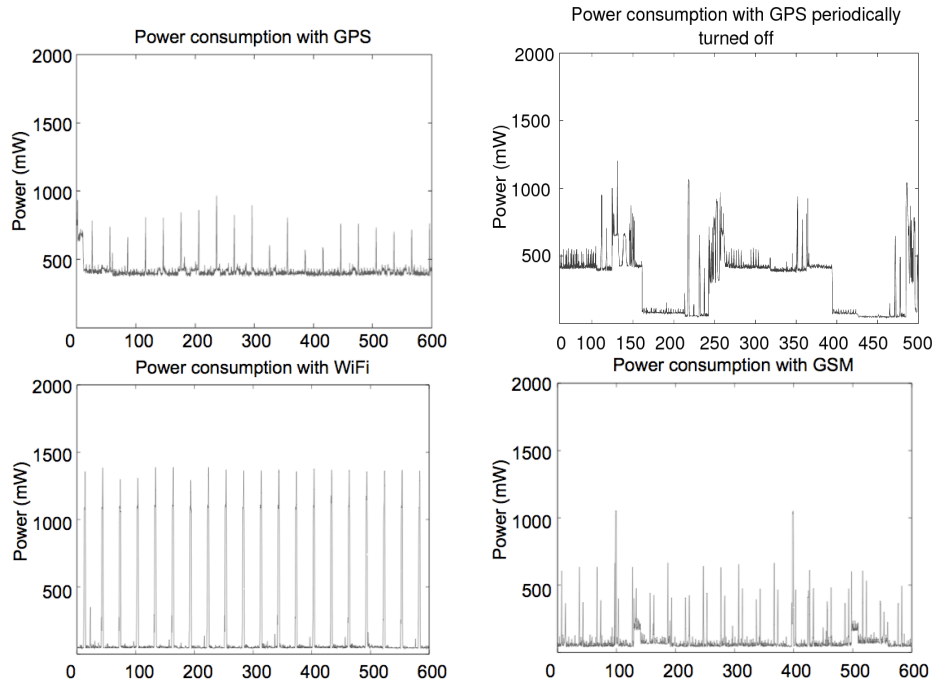
The rest of the paper expands on each of the contributions.

## 2 Motivation

We used the Nokia Energy Profiler [8] to measure fine-grained power consumption in Nokia N95 phones. With this monitor running in the background, we performed GPS, WiFi, and GSM based localization. This section reports the results recorded, and highlights key observations that motivate EnLoc. However, we first provide a brief background on WiFi and GSM localization techniques, originally proposed in [9].

### Localization Using WiFi and GSM

As a benchmark, we measured localization error with on-phone GPS hardware, and found an average accuracy of 10m in clear-sky, outdoor conditions. GPS was mostly unavailable indoors. To circumvent GPS hardware and its problems indoors, project Place Lab [9, 10] proposed interesting alternates to GPS localization. Specifically, authors create a wireless map of a region by war-driving in the area. The wireless map is composed of sampled GPS locations, WiFi access points and GSM towers audible at these locations, and their corresponding signal strengths. This wireless map is then distributed to phones. When a phone travels through the mapped area, it estimates its own location by matching its list of audible WiFi APs to the wireless map. Several matching schemes have been proposed, including *Centroid* Computation, *Signal Strength* based estimation, and *Fingerprinting*. Place Lab experiments in Seattle region exhibit a median positioning error of 13 to 40m with WiFi, and around 94 to 196m with GSM. When performed in Champaign, IL, and Durham, NC, the errors were higher. Specifically, WiFi accuracy ranged between 25 to 40m, while GSM ranged between 300 to 400m. The increase in GSM error was due to a smaller number of cells and using a single GSM provider.



**Fig. 1.** Power consumption in mW: (a) GPS localization every 30 seconds. (b) GPS periodically turned off. (c) WiFi localization every 30 seconds. (d) GSM localization every 30 seconds.

### Energy Measurements for GPS, WiFi, and GSM

We measured the energy consumption on Nokia N95 phones for each localization scheme. For this, we charged the phone to its full battery capacity, and turned off all applications and communication technologies. Then, we invoked the energy monitoring program and turned on only the location sensor we aimed to measure. Our program probed the location sensor at a chosen interval,  $T_{probe}$ , that varied from 30 to 300 seconds.

Figure 1(a) shows GPS power consumptions as a function of time, for  $T_{probe} = 30$  seconds. We see periodic spikes on top of a baseline power consumption at approximately 400 mW. The spikes correspond to the periodic probes which trigger a sensor read and a data write operation into the phone's file system. The baseline corresponds to the power consumed by the GPS receiver. To prove this, we periodically turned off all sensors, including GPS. Results in Figure 1(b) show a baseline power consumption of around 55 mW when all sensors are turned off.

Similar measurements for WiFi and GSM are reported in Figure 1(c) and (d). Observe that while the baseline power consumption for WiFi is low (55 mW),

it exhibits a high spike (of around 1300 mW) at every probe. GSM based samples also exhibit similar characteristics, however, their power consumption is less.

Based on these power measurements, we computed the total battery lifetime for each sensor as if the entire battery capacity is used for localization alone. The results are presented in table 1 together with the localization accuracy of each sensor. We record battery lifetimes of around 10, 40, and 60 hours, for GPS, WiFi and GSM, respectively. When viewed against corresponding localization accuracies of 10, 40, and 400 meters, the energy-accuracy tradeoff is evident.

Sensor	GPS	WiFi	GSM
Lifetime (h)	10	40	60
Loc. error (m)	10	40	400

**Table 1.** Energy-Accuracy Tradeoff

### 3 EnLoc: Framework Design

#### 3.1 Average Location Error (ALE)

The energy-efficient localization problem can be defined as follows. Given an energy budget  $E_B$ , time duration  $T$ , and a set of localization schemes  $S$ , design a strategy that will minimize the average localization error over the entire time,  $T$ . Formally, denote  $L_{reported}(t_i)$  and  $L_{actual}(t_i)$  to be the reported and actual locations of the phone at time  $t_i$ . Assuming  $T$  discrete time-points in the entire time duration, the average localization error  $\delta_{avg}$ , is defined as

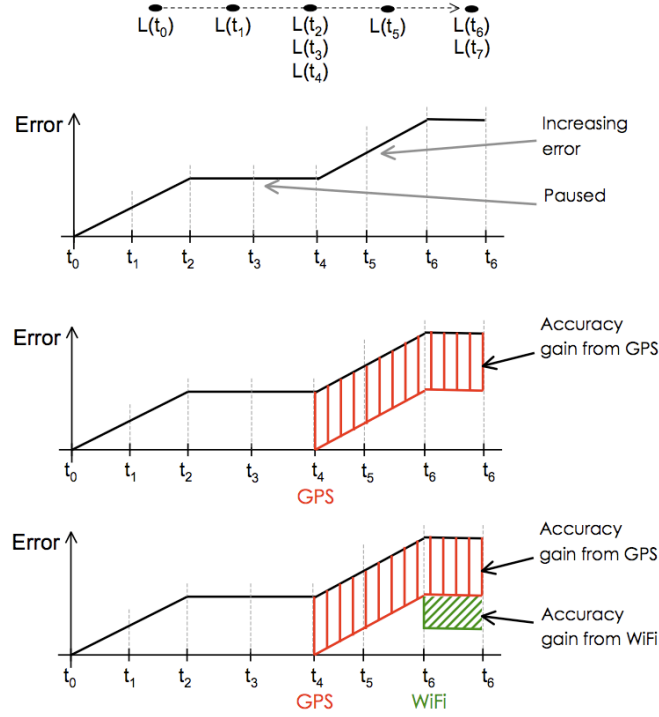
$$\delta_{avg} = \sum_{i=1}^T \left( \frac{L_{reported}(t_i) - L_{actual}(t_i)}{T} \right)$$

where  $L_{reported}(t_i) - L_{actual}(t_i)$  is the distance between the reported location and the actual location at time  $t_i$ .

Assuming GPS to be the ground truth, a GPS reading at time  $t_j$  implies that  $L_{reported}(t_j)$  is same as  $L_{actual}(t_j)$ . Similarly, a WiFi reading at  $t_j$  implies that  $L_{reported}(t_j) - L_{actual}(t_j)$  is in the order of 40m. The problem then is to minimize  $\delta_{avg}$  for a given energy budget. We model this energy-accuracy tradeoff as an optimization problem.

#### 3.2 Problem Formulation

Our goal is to determine a schedule with which the location sensors should be triggered such that the ALE is minimized under a given energy budget. We



**Fig. 2.** Error vs time model of physical movement.

envison a schedule to be a set of time instants,  $\{t_1, t_2, t_3, \dots\}$ , and corresponding sensors  $\{s_1, s_2, s_3, \dots\}$ , where  $s_i \in [GPS, WiFi, GSM]$ . The optimal schedule triggers a reading of sensor  $s_i$  at time  $t_i$  to minimize ALE. We elaborate with an example.

Consider the case in which a person begins moving at time  $t_0$ , from an initially sampled location,  $L(t_0)$ . Assume that until the next location reading, this location is periodically reported as the location of the phone. As the person moves away from the original location, the location error increases with time. When the person stops moving, the error remains static; when the person moves towards the  $L(t_0)$ , the error starts decreasing. Without loss of generalization, Figure 2(a) and (b) show an example trajectory in discrete time, and the corresponding error versus time graph. The phone moves away for 2 time steps, pauses for 2 time steps, moves further away for another two time steps, and then pauses again for one time step. The total error due to this movement can be computed as the sum of the 7 vertical lines, dropped from the error curve to the  $x$  axis. For easier explanation, let us assume this is the area under the curve.

Now, if the user takes a GPS reading at time  $t_4$ , the corresponding error graph is shown in Figure 2(c). Time  $t_4$  onwards, the error increases as if location  $L(t_4)$  is the new (known) starting location. If no other readings are taken, the reduction

in total error due to the GPS reading is the shaded area labeled “Accuracy gain from GPS”. If a WiFi reading is scheduled at time  $t_6$ , the instantaneous error, unlike GPS, *does not reach zero* – this results in error reduction shown by the diagonally striped rectangle. In general, the reduction depends on the time-point at which the reading is taken, as well as the sensor used. Additional location readings at subsequent time-points offer further error reductions, at the cost of expended energy. Hence, the problem of energy-efficient localization translates to finding the appropriate time-points and sensors that must be scheduled to minimize the localization error. More formally, for a given energy budget,  $E_B$ , energy consumption per-sensor,  $e_{s_i}$ , and a mobility trace,  $\tau$ , the algorithm must produce a  $\langle time_i, sensor_i \rangle$  schedule, that minimizes the total area under the error curve. We approach this problem using dynamic programming. Related problems in the context of satellite communication and user state recognition have been investigated in [11] and [12].

### 3.3 Dynamic Programming

The dynamic program computes the minimum error accumulated till the end of a given trace, and the corresponding  $\langle t_i, s_i \rangle$  schedule that achieves it. For this, the accumulated error is expressed as a sum of (1) error accumulated until some time  $j$  when the last location reading was taken, and (2) error accumulated from  $j$  till the final time instant,  $T$ . Now, any accumulated error between two consecutive location readings at time  $i$  and  $j$  can be computed as  $\sum_{k=i+1}^{j-1} (L_{reported}(k) - L_{actual}(k))$ , where  $L_{reported}(k) - L_{actual}(k)$  is the distance between the estimated/reported location, and the actual location. Without loss of generality, assume that  $L_{reported}$  returns the last known location. Thus, the accumulated error between  $i$  and  $j$  is  $\sum_{k=i+1}^{j-1} (L_{actual}(i) - L_{actual}(k))$ . Since this accumulated error is a function of the sensor used at  $i$  (recall Figure 2), we define three matrices  $E_{GPS}[i][j]$ ,  $E_{GSM}[i][j]$ ,  $E_{WiFi}[i][j]$ . Each matrix corresponds to the localization sensor in the subscript, and denotes the total error from  $i$  to  $j$ , when that sensor is used at time  $i$ . The last column of these matrices gives us the error from some last reading at time  $j$  till the end of the trace at time  $T$ . Table 2 presents the notations used in the dynamic program.

For the first term, we need to minimize the accumulated error till the (last) reading at time  $j$ . This problem can in turn be formulated as finding a second last reading, say at time  $i$ , which minimizes the error until  $j$ . This can be repeated under the constraints of a maximum number of feasible readings.

Let us consider the problem of finding the minimum error accumulated till time  $j$ , assuming a GPS reading is executed at  $j$ . There must exist a previous time  $i < j$ , at which the previous reading was taken. At time  $i$ , any one of GPS, WiFi or GSM sensors could have been used. Since there is no other reading between  $i$  and  $j$ , some error accumulates from  $i$  to  $j$ . This accumulated error is one of  $E_{GPS}[i][j]$ ,  $E_{WiFi}[i][j]$  or  $E_{GSM}[i][j]$ , depending on the location sensor

<i>Notation</i>	<i>Explanation</i>
$T$	Trace length in time units
$I_{WiFi}[j],$ $I_{GSM}[j]$	Instantaneous error at time unit $j$ when the WiFi/GSM sensor is used.
$E_{GPS}[i][j],$ $E_{WiFi}[i][j],$ $E_{GSM}[i][j]$	Accumulated error between $i$ and $j$ or sum of instantaneous errors at intermediate time units, if at $i$ GPS/WiFi/GSM sensor was used and until $j$ there is no other reading.
$\min_{GPS}[i][n1][n2][n3],$ $\min_{WiFi}[i][n1][n2][n3],$ $\min_{GSM}[i][n1][n2][n3]$	Minimum accumulated error till time unit $i$ , if $i$ was the last scheduled reading and used the GPS/WiFi/GSM sensor. The minimum is achieved with $n1$ GPS, $n2$ WiFi and $n3$ GSM readings.
$totalError_{GPS}[j][n1][n2][n3]$ $totalError_{WiFi}[j][n1][n2][n3]$ $totalError_{GSM}[j][n1][n2][n3]$	Total accumulated error till the end of the trace if at time unit $j$ the GPS/WiFi/GSM sensor was used and there are no further readings till the end of the trace. In total there are $n1$ GPS, $n2$ WiFi, $n3$ GSM readings

**Table 2. Summary of notation.**

used at time  $i$ . The problem of finding the minimum error till time  $j$  translates into solving the minimum sum of two terms – (i) the minimum accumulated error till some time  $i < j$  using some sensor (GPS, WiFi or GSM), and (ii) the error accumulated between  $i$  and  $j$ . However, the minimum accumulated error till time  $i$  has 1 GPS reading less (because one GPS reading is invested in estimating the location at time  $j$ ). Since, there are three possible sensors that can be used at time  $i$ , the optimization can be formulated as:

$$\min_{GPS}[j][n1][n2][n3] = \min_i \begin{cases} \min_{GPS}[i][n1-1][n2][n3] + E_{GPS}[i][j], \\ \min_{WiFi}[i][n1-1][n2][n3] + E_{WiFi}[i][j], \\ \min_{GSM}[i][n1-1][n2][n3] + E_{GSM}[i][j] \end{cases}$$

where  $j > i$ ,  $j \geq n1 + n2 + n3$ ,  $T > j$ .

The total error till the end of the trace (if the last reading is at time  $j$ ), is the minimum error achieved till  $j$  plus the accumulated error between  $j$  and  $T$ .

$$totalError_{GPS}[j][n1][n2][n3] = \min_{GPS}[j][n1][n2][n3] + E_{GPS}[j][T]$$

Similarly, we can write the above equations for the case in which WiFi or GSM is used for reading the location at time  $j$ . However, we must add an additional term to denote the instantaneous location error at time  $j$  (recall that WiFi and GSM sensors have an instantaneous error represented as  $I_{WiFi}[j]$  and  $I_{GSM}[j]$ ). The dynamic program for a WiFi reading at time  $j$  is

$$\min_{WiFi}[j][n1][n2][n3] = \min_i \begin{cases} \min_{GPS}[i][n1][n2-1][n3] + E_{GPS}[i][j] + I_{WiFi}[j], \\ \min_{WiFi}[i][n1][n2-1][n3] + E_{WiFi}[i][j] + I_{WiFi}[j], \\ \min_{GSM}[i][n1][n2-1][n3] + E_{GSM}[i][j] + I_{WiFi}[j] \end{cases}$$



and that for a GSM reading at  $j$  is

$$\min_i \begin{cases} \min_{GSM}[j][n1][n2][n3] = \\ \min_{GPS}[i][n1][n2][n3-1] + E_{GPS}[i][j] + I_{GSM}[j], \\ \min_{WiFi}[i][n1][n2][n3-1] + E_{WiFi}[i][j] + I_{GSM}[j] \\ \min_{GSM}[i][n1][n2][n3-1] + E_{GSM}[i][j] + I_{GSM}[j] \end{cases}$$

The total error till the end of the trace, if the last reading at time unit  $j$  uses the WiFi sensor, is:

$$totalError_{WiFi}[j][n1][n2][n3] = \min_{WiFi}[j][n1][n2][n3] + E_{WiFi}[j][T]$$

while for the GSM sensor, total error can be expressed as:

$$totalError_{GSM}[j][n1][n2][n3] = \min_{GSM}[j][n1][n2][n3] + E_{GSM}[j][T]$$

The final solution for the minimum accumulated error till the end of the trace, given a maximum of  $n1$  GPS readings,  $n2$  WiFi readings and  $n3$  GSM readings, is:

$$Sol = \min_{j,n1,n2,n3} \begin{cases} totalError_{GPS}[j][n1][n2][n3] \\ totalError_{WiFi}[j][n1][n2][n3] \\ totalError_{GSM}[j][n1][n2][n3] \end{cases}$$

The  $\langle t_i, s_i \rangle$  schedule can be retrieved by keeping track of timing and sensor readings for each minimization sub-problem. In this way, the entire solution sequence can be retrieved and the optimal schedule computed.

### 3.4 Optimal Localization Error

To obtain the best localization accuracy, we executed the DP on mobility traces from the UIUC campus. We first war-drove the campus [9] and generated a wireless map of the area (as described in Section 2). Then, we distributed phones to students to gather mobility traces. A custom simulator integrated the traces with the wireless map, and executed the dynamic program. We briefly present a few relevant details.

#### Collecting Mobility Traces

Our tracing software (installed on mobile phones) sampled the GPS location, as well as the WiFi and GSM base station IDs every 30 seconds. For purposes of anonymity, we requested the users to turn off their mobile phones when they did not wish to be tracked. The traces were analyzed offline. For each sample, the set of WiFi APs were obtained and the corresponding locations extracted from the (war-driven) wireless map. A centroid of all the WiFi AP locations was computed and declared to be the instantaneous WiFi-based location of the phone. Since the GPS location for this time instant is known, the instantaneous error from WiFi localization can be computed. This instantaneous error (say

at time,  $t_i$ ) populates the  $I_{WiFi}[i]$  value in the dynamic program formulation. The GSM localization error is identically computed, and used to populate the  $I_{GSM}[.]$  matrix. In summary, for a given mobility trace, the exact errors due to WiFi and GSM localization is known for each time instance<sup>1</sup>. These errors are fed into the DP for error computation. The energy budget is specified as 25% of the phone battery, while the required duration of operation is 24 hours.

We evaluate four schemes, namely, *Optimal GPS*, *Optimal WiFi*, *Optimal GSM*, and *Optimal Combined*. As the name suggests, *Optimal GPS* corresponds to the minimum *average localization error* achieved when only GPS readings are used. In the interest of space, Table 3 reports results from three representative mobility traces. Observe that *Opt WiFi* outperforms *Opt GPS* indicating that greater number of less accurate readings is better for mobile phone localization. Also, *Opt Combined* outperforms the others, and is close to *Opt WiFi* in many of the traces. However, it is surprising that even the offline optimal error (with knowledge of the entire trace) was typically more than 60m. Online versions of these schemes, that do not have the entire trace, will naturally perform worse. This warrants enrichments that would improve the limits of energy-constrained localization. We observe that “reporting the last known location between location readings” is a source of inefficiency, and seek solutions that exploit mobility prediction.

	OptGPS	OptWiFi	OptGSM	OptComb
Trace 1	164.999	78.52	352.909	78.52
Trace 2	105.35	75.16	327.116	58.66
Trace 3	125.848	62.134	370.621	62.134

**Table 3.** Optimal performance for different traces

### 3.5 Prediction Opportunity

The error model discussed thus far assumes that between two successive location readings,  $L(t_i)$  and  $L(t_j)$ , the location reported is the last known location,  $L(t_i)$ . In reality, human behavior/mobility is amenable to prediction [13, 14, 15, 16]. Driving on straight highways, turns on one way streets, habitual office hours, are examples of prediction opportunities. EnLoc aims to take advantage of them.

**Simple Linear Predictor:** We begin by considering a basic linear predictor. The location of a phone at time  $t_k$ , denoted  $L(t_k)$ , can be a linear extrapolation of the two previously sampled locations,  $L(t_i)$  and  $L(t_j)$ . This can be effective when a phone moves on a straight road. However, if the phone’s movement is not straight, or if  $L(t_i)$  and  $L(t_j)$  were highly erroneous, linear prediction may be unsuitable. Therefore, the performance of linear predictors may not be consistent, and needs evaluation. We have modified the DP to incorporate linear prediction; we omit the formal details in the interest of space.

<sup>1</sup> We assume that GPS is the ground truth.

**Human Mobility Patterns:** While linear prediction is a generalized approach, capturing individual human behavior may facilitate better predictions [14, 15]. The intuition is that humans have habitual activity patterns, and sampling the activity at a few uncertainty points may be sufficient for predicting the rest. For instance, given that a person goes to lunch at either 12:00pm, 12:50pm, or 1:00pm, the phone may trigger GPS readings just after these times. Learning that the person has started out for lunch, her subsequent locations can be predicted (i.e., locations along the habitual path from office to the cafe). In the next section, we augment EnLoc with such human-tailored predictions. Problems arise when the person deviates from her habitual pattern.

**Deviations:** To cope with deviations from habitual paths, we hypothesize that statistical behavior of large populations provide useful hints. At a traffic intersection, knowing that most of the vehicles take a left turn can be valuable for prediction. In general, if a “probability map” can be generated for a given area, an individual’s mobility in that area can be partially predicted. We describe the design of such a probability map later. However, assuming we have such a probability map, we are able to extend our DP to extract the optimal localization schedule for a given trace. Intuitively, the DP is expected to schedule location readings at points where the individual’s actions disagree with the population behavior. Where the behavior agrees with the majority of the population, the DP can save energy.

## 4 EnLoc: System Solution

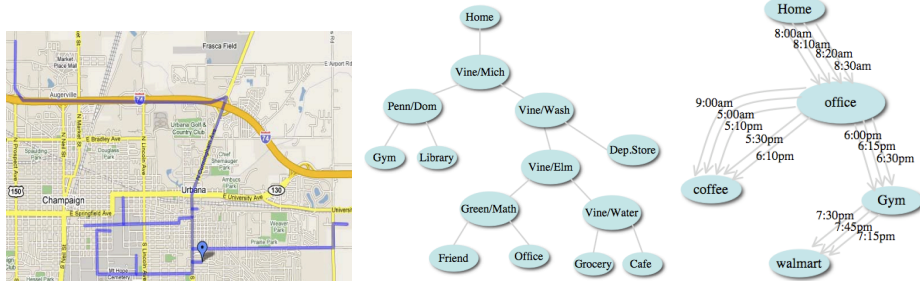
This section describes a complete system solution, EnLoc, for performing energy-efficient localization on mobile phones. The solution exploits both habitual mobility patterns and population-driven probability maps. EnLoc is an online solution, and unlike the DP, does not assume knowledge of the user’s entire trace. Results presented in Section 5 compare EnLoc with the offline optimal schemes.

### Exploiting Habitual Mobility

A study with 100,000 people has shown that individuals exhibit habitual space-time movements, with reasonably small variation [13]. If designed carefully, a few location samples may be sufficient to track the location/movement of the person over long time scales. To understand this better, we collected GPS-based mobility traces of several people over a month, and plotted them over Google Maps. Figure 3(a) shows a simplified example.

One may envision the Google maps plot as a tree, with branches at certain points – we call this the *logical mobility tree* (LMT)<sup>2</sup>. The vertices of this tree

<sup>2</sup> We are aware that the traces can very well form a graph, however, envisioning this as a tree facilitates easier explanation. The ideas we propose are general to *logical mobility graphs*.



**Fig. 3.** (a) An anonymous user’s movements over two weeks. (b) A spatial logical mobility tree. (c) A space-time logical mobility tree.

are the branching points on the person’s actual mobility paths. While tracking a phone along a path on the tree, uncertainty arises at these branching points, and hence, the nodes of the LMT are also called *uncertainty points*. The edges of the LMT represent physical traces that connect consecutive uncertainty points. Each edge is associated with (1) the starting time of that physical movement, (2) the average velocity on that path, and (3) the duration of travel on that physical path. Figure 3(b) shows the LMT corresponding to the physical mobility in Figure 3(a).

Our key idea is to schedule location readings right after the uncertainty points on the LMT. Such a location reading will resolve the uncertainty since the phone will be placed in one of the paths emanating from that uncertainty point. Thus, the phone’s location can be reasonably predicted until it encounters the next uncertainty point, at which time, another location reading will be necessary. For instance, a location reading at the Vine/Mich intersection can tell whether the person is headed towards the Penn/Dom intersection, or the Vine/Wash intersection. Of course, problems arise due to time variations.

Observe that the LMT in Figure 3(b) is a spatial representation of a person’s mobility profile. In reality, a person traverses the same edge on the LMT at many different times (e.g., one may leave for office at different times between 8:00 and 9:00am). Each of these possibilities translates into a distinct edge in the LMT. Figure 3(c) shows a hand-constructed example of such a space-time LMT representation. To accurately know when the phone leaves a particular node of the LMT, EnLoc will need to sample all the edges emanating from that node. Since a person can start moving across a large number of time points (i.e., a large number of edges), the energy budget may not permit sampling at all these times. Only a fraction of the emanating edges will need to be sampled. EnLoc designs a heuristic to sample a subset of the edges branching out of a node. We explain this with the example of Figure 3(c), which is not derived from the actual mobility trace.

Assume that current time is 8:00am, and the phone is located at home,  $H$ . Also assume that the remaining energy budget is  $B_{remaining}$ . The heuristic begins by identifying all the paths from  $H$  to the leaves of the tree, i.e.,  $P_1 = home \Rightarrow coffee$ , and  $P_2 = home \Rightarrow walmart$ . Then, the number of location readings  $N_i$ , necessary to track the phone with certainty, is computed for each path  $P_i$ . Thus  $N_1 = 4 + 6 = 10$  readings. Observe that 4 readings are necessary to track the phone leaving Home, and 6 readings for going from Office to Coffee. These 6 readings include the 5 edges from Office to Coffee (the latest being 6:10pm), as well as the 6:00pm edge from Office to Gym. If the 6:00pm edge is not included, EnLoc may not know if the phone has started moving towards the Gym. Similarly,  $N_2 = 4 + 8 + 3 = 15$  readings. Now, the heuristic computes  $M = \max(N_i)$ , a pessimistic estimate of the number of readings necessary in the future;  $M = 15$  for this example. Then, the heuristic computes  $F = \frac{e_H}{M}$ , where  $e_H$  is the number of emanating edges from Home. In this example,  $F = \frac{4}{15}$ . The phone is allocated  $F \times B_{remaining}$  amount of energy for detecting its departure from home. Assuming  $B_{remaining}$  is 10, there are approximately 2 GPS readings available. The heuristic randomly chooses 2 time-points out of the 4, and samples the phone's location. Once that phone is found to be on one of the paths going out of Home, the heuristic predicts the phone's location based on the habitual velocity on that edge. At the next uncertainty point, the phone recomputes  $F$  using the same scheme above. The overall system is reset at night when the phone is plugged into the power outlet.

#### Addressing Deviation from Habits

Users may deviate from their habitual paths, and the above scheme will not be able to predict their locations. Even though deviations are not the common case, they are important because several applications may be triggered due to deviation. For instance, micro-blogging [1] may be more active when people go for vacations; location-specific information may be necessary when people are driving down untraveled paths. EnLoc addresses the case of deviation – the main idea is to exploit mobility of large populations as an indicator of the individual's mobility. Consider a person approaching a traffic intersection from Street A. Since the person has not visited this street in the past, it is difficult to predict how she will behave at the imminent intersection. Now, if a large fraction of the population is known to take a left turn onto Street B, then the person's movement can be guessed accordingly. EnLoc creates, and takes advantage of, population-driven mobility maps. The details follow.

EnLoc detects a deviation when a scheduled location reading discovers the phone in an unexpected location. At this time, EnLoc switches to the *Deviation Mode* of operation. In this mode, the residual energy budget is divided into equally spaced GPS readings across the remainder of the day. If the current time is 5:00pm, the remaining energy budget is 48 GPS readings, and there are 6 hours left before a habitual battery recharge (say at 11:00pm), EnLoc schedules 8 equally spaced readings per hour. Now, once the first location sample has



Fig. 4. UIUC campus intersection with associated probability matrix

been obtained, say  $L(t_1)$ , EnLoc uses the population activity map to predict the phone’s movement. The velocity is estimated from the activity map, as well as turns at different intersections. Incorrect predictions obviously incur location error. The error accumulates until the next reading at time  $t_2$ , at which point EnLoc starts a new prediction using location  $L(t_2)$  as the starting point. Of course, this heuristic needs the population-driven activity map for a given area. We created such a map of the UIUC campus, described next.

Without loss of generalization, let us consider 4-way traffic intersections. EnLoc computes 4 probabilities for each intersection, i.e., an user entering the intersection from Street A, either turns left, turns right, continues straight, or takes a U-turn. One may envision this as a  $4 \times 4$  matrix, where element  $ij$  denotes the probability that the user entering street  $i$  exits through street  $j$ . Our first approach towards creating this matrix was to deploy sound/vibration sensors at traffic intersections, and count passing vehicles through simple signal processing. When this proved complicated, we adopted a much simpler approach at the expense of some inaccuracy.

On Google maps, we identified all segments of roads that border the UIUC campus. Further, we identified roads that intersect the bordering roads, and enter the campus. We call these feeder roads. Observe that all vehicles must enter the university campus through one of these feeder roads. We also identified all parking lots within the campus, their capacities, and the total number of active parking passes in the university. Now, we simulated vehicles that enter the campus through a feeder road, and drives to a pre-specified parking lot. The pre-specified parking lot is randomly chosen from the distribution of parking lot sizes (i.e., proportionally more cars are destined to bigger lots). For each vehicle, we obtained its driving direction through Google Maps APIs [17], and parsed it to extract the vehicle’s movement at each traffic intersection (i.e., left/right/forward/backward). Simulations of thousands of vehicles produces the probability matrix for each intersection. Figure 4 shows the intersections on the UIUC campus, and a probability matrix for one of them. We expect the

probability map to be installed in phones, and thereby used for online prediction and localization.

## 5 Evaluation

We evaluate EnLoc using realistic traces collected from mobile phones of UIUC students. Traces were processed offline to extract their time durations, GPS locations (every 30 seconds), as well as the WiFi APs and GSM towers overheard along the paths. Trace segments located within the war-driven campus were processed to obtain the WiFi and GSM localization errors for every 30 second time points (we assume GPS is ground truth). Further, at each traffic intersection, the mobility of the phone was predicted based on the maximum probability at the intersection. The error was computed whenever the prediction was inconsistent with the actual user’s movement. All the errors were systematically incorporated into our custom simulator (which includes the dynamic program module). The energy budget for localization was assumed to be 25% of the battery capacity, while the required duration of operation was 24 hours.

An ideal evaluation of EnLoc should characterize the average localization error (ALE) over a person’s complete mobility pattern (i.e., habitual and deviant paths). However, since the deviations in our traces extended far beyond the (war-driven, probability-mapped) UIUC campus, we were unable to perform an ideal evaluation. Instead, we first evaluated several trace segments confined within the UIUC campus (we pretend these are the deviant paths). Then, we evaluated the *habitual mobility profile*-based approach by pruning the deviations from a user’s LMT. We believe that in reality, the localization error will be close to the average of these two cases.

### Deviant Paths

Table 4 and 3 describe the different schemes examined in this section. Figure 5(a) reports the optimal localization error averaged over all mobility traces within the UIUC campus. Figure 5(c) and (e) show optimal results from two representative traces. We make the following observations from these graphs.

Opt-LP-GPS	Optimal with GPS + Linear Prediction
Opt-LP-WiFi	Optimal with WiFi + Linear Prediction
Opt-LP-GSM	Optimal with GSM + Linear Prediction
Opt-LP-Comb	Optimal with Combined + Linear Prediction
Opt-Map	Optimal using map predictor
Heu-Eq-GPS	Heuristic with Equally spaced GPS
Heu-Eq-WiFi	Heuristic with Equally spaced WiFi
Heu-Eq-GSM	Heuristic with Equally spaced GSM
EnLoc-Eq-Map	Heuristic with Equally spaced GPS on Map

**Table 4.** Optimal and heuristic schemes

(1) As mentioned earlier, OptWiFi consistently outperforms OptGPS. Evidently, this trend holds for the case of linear prediction as well. Opt-LP-WiFi averages an error of 28.94m, in comparison to 36.06m from Opt-LP-GPS. We conclude that, when scheduled carefully, WiFi offers better energy-efficient localization than GPS/GSM.

(2) Linear prediction performs well even for mobility traces that take frequent turns. This may appear surprising because linear prediction is expected to yield higher errors when an user has taken a right-angle turn. Examination of the optimal schedule for “manhattan type” traces offered useful insights. When the distances between consecutive turns were short, the linear predictor approximated the movement with a straight line cutting through the trace. The error is not large, as evident from Figure 6(a). On the contrary, if the trace has long stretches of straight lines, the Opt-LP scheme naturally predicts correctly (see Figure 6(b)).

(3) When using probability maps, the optimal ALE proves to be small. This is because the number of mis-predictions (at the intersections) are typically fewer than the number of location readings permitted by the budget. As a result, Opt-Map schedules a location reading wherever there is a mis-prediction. Of course, we assumed that the velocity of the phone can be perfectly predicted, and hence, errors arise only after mis-predictions. In reality, some errors will accumulate as a result of inaccurate velocity prediction. We plan to address this issue as a part of our future work, potentially using phone accelerometers to predict the velocity.

Figure 5(b)(d)(e) present the performance of online heuristics. Consistent with our earlier observation, Heu-Eq-WiFi outperforms both Heu-Eq-GPS and Heu-Eq-GSM. However, EnLoc-Eq-Map outperforms Heu-Eq-WiFi, except in rare occasions (such as in Trace B). The reason is as follows. In trace B, the phone encounters a road block, and takes a detour from its natural path towards the destination. As a result, it takes turns that are inconsistent with the map predictions. Since this detour causes many mis-predictions, the budgeted number of readings are not sufficient to correct them. The probability map can be updated when there are road blocks in the neighborhood, and a phone can periodically download a fresh copy of the map from a web service. In general, however, results indicate that heuristics based on probability maps are reasonably effective for achieving energy-efficient localization. We are unsure if similar results may hold for places unlike university campuses – we discuss these issues in Section 6.

Figure 7 shows the variation of ALE (averaged over all traces) for increasing energy budget. The error from optimal schemes decreases monotonically with increasing energy. The EnLoc-Eq-Map scheme follows a similar trend.



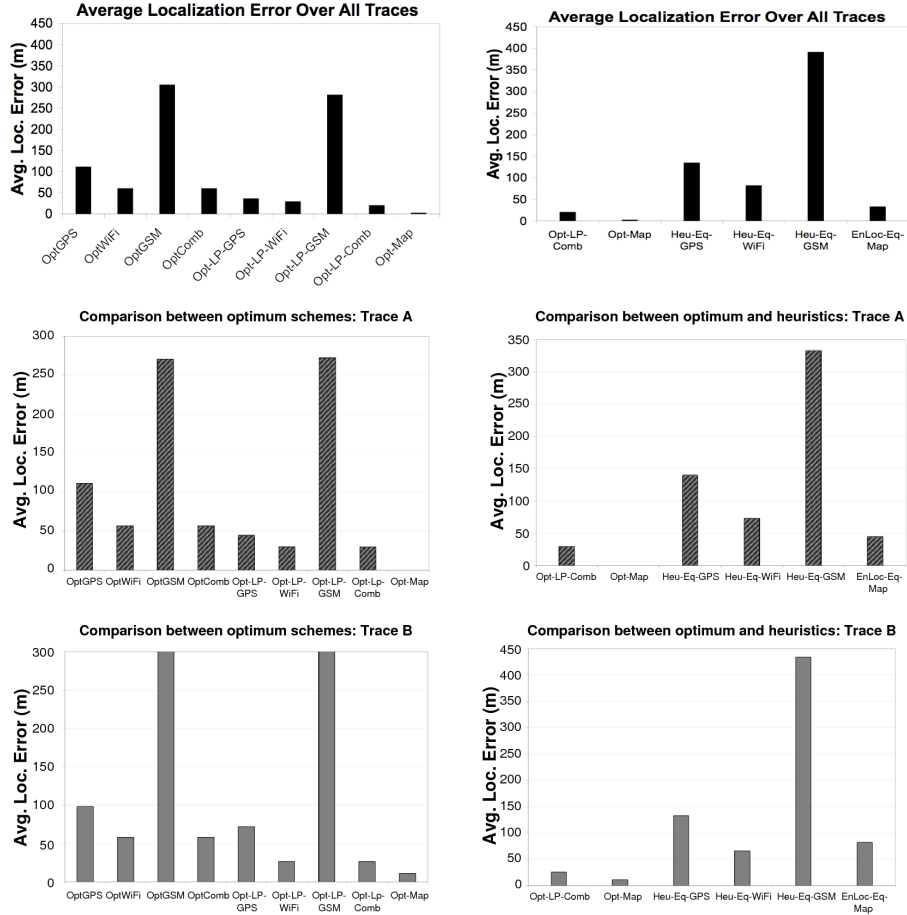


Fig. 5. Average localization error using optimal schemes: (a) Optimal over all traces, (b) Heuristic over all traces, (c) Optimal on Trace A, (d) Heuristic on Trace A, (e) Optimal on Trace B, (f) Heuristic on Trace B.

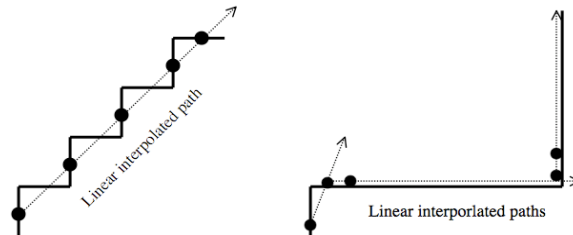


Fig. 6. Linear prediction performs well on (a) Manhattan, (b) Highway movement.

### Habitual Mobility

Next, we present the localization error when a person’s mobility profile is utilized for prediction. We use an anonymous student’s mobility profile derived from 30 days of traces. We processed her mobility traces and generated the logical

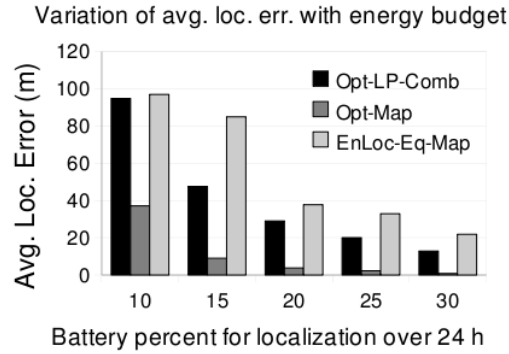


Fig. 7. ALE variation with increasing energy

mobility tree (LMT). Then, for each of the days, we executed the *Mobility Profile Heuristic*. Figure 8(a) shows that for the allocated energy budget of 25% for the entire day, the average localization error averages around 12m. This error is fairly small, particularly in view of its scalability for 24 hours. The variation of the average error with increasing energy budget is shown in Figure 8(b).

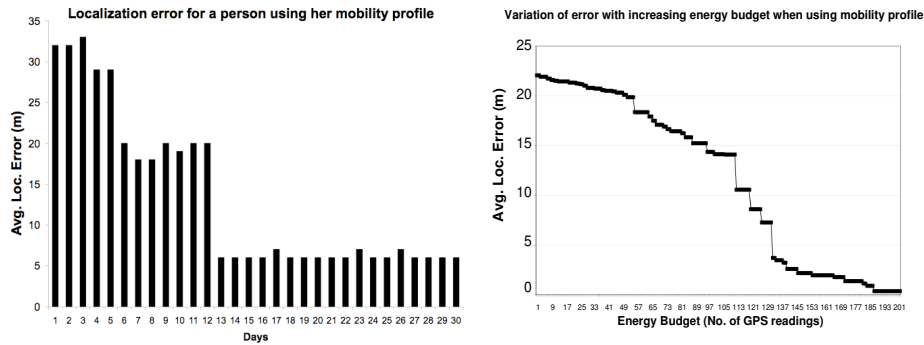


Fig. 8. (a) ALE using individual mobility profile. (b) Variation of ALE with increasing energy budget.

## 6 Limitations and Future Work

Several limitations need to be addressed before EnLoc is ready for realistic deployment. We discuss the key limitations here, and allude to potential approaches to address them.

(1) We assumed that while moving along a predicted path, the location of the phone can be accurately tracked. In reality, varying speeds, pauses, and other

random fluctuations in movement will cause this prediction to become imprecise. Our evaluation results do not account for these errors. Nonetheless, with accelerometers available on modern phones, some of these speed variations can be estimated, and used for prediction. We have investigated this in our early work in [18], and we plan to exploit it further in EnLoc.

(2) EnLoc does not proactively identify deviations from habitual paths. A person may leave office 4 hours before her habitual departure time; EnLoc will discover this after 4 hours. Techniques are necessary to quickly detect departures without investing excessive energy. A simple way could be to schedule periodic GSM readings – when the phone identifies that the GSM tower IDs have changed at an usual time, it can trigger the *deviation mode*. Accelerometers may also be effective. If the phone moves for more than a threshold duration at an unexpected time, EnLoc may suspect deviation, and schedule a GPS reading. We are investigating these possibilities in our ongoing work.

(3) Probability maps may be harder to generate for places unlike university campuses (say, a town, city, downtowns). Location updates gathered over time, from many mobile phones, can help in bootstrapping. Statistics from transportation departments, if available, can also be useful. We plan to investigate general ways of creating probability maps.

(4) The mobility profile used in EnLoc are those of graduate students, and may be less diverse (more predictable) than that of an average person. Thus, the reported errors in this paper may be optimistic. We plan to collect more (anonymous) traces, and evaluate EnLoc extensively. However, we believe that the algorithms/results reported in this paper validate the intuition that mobility profiling per-individual can be an effective tool for energy-efficient localization.

## 7 Conclusion

Emerging mobile phone applications will rely heavily on location technology. We show that energy can be a critical bottleneck for continuous localization, and that attempts to increase battery-life can degrade location accuracy. This paper presents a framework, EnLoc, that quantifies this tradeoff, derives optimal performance bounds, and develops practical schemes to achieve energy-efficient localization. Proposed schemes exploit prediction opportunities inherent in human behavior. Evaluations on traces from the UIUC campus show promising results. We believe that additional use of phone sensors, such as accelerometers, compasses, cameras or microphones [19, 20], may offer further benefits in localization accuracy for small energy costs.

## References

1. S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, "Micro-blog: Sharing and querying content through mobile phones and social participation," in *ACM MobiSys*, 2008.
2. S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G. S. Ahn, and A. T. Campbell, "Metrosense project: People-centric sensing at scale," in *Workshop on World-Sensor-Web*, 2006.
3. J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Workshop on World-Sensor-Web*, 2006.
4. T. Sohn, K. A. Li, G. Lee, I. E. Smith, J. Scott, and W. G. Griswold, "Place-its: A study of location-based reminders on mobile phones," in *UbiComp*, 2005.
5. J. Yoon, B. Noble, and M. Liu, "Surface street traffic estimation." in *ACM MobiSys*, 2007.
6. J. Eriksson, L. Girod, B. Hull, R. Newton, H. Balakrishnan, and S. Madden, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *ACM MobiSys*, 2008.
7. P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *ACM Sensys*, 2008.
8. Forum.Nokia.Com, "Nokia Energy Profiler," [http://www.forum.nokia.com/info/sw.nokia.com/id/324866e9-0460-4fa4-ac53-01f0c392d40f/Nokia\\_Energy\\_Profiler.html](http://www.forum.nokia.com/info/sw.nokia.com/id/324866e9-0460-4fa4-ac53-01f0c392d40f/Nokia_Energy_Profiler.html).
9. Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *ACM MobiSys*, 2005.
10. M. Chen, T. Soh, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky, "Practical metropolitan-scale positioning for gsm phones," in *UbiComp*, 2006.
11. A. C. Fu, E. Modiano, and J. N. Tsitsiklis, "Optimal energy allocation and admission control for communications satellites," *IEEE/ACM Trans. Netw.*, 2003.
12. 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," in *MobiSys*, 2009.
13. M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, 2008.
14. H. Lee, M. Wicke, B. Kusy, and L. Guibas, "Localization of mobile users using trajectory matching," in *ACM MELT*, 2008.
15. I. Burbey and T. Martin, "Predicting future locations using prediction-by-partial-match," in *ACM MELT*, 2008.
16. K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A mobility model for human walks," in *IEEE INFOCOM*, 2009.
17. "Google maps api," <http://code.google.com/apis/maps/>.
18. A. Ofstad, E. Nicholas, R. Szcodronski, and R. R. Choudhury, "Aampl: Accelerometer augmented mobile phone localization," in *ACM MELT*, 2008.
19. E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: The design, implementation and evaluation of cenceme application," in *ACM Sensys*, 2008.
20. M. Azizyan, I. Constandache, and R. R. Choudhury, "Surroundsense: Localizing mobile phones via ambience fingerprinting," in *ACM MobiCom*, 2009.