Experiences with eNav: A Low-power Vehicular Navigation System

Shaohan Hu[†], Lu Su[*], Shen Li[†], Shiguang Wang[†], Chenji Pan[†], Siyu Gu[†], Md Tanvir Al Amin[†], Hengchang Liu[‡], Suman Nath[§], Romit Roy Choudhury[†], and Tarek F. Abdelzaher[†]

[†]University of Illinois at Urbana-Champaign, [*]State University of New York at Buffalo, [‡]University of Science and Technology of China, [§]Microsoft Research

Email: {shu17, shenli3, swang83, cpan8, siyugu2, maamin2, croy, zaher}@illinois.edu,, lusu@buffalo.edu, hcliu@ustc.edu.cn, sumann@microsoft.com

ABSTRACT

This paper presents experiences with eNav, a smartphone-based vehicular GPS navigation system that has an *energy-saving* location sensing mode capable of drastically reducing navigation energy needs. Traditional navigation systems sample the phone's GPS at a fixed rate (usually around 1Hz), regardless of factors such as current vehicle speed and distance from the next navigation waypoint. This practice results in a large energy consumption and unnecessarily reduces the attainable length of a navigation session, if the phone is left unplugged. The paper investigates two questions. First, would drivers be willing to sacrifice some of the affordances of modern navigation systems in order to prolong battery life? Second, how much energy could be saved using straightforward alternative localization mechanisms, applied to complement GPS for vehicular navigation? According to a survey we conducted of 500 drivers, as much as 91% of drivers said they would like to have a vehicular navigation application with an energy saving mode. To meet this need, eNav exploits onboard accelerometers for approximate location sensing when the vehicle is sufficiently far from the next navigation waypoint (or is stopped). A user test-study of eNav shows that it results in roughly the same user experience as standard GPS navigation systems, while reducing navigation energy consumption by almost 80%. We conclude that drivers find an energy-saving mode on phone-based vehicular navigation applications desirable, even at the expense of some loss of functionality, and that significant savings can be achieved using straightforward location sensing mechanisms that avoid frequent GPS sampling.

Author Keywords

Smartphone; Dead-reckoning; Low-power; GPS; Navigation

ACM Classification Keywords

C.5.3 Computer System Implementation: Microcomputers–Portable devices

UbiComp '15, September 7–11, 2015, Osaka, Japan.

http://dx.doi.org/10.1145/2750858.2804287

INTRODUCTION

This paper describes evaluation and experiences with eNav, a smartphone-based GPS navigation system with a novel power-conserving mode. The purpose of the paper is to answer two questions. First, would drivers be willing to sacrifice some of the affordances of modern navigation systems in order to prolong the phone's battery life during navigation? Second, how much energy could be saved using straightforward alternative localization mechanisms, applied to complement GPS in vehicular navigation? We show that drivers indeed find the energy-saving mode desirable. Furthermore, a significant amount of energy can be saved. The general intuition why energy-saving vehicular navigation is effective is that location estimates do *not* have to be accurate at all times for navigation errors to be prevented. Rather, it is fine to have inaccurate location measurements, for example, when the vehicle is far away from the next navigation waypoint (We define a *navigation waypoint* as a point on the route where the driver must take an action, such as making a turn or taking an exit.). Hence, eNav judiciously switches between a cheap inaccurate location estimation mode and an energy-expensive accurate one, thereby saving energy while maintaining usability.

We should stress that the phone GPS energy saving problem has been widely studied. Various adaptive sensing schedules (e.g., [16,21]) and multi-modal approaches (e.g., [9,36]) were explored. Our energy saving approach is comparatively very simple and relies on dead-reckoning to interpolate between appropriately chosen GPS sampling times. While we do not claim a contribution in dead-reckoning, we would like to note that contrary to the prevalence of prior work, our eNav is the first system that actively exploits low localization accuracies as energy saving opportunities for GPS navigation tasks. We do not aim to achieve accurate localization at all times. Instead, as mentioned above, we need accuracy only near a few discrete points along the route. This variable accuracy requirement offers more opportunities for energy savings. It is the exploitation of such energy saving opportunities, the evaluation of such savings, combined with usability questions, that is the point of the paper.

The motivation for this paper comes from the observation that smartphones have become popular means for navigation in vehicles. Dedicated GPS navigation devices, such as Garmin, see a continued decline in market share [5], whereas integrated dashboard systems are still an expensive option, compared to smartphone applications. Unfortunately, the GPS module is one of the most power-hungry components on phones [21, 23, 26, 27, 30]. It may deplete batteries within hours (or less when the phone is not fully charged), running the risk of navigation loss while driving—Indeed, the "real" motivation for this paper arose when an author nearly missed a flight, after his phone battery ran out while navigating to the airport. The above observations beg the question: would an energy-saving mode be a useful addition to current phone-based GPS navigation applications used by drivers? If so, how much energy will it save? In this paper, we first show results of a survey user-study that answers the first question in the affirmative. We then present the design, implementation and evaluation of such a service, demonstrating significant energy savings.

Briefly, eNav allows the user to enter or exit an energy saving navigation mode at will. In that mode, two mechanisms are employed that reduce energy consumption; *adaptive GPS sampling* and *screen saving*. Adaptive GPS sampling refers to substituting actual GPS positioning with dead reckoning using cheaper sensors, whenever such a substitution is deemed safe. The substitution is deemed safe as long as it cannot lead to a navigation error (for example, it is safe when the vehicle is sufficiently far from the next navigation waypoint). Screen saving refers to turning the screen off, ostensibly to save energy, but in reality to mask the fact that location estimation is inaccurate at certain parts of the route. Please note that some existing phone-based navigation apps (e.g., Google Map Navigation for Android) do have the option of automatically dimming or turning off the phone screen, however, to the best of our knowledge, eNav is the first navigation system that incorporates low accuracy tolerant localization during navigations. As a waypoint approaches, the allowable location estimation error shrinks, GPS sampling restarts, and voice navigation alerts the driver to needed actions, making it look as if location estimation was accurate all along. To further enhance user experience, we also allow the user, at any time, to pause the energy-saving mode by waking up the phone screen, at which point eNav will restore GPS sampling, and present to the user an accurate location, masking the fact that location was ever inaccurate. We show that the above mechanisms contribute to improved energy-efficiency, while keeping intervals of location estimation inaccuracy largely transparent to the driver.

To assess the need for eNav, we conducted a user survey (Under IRB Approval #14266) using CrowdFlower.com, between Nov 20th 2013 and Jan 5th 2014, asking the participants about their preferences regarding phone-based vehicular navigation. The survey was terminated upon receiving 500 valid responses. The purpose was to answer questions such as: How widely used are the drivers' smart-phones for vehicular GPS navigation (despite the availability of non-phone-based alternatives)? Since the car has ample power, would drivers indeed welcome a power-saving navigation mode on a phone (or is it irrelevant because they can plug it in)? Finally, would turning off the phone screen in the energy saving navigation mode be acceptable to drivers? The answers, detailed later, confirm that phones are widely used for vehicular navigation, drivers overwhelmingly welcome an energy-saving phone-based navigation mode, and drivers mostly find it acceptable to turn the screen off when the purpose is to increase battery life during navigation.

The system presented in this paper has been implemented and empirically evaluated via a deployment study involving 33 external (non-author) drivers who used the service. Participants were given randomly chosen destinations to navigate to using our prototype eNav implementation. We made sure that the participants were not familiar with the locations of the destinations they were given, and hence had to rely on our navigator. A total of 6000 km of uncontrolled driving traces and 2000 km of navigation trips were logged, spanning various road, traffic and weather conditions. Results of the deployment study showed that our energy-efficient navigation system achieves almost 80% energy saving compared to standard GPS, without missing navigation waypoints.

MOTIVATION

The idea for eNav originated from the authors' own bad personal experience with loss of navigation while driving. To motivate saving energy in a phone-based vehicular navigation system, however, the authors needed to answer three questions: Do (other) drivers commonly use phone-based navigation in vehicles, as opposed to other in-vehicle navigation options? Would saving phone energy while navigating be useful to them, despite the abundance of energy in a car? Finally, how do drivers feel about falling back on voice navigation, instead of visual cues, in the energy saving mode? (The latter is a side-effect of needing to mask location estimation inaccuracy in our service.) These questions were meant to establish the desirability of eNav, before taking steps to implement such a system.

To answer these questions, we carried out a nation-wide online survey (in the United States) using CrowdFlower.com. All questions were multiple-choice. In order to filter out less reliable responses (possibly due to respondents not paying enough attention or simply providing random answers), each survey contained 5 randomly-placed repeated questions with reordered choices. We only considered responses that showed consistency across all the repeated questions. We ran our survey until 500 valid responses were received. Each participant was paid \$0.5 (US) for completing the survey, which is consistent with prevailing compensation rates on CrowdFlower. The survey engine had mechanisms to prevent repeated entries by the same user.

Demographics: Survey respondents covered 385 cities in 48 different states in the US, of whom 38.6% were male and 61.4% female, ranging from 18 to 64 years of age (mean 35.9 and standard deviation 11.7). Most respondents were frequent drivers. Specifically, 68.6% said they drove every day, 21.8% said they drove once or twice a week, whereas only 9.6% said they drove rarely. Most of the driving was associated with shopping (84.4%), occasional entertainment (69.6%), work commute (65.2%), and long distance travel (46%), in that order. More than 75% said their average commute was at least 15 minutes.

Prevalence of phone-based GPS: The majority of respondents said they used phone-based GPS in the car. More specifically, 80.4% said they used GPS in a car, and 73% said they used a phone-based GPS in a car. More than 59.4% also used a phone-based GPS in a rental car (which often implies being away on a trip and in need of navigation), and as much as 37.4% said that they experienced running out of phone battery while using the phone for navigation.

Interest in energy-saving navigation (eNav): When asked whether they would find an energy saving navigator useful, 91% of the respondents said they would like to have an energy-efficient phone navigation application, of whom, roughly 2/3 said they would choose the energy saving mode when their phones are running low on battery, while 1/3 said they would make it the default mode regardless of the phone battery status, which is an even stronger endorsement.

We were especially interested in finding out how the demographics correlated with interest in eNav. Taking survey responses as ordinal values, we computed the correlations between these responses and interest in eNav. *Statistically significant* positive correlations were found between interest in eNav and each of (i) being a frequent driver, (ii) using GPS, (iii) using GPS on a phone, and (iv) running out of battery while navigating. This means that individuals with more driving and GPS usage experience are precisely those who liked eNav more. Not surprisingly, individuals who suffered from a navigation outage thanks to battery depletion also appreciated the service more. What was more surprising was that interest in eNav was also found to have a statistically significant positive correlation with the frequency of using a phone charger in the car. This suggests that individuals who use the charger more frequently do not particularly like having to do so, and are thus more appreciative of eNav.

Voice versus screen: Finally, 75% of the respondents either considered voice to be more important than visual cues for navigation, or were fine with either mode. Moreover, 81.8% said they would be willing to rely on voice navigation in the car if their phone battery was running low. There was a statistically significant correlation between accepting voice navigation (in lieu of the screen) and liking eNav, as well as being a frequent driver.

We summarize three key observations from the above results. First, the intuition that smartphones are widely used for vehicular GPS navigation is corroborated by survey data. Second, while the car has ample power, the great majority of drivers do indeed welcome a power-saving navigation mode on the phone. Finally, turning off the phone screen in the energy saving navigation mode is acceptable to most drivers. The latter observation was important to us because eNav *has to* turn off the screen in order to mask the fact that its location estimate gets inaccurate (specifically, when the car is far from the next navigation waypoint). Hence, it was important to determine whether drivers will accept that. The above results complete our motivation for eNav. Next, we describe system design, implementation, and actual deployment-based evaluation of efficacy and usability.

ENERGY-SAVING NAVIGATION

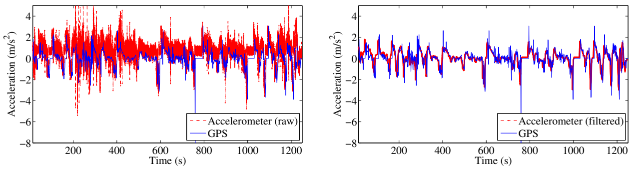
The goal of a GPS navigator is simple: instruct drivers to follow a specified route and minimize navigation errors. In order to build an energy-efficient navigation system, GPS usage needs to be reduced by replacing GPS with less expensive sensors while meeting the above goal. This naturally leads to the following two questions: (i) How to estimate the car's location when the GPS is off? (ii) When to turn GPS back on to prevent navigation errors? Below we outline the main ideas behind our energy-saving navigation.

How to estimate car's location with GPS off? We recognize that the phone's on-board motion sensors (e.g., accelerometers) are a natural candidate for estimating location when GPS is off: They consume only a tiny amount of energy (0.0488mW at 10Hz, according to our measurement) compared to the GPS (150mW at 1Hz) and can provide inertial motion readings, which can be used to estimate locations by carrying out dead-reckoning. Network-based localization (cellular triangulation and WiFi SSID signatures) has also been studied [16, 18]. We found from our experiments using Android's network-based localization implementation that its highest sampling rate was only at about 1 sample per 20s, which we considered too low for navigation.

When to turn GPS on? To answer this question, one might propose the naïve approach of simply lowering the GPS sampling rate during navigation while in the power saving mode. This approach *will* reduce the energy consumption, but at the expense of uniformly lowered localization accuracy and degraded navigation quality. As later shown in evaluation, due to the non-linearity of GPS energy consumption, saving about 80% energy (which our eNav system achieves) would dictate that the GPS sampling period of a traditional navigation application be increased from 1s to a whopping 83s, which according to our experiments would cause the user to miss most waypoints during navigation! We, on the other hand, take an adaptive approach derived from the simple intuition that high localization accuracy is needed when the car is close to a waypoint, but not when the car is still far away. Thus, given a rough estimate of the car's location, we can adaptively decide when to sample the GPS next.

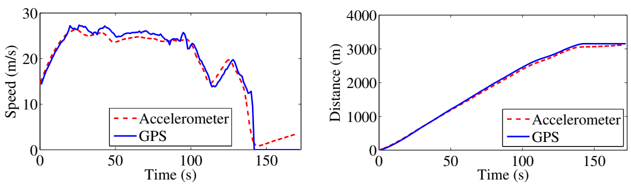
Low-Power Location Sensing

We estimate location using dead-reckoning based on accelerometer data when GPS is turned off. While dead-reckoning is not new, below we evaluate its performance in the specific context of vehicular navigation. We first conduct an ideal experiment, where the phone is carefully mounted in a car (driving on a horizontal road), such that a particular accelerometer axis is perfectly aligned with the car's direction of motion. In this experiment, a Galaxy Nexus Android phone was used. We drove the car on a predefined route, during which the phone continuously collected and logged on-board accelerometer (at 10Hz) and GPS (at 1Hz) readings. Since our goal is to mimic GPS, the collected GPS trace was treated as the ground-truth, offering position and speed measurements (from which acceleration was computed). These measurements were compared against the phone's accelerometer readings (which were integrated to obtain speed and



(a) Raw acc. data compared to acceleration computed from GPS trace (b) Car acceleration estimation using the phone's acc. data

Figure 1. Car acceleration estimation



(a) Speed estimate (b) Distance estimate

Figure 2. Example speed and distance estimation from acc. data

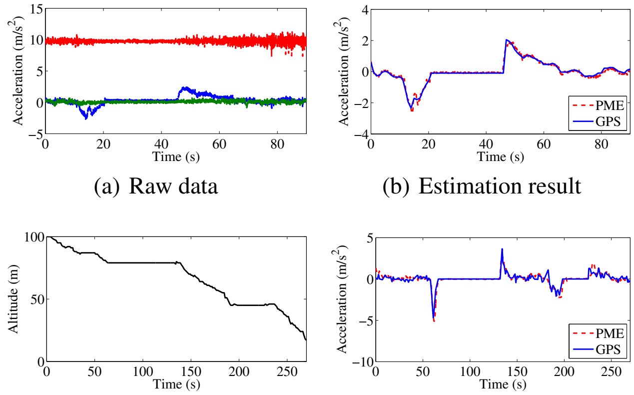
displacement). We repeated the experiment with 5 different driver-car pairs on 3 different routes, each consisting of around 10 different segments of various lengths. A total of 200 km of driving data was collected.

Fig. 1 shows an example trace, where a car's acceleration measured from the phone's accelerometer (raw, as well as filtered by a low-pass filter) is compared to that computed from the phone's GPS readings. The figure demonstrates the accuracy of acceleration measurements.

Given the measured acceleration, we integrate the time series to obtain the corresponding speed and distance estimates, as illustrated in Fig. 2 for an example road segment. The distribution of the distance estimation error (per unit of route segment length) is estimated using 200 km of driving data, with a mean close to 0 and a standard deviation of 0.02. Hence we deem phone-based dead-reckoning acceptable (for short periods of time), albeit imperfect.

Next, we consider arbitrary and unknown phone placement. To perform position estimation of a car on a road using a phone whose orientation is arbitrary and unknown, we need to solve two problems: (i) extract the acceleration measured along the car's direction of motion by using the phone's local sensor readings without prior knowledge of the phone's orientation relative to the car, and (ii) remove the effect of gravity component from the acceleration measurements along the car's direction of motion (when the car is not driven on horizontal roads). The Android phones that we used implement proprietary solutions, called Sensor Fusion [22], to solve these problems. We found them inaccurate, and opted to implement our own solution, based on Principal Component Analysis (PCA) [13]. We called it *Principal Motion Estimation* (PME).

We apply PCA to the phone's 3-axis accelerometer data. The first component derived by PCA captures the largest variability of the car's acceleration, which intuitively should correspond to the acceleration in the direction of driving. Note that, the principal component produced by the PCA has an ambiguous sign. Since cars spend most of their time moving forward, we simply assume that the initial acceleration after any stop (i.e., period of zero speed) is in the forward motion direction. We also considered using Nericell [25], which introduced another method for determining accelerometer orientation in cars. However, it requires the sampling of GPS, which introduces extra energy costs and hence did not serve our purpose of low power navigation.



(a) Raw data (b) Estimation result (c) Altitude trace (hilly road) (d) Estimation result (hilly road)

Figure 3. Principal Motion Estimation result demonstrations

	Mean		Variance	
Placement	PME	Fusion	PME	Fusion
Dashboard	0.0272	-0.3873	0.0269	0.3755
Seat	0.0063	-0.4002	0.0282	0.0748

Table 1. Comparison of the normalized distance estimation error distribution statistics: PME *vs* sensor fusion

Next, we show examples of PME-based estimation results. Fig. 3(a) and 3(b) show the 3-axis raw accelerometer data and the PME-estimated car acceleration as compared to the ground-truth, when the phone is on the dashboard.

Next, we consider a trace from a hilly road. Fig. 3(c) shows the altitude trace of the car as it was driven on that road. Since the road was not horizontal, the gravity had a non-zero component in the car's driving direction. Fig. 3(d) reports the car's acceleration estimate produced on that road by our PME method, showing good correspondence with ground truth.

To offer a more quantitative measure of accuracy, we use the raw accelerometer data collected from experiments involving driving trips on routes that spanned various road (e.g. horizontal, hilly, rural, urban) and weather (e.g. sunny, rainy, snowy) conditions, at different times of days, under varying traffic conditions. The experiment was repeated with 5 different driver-car pairs on 3 different complex routes, each consisting of around 20 different segments of various lengths. A total of 400 km of driving data was collected. The produced distance estimation error distribution statistics are shown in Table 1, comparing our approach to the phone's native fusion API. The table demonstrates that the average estimation error is small enough that the approach is sufficient for location extrapolation in between GPS samples. It remains to decide when to actually sample GPS in order to prevent navigation errors, which is the topic of the next section.

Adaptive GPS Sampling

Given the acceptable performance of dead reckoning illustrated in the previous section, we now introduce the idea of adaptive GPS sampling that switches between GPS sampling and dead reckoning depending on an estimated error bound and the current need for location accuracy. As is standard in dead reckoning, we first integrate the measured acceleration twice to compute a displacement along the current navigation segment. From the standard deviation in acceleration, we also compute the standard deviation in displacement and accordingly a confidence interval (confidence window) around the current displacement estimate using the obtained standard deviation in displacement. We use a window that extends two standard deviations around the mean, which corresponds to a 97% confidence interval. To decide when to sample GPS, we use the following simple rule: if the next navigation waypoint is at least T seconds (computed from the current location and speed estimation) beyond the 97% confidence interval window, then we are far enough away from the waypoint with high probability and hence no GPS sampling is needed. Otherwise, a GPS sample is taken and the accumulated location error is reset to zero. The configurable threshold T can be chosen to correspond to a comfortable warning distance for the driver. The chosen confidence interval is a tradeoff between energy savings (keeping GPS off longer) and false negatives (missing a waypoint).

As a practical consideration, when performing numeric integration to compute current speed, we bound it between zero and 15 mph above speed limit, as we consider speeds outside that box to be erroneous. This bounding prevents errors from accumulating, causing the speed to reach unrealistic numbers. For our service, we found a web resource, Wikispeedia [38], that hosts publicly available crowd-sourced road speed-limit data. We were able to crawl the data covering our regions, hence implementing the above feature.

Whenever eNav takes a GPS sample, it computes from the current location and speed how long it would take the vehicle to reach the next waypoint; If the waypoint is less than T seconds away, a voice notification is delivered to the driver. Empirically, from experimenting with existing navigation systems, we set T to be 15 seconds.

Enhancements to Location Estimation

We next discuss two enhancements for improving location estimation results during navigation.

Car-Idle Detection

Being able to detect car idling helps reset both the acceleration and speed estimations and prevent unnecessary growth of the corresponding distance estimation errors. Treating the detection task as a binary classification problem (where for each time slot we classify the accelerometer data as reflecting the car being idle or not idle), we initially experimented with a simple threshold-based method, for which we just took the magnitude of the raw 3-axis accelerometer data for each time slot (1s window) and compared its $mean$ to the learned threshold. Intuitively, when the car is idle, the magnitude of acceleration should be around 9.81, which is gravity, hence suggesting a threshold based approach. The approach yielded about 90% accuracy, with occasional misclassifications. This is because a good car on a good road offers a smooth enough ride that the accelerometer may not distinguish between being still and moving at constant velocity in a straight line.

Hence, in addition to the $mean$ acceleration magnitude, we computed the min, max, and $standard\ deviation$ to form a $4D$ feature vector for the classification task. Note that, for training, we simply labeled the time slots using speed readings of the corresponding GPS trace.

We experimented with several classification algorithms, using 10-fold cross-validation to compare their accuracy, defined as the average percentage of correctly labeled time slots among all slots tested. Our experiments show that the decision tree algorithm [31] achieves near perfect classification, as shown in Table 2. Therefore, we use a decision tree classifier in our final system design to detect idle time. The classifier is trained on the car's own data.

Classification Algorithm	Car-Idle (%)	Car-Turning (%)
Decision Tree	99.80	98.89
Support Vector Machine	96.35	69.48
Naive Bayes	98.17	63.63

Table 2. Car-idle and car-turning detection accuracy comparisons using various classification algorithms (10-fold cross-validation)

Car-Turning Detection

One other event we exploit is when a car turns. Combined with road intersection information, a turn gives us the opportunity to pinpoint a car's location without needing to sample the GPS. The intuition is straightforward. Whenever the car makes a turn, we check to see how many road intersections exist within the current location confidence window. If there is none or more than one, we are unable to determine accurate location of the car, so we sample GPS (to start a new segment). If there is only one, however, we can pinpoint the car as being at that intersection, without sampling the GPS. Road intersection information is obtained by processing the OpenStreetMap [28] (OSM) data, where intersections are identified as OSM nodes shared by multiple OSM ways. We carry out the extraction offline, and store the resulting intersection data locally on phones for use during real-time navigation. A single intersection is just a pair of (*latitude, longitude*) floats, thus caching even a large number of intersections (in the broad vicinity of the car) would not take much phone storage space.

The detection of car-turning is modeled as a binary classification problem, similarly to idle-detection, and uses the same set of features with the only difference being that each time slot is now 5s long (as we observed from data traces that turnings usually lasted about 5s). For training, each time slot is automatically labeled using the GPS bearing trace. Again, we observed that the decision tree classifier gives the best performance among all. We initially used a gyroscope for this experiment, which yielded similar results as the accelerometer. However, as gyroscope's energy consumption is about two orders of magnitude greater than that of the accelerometer (as shown later in evalution), we decided to use accelerometer for turning detection.

With turning detection enabled, eNav would suspend GPS sampling upon delivering the navigation notification to the user and relies on the to-be-detected turning motion to snap the car to the way-point, whose true location is known beforehand. Therefore, this approach reduces the sampling of GPS near waypoints, and introduces additional energy savings.

Deviation Detection & Handling

The handling of human mistakes is mission-critical for a navigation system. We assume the common case where users are honestly trying to follow navigation instructions, as opposed to trying to defeat their navigator. Three different user error scenarios are addressed:

1. The user makes a turn too early. This means that after being notified about an upcoming waypoint, there is at least one more intersection before the actual waypoint intersection. eNav will detect the turn immediately, as described earlier, and then try to localize the turn, either via map-based localization (if it is the only intersection in the uncertainty window) or a GPS sample (if it is not). As a result, the turn is identified as wrong, the user is notified, GPS is sampled, and a new route is computed.

2. The user makes a turn too late. In this case, the user, after hearing the navigation notification, misses the waypoint and takes a subsequent intersection. The detection of this type of deviation is exactly the same as the previous scenario if the wrong intersection is not far away from the waypoint. If it is, then it becomes identical to the next scenario.

3. The user fails to make a turn and keeps driving (possibly because there is no nearby subsequent intersections after the missed waypoint). In this case, eNav will keep updating the possible location range for the car to the point where the location confidence window moves beyond the waypoint intersection, at which point eNav recognizes that the user has likely missed the waypoint, and thus re-localizes by sampling the GPS and alerts the user.

IMPLEMENTATION

Our eNav prototype application was implemented on Android phones. When a user enables the energy-saving mode during navigation, adaptive GPS sampling is enabled and car localization is allowed to become inaccurate between waypoints. The phone screen will also turn off. As discussed previously, according to our survey results, the majority of people find it acceptable to rely on voice guidance to conserve phone battery.

We also made it such that the user can, at any time, pause the energy-saving mode by waking up the phone screen, at which point eNav will restore GPS sampling, and present to the user their accurate location, masking the fact that location was ever inaccurate. In our current prototype implementation, this interaction is done via user pushing the power button. More convenient interfaces (e.g., user voice command) are doable but are left for future implementation improvements. Also, as less experienced drivers may find some waypoints (e.g.,involving complicated intersections and junctions) confusing, we provide the option of automatically turning on screen, along with voice navigation guidance, at waypoints.

Our current eNav prototype uses Google Maps API to compute routes for specified source-destination locations. As several iOS and Android offline navigation apps already exist, our current eNav prototype energy saving mechanisms can be easily integrated into such offline systems to completely eliminate the need for cellular data connectivity.

Next, we discuss eNav's energy-efficient navigation flow, illustrated in Fig. 4. Nodes marked as Ox's and Dx's correspond to the basic operations and decisions, and eDx's are the enhancement components.

Basic Navigation

At the beginning of the trip, the GPS and accelerometer are both sampled until the car has gotten on the road and been driven for a short duration (empirically 1 min is enough). Data collected during this phase is used for initializing the various eNav models: the PME principal component vector, and the acceleration estimation error distributions. Then, eNav's energy-efficient navigation can kick in whenever the user decides to turn it on.

We first describe how the basic navigation flow works. In real-time, for each time slot, the car's principal motion is computed from accelerometer data (O1), and the speed and location estimation of the car are updated (O2). The estimated speed and the displacement confidence interval are then used to estimate distance to the next waypoint, which is then translated into time. If this time is smaller than a threshold (D1), GPS is sampled to get the accurate speed and location information (O3), which in turn is used to compute the time it takes to reach the next waypoint under the current GPS speed. If this time is again smaller than a threshold (D4), then the car actually *is* close to the next waypoint. In this case, eNav notifies the user about the upcoming waypoint (O5), and keeps sampling GPS continuously until the user passes through the waypoint (unless the turn-detection enhancement is used). Otherwise, the car is still far away, no special action is taken.

If the user fails to follow navigation instructions and drives past the waypoint (D3), deviation detection eventually fires, energy saving stops, and eNav immediately recalculates a new route using the car's current location (O4). Finally, eNav notifies the user and ends the navigation (O6) upon reaching the final destination (D2).

As mentioned, two threshold values are used in the navigation flow. We call the one in D4 the *critical notification time*. This controls how far ahead should the user be notified about the upcoming waypoint. If it is too high (e.g., "Turn right in 5 min after 10 km"), the user will probably have already forgotten about the notification by the time s/he actually reaches the waypoint. If it is too low, the user will likely not have enough time to react. By testing various commercial navigation applications/devices and interviewing our test users, we decided on the value 10s, which users can also adjust to better fit their personal preferences. Please note that

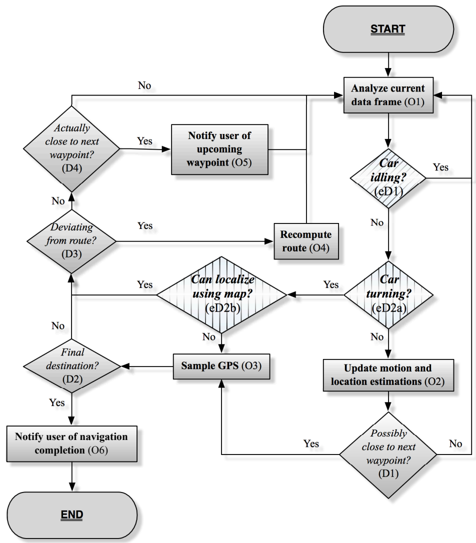


Figure 4. eNav's Navigation Flow

a navigation application can provide multiple notifications to the user about the same waypoint. As long as a notification is delivered at or before the critical notification time, we consider that the navigator succeeded in announcing the waypoint. The other threshold, as used in D1 is then set to be the time for GPS to get a fix under the current situation plus the critical notification time.

Enhanced Navigation

To incorporate car-idle detection (eD1), the acceleration data of each time slot is used to classify whether the car is idle or not. If it is, then eNav sets the car's estimated acceleration and speed to be 0 and does not modify the location estimate. Otherwise, eNav follows the rest of the basic navigation flow.

When car-turning detection (eD2a) is enabled, eNav checks for car-turning motion for each time slot. Upon detection, eNav tries to get the accurate turning location using map intersection location information (eD2b) by checking if a unique intersection can be identified within the car's location confidence window. If yes, an accurate location fix is obtained without using the GPS; and if not, GPS is sampled to get the accurate location information. Then, the rest of the basic flow is followed, except that eNav suspends all further GPS pull requests after notifying the user of the upcoming waypoint for the current segment when the turning detection module is enabled, and instead relies on turn detection to determine when the user has reached the waypoint.

EVALUATION

In this section, we evaluate expected energy savings in eNav, then report actual experiences with overall eNav's energy efficiency and navigation quality.

Energy Models

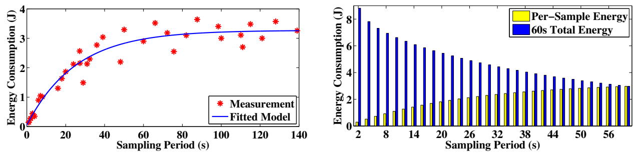
An ideal solution for measuring energy on phones while running eNav would be to physically connect a multimeter to the phone and battery, while subjects are driving, and measure how much energy is consumed. We opted against this approach because the extra wiring may interfere with the phone's motion sensor readings and potentially affect how users interact with the phone (e.g., where they might place it, and how they would use it during navigation). Hence, we instead log detailed event traces during eNav use and estimate energy consumption from that trace. For this approach to work, we first obtain energy consumption models for the phone's on-board sensors (accelerometer and gyroscope), GPS module, and the computational tasks that make up the eNav system. These models are then used to compute energy consumption from driving data traces.

Phone's On-board Motion Sensors

The gross energy consumption of each of the on-board sensors was measured at a 10Hz sampling rate. We used two multimeters for the concurrent measurements of voltage and current during phone operations. The voltage meter was connected in parallel to the phone's $+$ and $-$ connector pins, and the current meter in series between the battery's $+$ electrode and the phone's $+$ connector pin. To isolate the net energy consumption from the rest of the phone operations, measurements were made both with and without sampling the sensor. The difference then gave us the net energy consumption of the sensor. Each experiment was repeated 10 times and the average was taken. Also, all energy measurement experiments were carried out using the same phone within a span of 2 weeks, in order to avoid possible problems caused by different phones having different battery capacities or the same phone's battery capacity changing over a longer period of time. According to our measurements, accelerometer's and gyroscope's energy consumption rates were 0.0488mW and 2.14mW, respectively. This suggested that we not use the gyroscope.

Phone's GPS Module

For the phone's GPS module, a similar general approach was used to measure its power consumptions. However, due to its cold/warm/hot-start nature, discussed in previous work [20,24], GPS consumes energy at different rates under different sampling periods. The computation needed to acquire a single GPS fix increases as the time elapsed since the previous fix grows, because the information from the previous fix becomes less useful and eventually expires. We thus needed to measure GPS energy consumption at various sampling periods (from 1s to a couple of minutes, in our actual measurements) in order to be able to fit a quantitative model that we could use to compute the energy consumption of a particular GPS sampling trace. Our measurements are shown in Fig. 5(a). As seen, the per-sample energy consumption increases rapidly as the sampling period increases from 1s to about half a minute, and tends to saturate afterwards. We also notice that the general trend is gradual and smooth, showing no apparent GPS mode change. We therefore fit a simple continuous monotone function to model the GPS energy consumption, as shown by the curve superimposed over the measurements in Fig. 5(a). It is worth noting that, even though the per-sample energy consumption increases as the sampling period grows, it never gets to a point where, for a fixed sampling duration, sampling at a higher rate would consume less energy than at a lower one, as illustrated in Fig. 5(b).



(a) Phone's GPS module energy consumption measurements and the fitted model

(b) Phone's GPS energy consumption trends with increasing sampling period: per-sample vs 1min total

Figure 5. GPS energy consumption

Computation

The only computation that is eNav-specific and carried out continuously for every time slot (set to be 1s in our experiment) throughout the entire navigation is the car-idle and car-turning detection and the principal motion calculation. Both operations involve computing simple statistics (e.g., $mean$) of the accelerometer data segment (of size 10×3). The principal acceleration calculation then only involves taking the dot product of two 3D vectors. The idle/turning detection involves decision tree prediction computations, where the decision trees are usually of depth < 10. Therefore, each such detection involves comparing a pair of numbers at most 10 times. These computations are lightweight, and their energy consumption as measured are negligible compared to the sensors and the GPS.

The training of the decision tree and the estimation of the principal motion vector using PCA are carried out once within the first minute of the trip instead of continuously throughout the entire navigation. Our measurements confirmed that the corresponding energy consumption is negligible compared to the navigation's total energy consumption throughout the entire trip.

Energy Model Verification

We validated our energy models using data collected from about 300 km of driving traces, with two phones placed in the car under similar settings (both on the seat). One phone ran eNav and performed navigation, while the other phone simply logged the GPS and accelerometer data. The energy consumption of eNav during the navigation was physically measured and recorded, and the data collected from the other phone was used to simulate the running of eNav and predict the energy consumption by applying our energy models. We observed that, over the aggregated driving trip length, the computed value remained within an 5% error of the measured energy consumption. Hence, the results below have a 5% error margin.

User Studies

We aimed to carry out a thorough analysis of both the energy saving and the navigation quality aspects of eNav through user studies. We recruited 33 external (non-author) volunteer participants (from multiple departments of the university; of both genders; ages ranging from 20s to 40s) for our studies. Ideally, we would need to carry out carefully controlled experiments, where each participant uses eNav on a large number of navigation trips to previously unvisited destination locations (otherwise the participant would not need to use navigation systems in the first place), and have the participant repeat with varying eNav settings. We would also ideally repeat the experiment for all participants. This, however, would require an unrealistically large amount of time and effort from the participants, and thus would not be practical.

Faced with this constraint, we divided our user study into two phases. In Phase I, we simply asked all users to drive as they wished, while vehicle-resident phones logged the GPS and accelerometer data traces. From the collected trace, we could later carry out thorough analysis of energy consumption by simulating the running of eNav using the actual driving data collected. This simulator would use the accelerometer and GPS traces to compute location estimates per the algorithms described in this paper. The algorithms would decide when to sample GPS. The closest pre-recorded GPS sample would be used. The others would be hidden from our algorithm. Energy would then be estimated based on the number of GPS samples used, as well as the energy consumption of other sensors for the duration of each trip. As mentioned above, this approach generates less than 5% error. Using this method, within a span of two months, we were able to collect a total of over 6000 km of driving data, including various road, traffic, and weather conditions (urban, rural; rush hour, non-rush hour; daytime, nighttime; sunny, rainy, snowy), where each participant contributed about 3-week's data. In subsequent subsections, we report the energy savings and navigation qualities derived from analysis on this dataset.

In Phase II, we asked all participants to use eNav for actual navigation. We explained to them beforehand that eNav has an energy saving mode that they were to test. Each participant was given 3 routes (varying from 6 to 30 km) to drive on, selected from a pool of source-destination pairs, where we made sure the destinations had not been previously visited by the participant. All participants were also informed that eNav would by default keep the phone screen off in the energy saving mode, but they could turn on the display at any point they felt like to by a button push. On the other hand, the turn-by-turn voice guidance would remain active throughout the entire trip. A total of about 2000 km of eNav navigation trips were logged, and we report the participants end-to-end experiences of using eNav for navigation.

Energy Savings

As indicated by our survey results, the majority of drivers found it acceptable to rely on voice guidance during navigation, and most found it acceptable to turn off the screen to preserve phone battery when running low. Therefore, to be fair, we considered the following as the baseline navigation strategy: sampling GPS constantly at 1Hz and having the phone display turned *off* during the navigation. We report the energy savings of eNav compared to this baseline, with and without the enhancement modes enabled. Results are shown in Table 3. As seen, compared to the base navigation strategy, the basic eNav scheme reduces the navigation energy consumption by about 65%. Enabling the idle and turning detection enhancement modules each introduces an additional $5 \sim 8\%$ energy saving. And finally, with both detection modules enabled, eNav cuts down the navigation energy by about 78%.

	Energy Savings (%)
eNav with Idle & Turn Detections	78.37
eNav with Turn Detection	73.42
eNav with Idle Detection	70.59
Basic eNav	65.64

Table 3. Energy savings of various navigation schemes as compared to the baseline navigation strategy

Note that, the savings would be higher if the baseline was a mode with the screen turned on.

We also take a look at the energy saving breakdown as eNav (with both enhancement modules enabled) operates on road segments of different lengths. As shown in Fig. 6(a), longer segments lead to higher energy savings. This is expected because a car, driven on a longer segment, remains “far away” from the next waypoint a higher percentage of time than on shorter segments, thus sampling GPS less often. Inspecting the driving traces reveals that instances of long-distance highway driving were few in our experiments. Therefore, we expect the energy savings to be even higher for navigation trips that involve long-distance highway driving, spanning tens or hundreds of kilometers.

Note that, even though eNav achieves lower energy savings on shorter segments than longer ones, it does not mean that eNav navigation in urban settings would necessarily lead to much poorer energy savings. There are two reasons for that. First, cities are more congested, leading to more savings when the car is stopped (due to idle-detection). Second, when navigation services compute routes, they rarely choose zigzag-shaped routes. Instead, they prefer simpler ones with less turns. We observed this from using Google Maps (which we use as the routing engine for our eNav implementation as well) to compute navigation routes for a large number of randomly selected pairs of locations within urban areas. We tried this in Urbana-Champaign IL, New York City, and Seattle WA, using an average trip length of 10 km (ranging from 3 to 15 km). The navigation routes computed rarely ($< 1\%$) contained more than 6 waypoints.

Navigation Quality

By navigation quality we refer to the ability of the navigation system to successfully deliver a navigation notification within a critical notification time window needed for the driver to react and make the turn. In our experiment, we set this critical time window to 10-15 seconds before a waypoint. Notifications that occur more than 15 seconds prior or less than 10 seconds prior to the waypoint therefore do not count. If no notification of a turn was delivered within the window, it is considered a miss. Navigation quality decreases as the number of missed notifications increases. Traditional navigation applications are able to provide the highest possible navigation quality as they have constant access to high-accuracy location information. eNav, on the other hand, does not have spot-on location information of the car at all times. Thus, it is natural to question whether navigation quality is impacted.

The driving trace data collected from our 33-user deployment study in Phase I was used to estimate navigation quality as defined above. Specifically, we computed all notification times based on driving traces and checked whether or not they fell within the respective windows. The evaluation revealed that eNav *never* missed a single delivery of navigation notification throughout the study. This means eNav is able to save energy during navigation without sacrificing quality, as defined above, compared to traditional navigation apps. To compare to a traditional navigation app that samples GPS at a constant rate, we ran the navigation component of eNav at a constant rate and reduced the rate until its energy matched that of eNav’s adaptive GPS sampling. The sampling period had to be increased from 1s to 83s. We then simulated running navigation using this low GPS sampling rate on our collected driving traces, and determined the timing of waypoint notifications. Only 16.8% of all way-points notifications were delivered within their windows. Hence, the constant sampling rate service was poor on quality.

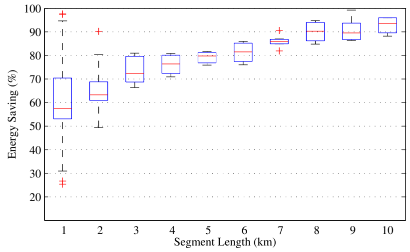
To more directly illustrate how eNav navigation works, Fig. 6(b) shows a complete example driving trip, which consists of both relatively long (around 10 km) and short segments (a couple of kilometers or about a few hundred meters). As seen, eNav samples the GPS very sparsely, and only when high location accuracy is needed. The traces for both the actual displacement estimation (computed using dead-reckoning, for estimating the car’s location) and the displacement confidence interval (maintained for determining when GPS should be used) are shown. Car-idle in the middle of a segment can also be observed, for example, the I_a, I_b, and I_c as annotated in the figure. At these times eNav successfully detected that the car was not moving, reset its motion estimation, and prevented the unnecessary increase of the displacement estimation and the error confidence bound. The phenomenon that longer segments lead to higher energy savings is also evident.

End-to-End User Experience

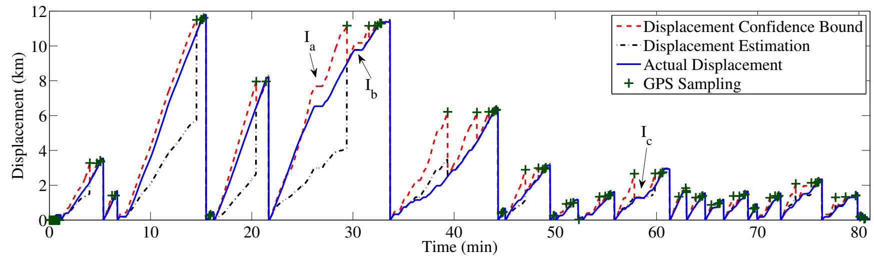
We now discuss users’ actual end-to-end experiences as they used eNav for navigation. Specifically, we look at their real-life energy savings, their actual usage patterns, and how they felt about using eNav for navigation.

First and foremost, no miss of navigation waypoints occurred in any of the eNav navigation driving trips for any participant, which agrees with the results we derived from our larger scale data analysis, previously discussed. The empirical CDFs of eNav’s energy savings (over the aforementioned baseline navigation strategy of sampling the GPS at 1Hz and keeping the phone screen off) are shown in Fig. 7(a). As can be observed, the median energy saving is at round 75%, where the per-trip energy savings range from about 55% to 85% (agreeing again with our average energy saving of 78.37% from the previous larger scale analysis).

Regarding the actual users’ usage patterns, we observed among all the 99 navigation trips (33 users each making 3 trips), about 50% of them have total screen-on time below 2% of the entire trip duration, and about 90% trips have less than 5% total screen-on time, as shown in Fig. 7(b). Notice that, this number accounts for the initial screen-on times when users are starting the eNav app and interacting with it at the beginning of the navigation trip. If we exclude this

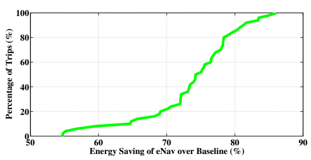


(a) Energy saving distributions over road segment lengths (box edges are the 25th and 75th percentiles, whiskers extremes)

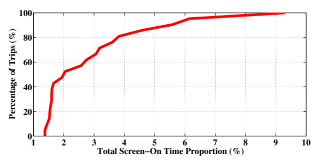


(b) A complete example navigation session for an entire trip using eNav with both car-idle and turn detection modes enabled. Every single segment's initial displacement value is set to 0 for ease and clarity of illustration.

Figure 6. eNav Energy Savings



(a) The empirical CDF of energy savings of eNav in real navigation trips



(b) The empirical CDF of total eNav screen-on time proportions (i.e., $\frac{\text{total screen-on time}}{\text{total trip time}}$)

Figure 7. eNav navigation empirical results

start-up screen-ons, then only 32% of users had ever actively turned on the phone screen during navigation.

As users completed our study, we conducted a simple exit interview to ask them about their general experience regarding using eNav for navigation, expressed in their own words. Of 33 users, two told us that they were nervous without the screen on, but felt reluctant to actually turn on the phone screen because they wanted to preserve the phones' battery power to complete the navigation session. Three users mentioned that they felt they could not talk in the car for fear of missing voice navigation notifications, which they perceived as a disadvantage. The rest (28) all expressed that they did not have any complaints. Two users pointed out that pressing a phone button to wake the screen was inconvenient for navigation, to whom we clarified right away that voice activation was the original implementation plan and would be in the final version. Finally, regarding the actual navigation quality of eNav, the one that best summarizes all user comments we received was *"It's hard to tell the difference between your service and real GPS,"* which we think well reflected eNav's navigation quality and user experience, and was as our original design goal indeed.

RELATED WORK

Mobile phone-based energy-efficient location sensing has been widely studied, including static [23] and adaptive GPS duty cycling [3,16,29,39], motion sensor-triggered GPS sampling [21, 25], network-based localization [14, 16, 33], and other multi-modal approaches [9,15,36]. However, they generally optimize for high average localization accuracy at all times, where under navigation scenario only spot-on localization near waypoints is crucial, which eNav exploits. This notion of variable localization accuracy was also studied [18], where, however, navigation was specifically considered as an inapplicable scenario.

Related studies on navigation also exist; They are, however, mainly for indoor and/or pedestrian scenarios [2, 4, 7, 36]. Vehicles and road-sensing have been a target for several recent smartphone-based systems [1,6,11,12,17,19,25, 32,37,40] that provide road/traffic advisories and services, or sense the dynamics of people (e.g., driver phone use), the vehicles (e.g., speed, etc), and the road/traffic environment. SenSpeed [10] is closely related to our work in that both involve using accelerometer to estimate vehicle speed. However, SenSpeed tries to get high estimation accuracy at all times, where eNav targets high accuracy around waypoints only. Also, SenSpeed involves the constant sampling of gyroscope, which eNav avoids due to its much higher energy footprint compared to accelerometer.

For vehicular navigation, multiple approaches exist that try to combine GPS and motion sensors (accelerometer, gyroscope, etc.) [8,34,35]. However, they focus on improving the vehicle location tracking accuracy by building dedicated systems that use motion sensor data as supplements to GPS sensing. eNav, on the other hand, embraces a completely different design principle in approaching the navigation problem: the entire system is based on off-the-shelf mobile phones, and focuses on energy efficiency by trying to replace GPS accurate location sensing with dead-reckoning-based rough location estimation as much as possible, without compromising perceived navigation quality.

CONCLUSION

In this paper we presented the design, implementation, and evaluation of eNav, a smartphone-based energy-efficient vehicular navigation system. We focused the investigation on answering two questions. One investigates the willingness of drivers to sacrifice some navigation features (e.g., visual cues) in order to prolong the phone's battery life during navigation. The other computes energy saved using simple accelerometer-based mechanisms that complement GPS in vehicular navigation. The answers show that the service is both desirable by drivers and effective at saving energy. The navigator behaves largely the same as traditional navigation applications in terms of quality and usability, while achieving around 80% energy savings.

ACKNOWLEDGMENTS

This work was supported in part by NSF grant CNS 1040380.

REFERENCES

1. G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen. Tracking vehicular speed variations by warping mobile phone signal strengths. In *PerCom*, 2011.
2. D.-K. Cho, M. Mun, U. Lee, W. J. Kaiser, and M. Gerla. Autogait: A mobile platform that accurately estimates the distance walked. In *PerCom*, 2010.
3. Y. Chon, E. Talipov, H. Shin, and H. Cha. Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *SenSys*, 2011.
4. I. Constandache, R. R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *INFOCOM*, 2010.
5. Electronista. Garmin profits drop as hardware sales continue decline http://goo.gl/8YVH8p, 2013.
6. R. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. Abdelzaher. Greengps: A participatory sensing fuel-efficient maps application. In *Mobisys*, 2010.
7. Y. Gao, J. Niu, R. Zhou, and G. Xing. Zifind: Exploiting cross-technology interference signatures for energy-efficient indoor localization. In *INFOCOM*, 2013.
8. M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global positioning systems, inertial navigation, and integration*. Wiley-Interscience, 2007.
9. S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar. Autowitness: Locating and tracking stolen property while tolerating gps and radio outages. In *SenSys*, 2010.
10. H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, Y. Zhu, G. Xue, and M. Li. Senspeed: Sensing driving conditions to estimate vehicle speed in urban environments. In *INFOCOM*, 2014.
11. S. Hu, H. Liu, L. Su, H. Wang, T. F. Abdelzaher, P. Hui, W. Zheng, Z. Xie, and J. Stankovic. Towards automatic phone-to-phone communication for vehicular networking applications. In *INFOCOM*, 2014.
12. S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher. Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification. *ACM Transactions on Sensor Networks (TOSN)*, 11(4):55:1–55:27, July 2015.
13. I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
14. A. J. Khan, V. Ranjan, T.-T. Luong, R. Balan, and A. Misra. Experiences with performance tradeoffs in practical, continuous indoor localization. In *WoWMoM*, 2013.
15. D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *SenSys*, 2010.
16. M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær. Entracked: energy-efficient robust position tracking for mobile devices. In *Mobisys*, 2009.
17. E. Koukoumidis, L.-S. Peh, and M. R. Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Mobisys*, 2011.
18. K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy aware localization for mobile devices. In *MobiSys*, 2010.
19. J. Lindqvist and J. Hong. Undistracted driving: A mobile phone that doesn't distract. In *HotMobile*, 2011.
20. J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. Loureiro, and Q. Wang. Energy efficient gps sensing with cloud offloading. In *SenSys*, 2012.
21. H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys*, 2010.
22. G. Milette and A. Stroud. *Professional Android Sensor Programming*. Wrox, 2012.
23. 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 the cenceme application. In *SenSys*, 2008.
24. P. Misra, W. Hu, Y. Jin, J. Liu, A. S. de Paula, N. Wirström, and T. Voigt. Energy efficient gps acquisition with sparse-gps. 2014.
25. P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys*, 2008.
26. M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Mobisys*, 2009.
27. S. Nath. Ace: exploiting correlation for energy-efficient and continuous context sensing. In *Mobisys*, 2012.
28. OpenStreetMap. http://www.openstreetmap.org/, 2015.
29. J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Mobisys*, 2010.
30. J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan. Energy-efficient positioning for smartphones using cell-id sequence matching. In *MobiSys*, 2011.
31. J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
32. F. Saremi, O. Fatemieh, H. Ahmadi, H. Wang, T. Abdelzaher, R. Ganti, H. Liu, S. Hu, S. Li, and L. Su. Experiences with greengps — fuel-efficient navigation using participatory sensing. *IEEE Transactions on Mobile Computing (TMC)*, PP(99):1–1, 2015.

33. S. Sen, J. Lee, K.-H. Kim, and P. Congdon. Avoiding multipath to revive inbuilding wifi localization. In *MobiSys*, 2013.
34. I. Skog. A low-cost gps aided inertial navigation system for vehicular applications. *KTH Signals Sensors and Systems*, page 49, 2005.
35. M. L. G. Thoone and H. P. M. Krukkert. Adaptive inertial vehicle navigation system, Mar. 7 1990. EP Patent 0,181,012.
36. H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *MobiSys*, 2012.
37. Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin. Sensing vehicle dynamics for determining driver phone use. In *MobiSys*, 2013.
38. Wikispeedia. http://www.wikispeedia.org/, 2015.
39. C.-L. Wu, Y.-T. Huang, C.-L. Wu, H.-h. Chu, P. Huang, and L.-J. Chen. An adaptive duty-cycle scheme for gps scheduling in mobile location sensing applications. *PhoneSense*, 2011.
40. J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin. Detecting driver phone use leveraging car speakers. In *MobiCom*, 2011.