Geo-Fencing: Geographical-Fencing based Energy-Aware Proactive Framework for Mobile Devices

Jihoon Ryoo†, Hwangnam Kim§, and Samir R. Das†
†Computer Science Department, Stony Brook University, Stony Brook, NY
§School of Electrical Engineering, Korea University, Seoul, Korea
Email: jiryoo@cs.stonybrook.edu, hnkim@korea.ac.kr, and samir@cs.stonybrook.edu

***Abstract*—Location-based services (LBSs) are often based on an area or place as opposed to an accurate determination of the precise location. However, current mobile software frameworks are geared towards using specific hardware devices (e.g., GPS or 3G or WiFi interfaces) for as precise localization as possible using that device, often at the cost of a significant energy drain. Further, often the location information is not returned promptly enough. To address this problem, we design a framework for mobile devices, called `Geo-fencing`. The proposed framework is based on the observation that users move from one place to another and then stay at that place for a while. These places can be, for example, airports, shopping centers, home, offices and so on. `Geo-fencing` defines such places as geographic areas bounded by polygons. It assumes people simply move from fence to fence and stay inside fences for a while. The framework is coordinated with available communication chips and sensors based on their energy usage and accuracy provided. The essential goal is to determine when users check in or out of fences in an energy effiecient fashion so that appropriate LBS can be triggered. Windows based smartphones are used to prototype `Geo-fencing`. Validations are conducted with the resulting traces of outdoor and indoor activities of several users for several months. The results show that `Geo-fencing` provides an effective framework for use with LBSs with a significant energy saving for mobile devices.**

I. Introduction

Mobile users experience a considerable amount of battery drain when they use location-based services (LBSs). This results from the use of commodity GPS and WiFi interfaces that are typically power hungry but are widely used to implement LBSs. To optimize energy expenditure, many researchers have proposed schemes that focus on using less power hungry chips or subsystems more often than subsystems that require a higher power. One of the representative low-power subsystem that can aid location determination is the accelerometer. Accelerometer can effectively triggering location update procedures. For example, when the device detects any motion, it can activate more power hungry subsystems such as the GPS to determine the location. However, in current state of the art of mobile software systems, device enable/disable needs to be *explicitly* handled by the LBS applications in the resource framework. The framework manages all the resources in the mobile devices, and it controls the state of installed sensors and devices. Since devices are enabled on demand, it may take time for the devices to come to an active and ready state so that they can be sampled. In case of the GPS, it takes about 10 seconds on average to 'warm up' (of course, the cold and hot state requires different delays) [1]. WiFi takes about 3 seconds once it is enabled to scan all the channels and retrieves the signal strength of each AP in the vicinity so that localization can be enabled . Communication chips such as 3G – normally enabled all the time – take about 9 milliseconds. The low power sensors such as the accelerometer and the digital compass take 140 milliseconds and 36 milliseconds for enabling and retrieving data, respectively. This response time can be minimized to negligible values by making passive transition to proactive one. As the resource framework keeps track of the location and motion information, it can respond instantly to any location change and it can update the location-related data when the data needs to be updated. Note that our previous work [2] showed the required accuracy is changed only when user moves, and therefore, the framework does not need to hold always new information.

978-1-4673-1298-1/12/$31.00

These observations prompt us to devise a new resource framework. This new framework can be used where the mobile platform may have many devices and sensors that may have unexpected delays for activation and consume different amounts of energy. We propose a new geographic fencing concept, called `Geo-Fencing`. This is motivated by the observation that mobile users often dwell in a place (e.g., a bounded area such as home, office, school, shopping area etc.) for an extended period of time and, in between, simply move from place to place. Many LBSs do not need to know the precise location of user or track the user at all times, and simply use location just to determine whether the user is in such a place, or users' history of being in such areas.

The `Geo-Fencing` identifies possible places (where interested persons could be found) by geographically fencing them with the global coordinates, then it establishes different service contexts for each fenced area, and it uses different hardware resources (sensors and devices) to save energy usage and recommends a different suit of applications for users according to the location context. Note that the size and complexity of the fence can be diverse. Fences can even be hierarchical as a fence can be included in other higher level fences. For example, a shoe store which offers discount coupons for users can be categorized in various ways, such as shoe store in the eastern wing of the mall that is again part of the entire mall.

Fences are tagged with well-known names. Fences are typically represented by polygons of arbitrary shapes. While using rectangles simplifies computation to determine when the user checks in or out of the fence, polygon offers a

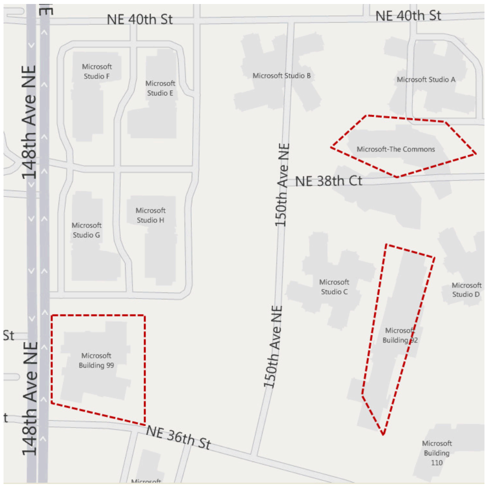


Fig. 1. Geo-fence examples.

Fig. 2. Resource frameworks on smartphone. Current state of the art is on the left. The proposed solution is proactive and is on the right.

more accurate representation, especially when there are many fences in a region that could be overlapping. See Figure 1 for examples of fences.

We now describe how the proposed `Geo-Fencing` framework can be used. When a user checks in to a fence (called *incoming fence*), the proper application can be turned on proactively, and then makes proper services based on his (or her) personalized profile. For example, a newly stocked XBOX game title can be recommended to users who are interested in XBOX titles. Additionally, the proposed framework may turn off the GPS chip (in a mobile device) to stop the battery drain by repeatedly accessing the GPS chip inside fences and may turn on the WiFi for faster communication if it is needed and available. When the user leaves the current fence (called *outgoing fence*), the 3G based triangulation confirms his (or her) leaving, and the accelerometer data predicts the current state. When the user is moves to another fence (called *between fences*), the proposed framework mainly makes a state prediction, and also retrieves the location data when it decides to update the current location information. In section III, we discuss more in detail about this incoming, outgoing fence, between fences and state prediction.

To summarize, this work makes the following contributions:

- introducing the geographical fencing concept for mobile device's accurate, fast, and energy efficient framework;
- studying the classification of human activity based on low power sensors that allows us to achieve a good accuracy in detecting human activities while still being energy efficient;
- developing a prototype testbed using a windows mobile smartphone.

The remainder of the paper is organized as follows. Section II discusses the problems with state of the art resource frameworks. Section III presents key design principles, and introduces our framework operation that is described in section IV. Then in section V, we ad some discussions. Section VI describes our prototype implementation. Section VII discusses related work, and section VIII concludes the paper.

II. Current Framework

The major causes of battery drain in the current framework (for applications like LBS) are: (*i*) on-demand (as opposed to proactive) activation and operations, (*ii*) limited energy environment, and (*iii*) always demanding the highest possible accuracy that also spends significant energy.

A. Resource framework

In current applications such as LBS, the resource framework operates in the on-demand mode as you can see on the left side of Figure 2. The LBS application requests the location information from the resource framework. The framework in turn queries the resource and delivers the information to the application that requested it. The framework does not maintain any location related state on its own.

B. Resource cost

We expect increased hardware variety and software complexity in smartphones as they become more prevalent than ever. Better sensors, CPUs, communication chips and batteries as well as new device yet unforeseen will become available. For example, 2G or 3G has given way to 4G; WiFi, Bluetooth, GPS, and gyroscope devices have become common, and battery technology has fed more powerful CPUs and displays including brighter multi touch screens. In the `Geo-fencing`, we are able to utilize many of these chips and sensors to determine the current location directly or indirectly, and ultimately to build up a smarter framework. We can do several levels of localization with different devices or sensors with varying degrees of accuracy and energy costs.

1) Absolute Position: Existing frameworks can determine the user's location reasonably well using GPS, 3G or WiFi-based localization.

GPS: GPS is obviously the most accurate for localization. There have been many studies on its accuracy and energy usage. In our testbed, its average error is found to be 10 meters and its energy expenditure per sampling is around 29.15 joules.

3G: 3G is the most available resource among all devices used for localization as the interface is always on. The cell-tower ID can serve as a simple coarse-grain location information. We

Fig. 3. Relative angle in digital compass

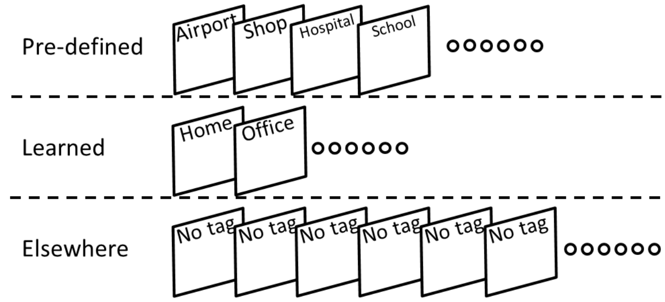


Fig. 4. Fence areas registered with the Geo-fencing

will also augment this later with signal strength information. In our experimentation, the average error is around 300 meters and its energy expenditure per sampling is around 1.2 milli joules.

WiFi: WiFi provides the most valuable location information in the indoor environment where it is prevalent and GPS may not work very well. However, in our work, we have used it outdoors. As a possible localization method, SSID matching can be used, but it is required to access SSID database since the database is necessary to do the large-scale location. Unfortunately, many areas do not have such the database, and so it is hard to utilize the SSID matching. In our experimentation, its average error is around 30 meters and its energy usage for a sampling is around 33.66 joules.

2) Relative Movement: Existing frameworks can detect relative movement using the accelerometer and digital compass.

Accelerometer: The accelerometer is the most efficient and sensitive sensor for motion detection among sensing devices, it is accurate and very energy efficient. Many researches proposed their own methods to effectively and accurately identify the pattern of human behaviors [3]–[5]. In those approaches, their hit ratio is around 40∼90 % in 6∼20 activities. The high hit ratio –the successful prediction– comes from determination of distinctive activities. In the proposed framework, we simplify these behaviors to three activities (walking, driving, and stationary). Additionally, walking is divided into indoor and outdoor walking, driving to highway driving and street-way driving. In such the simplified classification, we can achieve around 90% hit ratio. Note that "street-way" driving refers to driving outside the highways.

Digital Compass: The use of digital compass is limited. It only provides the angle of orientation of the mobile device's main axis. If the mobile device is fixed relative to the user, the change in compass angle, possibly along with accelerometer data, can detect certain activity. For example, during driving the mobile is expected to be in a fixed orientation with the user (say, in the pocket or a purse)figure 3. Then, the compass can detect turns, e.g.

Several other sensors and communication interfaces are available in current smartphones. However, they are limited in some perspective. For example, Bluetooth suffers from the fact the communication range is small and there are typically no bluetooth devices that are fixed , like WiFi APs. The ambient light sensor needs a deeper examination to be used for detection of human activity as the same environment can vary widely in terms of lighting, and also the phone can often be in a pocket or purse. The gyroscope sensor can be represented with a combination of accelerometer and compass. We have described aforementioned resources with variant accuracy and cost in Table I, and based on the table we can improve the current framework based on the table.

C. Approach

The proposed proactive framework will provide efficient energy management, increasing battery life. This can be done without compromising the accuracy of the underlying devices, and without imposing latency in performance.

III. System Overview

In this section, we explain the fundamental building blocks of `Geo-fencing`. These are *fence*, *activity classification* and *energy consumption profiles*. These together provide a solid grounding for the proposed energy aware proactive resource framework for mobile devices.

A. Fence

We first present the concept of `fence` in `Geo-fencing`. The `fence` is divided into three categories `predefined fence`, `user-learned fence`, and `elsewhere`.

`fence`: A fence surrounds a polygonal area, and is defined by a set of GPS points. The fence is simply defined by the polygon and a buffer area on all sides of the polygon (see section V). When the framework detects that the user is checking-in a certain fence, the framework can enable appropriate applications. For example, when the user checks

TABLE I
ACCURACY AND ENERGY COSTS IN SOME OF THE AVAILABLE RESOURCES ON THE SMARTPHONE

	Accuracy	Energy	Comment
GPS	High	High	open sky needed
3G	Low	Low	good coverage in urban areas
WiFi	Middle	High	limited coverage (predominantly indoors)

Fig. 5. Tracing around a home fence.

at the airport fence, the framework initiates airport-related applications. E-tickets can be popped up, information about delays or gate information can be displayed, shopping or cafe specials can be announced etc.

`predefined fence:` A predefined fence is created by a person, institution other than the phone's owner, or the framework itself. These are typically well-known fences that are widely used. For example, an airport, a shopping center, a hospital, or a school. Many areas can be defined as pre-defined fences. This information is tagged with GPS, Cell-tower ID, and MAC address of AP. Since many map applications have already had this information, plugging any map database into our framework extends the predefined fences. In our experiment, we registered some predefined fences with the `Geo-fencing` in the tested area.

`user-learned fence:` The user-learned fence is an area the user periodically visits. For example, the certain area that user visits at work hours in working days can be defined as "office". Similarly, "home", "friend's house" and many other fences can be defined based on the user's moving patterns. Figure 4 describes a collection of fence areas in the `Geo-fencing`. Figure 5 shows an example of user-learned fences, which is the home fence for one of authors. The home labeled spot is the actual location of the home, and most of GPS data is obtained around a window in the bedroom where the mobile device is placed. Because the `Geo-Fencing` holds the last location information, it indicates that last GPS data are mostly gathered around the entrance of the home. The majority of information comes from the home labeled spot, but taking trash to dumpster and picking up the mail at the mail room are also used to determine the home fences. This example is illustrate the concept of fences alluded to before. The mailroom and dumpster are also part of the home fence. Thus, one can define the concept of a `sub-fence`. As an example, when the user visiting shopping mall, various stores can be `sub-fences`. We leave these `sub-fences` as future work in order to simply the presented framework.

`elsewhere-fence:` The elsewhere-fence is an area where it does not belong to any predefined fence, or so that we leave this as the elsewhere-fence and it can be manually tagged by the user.

B. Activity Classification

We utilize the phone's sensors for classifying user's activity in a coarse fashion. This in turn is used to prompt more accurate location determination that perhaps requires more energy. For example, change in activity (e.g., brisk walking after a period of relatively sedentary activity) can possibly be followed by a checking out from a fence (say, office). Thus, the framework can now trigger GPS to determine the correct location. This policy effectively reduce energy demand and activity sensors such as accelerometer are relatively inexpensive in terms of energy.

In our work, we mainly used three activities to demonstrate the power of our method – Walking, Driving and Stationary – but we also explored additional activities to verify if they can be used in estimating the current state of the user. Figure 6 shows the accuracy of our activity detection technique which is based on the standard deviation of sampled accelerometer and compass data.

In Figure 6 we present the accuracy results for activity classification. We can identify 98.7% of the mobile activity from immobile activity based on the standard deviation of accelerometer data – the small portion of miss comes from small movements that are possible even in the stationary state. The accuracy of distinguishing between walking and driving detection is 92.5% – stops for traffic results in some misses. However, in our method distinguishing between indoor and outdoor walking was hard (not much better than random). More is explained in section V.

The compass data is used to discriminate between the city street (street-way) and the highway driving. The classification as done follows. We classified the street-way driving with 90 degree turns and stops/moves at various traffic signs, and we identified the highway driving with straight line driving, possibly with smooth curvatures. The overall accuracy is not very high (69.4%). A confusion matrix is presented in Table II. This shows that highways are almost always correctly labeled. But often street-ways are labeled as highways, likely because

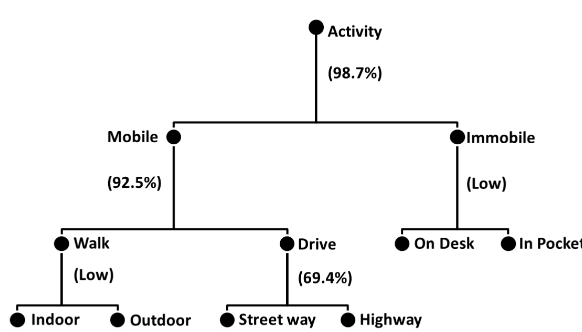


Fig. 6. Classification of user activities and their accuracies.

TABLE II
CONFUSION MATRIX FOR DRIVING

		Prediction	
		Street-way	Highway
Truth	Street-way	22.2%	77.8%
	Highway	3.0%	97.0%

TABLE III
OVERALL CONFUSION MATRIX

		Prediction		
		Stationary	Walking	Driving
Truth	Stationary	97.1%	0	2.9%
	Walking	0	76.9%	23.1%
	Driving	0	0	100%

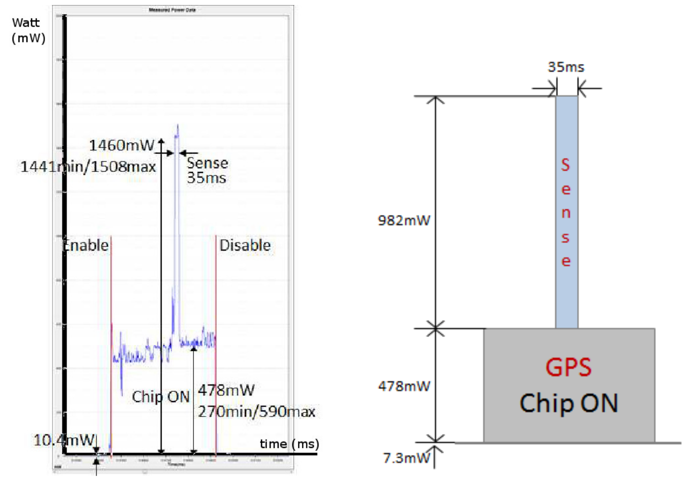
less stops are no turns. More work is clearly needed for a better classification.

Based on these results, we finally chose *Stationary*, *Walking* and *Driving* as the *motion-level* states to detect user motion, then determine the current *fence-level* state, and consequently the current fence. The motion-level states play the role of a trigger to detect the fence-level state (which is one of *incoming fence*, *outgoing fence* and *between fences*, and is explained in section IV). Table III summarizes the accuracy of overall prediction for those motion detection methods, which gives the basis to construct the system operations in section IV. Note that all the results are the result of using low energy sensors and no high energy chips are used to increase the accuracy.

C. Energy Consumption

We primarily take into account the energy usage as well as efficiency. To monitor accurate energy consumption, we use the *power monitor* as a monitoring tool [6]. Once the mobile device is connected to the probe of the measuring device, it shows watt/current usage and records it into a file. Figure 7 (a) shows the measured device's watt usage when only GPS chip is enabled in the background.[1] The power usage requires additional 330mW when the display device is turned on. In this figure, 10.4mW (min 7.1mW/max 11mW) is used for the system ready state, and 478mW (min 270mW/ max 590mW) is used for the GPS chip enabled state, and 1460mW (min 1441mW/ max 1508mW) is used for the sensing state. The power usage returns to 10.4mW when the chip is disabled. This is obtained from a windows phone [7] and the values reported here could be different from other devices, but it is expected that qualitative patterns are similar. Figure 7 (b) shows the simplified energy profile. Figure 8 shows energy blocks for the GPS, 3G, WiFi, Accelerometer, and Digital compass. Notice that the GPS and WiFi need chip-on state while 3G, Accelerometer, and Digital compass do not need the chip-on state but they simply can read values from each sensor. This simplified energy profiles are used in our framework rather than fluctuating real values. Figure 7 shows the shape of chip-on and that of sensing GPS signal. From

[1]The term *background* refers to periods when the display of the smartphone is turned off and no foreground application is activated.



(a) energy usage in mobile device (b) simplified energy block of energy usage

Fig. 7. Energy usage of the GPS chip in a mobile device.

the figure, we can observe the sensing period is very short in comparison to chip-on period, which means the energy usage for a sampling is a small amount. Therefore, number of sensing is not substantial after the chip is on. We call a *frequency* for the number of sensing in one chip-on period.

IV. SYSTEM DESIGN

In this section, we introduce the `Geo-fencing` in detail. As we discussed in the previous section III-B, we divide the human activities into three states or categories: driving, walking and stationary. In our prototype we could distinguish between these three states with a high degree of accuracy. With this accurate activity classification (motion-level state), we can determine a fence-level state. We discuss the fence-level states now.

A. Fence-level State Transition

The `Geo-fencing` work based on three fence-level states – incoming fence, outgoing fence, and between fences.

1) Incoming Fence: The framework changes its fence-level state to incoming fence when a user checks in to a known fence. In this fence-level state, the framework turns off the GPS chip to save energy, and only looks for the associated cell tower's signal strength variations. The framework does not attempt to verify or change its fence-level unless its cell-tower's signal strength is 40dB weaker than the value at check-in. In urban regions with small cells, the matching of last few cell-tower's IDs could be used, but, in the rural region, cells could be substantially large. Therefore, the framework uses the cell-tower's signal strength variation for the state transition. The last three cell-tower IDs are recorded anyhow to prevent the fence-level state from being frequently changed when the user is located at the boundary of a cell. The value of 40dB is large enough to prevent false transitions.

2) Outgoing Fence: The framework changes its fence-level state to outgoing fence when the signal strength of each cell-tower falls below 40dB than the recorded values at check in.

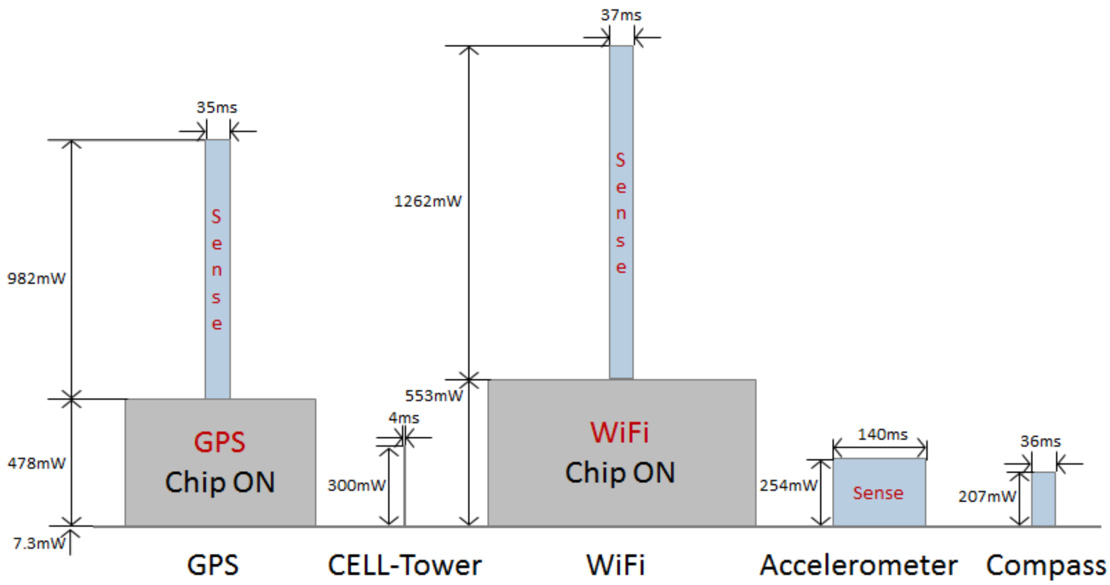


Fig. 8. Energy usage profiles for GPS, 3G, WiFi, accelerometer and compass.

Then the framework turns on GPS and confirms its location to check whether it is out of the current fence or not.

3) Between Fences: The framework checks its speed based on the GPS information – the speed information is available from the GPS chip. Then it calculates the estimated time of arrival (ETOA) to any target fence (which can be changed in the course of movement). When the framework is in the fence-level state of "between fences," it checks its motion-level state. If the motion-level state is changed, the ETOA needs to be recalculated because the current speed is changed. We use the interval of C·ETOA to turn on the GPS and recalculate a new ETOA, where C is a constant coefficient less than 1. Since the coefficient is less than 1, the framework can reconfirm whether or not the previous target fence is changed by recalculating new ETOA before it arrives at the target fence. This step is needed because people can change its direction or speed unpredictably.

B. Motion-level State Transition

The aforementioned fence-level state transition is activated based on the motion-level state transition. The motion-level states are described in Table IV, and the current state is decided according to the activity classification presented in section III-B.

Stationary: In the stationary state, we have three transitions – Loop, Incoming, Outgoing. To join the stationary state, the standard deviation of the accelerometer data should be smaller than S-Threshold (the expected deviation in being stationary). If the previous state was stationary, its S-Threshold becomes double. Then it stays at its loop, the tendency of state is more likely to stay, and this prevents frequent leaving and returning. To leave this state, the accelerometer should be vibrated bigger than (doubled) S-Threshold.

Algorithm 1 `Geo-fencing`

```
// Incoming Fence
if Recorded fence then
    Turn off GPS chip
    Check recorded cell-tower ID' SS
    if STATE is WALKING or DRIVING then
        @2 min period
    else if STATE is STATIONARY then
        @10 min period
    end if
end if

// Outgoing Fence
if Recorded Cell-tower ID's SS < threshold then
    Turn on GPS chip and confirm
end if

// Between Fences
Check speed
ETOA w/ current speed
if STATE change then
    Recalculate TOA
end if
@ C·ETOA turn on GPS and calculate TOA from current
location to nearest fence
```

Walking: To join the walking state, the standard deviation of the accelerometer data should be larger than W-Threshold (the expected deviation in being walking). If the previous state was walking its W-Threshold is halved. Then it stays at its loop. To leave this state, accelerometer should be vibrated smaller than (halved) W-Threshold.

TABLE IV
STATE TRANSITION ACTIVITY CLASSIFICATION.

State Flow	Activity Classification and Description
S - Loop	continuous stationary
- Leave for P	leave the stationary spot by walking
- Leave for D	leave the stationary spot by driving (direct transition from S to D is unlikely occurred)
W - Loop	continuous walking
- Leave for D	checks in the car, and leave the spot by driving
- Leave for S	stays in the rest area, or stays in the car without moving
D - Loop	continuous driving
- Leave for S	stays in the car without moving (longer traffic sign or stays in the car at parking lot)
- Leave for P	checks out the car, and start walking

Driving: To join or stay in the driving state, the standard deviation of the accelerometer data should be larger than S-Threshold, and smaller than W-Threshold. To leave the state, its accelerometer should be vibrated out of thresholds (bigger for Walking or smaller for Stationary).

The two threshold values based state-prediction achieves high hit ratio from the ground truth and prevents the battery from being drained too much by minimally accessing high-profile sensors in a fence.

V. DISCUSSION

As stated before, a fence includes a marginal area around the line between GPS points. The size of this margin should be carefully considered. In predefined fences, we believe the framework needs this buffer area, but, user-learned fences do not require it because the user's moving patterns themselves can generate the boundary. The determination of the marginal area affects the accuracy of checking-in. For example, if the marginal are is too wide, checking-in occurs more frequently than it should – these are false positives (FP). If the marginal area is too tight, checking-in occurs less frequently than it should – called false negatives (FN). Checking-out is also affected by this marginal area. Figure 9 shows a predefined office building; the solid line is actual building shape and dashed lines are added marginal area of -5,+5,+10...+25 meters.

Consequent results are depicted in Figure 10. If it is too wide, FP will increase; otherwise, FN will increase. Figure 10 depicts CDF and CCDF. We observe from the figure that (*i*) FP increase as the marginal widens in CDF, (*ii*) FN decrease as it grows in CCDF. Note that the sum of these values is also depicted as solid line in CDF. We found that if the marginal area is too tight we experienced frequent FN, and if the marginal area is too wide we experienced frequent FP. When the marginal area is wider than 5 meters, we can see that there is no FN and the increasing pattern of FP in +5 +15meters is smooth. As a result, +5 meters is reasonable choice and that choice incurs smooth checking-in and checking-out.

VI. EXPERIMENT AND EVALUATION

We now evaluate the performance of the proposed framework in terms of the energy savings and the efficiency.

A. Implementation for trace file gathering

We implemented a measurement tool on an HTC Windows Phone [7]. We collected experimentation data of several months from multiple users. The users are categorized into researchers, their family and friends. They carry devices on the front of their car, in back pocket, or in purse. The users are mostly in the region of Redmond, Seattle, and Bellevue city, but some files are traced at different cities - when users

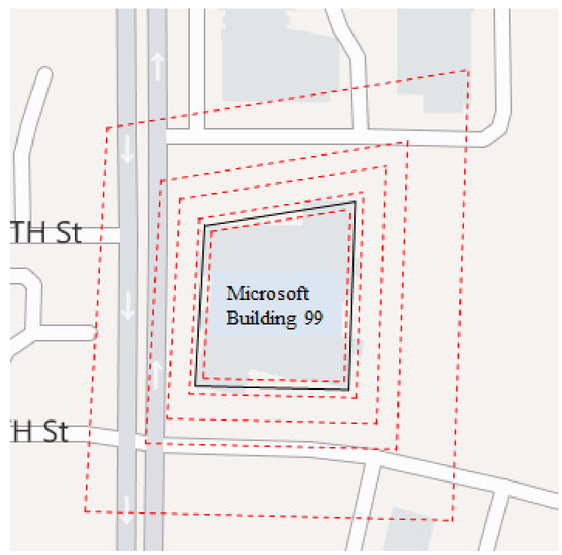


Fig. 9. Various sized fences according to marginal (buffer) area.

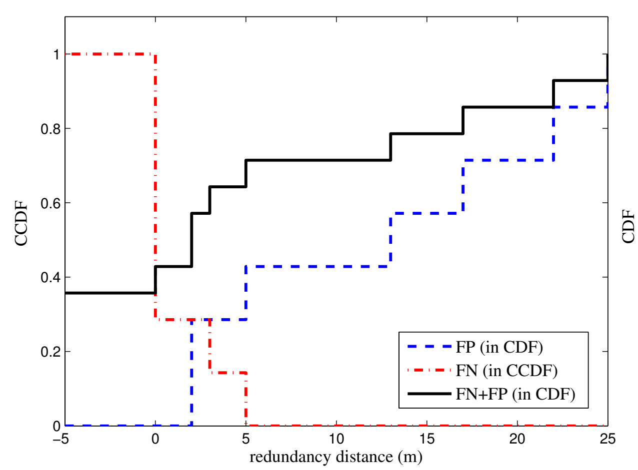


Fig. 10. Distributions for the false positives and false negatives.

are on a business/personal trip. The trace application runs in the background, and it automatically reports its trace file at the end of day through wireless networks. The user neither needs any additional action to report nor hinders the measurements by installed application. In the measurement tool, it traces its GPS (number of satellite, Latitude, and Longitude), 3G (cell-tower ID, region code, and its signal strength), WiFi (number of neighboring AP, and these AP's name, mac address, and RSSI), accelerometer (x,y,z value), and digital compass (angle value) with absolute time stamp.

B. Trace files and its validity

The traced file is available at [8], for the privacy matter, the 4 digit ID is randomly generated from local device, then we make sure the trace file is not used for tracking the person or any other purpose except the current research. In addition to the real world trace files, we implemented the framework to estimate the energy savings based on these collected files, and those can be used for comparison between the current framework and the proposed framework.

C. Operation of framework

We now demonstrate the energy usage and the performance of `Geo-fencing` based on the first author's trace data from his office to his home. In Figure 11, as he leaves the office, (a) shows the GPS sampling, which can trace accurately but samples 64 times by enabling the GPS chip all the time; 322.42894 Joules are used. However, when accelerometer and compass sensors are used for the framework, (b) shows the 14 times of the GPS sampling from office fence (left-bottom) to home fence (right-top, blue); 73.49 Joules are used. The 77.21% of energy is saved in this scenario. The five fences are shown in Figure 11 (b), where the red fences are predefined fences – B99, B92, Commons, and Redmond town center from left-bottom to right-top, and the blue fence is a learned fence – the home fence at the right-top. More scenarios are examined, and roughly around 60$\sim$80% of energy is saved.

D. Application of framework

The `Geo-Fencing` can also work great as security related applications. For example, industrial companies are sensitive on their documentation leaks by any photography. As a solution, many company disable workers' camera functions by affixing stickers on the camera. More advanced solution for smartphone is available such as disabling-camera function when a worker's ID card is checked at the entrance, the server sends the message of disabling camera function. However, the `Geo-Fencing` can work in this application, when worker checks in the secured office fence, it can automatically disables the camera, then mobile device can send a confirmatory message to the server. Without ID card, `Geo-Fencing` can work great with the required minimum message exchange with the server. Many other systems and devices for vehicles can be improved by our `Geo-Fencing`. Majority of those systems are wired or battery powered. As `Geo-Fencing` offer a smarter power management system, those can be *on* only when it needs to and *off* most of the standby time.

VII. Related Work

Detection and prediction techniques of human activity with the accelerometer in the mobile devices have been widely developed because it is cheap and accurate. Attaching multiple sensors or singular sensor to human body increases the hit ratio with the proposed technique. However, it is obvious that, as they classify human activity into many activities, the hit ratio decreases. Discriminating human activity into 20 activities, the hit ratio spans between 41.42$\sim$97.49% [3], 10 activities 32.8$\sim$ 100% [4], 8 activities 32.63$\sim$87.18% [9], 6 activities 55.5$\sim$ 96.5% [10], and 5 activities 49.4$\sim$95.5% [5]. They also used mathematical methods to increase hit ratio such as statistical approaches. The common goal of all these activity detection and prediction methods is to increases the hit ratio in classifying human activity.

Many localization techniques have been proposed and some of them have been commercialized. Localization is undoubtedly necessary because people are always curious where they are regardless of outdoor or indoor environment. Since the GPS chip drains battery, it is the key for GPS-based localization to use the GPS as minimal as possible [11]. The social network based techniques that predict user's moving pattern based on the previous path were studied so as to minimize GPS usage [12], [13].

The WiFi's SSID encoding technique [14], cell-tower signal strength [15], [16], FM radio station signal strength [17], and camera and microphone (on the cell phone) based techniques [18]–[20] are widely studied and validated. The energy profiling based techniques are also proposed [2], [21].

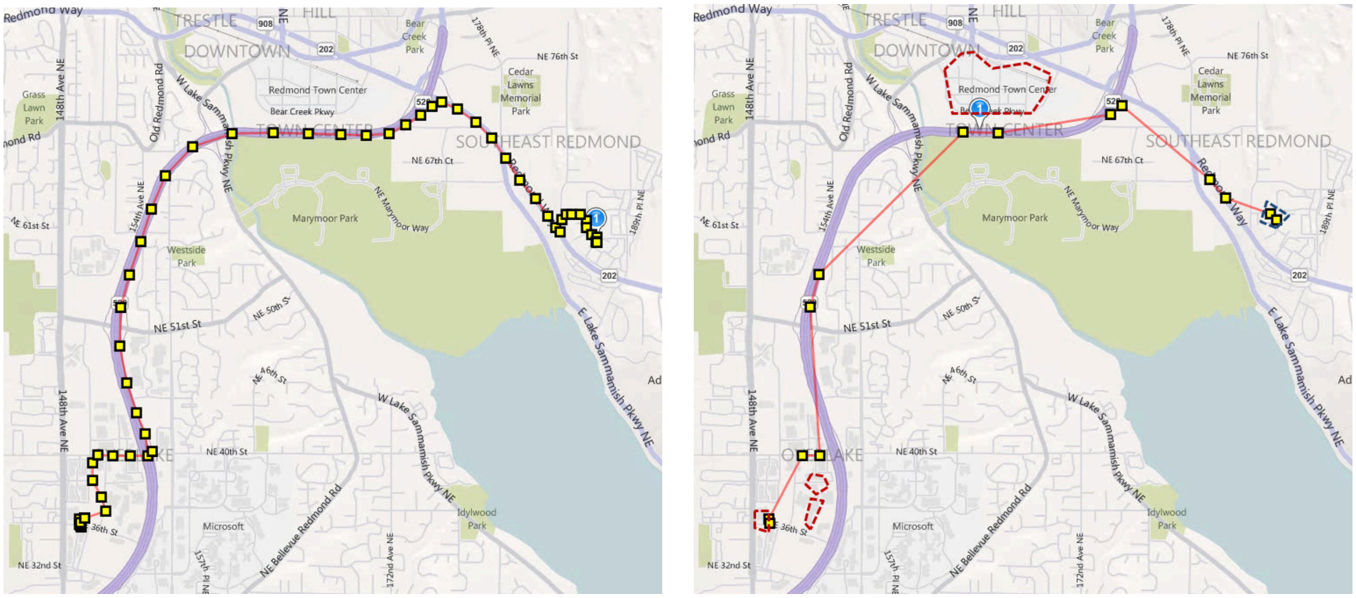
The predefined tags are abundant and its growth is fast, more than 3 Billion tags are created in last 6 months [22]. To utilize these resources efficiently, the proposed framework can be a great solution; for example, plug-into *Goolge*'s checkin service can expand our work [23].

VIII. Conclusion

In this paper, we first identified inefficient resource coordination in the current framework that affects battery drain. Secondly, we define the fence concept for human moving patterns. Lastly, we detected human motion based on low cost energy sensors. These findings are fed into the `Geo-Fencing`. We then implemented the proposed framework on windows phone and executed a performance evaluation study in order to validate both the accuracy and the efficiency. The experiment results present that the `Geo-Fencing` provides an energy effective framework in the mobile devices. Several challenges still remained to be explored. We plan to elaborate the hierarchical fencing concept and verify if it is feasible and achieves better accuracy. We also would like to enhance the technique of detecting human motion to increase the hit ratio within the proposed framework. Also, we will accumulate more trace data to sophisticate the proposed framework.

Acknowledgment

We thank Aman Kansal of Microsoft Research, Redmond, for help and support and the anonymous reviewers for their



(a) current framework

(b) `Geo-fencing`

Fig. 11. Energy usage of the GPS samplings in smartphones.

feedback. We are grateful to Sensing and Energy Research Group in Microsoft Research for providing us with an initial code and helping us understand the smart phone frameworks. This research was supported by Microsoft Research, in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MEST) (No. 2010-0014060), and in part by the KCC (Korea Communications Commission) Korea under the R&D program supervised by the KCA (Korea Communications Agency) (KCA-2012-08-911-05-001).

References

[1] M. Lehtinen, A. Happonen, and J. Ikonen, "Accuracy and time to first fix using consumer-grade GPS receivers," pp. 334–340, 2008.

[2] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proc. MobiSys*, ACM, 2010.

[3] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Proc. PerCom*, ACM, 2004.

[4] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *Proc. IJCAI*, 2005.

[5] X. Long, B. Yin, and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in *Proc. EMBC*, IEEE, 2009.

[6] Power Monitor, "http://www.msoon.com/labequipment/powermonitor."

[7] Windows Phone, "http://www.microsoft.com/windowsphone."

[8] Trace files, "http://research.microsoft.com/project/geofencing."

[9] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proc. PerCom*, ACM, 2006.

[10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," in *Proc SensorKDD*, ACM, July 2010.

[11] M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer, "Entracked: energy-efficient robust position tracking for mobile devices," in *Proc. MobiSys*, ACM, 2009.

[12] S. Gaonkar, R. R. Choudhury, L. Cox, and S. A, "Micro-blog: Sharing and querying content through mobile phones and social participation.," in *Proc. MobiSys*, ACM, 2008.

[13] I. Constandanche, M. Sayler, S. Gaonkar, R. R. Choudhury, and L. Cox, "Energy-aware localization using mobile phones.," in *Poster, MobySys*, ACM, 2008.

[14] J. Barrus, "Geofi: Global positioning for wifi-enabled devices," in *where2.0*, 2008.

[15] G. Sun, J. Chen, W. Guo, and K. J. R. Liu, "Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 12–23, 2005.

[16] A. Lamarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place Lab: Device Positioning Using Radio Beacons in the Wild," in *Proc. PerCom*, ACM, 2005.

[17] J. Krumm, G. Cermak, and E. Horvitz, "Rightspot: A novel sense of location for a smart personal object," in *Proc. UbiComp*, ACM, 2003.

[18] G. Schindler, M. Brown, and R. Szeliski, "City-Scale Location Recognition," in *IEEE CVPR*, 2007.

[19] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proc. MobiCom*, ACM, 2009.

[20] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: a high accuracy acoustic ranging system using cots mobile devices," in *SenSys*, ACM, 2007.

[21] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proc. MobiSys*, ACM, 2010.

[22] Tagging, "http://www.microsoft.com/tag."

[23] Checkin, "http://www.google.com/mobile/checkin."