Algorithms for Detecting Motion of a GSM Mobile Phone

Ian Smith, Mike Chen
Intel Research Seattle
1100 NE 45th St
Seattle WA, 98105

ian.e.smith@intel.com
mike.y.chen@intel.com

Alex Varshavsky
University Of Toronto
40 St. George St. Toronto, Ontario
M5S 2E4, Canada
+1. 416.946.0241
walex@cs.toronto.edu

Timothy Sohn
UC San Diego
9500 Gilman Dr.
La Jolla, California, 92093, United States
+1.858.534.9669
tsohn@cs.ucsd.edu

Karen Tang
Carnegie Mellon Univ.
5000 Forbes Ave
Pittsburgh, PA 3
+1. 412.268.9288
kptang@cs.cmu.edu

ABSTRACT
In this paper we detail two algorithms for detecting that a GSM mobile phone is moving (or stationary, since this is when the phone is simply not moving). These algorithms, in early experiments, show excellent promise and require nothing from the mobile phone other than radio signals that the phone must have to perform its normal function. We also discuss our approach to determine not only if a phone is mobile but also to differentiate motion based on walking from motion caused by traveling by car.

Keywords
Mobile phones, social mobile applications, GSM.

1. INTRODUCTION
A number of authors have discussed the value of context in ubiquitous computing [1, 2] with location being a key component of context. Of course, a user's location is not fixed and rooted to one spot—they are mobile actors in their social *milieu*. For many people, especially in the western world, a key element of their context and their "location" is their use of transportation. The means of transportation can vary from the above-ground street trams of Zurich to the deeply buried central London Tube or from the blinding speed of 250kph on a German *autobahn* to a gentle 3kph walk on Champs Elysées. In any of these cases, the people engaged in this means of transport view this as key component of "what they are doing right now", and their location context.

When people are moving around in their world, they often use mobile phones both to coordinate their activities at the other end of the transportation and as a way to convert the otherwise "wasted" time in transit into useful work or social time. Building on the work of the Place Lab project [3] we have been studying techniques that can be used to automatically detect when a mobile phone is moving, and when a phone is moving what means of transportation is being used. We are not assuming the use of GPS on the mobile phone, both because this makes our results more widely applicable but also because many means of transportation do not allow a view of the sky to receive GPS signals (e.g. subways, in urban canyons).

Clearly, our efforts to detect mobility need to be combined effectively with other location technologies that can detect places [4, 5], so that the user's location can be detected at points in time when the user is not mobile. In the interest of space, we do not detail our efforts to effectively recognize previously visited places [6].

It is important to note that the work detailed in this paper does all of its computation *on the mobile device itself* and does not need to coordinate in any way with the network provider or the network infrastructure. This has been a key assumption in all the Place Lab work [7], and it is preserved in this work in an effort to not compromise the end user's privacy. Any GSM mobile phone user *can* be tracked by their network provider, although the laws governing this tracking or the disclosure of obtained location information varies by locale. We assume that the end-user is running applications that need to detect whether or not the user is in motion and will use the principles of informed consent [8]

The key contributions of this work are:

- Two practical algorithms for the detection of mobility using only the GSM radio present on any GSM mobile phone. Naturally, these algorithms can also detect when a GSM mobile phone is stationary, since this simply the cases when the phone is not mobile.
- An approach for detecting the mode of transportation employed, when the mobile phone is in motion, again using only the GSM radio signals available on any GSM mobile phone.

2. Detecting Mobility
With a GSM mobile phone, there are basically four properties of the cellular network that can be measured and that vary as the mobile phone moves through its radio environment.

- Currently connected cell: This is the identifier of the cell to which the mobile phone is currently "connected." (Although this identifier represents a specific antenna on a specific, typically nearby, tower, we will not say antenna but "tower" as this is the common usage). The connected cell is the one that the mobile phone communicates voice data to and from should a call be started, or be in progress. Associated with the connected cell is a measurement of how strong the signal from that cell is being registered, according to the handset (signal strength).
- Neighbor cells: This is a list of identifiers that represent "nearby" cells within the cellular network of a given provider. These are used to accomplish the "hand off" between cells as the phone moves through its environment. Put another way, when the connected cell becomes faint one of the neighbor cells is chosen to

become the connected cell. As with the currently connected cell, a signal strength value is measured by the handset for each neighboring cell.

- Channel information: A GSM mobile phone can see many "channels" in its environment, with each channel representing a particular cellular tower in the environment. This data is very similar to neighbor cells with the exception that the mobile phone does not know the identity (or mobile carrier) of the cell towers listed in this set of channels. For example, there may be 100 active channels in the GSM environment and a particular handset on a particular network may know that it's connected cell (id: 4223) is on channel 29 and the seven nearby neighbor cells are on channels 30-36, but it has no information about the other 92 channels in use by other users and networks. Again, for each channel a signal strength value is calculated by the handset.

- Timing advance: For efficiency reasons, a GSM phone measures the amount of time it takes for its signal to reach the connected cell. This data is used to coordinate between multiple, simultaneous users of the same cell tower. Since the speed of radio transmissions is known, this timing information is effectively a distance measurement from the cell tower. In practice this has a resolution of 550m. Although this can be measured by the handset, the phones used in our experiments do not expose this information, so we do not use this data in our mobility calculations.

The algorithms presented are derived from a very simple principle: when a phone is moving, discernable amounts of variation occur in the observed signal strengths as opposed to when the phone is stationary and the signal strengths are "stable." Of course, in practice the radio environment is quite noisy and therein lies the challenge of distinguishing changes in the radio environment caused by the user's motion from random, background noise.

2.1 Detecting Motion

In this section, we present two of our initial algorithms we have designed to distinguish between 2 motions states {MOBILE, STABLE}.

The first algorithm is based on the RSSI (Receiving Signal Strength Indicator) "spread" of each GSM reading. As indicated in the previous section, each cell tower has both a channel and a signal strength associated with it. However, a GSM mobile phone does not typically have information about channels or towers of network providers not associated with the end-user's monthly bill [1].

Our first algorithm tracks the difference between the maximal and the minimal signal strength values per channel observed in a given window of GSM readings. This metric gives a coarse estimation of the variation in the RSSI, the "spread," over all channels. Intuitively, large variance is an indicative that the GSM mobile phone is seeing vastly different signal strength readings, and is therefore being mobile. While such differences may periodically occur while the phone is stationary, in most cases high variance is an indicative of mobility.

Figure 1 presents the pseudo-code for the Spread Based Mobility Algorithm. The algorithm monitors the amount of RSSI spread within a sliding window of GSM readings. When the average RSSI spread exceeds a certain upper threshold, the algorithm labels the phone as being {MOBILE}. Whenever the average RSSI spread falls below the threshold, the algorithm follows a stepwise gradual decay pattern until it falls below a minimum threshold, whereupon the algorithm transitions from {MOBLE} → {STATIONARY}. Currently, {STATIONARY}→ {MOBILE} transitions are done instantaneously: the instant the average signal strength difference exceeds the upper threshold, the {MOBILE} state is instantaneously fired.

```
for (every GSM reading within the window)
{
        Ignore signals which are "noise";
        for (every sensed channel)
        {
                Sum RSSI spread;
        }
        Average RSSI spread;
        If (RSSI spread > UpperThreshold)
                Return "MOBILE";
        Else
                Begin stepwise decay counter
        If (counter < MinimumThreshold)
                Return "STATIONARY";
}
```

Figure 1. Spread Based Mobility Algorithm

The second algorithm tracks the difference in Euclidean distance between every consecutive pair of GSM readings within a given window. Conceptually, Euclidean distance correlates to how similar these consecutive GSM readings are to each other. The smaller the Euclidean distance between the two readings the more similar these readings are.

Figure 2 presents the pseudo-code for Window Based Mobility Algorithm. Whenever the Euclidean distance between two consecutive readings is larger than a DistanceThreshold, the algorithm treats the pair as "bad" and increments a counter. A large number of "bad" readings is an indicative of instability in signal strength, which is assumed to be caused by mobility. Once the count of bad readings exceeds BadReadingsThreshold, the phone is labeled as being {MOBILE}. To avoid a succession of transitions between {MOBILE} and {STATIONARY} states, the algorithm requires the number of "bad" readings to be either beyond or above the BadReadingsThreshold for a certain period of time (this is not reflected in the pseudo-code).

```
numOfBadReadings = 0;
```

[1] This is the normal case. When a user's phone is "roaming" onto other networks, a mobile phone *can* see channel and signal information for other providers. For simplicity, we have ignored this case in this document although the same class of techniques can be used in this case as well.

```
for (every consecutive pair of GSM readings
within the window)
{
 Calculate Euclidean distance
 if (Euclidean distance > DistanceThreshold)
       numOfBadReadings++;
}

if (numOfBadReadings > BadReadingsThreshold)
       return “Mobile”;
else
       return “Stationary”;
```

Figure 2. Window Based Mobility Algorithm

2.2 Differentiating Driving vs. Walking

The same methods described for detecting motion can be adapted to differentiate types of movement such as between driving and walking, two key means of transport for many in the urban, western world. As one would expect, the rate of change in GSM signals is significantly greater when driving as opposed to walking. However, there are many times when the activity of driving can look very similar to walking. While walking, buildings, buses passing by, or other obstructions can cause drastic changes in signal strengths, which may may look similar to driving. In addition, city driving is typically stop-and-go traffic which can cause signals to seem similar to those seen in a walking activity. We use a two-level approach for detecting these types of movements to account for these types of anomalies.

The first level uses the RSSI spread and Euclidean distance techniques to determine if the rate of change in signal strengths is {STILL, SLOW, FAST}. As described before, we use a separate algorithm to determine if the device is stationary or mobile. The device being stationary is equivalent to the STILL value. If the device is considered to be mobile, we then apply a second algorithm. Instead of calculating the average spread across the various channels in the RSSI case, we look at the minimum and maximum of the averages in each channels’ signal strength. This allows us to see the actual change in each channel, important feature that can sometimes be masked out when averaging the values together. Using these calculated differences, if the value is above a threshold we consider movement to be FAST. Otherwise, movement is considered to be SLOW. Similarly, in the Euclidean distance approach, we look for several values above a threshold within a window of measurements. If we see the amount of measurements, we declare FAST movement. Otherwise, if our stationary algorithms determined we were mobile, the algorithm will output SLOW movement.

Second, we use the three different outputs from the basic movements in the first step to determine a final activity estimate. This second level is a state machine with three different states: STATIONARY, WALKING, and DRIVING. This state machine is designed to address the problem or different rates of movement in walking and driving activities (*e.g.* stop-and-go). The initial state is stationary. A slow movement will cause the state to transition to walking. Any slow or still movement input will keep the state in walking, unless movement is still for longer than a certain amount of minutes, we use 3 minutes for our threshold. A fast movement input from any state will always cause a transition to the driving state. Any still, slow, or additional fast movements will keep the activity as driving, unless movement is still for longer than our described threshold. Initial results of this approach are promising in capturing the human perceived activities of driving and walking.

3. Results

To test the accuracy of our mobility detection algorithms, we developed a set of tools running on a commodity mobile phone (Audiovox SMT 5600). The tools allow capturing series of labeled GSM measurements, or in other words, snapshots of visible GSM towers along with their respective signal strength values, as reported by the mobile phone.

Using our system, we recorded labeled traces of GSM measurements, consisting of periods of about 10 minutes of stillness followed by about 2-5 minutes of movement between places. The places in the trace include two places in our laboratory, three nearby coffee shops, a restaurant and a bookstore.

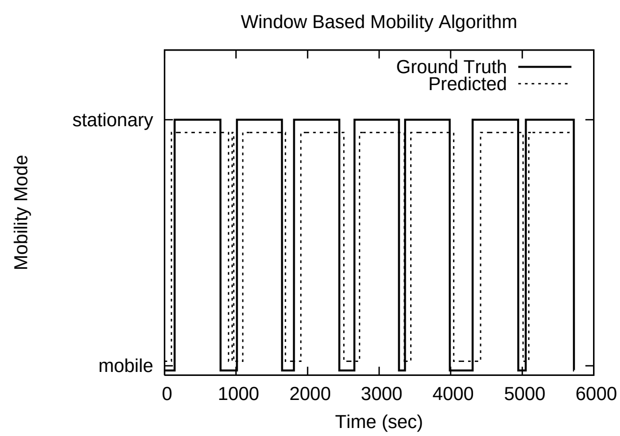

Figure 3. Results for Window Based Mobility Algorithm

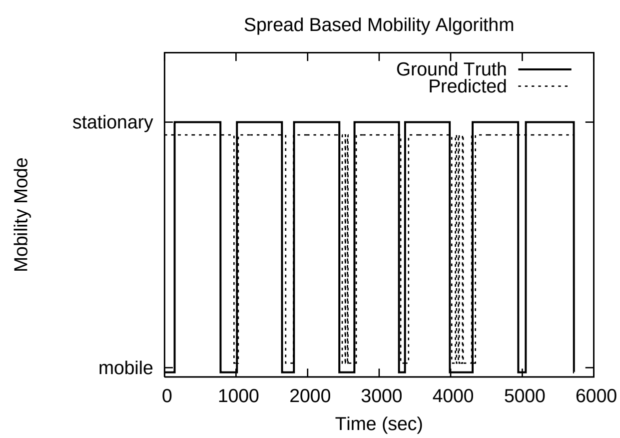

Figure 4. Results for Spread Based Mobility Algorithm

The accuracy of Window Based and Spread Based algorithms applied on one of the traces we collected is depicted on Figure 3 and Figure 4, respectively. The solid lines represent the ground truth and the dashed lines represent the output of our mobility detection algorithms. Both algorithms perform well, recognizing most of the periods of stillness and mobility correctly. Window Based algorithm successfully differentiates between all periods of stillness and movement, but has a slight lag in making the decision. The lag is the consequence of a larger window size, necessary for accumulating enough measurements to make the correct decision. Spread Based algorithm, on the other hand, does not suffer from the lag, having much smaller window size. However, it is more jumpy in the presence of non-perfectly stable signals and sometimes misses short periods of mobility. The results of applying the algorithms on additional data traces we collected looks similar and are therefore not included in the paper.

Although our algorithms performed well on our perfectly crafted data traces, it is still to be discovered whether they to perform as well in the real world. We are in the process of deploying the algorithms to be used by several location-aware applications currently developed in our lab.

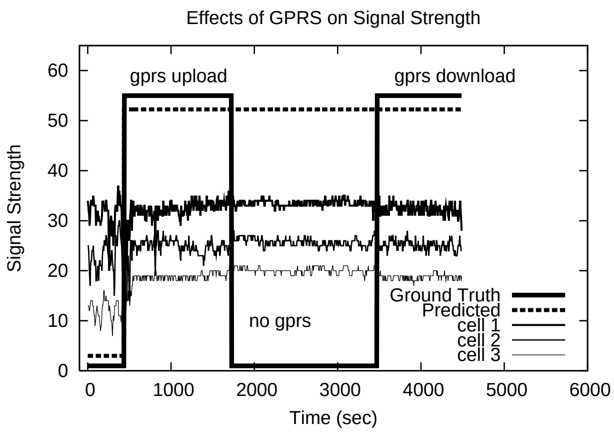

Figure 5. Effect of GPRS on Signal Strength

3.1 Effects of Using GPRS on Mobility Detection Algorithms

Applications running on a mobile phone may periodically use GPRS (General Packet Radio Service, a data service that runs in conjunction with GSM mobile phone networks) for data communication. For example, location-aware application may send a message from one user to another, download a web page, or send aggregate statistics back "to the web" for later browsing. Since accuracy of our mobility detection algorithms depends on stability (or predictability) of signal strength values, we tested the effects of downloading and uploading GPRS data on signal strength as measured by the mobile phone. Figure 5 depicts the measured signal strength values of the 3 strongest cells while not using GPRS, uploading data and downloading data. The phone was lying still on the table during the testing period. The thick solid line differentiates between the different states and the thick dashed line represents the output of the Window Based mobility algorithm.

The results show that there is a small local instability of signal strength when an upload or a download processes starts running, but the signal strength values stabilize shortly thereafter. There also appears to be no difference in reported signal strength values between the period when the GPRS connection was open and closed. The Window Based algorithm performed perfectly on this data set, reporting that the phone is stationary. In the future, we plan to investigate effects of GPRS on signal strength values in mobile environments, again using applications that we are developing in house.

4. Conclusions and future work

In this paper, we presented two mobility detection algorithms. Both work reasonably well in the environment that they have been designed for—urban areas with many cellular towers. In addition, both seem to suffer little in the face of GPRS usage by the user. In the future, we plan to continue working to evaluate our algorithms in different, real-world settings.

We also plan to derive a scheme for automatic place naming such that when the phone is stationary, we can attempt to automatically generate a sensible "name" of the place for the user. This may be accomplished in a variety of ways. First, we may use simple rule-based naming. For example, at 1:00am you are most likely at home, and at 11:00am on the weekday you are most likely at work. Some history information about the human carrying the phone may help in this approach as well. Second, we may utilize MapPoint like web-services for translating latitude/longitude coordinates into meaningful names. Of course, for this approach to work, the phone has to somehow obtain its latitude/longitude coordinates, although the techniques for this type of wide area localization have been explored [3]. We hope in the future to be able to combine motion detection of a mobile phone, automatic place naming and wide area localization services.

5. ACKNOWLEDGMENTS

We would like to thank all the members of the Place Lab team for their work in helping develop these algorithms, test them, and make them practical for use in real applications: Mike Chen, Jon Froehlich, Jeffrey Hightower , James Howard, Anthony LaMarca, and Fred Potter.

6. REFERENCES

[1] A.K. Dey, D. Salber and G.D. Abowd. *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications*. Anchor article of a special triple issue on context-aware computing. Human-Computer Interaction (HCI) Journal 16(2-4). pp. 97-166. 2001. Also published separately by Lawrence-Erlbaum as an edited book in January 2002, ISBN 0805896856.

[2] Gregory D. Abowd and Elizabeth D. Mynatt *ACM Transactions on Computer-Human Interaction*, Special issue on HCI in the new Millenium, **7**(1):29-58, March.

[3] Place Lab: Device Positioning Using Radio Beacons in the Wild. Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello and Bill Schilit. *In proceedings of Pervasive 2005, Munich, Germany).*

[4] Learning Significant Locations and Predicting User Movement with GPS. Daniel Ashbrook and Thad Starner. Proceedings of IEEE Sixth International Symposium on Wearable Computing (ISWC02). Seattle, WA. October 2002.

[5] Laasonen, K., Raento, M., Toivonen, H.: Adaptive On-Device Location Recognition. In: Proc. Pervasive 2004, LNCS 3001, Springer Verlag (2004) 287–304.

[6] Learning and Recognizing the Places We Go. Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian Smith and Jeff Hughes. *In Proceedings of Ubicomp 2005, Tokyo, Japan. September 2005.*

[7] Social Disclosure Of Place: From Location Technology to Communication Practices. Ian Smith, Sunny Consolvo, Anthony LaMarca, Jeffrey Hightower, James Scott, Timothy Sohn, Jeff Hughes, Giovanni Iachello and Gregory D. Abowd. *In proceedings of Pervasive 2005, Munich, Germany).*

[8] Friedman, B., & Thomas, J. C. (2001). Is ignorance bliss? Informed consent online. *Extended Abstracts of the CHI 2001 Conference on Human Factors in Computing Systems* (pp. 215-216). New York: Association for Computing Machinery.