AAMPL: Accelerometer Augmented Mobile Phone Localization

Andrew Ofstad Emmett Nicholas Rick Szcodronski Romit Roy Choudhury
Dept. of Electrical and Computer Engineering
Duke University
{andrew.ofstad, emmett.nicholas, ras18, romit}@ee.duke.edu

ABSTRACT

A variety of mobile phone applications are on the rise, many of which utilize physical location to express the *context* of information. This paper argues that physical location alone, unless remarkably precise, may not be sufficient to express this context. Even slight localization errors may cause a mobile phone to be placed in a grocery store, as opposed to its actual location in an adjacent coffee shop. Applications such as location specific advertisements, can get affected. This paper proposes accelerometer augmented mobile phone localization (AAMPL), a system that uses accelerometer signatures to place mobile phones in the right context. Early evaluation on Nokia N95 phones shows that AAMPL can correct locations derived from Google Maps. We believe that AAMPL can be extended to additional sensors (like light and sound) to further aid GPS-free localization.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems; C.2.4 [**Computer Communication Networks**]: Network Protocols

General Terms

Design, Experimentation, Measurement, Human Factors

Keywords

Mobile phones, accelerometers, localization, energy, sensing

1. INTRODUCTION

The proliferation of mobile phones is motivating a variety of pervasive, context-aware, social applications. Examples include *Micro-Blog* [1], *MetroSense* [2], *Place-Its* [3], *PeopleNet* [4], *MyExperience* [5], and several others. Many of these applications exploit the location of the user as a primary indicator of context. While most applications assume GPS based localization, recent investigations are beginning to expose several tradeoffs when using GPS. Poor indoor coverage with GPS, alongside its high energy consumption [1], warrants alternate localization methods. Some alternates, proposed by Place Lab and others [6, 7], utilize WiFi/GSM based fingerprinting and triangulation to localize mobile phones. While the improvements in coverage and energy are encouraging, they arise at the expense of higher localization error ranging from 50 to 500m. Such error margins may exceed the tolerance thresholds of several applications. More importantly, even if these error bounds are reduced to a few meters, they may still be insufficient to capture the user's context. We present our argument next.

Consider an example application – *Micro-Blog* [1]. One feature of Micro-Blog is that location-specific queries are geocast to mobile phones present at the corresponding location. For example, an Internet user may query about the "availability of free WiFi" at a particular coffee shop. If this query reaches phones in the adjacent grocery store, the replies to this query may be inapplicable. Existing localization schemes, even with accuracy of few meters, may not be able to avoid this. The physical separation between two phones may be small (few meters), and yet they may be in logically different contexts (opposite sides of the wall separating the coffee shop and the grocery store). We argue that localization needs to be performed across two domains, namely *physical* and *logical*. This paper presents a framework, AAMPL, that accepts the approximate physical location of a mobile phone, and augments it with context-aware logical localization. The main idea is described as follows.

Modern phones are equipped with a large number of sensors, including cameras, microphones, accelerometers, and health-monitors. These sensors are natural candidates for sensing the context in which a user is situated. Automatic access to such context information can be exploited towards localization. For instance, a user's movement (derived from the phone's accelerometer) may be effective for predicting whether the user is in a coffee shop or a grocery store. Since geographical localization can narrow down the choices to a few nearby contexts, accelerometer readings may be effective in selecting the correct one from among them. Hence, the "WiFi availability" query can be correctly guided to phones in the appropriate coffee shop. While classifying accelerometer signatures is one axis of augmentation, one may envisage multi-dimensional context sensing through light and noise signatures. This paper develops a framework for combining physical and logical localization via real-time classification of accelerometer readings. Evaluation on Nokia N95 phones shows that AAMPL was able to correct physical locations derived from phone GPS and Google Maps.

MELT'08, September 19, 2008, San Francisco, California, USA.

2. RELATED WORK

We divide the related work into two following branches:

Activity Recognition: Several papers have studied activity recognition using accelerometers. Bao and Intille showed that it is possible to detect 9 everyday activities using 2 biaxial accelerometers mounted on a user [8]. Other research has investigated augmenting accelerometer data with on-body audio sensors to detect recurring human behaviors [9]. Project SATIRE [10] extends these results by implementing a sensor mote based architecture that takes advantage of recognizable accelerometer signatures. The authors attempt to develop accelerometer-equipped smart clothes that could trigger alerts or record daily activities. Several projects have explored applications of context-aware mobile devices [11, 12, 13]. Project SenSay [14] aims to provide a context-aware phone that adapts its state based on the environmental and physiological changes. It uses externally mounted light, motion, and sound sensors to provide the contextual information. Inspired by these findings, AAMPL leverages accelerometer signatures towards augmenting localization. Readings from the on-board accelerometer of a mobile phone are transmitted over WiFi/GSM connections to a central server, which then localizes the phone in real time. The architecture is made energy-aware, permitting AAMPL to be a deployable underlay to next generation applications.

Localization: With increasing location based applications for mobile phones, localization has been an exciting topic of research. Since GPS is energy-hungry and has poor accuracy in indoor and dense urban areas, other localization techniques have been explored [6, 7]. Augmenting GPS with WiFi and/or GSM data has been shown to aid in localization. Also, external sensors can aid in placing a user in a specific location [12]. The advantage of AAMPL is that it unifies GPS based localization with other sensory inputs (accelerometer in the case of our implementation). While GPS offers an approximate location, an accelerometer based classifier effectively discriminates between possible logical locations. We present the design, implementation, and the evaluation of the system, next.

3. ARCHITECTURE AND DESIGN

Figure 1 presents the client-server architecture of AAMPL. Without loss of generality, we describe AAMPL for localizing phones located in urban business areas (shops, stores, malls). However, AAMPL may be equally effective in many non-business indoor environments, such as homes, schools, or workplaces.

3.1 Client-Server Architecture

The AAMPL client runs on a Nokia N95 smartphone, and uses a 3G or WiFi Internet connection to communicate to the server. The client and server operate in conjunction to first classify the type of business a user is in, based on accelerometer data. The server then uses this classification to choose from a list of businesses near the phone's physical location. The main operations are as follows.

The client collects X, Y, and Z-axis accelerometer data at one second intervals. These data points are logged in memory over the duration of a minute, and are then passed

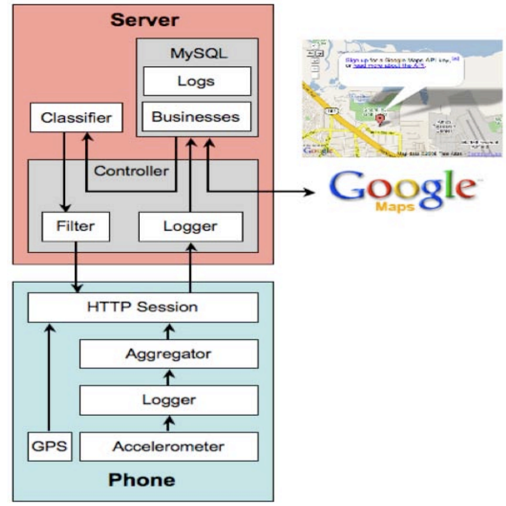

Figure 1: Block diagram of the AAMPL architecture

through a filtering layer that classifies each accelerometer data point into an "action state", which, in our implementation, is either "sitting" or "standing". This classified accelerometer data is then aggregated into three features that can be used at the server to classify a location (details presented later). The aggregated features are included in a data packet, along with a timestamp and the physical coordinates derived from GPS, WiFi, or GSM. The data packet is sent to the server via an HTTP POST request.
The server is mainly responsible for classifying the accelerometer data, and using the results to refine the physical location of the device. Logically, the server application is divided into three different components: (1) A MySQL database stores business and mobile activity information, keeping an updated list of nearby businesses and client data logs. (2) A classifier is responsible for classifying the business category based on a set of recent points in the "logs" database. (3) A controller coordinates events across the different components, and communicates to the mobile client when necessary. When a packet first arrives, it is added to the "logs" table, which contains a list of recent accelerometer data points and location coordinates, provided by the mobile client. The server dynamically queries Google Maps for an updated list of businesses near these location coordinates. The controller then examines the database to determine whether the latest accelerometer data point is within a "span," which is a sequence of data points with close spatial and temporal proximity to one another. The new data point is appended to the most recent span if appropriate. If not, a new span is created. Note that it is assumed that all the points in a single span come from the same business location. The controller then sends all the points from the latest span to the classifier, which uses the accelerometer features to classify the business category of the entire span. The controller can then use this classification to filter out the nearby businesses in the database, only returning businesses in the given category. The filtering module also uses other relevant pieces of information to filter out unlikely businesses. For example, business hours, stored in the "businesses" database table, are used to filter out locations that are not open. This list is returned to the phone and displayed on the screen.

3.2 Energy Awareness

Sending a single bit of information through a wireless medium (Wi-Fi or GPRS) uses the same amount of power as 800 clock cycles of processing [15]. Thus, it is important to limit the amount of data sent back to the server for classification. These considerations motivate the need for data aggregation at the client side. In AAMPL, the accelerometer classifier is set up such that 60 128-bit accelerometer data points (960 bytes in total) can be aggregated into 24 bytes that provide most of the information necessary for business classification. An improvement factor of 40 is immediate, substantially reducing the energy costs.

3.3 Classifier Design

The classification system is responsible for classifying accelerometer data from a span (where each span corresponds to a business category). In the current implementation, we decided to classify each span as belonging to one of three classes: restaurant, fast food, or retail store. This classification system required the collection and analysis of a training dataset. So, an accelerometer data-logging application was programmed for the N95 that collected samples from the 3-axis accelerometer every second. (Each accelerometer data point contains raw signed acceleration integers for each of the X, Y, and Z axes, with a conversion rate of $decimal64$ to $1g$ of acceleration.) Data was collected from 16 locations in the Raleigh/Durham metropolitan area. For each business, logging began upon entering the business, and was stopped upon exit. The phone was placed in the right pocket of a pair of jeans. In addition to business-specific data collection, measurements were taken for two different action states that are used in the classification. About 1000 data points (spanning 1000 seconds) were collected while standing (i.e. walking around and standing still) and sitting. The phone was placed in the pocket in a variety of different angles and configurations, although each time it was approximately upright in the pocket with the screen facing outward. When collecting standing data, the tester stood and walked with a variety of different gaits. Figure 2 shows this training data. These two states, as can be seen from the data, appear as distinct, separable distributions. Training data from the three business categories also had interesting distributions with distinguishable patterns (Figure 3).

The classifier design was faced with a energy versus complexity tradeoff. While classifying patterns in different business distributions demand high computation cost (favoring server side classification), reducing the energy cost of data transmission favors classification at the phone client. In view of this, AAMPL adopts a two stage classifier, with the first stage occurring on the phone and the second on the server. In the first stage, each accelerometer data point is classified as either sitting or standing, using a Bayesian classifier. The classified data allows for the extraction of three aggregate features that can be used in the second stage of business category classification. The first feature is the percentage of points that are in the standing state, the second is the average variance over all three axes for points in the standing state, and the third is the total number of points taken from the business, which translates directly to the amount of time in seconds. These aggregate data features describe the data in a concise manner, and provide most of the information needed to make a classification. For each such classifica-

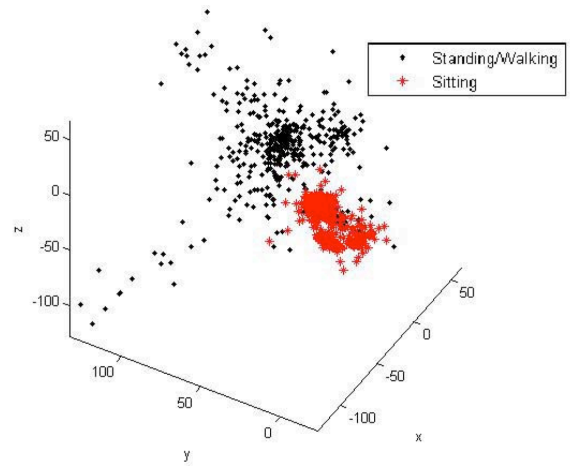

Figure 2: X-Y-Z accelerometer training data for standing (black) and sitting (red)

tion, only a small amount of meta-data needs to be sent to the server. The server, which keeps a database of business category training points, then performs a multi-class, three-dimensional classification on the test point. This second stage of classification is done with a k-nearest neighbors (KNN) classifier. Figure 4 shows a flow chart of this complete classification scheme. A Bayesian classifier was used for the first stage. Once a class conditional probability distribution function is created, the Bayesian classifier runs in linear time for a set of data points. In addition, the Bayesian classifier is easily extended to multi-class, multidimensional classifications with little additional computation complexity.

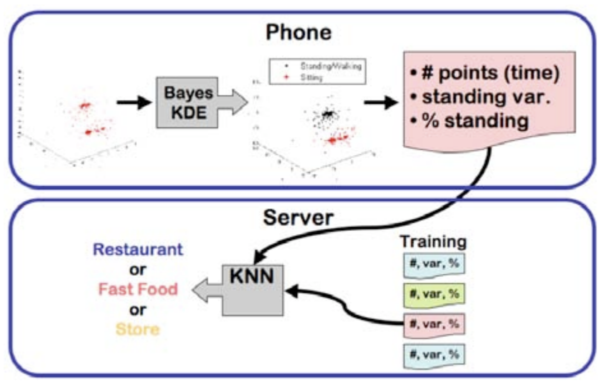

Figure 4: Business classification flow chart

For the Bayesian classifier, it was assumed that the prior distributions of sitting and standing are equal ($P(\omega_i) = .5$). The classification then degenerates to a simple comparison between the two class conditional probabilities for sitting (Eqn. 1) and standing (Eqn. 2):

$$If\ \frac{P(x|\omega_{sitting})}{P(x|\omega_{standing})} > 1,\quad then\ choose\ sitting \qquad (1)$$

$$If\ \frac{P(x|\omega_{sitting})}{P(x|\omega_{standing})} < 1,\quad then\ choose\ standing \qquad (2)$$

To obtain the above class conditional probabilities, non-

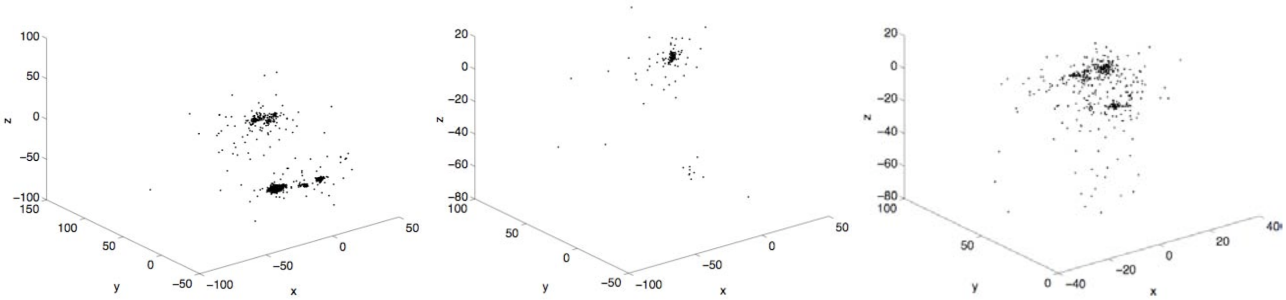

Figure 3: X-Y-Z accelerometer measurements in (a) restaurant, (b) fast food joint, and (c) retail store

parametric kernel density estimation (KDE) was used with a Gaussian window function on the training set in Figure 2. 50x50x50 three dimensional class conditional probabilities for sitting and walking were saved as text files for use as Bayesian classifier lookup matrices on the phone. After every point in a dataset is classified as sitting or standing, the percent sitting, number of points, and variance features are calculated with the aggregator. The variance of standing points is approximated by the average of the X, Y, and Z axes variances. The second stage classifier is responsible for taking an aggregate feature point, as produced by the first stage classifier and aggregator, and classifying it into a single business category. While this classification lends itself to a number of different multi-class classification techniques, the small data set limited the effectiveness of more complicated methods. For this reason, a relatively simple KNN classifier was chosen for this stage. The training set for this classifier can be seen below in Figure 5. This training set was classified and aggregated using the first stage classifier described above. The set is first normalized such that all the dimensions $\in$ [0,1].

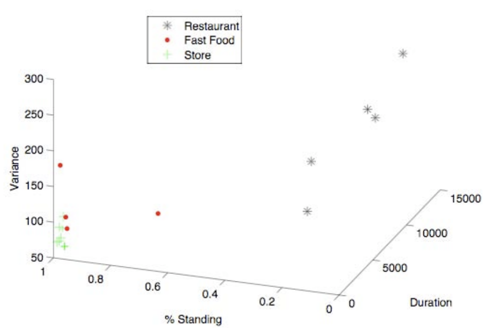

Figure 5: Business classification training data for restaurants, fast food, and retail stores.

While our dataset is very limited, one could imagine a larger dataset that occupies this feature space more completely. This larger training set could be used for any number of advanced classification techniques that are trained to minimum validation error, and these techniques could be implemented on the server without having to change anything on the mobile phone client. However, for this particular application with such a small dataset, a KNN classifier was sufficient. If a larger training dataset or more features becomes available, this information could be easily incorporated to enhance the accuracy of the classifier.

4. SYSTEM IMPLEMENTATION

The AAMPL phone client was implemented on the Nokia N95 smartphone in approximately 3300 lines of code using the Carbide C++ IDE and compiler for Symbian OS. The server side was implemented in approximately 1650 lines of code across PHP, Perl, and MySQL. The implementation details are presented in the next two sections.

Phone Client: The Nokia N95 smartphone, currently the flagship of the Nokia product-line, runs on the S60 3rd Edition platform, using Symbian OS v9.2. The AAMPL software requires access to the phone's location and network services, which are provided by the S60 SDK. We obtained an Open Signed Online certificate, to be used for development purposes only, from the Symbian Signed website [16].

Web Infrastructure: We installed a lighttpd 1.4.18 server to offer HTTP access to the client, as well as to interface with the back-end MySQL database server. Relevant installations on the server include PHP 5.2.4, Perl 5.8.8, and MySQL 5.0.24a. Logs, sent from the client every minute via POST requests, are received by the server and communicated to the database server through PHP. After each log is added to the database, the entire collection of logs is analyzed using PHP and Perl scripts. More specifically, Perl is used for the two classifiers, and all other server-side logic is implemented in PHP. After a request is received and processed by the server, a response is sent to the query-originator (the AAMPL phone client). For testing purposes, the response contains a list of possible businesses, ordered by likelihood, in which the client is currently located.

5. EVALUATION

We report early evaluation of AAMPL from a set of local businesses in the Duke University campus. We compare AAMPL to a GPS-only approach, in which the client is assumed to be in the business geographically nearest to the most recent GPS measurement. The metric for success is business prediction accuracy.

Classifier Evaluation: Overall, AAMPL serves its specific purpose well. The first stage classifier was tested for its ability to classify whether each accelerometer point was from the standing or sitting state. A 50-50 cross-validation was used for this, i.e., half of the dataset was randomly selected as training data, leaving aside the other half as test data. This validation produced classifications that were correct 98.9% of the time. While this classifier was not tested with data from outside the training set, our inspection/verification of the classified training data gives us confidence that the accelerometer will be able to well discriminate sitting from standing. Further, these results help to validate our decision to use a Bayesian classifier instead of a Hidden Markov Model for this stage of classification. The next stage of classification also produced encouraging results. Leave-one-out validation was used to train and test the classifier with the data collected. Each business was taken out of the training set, and then classified with the remaining data in the set. Each business was classified correctly using a simple nearest neighbor classification (KNN with $k = 1$). Of course, if the dataset grew larger, this would probably not be the case – k would need to be increased or another classification method would need to be employed.

Experimental Setup: The back-end "businesses" database contains geographic information for 25 businesses near the Ninth Street business district in Durham, NC. Figure 6 shows the stretch of Ninth Street on which we conducted part of our experiments. The (red) pushpins correspond to the *actual* locations of three adjacent businesses: Regulator Bookshop, Blue Corn Café, and Bean Traders Coffee. The (blue) balloons correspond to the positions of each business as reported by the *Google* Maps API. The Google-reported physical coordinates are moderately accurate, but the logical discrepancies with the actual businesses can be significant. For example, the location of Regulator Bookshop is reported as in front of the Blue Corn Café. We carried the AAMPL-equipped phone in our trouser pockets and spent varied amounts of time in or around the three locations. Since the businesses were of different types (store, restaurant, and fast food, respectively), the accelerometer signature provided useful information for localization.

Figure 6: Google satellite view of Regulator store, Blue Corn restaurant, and Bean Traders fast food. Coordinates from Google Maps marked with balloons, and actual coordinates marked with pushpins.

5.1 Experimental Results

Overall, AAMPL performed well even with our simple classification schemes. Table I summarizes a subset of our

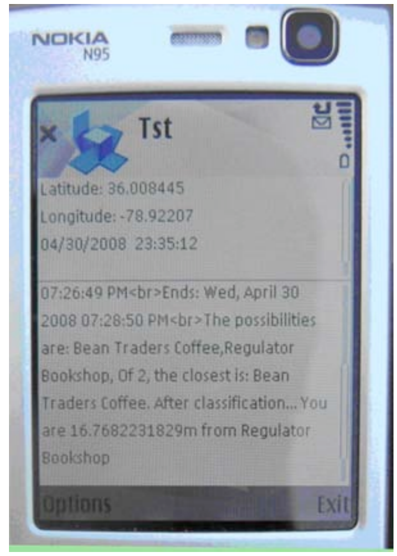

Figure 7: AAMPL classification on Nokia Phone

experiment results. Trials 1 and 2 constitute a single visit to Bean Traders Coffee. The GPS signal was lost upon entering the shop, so the measured distances to the three businesses are equal in the two trials. Though the location information remained constant, the accelerometer signature continued to provide useful information. After first entering the shop, AAMPL classified the signature as that of a *store* – incorrectly predicting Regulator Bookshop. However, after several minutes, the classifier recognized the accelerometer signatures as those of a *fast food* shop, thus correctly identifying Bean Traders Coffee. Trials 3 and 4 occurred during a sit-down dinner at Blue Corn Café. As was the case with Bean Traders Coffee, the GPS signal was initially lost upon entering the restaurant. As a result, the ensuing logs defaulted to the last valid set of GPS coordinates, but the accelerometer signature continued to change. In trial 3, the GPS-only approach indicated that Regulator Bookshop was closest. Since we had been sitting down for several minutes, AAMPL successfully recognized the signature of a restaurant and predicted Blue Corn Café as well. About 20 minutes into the dinner, a stray GPS signal was perhaps received, yielding the inaccurate distance measurements of trial 4. In this trial, the GPS-only approach correctly identified that we were currently in Blue Corn Café. AAMPL continued to identify the restaurant setting, and remained correct as well. Trial 5 was conducted in The Regulator Bookshop, which is classified as a store. The nearest location, as predicted by GPS-only localization, was Bean Traders Coffee. However, AAMPL correctly classified Regulator as a store, which enabled a correct prediction of our location. Figure 7 shows the AAMPL prediction screenshot from our Nokia N95 phone implementation. This was taken immediately after leaving Regulator Bookshop.

6. LIMITATIONS AND FUTURE WORK

Strengthening AAMPL: Clearly, reported results are preliminary and hence indicative (not conclusive) of the potentials of AAMPL. Our ongoing work is strengthening AAMPL in a variety of ways, including (1) larger action states (be-

Trial	Distance to Regulator Bookshop	Distance to Blue Corn Café	Distance to Bean Traders	Classification	Actual Location	GPS Prediction	AAMPL Prediction
1	11.8m	36.2m	15.1m	Store	Bean Traders	Regulator Bookshop	Regulator Bookshop
2	11.8m	36.2m	15.1m	Fast Food	Bean Traders	Regulator Bookshop	Bean Traders
3	10.7m	18.7m	25.6m	Restaurant	Blue Corn Café	Regulator Bookshop	Blue Corn Café
4	85.9m	59.4m	95.3m	Restaurant	Blue Corn Café	Blue Corn Café	Blue Corn Café
5	14.3m	26.2m	11.9m	Store	Regulator Bookshop	Bean Traders	Regulator Bookshop

Table 1: Experimental Results ("Distance" indicates difference between GPS reading and Google estimate)

yond sitting and standing), (2) addition of more location categories including non-business locations, (3) addition of more features derived from accelerometers and other sensors. Moreover, ladies may carry mobile phones in their handbags, and accelerometer signatures may be less indicative of activity. We intend to investigate these issues in future.

Light, Sound, and Compasses: We are investigating the applicability of light and sound signatures towards context identification. Our belief, partially validated by preliminary measurements, is that different contexts may exhibit dissimilar photo-acoustic ambiances. While any individual signature is too noisy an indicator, their intersection may provide stronger correlation with the context. Of course, cameras may be mostly in pockets, and the opportunity may arise only when the user takes it out for a phone call, or places it on the table. Hence, acoustics may be a more reliable source, and should perhaps be opportunistically augmented by light signatures. We are also investigating the utility of compasses to understand a user's orientation, and correlate that to the sitting layouts in restaurants and coffee shops.

7. CONCLUSION

Many mobile phone applications require a user's context to execute its functions. Rough location coordinates may not be a flawless indicator of this context because the spatial separation between these contexts may be arbitrarily small. As a result, marginal errors in physical coordinates may result in incorrect contextual information. This paper proposes AAMPL – a framework that utilizes phone accelerometers to augment physical localization services. The main idea is to shortlist a list of potential logical locations from GPS/alternate localization methods, and then choose from these logical locations by exploiting dissimilar human movements from each of them. In the restricted case of business localization, we show that accelerometer signatures from restaurants, coffee shops, and retail stores can be separable, thereby refining coordinates available from GPS or Google Maps. In conjunction with other sensor signatures, such as light and sound, the confidence of localization may be further increased. Our ongoing work is investigating these possibilities towards GPS-free context-aware localization.

8. REFERENCES

[1] Shravan Gaonkar, Jack Li, Romit Roy Choudhury, Landon Cox, and Al Schmidt, "Micro-blog: Sharing and querying content through mobile phones and social participation," in *ACM MobiSys*, 2008.

[2] Shane B. Eisenman, et. al. "Metrosense project: People-centric sensing at scale," in *First Workshop on World-Sensor-Web (WSW'2006)*, Oct, 2006.

[3] Timothy Sohn, et. al. "Place-its: A study of location-based reminders on mobile phones," in *Ubicomp*, 2005.

[4] Mehul Motani, Vikram Srinivasan, and Pavan S. Nuggehalli, "Peoplenet: engineering a wireless virtual social network," in *ACM MobiCom*, 2005.

[5] Jon Froehlich, et. al. "Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones," in *ACM MobiSys*, 2007.

[6] Anthony LaMarca, et. al. "Place lab: Device positioning using radio beacons in the wild," in *Pervasive*, 2005.

[7] Y. Chen, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *ACM Mobisys*, 2005.

[8] L. Bao and S. Intille. Activity Recognition from User-Annotated Acceleration Data. In Proceedings of Pervasive Computing (PerCom), 2002.

[9] D. Minnen, et al. Recognizing and Discovering Human Actions from On-Body Sensor Data. In International Conference on Multimedia and Expo, 2005.

[10] R. Ganti, et al. SATIRE: A Software Architecture for Smart AtTIRE. In Proceedings of ACM Mobicom, 2003.

[11] G. Chen and D. Kotz. A Survey of Context-Aware Mobile Computing Research. Dartmouth CS Tech. Report 2000.

[12] C. Randell and H. Muller. The Well Mannered Wearable Computer. Personal and Ubiquitous Computing 2002.

[13] G.D. Abowd, et al. A mobile Context-Aware Tour Guide. ACM Mobile Networks 3, 1997.

[14] D. Siewiorek, et al. SenSay: A Context-Aware Mobile Phone. In IEEE International Symposium on Wearable Computers, 2003.

[15] S. Madden, M. Franklin, J. Hollerstein, and W. Hong. TAG: a Tiny AGrigation Service for Ad-Hoc Sensor Networks. In USENIX OSDI, 2002.

[16] Symbian Signed. <https://www.symbiansigned.com>.