Pocket, Bag, Hand, etc. - Automatically Detecting Phone Context through Discovery

Emiliano Miluzzo[†], Michela Papandrea[§], Nicholas D. Lane[†], Hong Lu[†], Andrew T. Campbell[†]
[†]CS Department, Dartmouth College, Hanover, NH, USA
[§]SUPSI, Manno, Switzerland

ABSTRACT

Most top end smart phones come with a handful of sensors today. We see this growth continuing over the next decade with an explosion of new distributed sensor applications supporting both personal sensing with local use (e.g., healthcare) to distributed sensing with large scale community (e.g., air quality, stress levels and well being), population and global use. One fundamental building block for distributed sensing systems on mobile phones is the automatic detection of accurate, robust and low-cost *phone sensing context*; that is, the position of the phone carried by a person (e.g., in the pocket, in the hand, inside a backpack, on the hip, arm mounted, etc.) in relation to the event being sensed. Mobile phones carried by people may have many different sensing contexts that limit the use of a sensor, for example: an air-quality sensor offers poor sensing quality buried in a person's backpack. We present the preliminary design, implementation, and evaluation of *Discovery*, a framework to automatically detect the phone sensing context in a robust, accurate and low-cost manner, as people move about in their everyday lives. The initial system implements a set of sophisticated inference models that include Gaussian Mixture Model and Support Vector Machine on the Nokia N95 and Apple iPhone with focus on a limited set of sensors and contexts. Initial results indicate this is a promising approach to provide phone sensing context on mobile phones.

1. INTRODUCTION

The recent explosion of smartphones (e.g., Nokia, Apple iPhone, and Android phones) with embedded sensors is enabling a new generation of personal and environmental sensing applications [1, 2, 3, 4]. These applications are built on multi-faceted real-time sensing operations that require increasing computation either on the phone [2] or backend servers [3], or a combination of both [1]. As the demands of these new distributed sensing applications built on commercial phones is better understood in terms of their needs for on-phone sensors, computation and communication resources, a number of important challenges are emerging. Because these continuous sensing applications are extremely resource hungry in terms of sensing, computation and communications (with backend servers) there is need to drive the operation of the phone in a more intelligent manner. We believe efficiently computing the low level context of the phone, that is, the position of the phone carried by a person (e.g., in the pocket, in the hand, inside a backpack, on the hip, arm mounted, etc.) in relation to the event being sensed - which we call the *phone sensing context* - is a fundamental building block for new distributed sensing applications built on mobile phones. These observations have grown out of our implementation of CenceMe [1] and SoundSense [2], two continuous sensing applications implemented on Nokia and Apple phones. While there has been significant research in the area of context aware applications and systems, there has been little work on developing reliable, robust, and low cost (i.e., in terms of energy efficient and computational costs) algorithms that automatically detect the phone sensing context on mobile phones. We envision a future where there are not only personal sensing applications but we see the mobile phone as enabling global sensing applications where the context of the phone in relation to the sensing event is crucially important.

The different context impacts the fidelity of a sensing application running on mobile phones. For example, the camera is of little use in the pocket but the microphone might still be good [2]. Researchers are developing new sensors for the phones that we imagine will be available over the next decade, these include CO_2 and pollution sensors [5]. If the phone is carried inside the pocket or a backpack, an application relying on CO_2 or pollutants measurements would perform very poorly given that the phone is not exposed to open air. A better position for such sensing would be out of the pocket when the phone is exposed to a more suitable context for sensing. Similarly, if the accelerometer readings of the phone are used to infer the person's activity, the accelerometer would report different data if the phone is mounted on the arm or clipped to the belt. This is because, given the same activity, such as walking for example, arm swings would activate the accelerometer much more strongly for an arm-mounted phone than on the belt, where the phone oscillates more gently. In both cases a mechanism to infer the context of the mobile phone is needed in order to make the applications using the CO_2 or pollution sensor and the accelerometer, respectively, react appropriately. We envision a learning framework on the phone that is more sophisticated than what is implemented today. For example, when sensors report different sensor readings according to the position on the body, such as the accelerometer, the application's learning engine should switch to different classification algorithms or sensor data treatment policy in order to meet the application requirements.

Today the application sensing duty-cycle is costly because it is not driven by the phone sensing context, therefore, it is costly in terms of energy usage for sensing, computation and potentially communications if the inference is done on the backend, as in the case with split-level classification [1]. By offering system developers accurate phone sensing context prior to running classification algorithms, very low duty-cycle continuous sensing application systems are possible. In this case, the phone sensing context mechanism would refrain the application from activating a power hungry sensor if the context is unsuitable (e.g., don't activate the pollution sensor if the phone is not out of the pocket) or it may weight real-time sensor readings or inferences based on knowledge of where the phone is on the body (e.g., if the microphone is needed to measure human activity [2] and it is in the bag).

In this paper, we discuss Discovery, a framework that addresses the context problem supporting mobile phone-based sensing with improved accuracy and lower duty-cycle systems. Discovery is designed to automatically detect the phone sensing context as people move about in their everyday lives. Automatic context detection is a primary issue for mobile phone sensing applications because prompting the user to provide information about the position of the mobile phone on the body is not a viable and scalable solution. Phone sensing context is an important building block toward the successful implementation of personal, social, and public sensing applications on mobile phones and the work in this paper, while preliminary, provides important steps towards the goal of providing reliable phone sensing context. This paper is organized as follows. Section 2.1 contains the motivation of this work, while details of the approach taken in Discovery are discussed in Section 2.2. Preliminary evaluation results are discussed in Section 3. Future directions are reported in Section 4 and the related literature in Section 5, before concluding in Section 6.

2. DISCOVERY FRAMEWORK

In what follows, we discuss some challenges phone sensing context presents, its preliminary design and implementation as part of the Discovery framework, as shown in Figure 1.

2.1 Phone Sensing Context

Accurate, robust and low duty-cycle detection of phone sensing context is an important enabler of distributed sensing applications on phones, in particular, continuous sensing applications that sample sensors, make inferences, and communicate with the backend services in real-time.

Assume mobile phones are equipped with pollution, CO_2, or more specialized environmental sensors as we imagine [5]. Measurements from any of these sensors would most likely be impeded by the presence of clothing or fabric (e.g., phone inside the pocket or backpack) or by a short time interval the sensors are exposed to an ideal sensing context (i.e., phone in hand or exposed to open air). Therefore, phone sensing context detection would improve the sensing system performance. We could stop the system from activating the sensors when the quality of the sensor data is likely to be poor (e.g., phone inside the pocket). This would help reduce the sensing duty-cycle improving the battery lifetime of the phone, which continuos sensing application significantly limit today (e.g., phones running CenceMe [1] were initially limited to only 6 hours of operation). We could inform the system when a suitable sensing context is triggered or detected (e.g., phone taken out of the pocket) to maximize the accuracy and robustness of the sensing application which would then take advantage of the new context for collecting as many sensor readings as possible. It is evident the importance of the phone sensing context role in driving mobile phones sensors duty-cycle lower.

Another reason to provide phone sensing context as a low level service on phones is to improve the inference fidelity of distributed sensing applications. Although previous work [6] shows that it is possible to obtain reasonably good activity classification accuracy when using training data from sensors mounted on different parts of the body, it is not clear how an activity classifier would perform when the device is a phone, not specifically mounted (but moving as a dynamic system), and operates in noisy, everyday environments that people find themselves in, rather, than under laboratory test conditions. Many questions remain. Would training data from many activities and different parts of the body make a single classification model accurate enough? To avoid excessively diluting the training data set, would it not be preferable building a classification model for each single activity and position of the mobile phone on the body and then switch models according to the detected phone sensing context? For example, a system could have a "walking" activity classification model for when the mobile phone is in the pocket, in the person's hand, and in the backpack and use one of the models according to the detected phone sensing context. Results obtained from experimentation in [1] show, for example, that activity classification accuracy varies when the phone is carried in the pocket or in the hand. A system that used phone sensing context to drive the classification model by switching in the right technique would alleviate this problem. We believe this is of importance now that smart phones are growing in sensing and computational capability and new demands are emerging from different sectors such as healthcare. It is important to note that in the case of health care sensing applications it is fundamental to limit the classification error. Sensing context detection could drive inference model switching in order to achieve better classification accuracy.

We argue that phone sensing context detection could also be exploited by existing phone applications and services. For example, by inferring that the phone is in the pocket or bag, a caller might be informed about the reason the callee is not answering the phone call while the callee's phone ring tone volume could be increased so the callee might pick up. One could imagine people enabling this type of additional presence provided to legacy phone service through Discovery. By using the gyroscope (which measures the angular rate change of the phone) to detect the user taking the phone out of the pocket and moving it upwards, the screen saver could be disabled and the phone's keypad made automatically available. One could imagine many such adaptations of the UI with phone sensing context enabled. Similarly, the action of moving the phone towards the lower part of the body could trigger power saving mode. The camera application on the phone could be automatically started as the phone is detected in the user's hand and moved in a vertical position, which is the condition that normally precedes the action of taking a photo. One could imagine phone sensing context provided by the Discovery framework discussed in the next section being applicable to many emerging applications finding their way on to smartphones. For example, reality mining using mobile phone sensor data is starting to be explored as an enhanced form of communication and for social purposes [7].

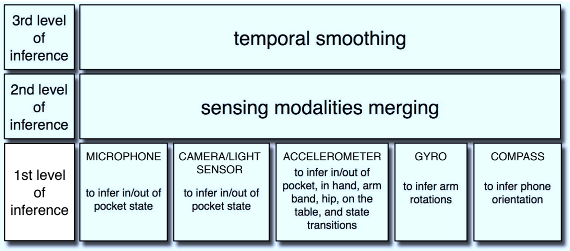

3rd level of inference
temporal smoothing
2nd level of inference
sensing modalities merging
1st level of inference
MICROPHONE
to infer in/out of pocket state
CAMERA/LIGHT SENSOR
to infer in/out of pocket state
ACCELEROMETER
to infer in/out of pocket, in hand, arm band, hip, on the table, and state transitions
GYRO
to infer arm rotations
COMPASS
to infer phone orientation

Figure 1: Discovery inference steps.

2.2 Design

The idea behind Discovery is to use the entire suite of sensing modalities available on a mobile phone to provide enough data and features for context discovery at low cost and for increased accuracy and robustness. Many research questions arise in response to the challenges discussed above: how do we combine the input from multiple sensors, such as, accelerometer, microphone, gyroscope, camera, compass, etc., to infer the phone sensing context? What are the best learning approaches and feature selection policies in order to provide a reliable and scalable context inference system? How do we design low duty-cycling policies with acceptable accuracy when employing phone sensing context? What is the inference accuracy and energy cost tradeoff between using all the possible sensors and only a subset of them according to their availability on the mobile phone? Which sensor set is more responsive to the type of noise in the system (i.e., classification outside controlled laboratory environments)? We believe that Discovery in its totality needs to ultimately address these demanding challenges. However, our preliminary work focuses on a simple phone sensing context: is the phone in the pocket or out. This sounds like a trivial context that could be solved by a number of different sensors. We focus on the microphone - a powerful and ubiquitous sensor on every phone on the market - making Discovery suitable to potentially all phones not just the smart ones. In what follows, we outline out initial framework design.

Discovery consists of a hierarchical inferences pipeline, as illustrated in Figure 1:

First Level Inference - Uni-sensor inference: In this phase, the sensor data from individual sensors is used to operate a first level of inference. Features extraction is tailored to each sensor. This first inference step provides hints about the nature of the current phone sensing context, which, however, might not be conclusive. For example, the use of the camera or light sensor to infer if the phone is in or out the pocket could be misleading because a phone out of the pocket could be in a dark environment, the camera could be covered by the person's hand or by the surface where the phone is positioned. For this reason, a second level of inference built on top of the first is needed.

Second Level Inference - Multi-sensor inference: In this phase, the inference process is based on the output of the first phase. Hence, the first level of inference provides the features to the second level. At this stage, the combination of the camera/light sensor and microphone output would provide better confidence about the actual sensing context. The accelerometer as well could be used as a hint to determine if the phone is inside or outside the pocket given the different accelerometer data signatures when the phone is in a person's hand versus when it's in the pocket. Similarly, by measuring the angular rate change, the gyro could provide indications that the phone has been taken out of the pocket considering that the arm rotation would be picked up by the gyroscope.

Third Level Inference - Temporal smoothing: In this phase, temporal smoothing and Hidden Markov Model (HMM) techniques are used on the output of the second level inference. This step exploits the correlation in time of sensed events when a phone experiences a certain context.

2.3 System Implementation

For our initial implementation of Discovery context classifiers are implemented on the Nokia 95 and Apple iPhone. The preliminary system implements a set of sophisticated inference models that include Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) on the Nokia N95 and Apple iPhone with focus on a limited set of sensors and inferences; that is, we uses the microphone sensing modality to infer the phone sensing context of in the pocket and out of the pocket. We discuss our initial results in the next section. Further modalities, such as accelerometer, compass, and light sensor, are going to be used in combination with the microphone to infer a larger set of sensing context as part of our future work. The initial idea is to evaluate which learning technique (between GMM and SVM) is better suited to the problem and, at the same time, to investigate the adoption of more than one learning strategy in concert to perform the final classification. More learning strategies will be evaluated in the following phase of this work. The challenge with GMM and SVM is that the phone has not been developed to run these computationally demanding models. Part of our efforts is to implement light weight versions of these models as a way forward to do more sophisticated multi-inference classification, as called for by Discovery. In particular a 20-component GMM is adopted, where the number of components is chosen by evaluating the model over the test data set varying the number of components and picking the number of components returning the best classification accuracy.

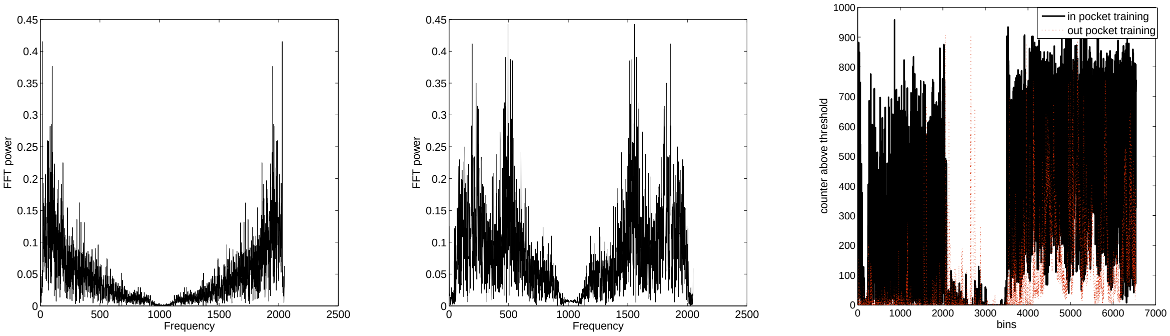
Feature Selection. The selection of an appropriate set of features is a key step to good classification performance. At the moment, a supervised learning approach is adopted and Discovery relies on a 23-dimensional feature vector extracted from an audio clip. A richer selection of features will be evaluated as part of our future work. The current features are:

1st-19th: Mel-Frequency Cepstral Coefficients (MFCC), which have been proven to be reliable features in audio signal classification problems. For the MFCCs extraction we rely on a well-known Matlab libray [8] which is largely used by the research community. We also developed a C version of the MFFC extractor library that can run on the phone;

20th: power of the audio signal calculated over the raw audio data;

21st, 22nd: mean and standard deviation of the 2048-point FFT power in the 0-600 Hz portion of the spectrum. The reason for focusing on this portion of the spectrum can be seen from Figures 2(a) and 2(b), where the presence of a pattern between the two FFT distributions - for in pocket and out-of-pocket recording - is clear. It can be seen that such a pattern is more evident in the 0-600 Hz portion of the spectrum rather than in the whole 0-1024 Hz range;

23rd: this feature is the count of the number of times the



(a) (b) (c)

Figure 2: (a) FFT power of an audio clip when the phone is inside the pocket; (b) FFT power of an audio clip when the phone is outside the pocket; (c) Count of the number of times the FFT power exceeds a threshold *T* for both the in-pocket and out-of-pocket cases.

Table 1: Sensing context classification results using only the microphone. Explanation: when a result is reported in X/Y form, X refers to the *in pocket* case, and Y refers to the *out of pocket* case. If the column reports only one value, it refers to the average result for both *in* and *out* of pocket. Legend: A = GMM; B = SVM; C = GMM training indoor and evaluating indoor only; D = GMM training outdoor and evaluating outdoor only; E = SVM training indoor and evaluating indoor only; F = SVM training outdoor and evaluating indoor only; G = GMM training using only MFCC; H = SVM training using only MFCC.

Classification results	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Accuracy	84% / 78%	80%	75% / 84%	84% / 83%	68%	81%	77% / 79%	71%
Error	16% / 22%	20%	25% / 16%	16% / 17%	32%	19%	23% / 21%	29%

FFT power exceeds a certain threshold *T*. This threshold is determined by measuring the Euclidean difference between the count of the in-pocket and out-of-pocket cases and picking the threshold that maximizes such a distance. An example of the count for both the in-pocket and out-of-pocket cases is shown in Figure 2(c) where it can be seen how these features can be used to discriminate between the in pocket and out of pocket cases. The x-axis of Figure 2(c) reports the number of bins the clip has been split in to.

Consequently, for the mixture model, a 20-component, 23-dimensional GMM is used. The SVM classifiers adopts the 23 dimensional feature vector.

Training. The training phase is performed using audio data collected with a Nokia N95 and Apple iPhone in different settings and conditions from a person going through different environments for several hours. Namely, the audio is recorded in a quiet indoor office environment and an outdoor noisy setting (along a road with cars passing by). In both scenarios the phone is carried both in the pants pocket and outside the pocket in the hand. The choice of these scenarios, i.e., indoor and along a road, is motivated by the fact that they are representative of classes of locations where most likely people spend a lot of their time while carrying their phone both inside and outside the pocket. For each configuration 14 minutes of audio are recorded at different times. Half of each clip (i.e., about 7 minutes of audio) is used to train the classifiers. The training data is finally labeled accordingly.

Prediction. For prediction, the remaining half of each audio clip not part of the training set (i.e., duration of about 7 minutes) is used. Each sample consists of a 96 msec chunk from which the 23 features are extracted. For each configuration there are about 58000 samples available for training and 58000 for evaluation.

3. PRELIMINARY SYSTEM EVALUATION

In what follows, preliminary results from using both the GMM and SVM classification techniques are reported. The results highlight that the audio modality is effective in detecting the in/out of pocket context with reasonable accuracy. Higher accuracy can be achieved by combining further modalities such as accelerometer and light sensor. Columns A and B in Table 1 show, respectively, the classification results for GMM and SVM when the training data combines both indoor and outdoor audio and the phone is carried in and out the pocket. The results are quite encouraging, since we obtain about 80% accuracy (see the accuracy values in columns A and B) adopting a non sophisticated feature set and using only one sensing modality, i.e., the microphone. We are confident that by bringing into the classification process more modalities, for example the accelerometer and light sensor, a more accurate selection of the feature vector, and temporal smoothing it might be possible to achieve a much higher classification accuracy. We then train and evaluate the models for only one scenario, i.e., either indoor or outdoor. The results using GMM are in Table 1 column C and column D. The results for SVM are in column E and column F. In the case of SVM trained and evaluated for the indoor scenario only (see column E) the accuracy is lower than the other cases because Libsvm (the well known SVM library implementation we adopt) is running with the default settings with the kernel optimization being disabled. From these results it is interesting to see that training the models with both indoor and outdoor data does not dilute the training data and the final classification accuracy does not drop significantly compared to the case when the models are trained for a single scenario only and evaluated for the same scenario. In fact, the accuracy in columns C, D, and F is on average close to 80% as in the case of indoor and outdoor training data set (see columns A and B). Columns G and H in Table 1 show, respectively, the classification results for GMM and SVM when the model is trained using only MFCCs (hence a 19-dimensional feature vector). It is evident that the addition of the 4 extra features (i.e., signal power, FFT mean, FFT stddev, and number of times a threshold is exceeded by the FFT power) boosts the classification accuracy. The improvement can be seen by comparing the results in columns G and H with the ones in columns A and B.

4. FUTURE WORK

After the initial promising results, the goal is to implement a working prototype for the Android platform as well. More sensing modalities are going to be used in combination with the audio modality. In particular, the accelerometer, magnetometer, and light sensors. Research is going to be needed in order to identify the most suitable feature vector elements that combine the characteristics of all the sensing modalities. Temporal correlation between events is also going to be taken into consideration to improve the overall accuracy. Techniques such as HMM or voting strategies will be taken into account. We will also pursue the idea of letting people customize the Discovery classifiers to accommodate their habits and needs.

5. RELATED WORK

In the literature, context awareness follows the definition that Weiser [9][10] and others [11][12] provided when introducing or evolving ideas and principles about ubiquitous computing. In that case, context awareness is intended as either the awareness of situations and conditions characterizing sensor devices surroundings or the behavior, activity, and status of the person carrying the sensors in order to provide smart ways to facilitate and explore interaction between machines and humans. Thus, context is seen as the collection of happenings around a monitored subject and the response of the subject to such those happenings. The work in [13, 14, 15, 16, 14] are examples of how sensing systems are adopted to infer such a context and/or leverage context awareness. In some cases external sensors, i.e., not part of the mobile phone itself, are also needed [14][13] in order to perform accurate context inference. The authors of [17] use the word context to mean location awareness and propose applications that efficiently build on top of it. A very large body of work focuses instead on the use of various sensing modalities such as accelerometer, magnetometer, gyroscope to infer a person's activities for different applications [18, 19, 6, 20, 1, 21, 22, 23]. The authors in [24] present an approach to help discover the position of the phone on a person's body. The work highlights two limitations: it uses simple heuristics derived from a small training data set to determine the classification rules, and it uses a single modality approach, i.e., the accelerometer. We instead rely on a systematic design using machine learning algorithms that are more scalable and robust than simple heuristics and consider a larger training data set from multiple positions on the body and different scenarios while using a multi-sensing modality approach.

6. CONCLUSION

In this paper, we argued that phone sensing context is a key system component for future distributed sensing applications on mobile phones. It should be designed to be accurate, robust, and low cost. We discussed our initial work on the Discovery framework that grew out of our work on the deployment of two continuous sensing applications implemented and deployed on Nokia and Apple phones. Our initial implementation and evaluation only focuses on a limited set of sensors/contexts, but looks promising and, as an idea, it has potential, when implemented in its full form, to become a core component of future mobile sensing systems.

7. ACKNOWLEDGMENTS

This work is supported in part by Nokia, Intel Corp., Microsoft Research, and NSF NCS-0631289.

8. REFERENCES

[1] Miluzzo E. et al. Sensing Meets Mobile Social Networks: the Design, Implementation and Evaluation of the CenceMe Application. In *SenSys'08*, 2008.

[2] Lu H. et al. SoundSense: Scalable Sound Sensing for People-Centric Applications on Mobile Phones. In *MobiSys'09*, 2009.

[3] M. Mun et al. PEIR, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research. In *MobiSys'09*, 2009.

[4] Azizyan M. et al. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *MobiCom'09*, 2009.

[5] Honicky R.J et al. N-SMARTS: Networked Suite of Mobile Atmospheric Real-Time Sensors. In *NSDR'08*, 2008.

[6] J. Lester, T. Choudhury, and G. Borriello. A Practical Approach to Recognizing Physical Activities. In *Pervasive'06*.

[7] Your mobile phone as an intelligent sensing machine. http://tinyurl.com/m3hgby.

[8] Rastamat. http://labrosa.ee.columbia.edu/matlab/rastamat.

[9] M. Weiser. The Computer for the 21 st Century. In *Mobile Computing and Communications Review*, 1999.

[10] M. Weiser. Some computer science issues in ubiquitous computing. In *Mobile Computing and Communications Review*, 1999.

[11] B. Schilit, N. Adams, and R. Want. Context-Aware Computing Applications. In *WMCSA'94*, 2004.

[12] P. Dourish. What we talk about when we talk about context. In *Personal and ubiquitous computing*, 2004.

[13] Siewiorek D. et al. Sensay: A context-aware mobile phone. In *ISWC'03*, 2003.

[14] Gellersen H.W. et al. Multi-sensor context-awareness in mobile devices and smart artifacts. In *Mobile Networks and Applications*, 2002.

[15] The Context Aware Cell Phone Project. http://www.media.mit.edu/wearables/mithril/phone.html.

[16] J.E. Bardram and N. Nørskov. A context-aware patient safety system for the operating room. In *Ubicomp'08*, 2008.

[17] Lukkari J. et al. SmartRestaurant: mobile payments in context-aware environment. In *ICEC'04*, 2004.

[18] Harrison B. et al. Using Multi-modal Sensing for Human Activity Modeling in the Real World. In *Handbook of Ambient Intelligence and Smart Environments*, 2010.

[19] Choudhury T. et al. The Mobile Sensing Platform: An Embedded Activity Recognition System. *IEEE Pervasive Computing*, pages 32–41, 2008.

[20] L. Bao and S.S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive'04*, 2004.

[21] Parkka J. et al. Estimating intensity of physical activity: a comparison of wearable accelerometer and gyro sensors and 3 sensor locations. In *EMBS 2007*.

[22] Li Q. et al. Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information. In *BSN'09*, 2009.

[23] S.W. Lee and K. Mase. Activity and location recognition using wearable sensors. In *IEEE Pervasive Computing*, 2002.

[24] Kawahara Y. et al. Recognizing User Context Using Mobile Handset with Acceleration Sensors. In *Portable'07*, 2007.