Coarse In-Building Localization with Smartphones*

Avinash Parnandi[2], Ken Le[1], Pradeep Vaghela[1], Aalaya Kolli[2], Karthik Dantu[1], Sameera Poduri[1], and Gaurav S. Sukhatme[1,2]

[1] Computer Science Department
[2] Ming Hsieh Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089, USA
{parnandi,hienle,pvaghela,kolli,dantu,sameera,gaurav}@usc.edu

Abstract. Geographic location of a person is important contextual information that can be used in a variety of scenarios like disaster relief, directional assistance, context-based advertisements, *etc*. GPS provides accurate localization outdoors but is not useful inside buildings. We propose an coarse indoor localization approach that exploits the ubiquity of smart phones with embedded sensors. GPS is used to find the building in which the user is present. The Accelerometers are used to recognize the user's dynamic activities (going up or down stairs or an elevator) to determine his/her location within the building. We demonstrate the ability to estimate the floor-level of a user. We compare two techniques for activity classification, one is naive Bayes classifier and the other is based on dynamic time warping. The design and implementation of a localization application on the HTC G1 platform running Google Android is also presented.

1 Introduction

Indoor localization is a challenging problem facing the ubiquitous computing research community. Accurate location information is easily obtained outdoors using GPS. However, when we enter a building the reception of GPS signals become weak or is lost causing the inability to determine additional location information within the building. Existing solutions for indoor localization include techniques that use WiFi [4], RFID, Bluetooth [2], ultrasound [16], infrared [19] and GSM [17] [13] etc. Many of these solutions rely on external infrastructure or a network of nodes to perform localization. Note that improving the localization accuracy or increasing the availability of the services provided by these systems require the scaling of infrastructure which can be costly and a challenge on its own.

Our goal is to develop a localization system that is independent of external infrastructure. To this end, we built an indoor localization application that relies primarily on the sensors embedded in a smartphone to coarsely locate a person within a building. Our approach is based on user activity modeling. We use the GPS receiver to determine which building the user entered, and we use the accelerometer to determine what activities the user is performing within the building. We localize the user by coupling

* This work was supported in part by NSF grant CCR-0120778 (CENS: Center for Embedded Networked Sensing), and by a gift from the Okawa Foundation. It was initiated as a project for the graduate course CS 546: Intelligent Embedded Systems taught at USC in Spring 2009.

S. Ystad et al. (Eds.): MobiCASE 2009, LNICST 35, pp. 367–378, 2010.

successive activities like climbing the stairs with some background knowledge. For example, using timestamps we can determine how long the user traversed the stairs or the elevator allowing us to infer which floor he/she is on. The accuracy and capabilities of our localization thus depends on how well we can identify and describe the user's contexts and the knowledge we have about the building.

The rest of the paper is as follows. Section 2 discusses the related work. Section 3 describes the implementation and data collection. Section 4 describes our feature set and classification algorithm, Dynamic Time Warping (DTW) – an alternate technique we explore for better classification at a greater computational cost, and our methodology to detect elevator usage. Section 5 discusses our experimental results in detail. Section 6 lists our conclusions and some directions to extend our work.

2 Related Work

As mentioned in Sec. 1, indoor localization has been the subject of much research. We sample some representative pieces of work from this literature.

Pedestrian indoor localization system developed in [20], does localization using particle filter estimation based on inertial measurement units' data. This work, though requires a detailed map of the building, is close to ours because both use on-board inertial sensors while most other systems depend heavily on external infrastructure. An active badge location system was developed in [18]. In this system, the badges emit a unique id via infrared sensors and the building is instrumented with infrared receivers. All these received ids are relayed to a central server that then tracks every active badge. The RADAR system [4] was a building wide tracking system using wireless LAN. The work in [13] relies on the GSM cell towers id and signal strength for indoor localization. The E911 emergency tracking system locates people who call 911 by doing sophisticated radio signaling and computing the time of arrival,and time difference of arrival from cellphone to perform accurate triangulation. The major difference between all these system and ours, is the fact that all of these require extensive external infrastructure and information, like cell tower id, signal strength measurement capability, access point locations which are not readily available to be used on a mobile phone. Our work does not depend on external infrastructure and uses embedded sensors to solve the localization problem. While in [20] they use the IMU data and estimate the current location using a particle filter on an offline system, we use an accelerometer for activity recognition with time stamps and utilize this information to do online indoor localization. Another important contribution of our work is the elevator ride detection, which to the best of our knowledge has not been greatly explored before.

Activity recognition using inertial sensors has been a topic of great interest to the research community for the last two decades [7]. It promises to have great impact on a variety of domains like elder care [6], fitness monitoring [9, 14], and intelligent context-aware applications [5]. Recently several researchers have used the accelerometers embedded in smartphones to detect activities such as walking, standing, running and sitting with the goal of developing context-aware applications [8, 10]. Our goal in doing activity detection is localization. We focus on activities that may cause a person's location (floor) to change within a building.

3 Implementation

3.1 Hardware

The sensing device we use is the HTC G1 smartphone equipped with GPS receiver and tri-axial accelerometer. The software was implemented on the Android platform, Google's operating system for mobile devices which includes an extensive API for application development. Notably, the API includes ways for accessing and using the phone's sensors [1].

3.2 Software

In this section we'll describe the localization application that was implemented in the smartphone. The application is made up of multiple services that run ubiquitously in the background. Each service provides a specific functionality and each turns on or off automatically depending on which state the application is in. The *Main* service provides the state machine functionality of the application, this is displayed in Figure 1. Its primary responsibility is to turn on and off other services based on the current state of the application and the actions that occur. Lets briefly walk through each of the states.

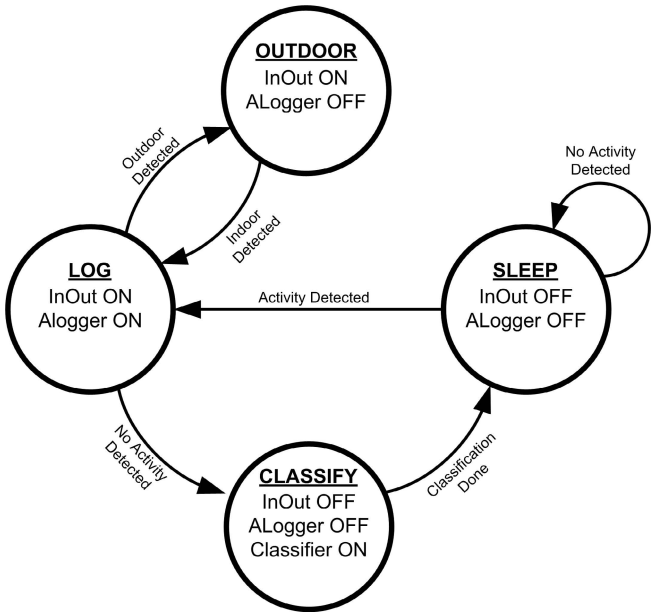


Fig. 1. State diagram of context sensing application

When the user is outdoors, the application is in the OUTDOOR state. In this state the *Indoor/Outdoor Transition Detection* service, abbreviated as InOut in Figure 1, is on to determine when an outdoor to indoor activity occurs using the phone's GPS receiver. When the user enters a building the *Indoor/Outdoor Detection* service logs the location

(latitude and longitude) of where the outdoor to indoor activity occurred and transmits a signal to the Main service.

When an outdoor to indoor signal is detected the application goes into the LOG state. The *Unlabeled Activity Logger* service (ALogger) turns on and begins logging "Unlabeled" accelerometer data into a file.

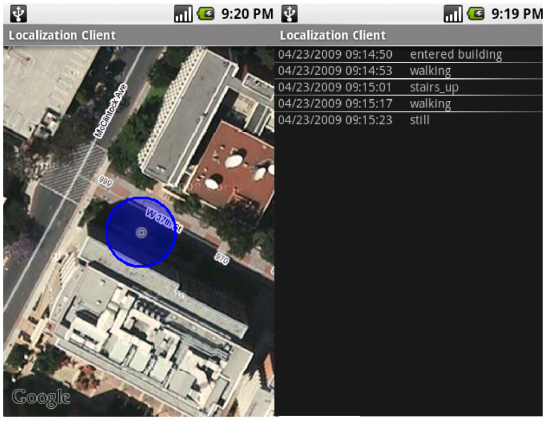


Fig. 2. Outdoor to indoor activity indicating which entrance of the building was used

When the application is in the LOG state and the user becomes still for a specific duration, the application transitions into the CLASSIFY state. Stillness is detected by the *Simple Activity Detection Service*, which is a lightweight module that determines whether the user is performing an activity or not. This service remains on throughout the life of the application for the purpose of triggering various state transitions.

In the CLASSIFY state the *Classification+Filter* service (Classifier) is turned on. This service retrieves the log file created by the *Unlabeled Activity Logger*, analyzes the data sequence, and classifies/labels each instance in the sequence with the most probable activity (e.g. *stairs_up*, *walk*). Occasionally, the classification will be wrong. For this reason, we implement an in-line filtration algorithm that removes misclassified activities. The result is a sequence of identified activities tagged with timestamps indicating when the activities occurred as shown in Figure 2.

Once classification is done, the application transitions to the SLEEP state where everything is turned off except for the *Simple Activity Detection* service. This is the most efficient state, and occurs when the user is staying still.

When the *Simple Activity Detection* service detects an activity again, the application transitions back to the LOG state where the processes described above repeat.

3.3 Data Collection and Analysis

Table 1 shows the list of activities that we focus on perceiving for our initial implementation.

We are interested in these activities since they all cause location change within a building except for the *still* activity. We began by collecting raw accelerometer data

Table 1. List of activities used to estimate floor-level

Activity Label	Description
Still	The user is still
Walk	The user is walking
Stairs_up	The user is going up a staircase
Stairs_down	The user is going down a staircase
Elevator_up	The user is going up in an elevator
Elevator_down	The user is going down in an elevator

using a simple logging application. To keep the data collection trials consistent we strapped the phone to the user's ankle as shown in Figure 3; this kept the y-axis of the phone aligned to the lower leg at all times. Figure 4 shows acceleration data in the y-axis while performing various activities. As you can see from the activity labels, we found that there were distinguishable characteristics in the accelerometer data between the different activities. Note that the still activity is represented by the flat sections in the graph. Our next task was to find a way to quantize the data into various features that would allow us to distinguish the different activities in numerical terms.

The section on Classification will go into further detail on the other features that were explored.

Fig. 3. Position of the phone for data collection with the axis

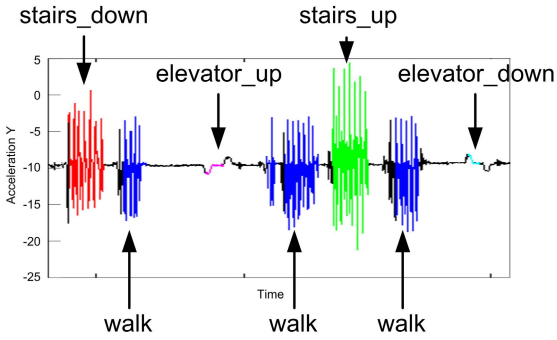


Fig. 4. Acceleration data along the Y axis

3.4 Feature Extraction and Classification

Features are extracted from the accelerometer data every X amount of samples, X was chosen to be 50 within our implementation. There were four sampling rates provided by the Android API.

We chose to operate at the fastest rate which from our observations sampled at approximately 75 samples/sec max and 40 samples/sec on average.

We used the Weka machine learning library in Java for activity classification on the phone. The classifier is used within the Classification+Filter service to classify/label the unlabeled log file, which is stored in *Attribute Relation File Format*. This file is created during feature extraction in the Unlabeled Activity Logger service.

4 Classification

In this section we describe the classification algorithms used to differentiate between the activities listed in Table 1. Two different approaches are presented and compared. The first approach, using a naive Bayes classifier, computes the posterior class probabilities based on a set of carefully chosen features. While this method gives good results for detecting *stairs_up*, *stairs_down*, and *walking* activities (average accuracy of 97% for a user specific trained classifier and 84% for a generally trained classifier), it does not perform well for elevator activity detection. The second approach, Dynamic Time Warping (DTW), classifies activities based on matching raw data with a reference *template* for each activity. This approach gives competitive results but is computationally expensive and not suited for implementation on a phone. We designed a light-weight DTW inspired approach that uses templates and a convolution operation to detect *elevator_up* and *elevator_down* activities.

4.1 Naive Bayes Classification

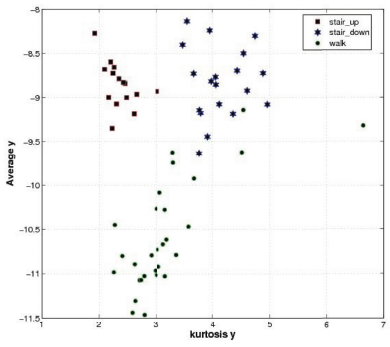
Naive Bayes classification [11] is a simple and well-known method for classification. Given a feature vector f, the class variable C is given by the *maximum aposteriori* (MAP) decision rule as

$$C(f) = argmax_C \Big\{ p(C) \prod_{i=1}^{k} p(f_i|C) \Big\} \tag{1}$$

The accuracy of the classifier depends on the choice of features. After testing a wide range of features [3] [15], we selected the following four features based on classification accuracy and computational cost. Details are shown in the table in Figure 5 . The raw acceleration data is divided into windows of fixed size samples. Each window gets feature extracted and classified independently. As expected, the acceleration data along the y-axis (vertical direction) contained most information about the activity.

We use the following four features for the classification:

1. mean (along y).
2. variance (along y).



Feature	stair_up vs stair_down	stair_up vs walk	stair_down vs walk
kurtosis y	3.92	31.11	11.32
mean y	44.92	9.55	16.28
energy y	43.00	14.66	19.31
skew y	15.53	49.77	20.12
eccentricity y	45.00	23.11	32.05
variance y	47.00	36.44	21.83
correlation yx	31.30	34.00	49.23
correlation yz	31.30	27.55	52.47
kurtosis x	21.84	48.89	18.58
average x	43.46	50.00	36.79
energy x	33.53	28.89	16.58
skew x	35.38	48.00	26.50
eccentricity x	42.69	52.88	47.13
variance y	45.30	38.66	46.92
correlation xy	31.30	34.00	49.23
correlation xz	41.15	38.66	31.45

Fig. 5. Cluster plot and Misclassification rate of features

3. skewness (along x) which is the degree of asymmetry in the distribution of the acceleration data.
4. kurtosis (along y) is a measure of how much a distribution is peaked at the center of the distribution.

Figure 5 shows a cluster plot of a set of training data along a pair of selected features (average y and kurtosis y). It can be seen that the feature pair confidently separates the three activities. We estimate the likelihood distributions to be used in equation 1 as Gaussian distributions that minimize the least squares error of the training data.

4.2 Dynamic Time Warping

Dynamic time warping (DTW) is an algorithm for measuring similarity between two signals which may vary in time or speed [12]. It has been used successfully to classify activities that have a cyclic pattern. A reference signal called a *template*, which is one cycle of the activity, is generated for each activity based on training data and new data is compared with the templates to find the best match on the basis of their Euclidean distance.

We extracted templates manually from the training data. Figure 6 shows an example of the stairs_up template. Classification of data is performed by using a sliding window of y acceleration data and finding its (Euclidean) distance to each of the activity templates. The window is labeled with the activity that gives the least distance. The DTW based approach had an average accuracy of 87.5% .

4.3 Elevator Activity Detection

As mentioned earlier, when using naive Bayes classifier, it was difficult to find features that allow classification of *elevator_up* and *elevator_down* activities. Preferring to use naive Bayes over DTW for efficiency, we needed to develop an auxiliary method for detecting elevator rides. The method we developed specifically for elevator detection

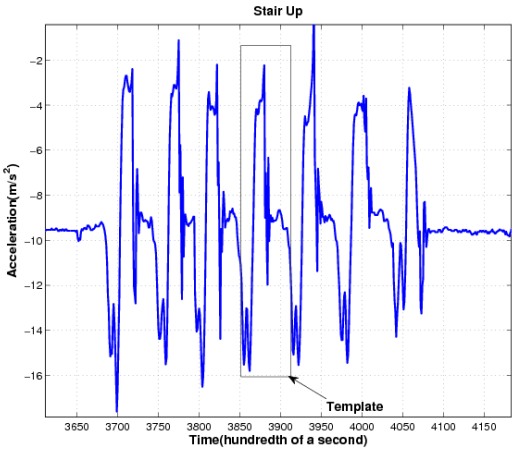


Fig. 6. *stairs_up* activity template

performs template matching using a convolution operator. The first step in this method is to filter out all data that has a standard deviation above a certain threshold. Observing Figure 4, we note that elevator activity has significantly less variance than other activities. With a correctly chosen threshold, the filtered data leaves us only with elevator peaks as shown in Figure 7.

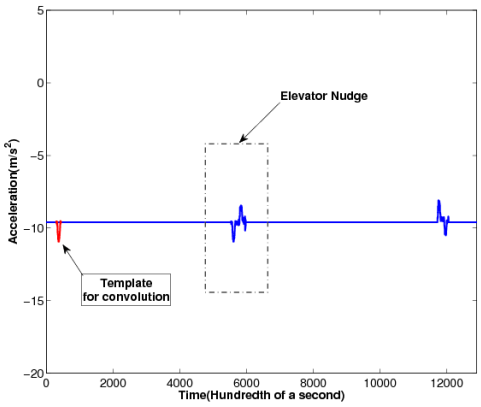


Fig. 7. Acceleration data post filtering

The filtered data is then convolved with the elevator activity template. A typical output of the convolution step is shown in Fig. 8. When the incoming elevator data is convolved with the standard elevator template, the points in the output curve corresponding to the elevator minima get amplified and are shown as the maxima in Fig. 8 (the minima gets amplified because our template has downward facing tip) while the elevator maxima gets registered as a minima in the convolution output.

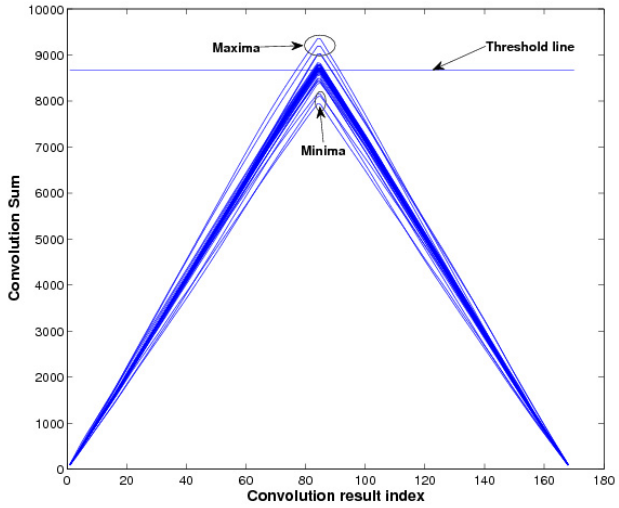


Fig. 8. Convolutional elevator detector: the minima and maxima with respect to the threshold

5 Experiments and Results

This section goes over the various experiments we carried out and their results.

5.1 Indoor Localization

We performed our indoor localization experiments in a five story building named Ronald Tutor Hall (RTH). The table in Figure 9 shows an example of pre-observed times taken to transition between various floors within the building using the stairs.

Floor Transition	Duration
First to Third	51 sec
Third to First	42 sec
First to Second	24 sec
Second to First	21 sec

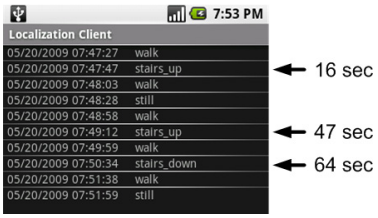


Fig. 9. Pre-observed times and Output of user's context history

The snapshot in Figure 9 shows the output of our localization application after a user performs a trial walk throughout the building. Coupling our knowledge of the approximate times taken to transition between floor levels with the context history outputted from the application we can infer what floor the user is on and at what time. For example, assuming the user is initially on floor 1 at time 7:47:27, we can infer that he/she is on floor 1 at time 7:48:03, on floor 3 at 7:49.59, and on floor 1 again at time 7:51:38. Here we simply divide the duration of *stairs_up* or *stairs_down* activity with the average

time to transition up or down one floor level to determine the number of floor transitions. The inferences made match with the actual floors the user was on during the trial run. The results here show the potential possibilities of performing localization using inference with dynamically retrieved contextual knowledge about the user and static background knowledge about the building.

5.2 Activity Classification

To test activity classification we asked 10 users with different physique and walking styles to perform composite activities. We tested two cases. The first case, *User Specific*, is where the classifier is trained and tested uniquely for each user. In this case, the trial begins by placing the application in training mode and asking the user to perform each activity several times. After training the classifier specifically for the user, the application is switched to operational mode and the user performs the same set of activities to see how well they can be classified. In the second case, *General*, the classifier is generically trained with a couple people and tested on all. In this case, we begin by selecting only a couple of users to train the phone. We then switch to operational mode, and ask every user to exercise the set of activities to see how well the generically trained classifier performs.

Results from our tests show that the User Specific case classified with 97% accuracy and the General case classified with 84% accuracy. In either case, after running the classified activities through the filter in the Classifier+Filter service all misclassified activities were removed.

5.3 Elevator

Using the elevator detection method we developed, we were able to detect elevator rides pretty well. The key part about an elevator ride (whether going up or down) is that convolution maxima and minima always occur in pair, at the start and end point. This is evident from Figure 7. By keeping track of these max-min pairs and their corresponding timestamps we can find out how much time the user spent in the elevator. To test elevator detection we began by extracting an elevator ride template from a trial ride of one user. We then convolved this template with the filtered acceleration data of test runs by 10 other users. Our algorithm detected the elevator rides in each of these runs with no errors.

5.4 Analysis

In Figure 10 we compare the performance of the generally trained classifier, user specific trained classifier, and DTW for each of the 10 users. For the classifiers we use the four features mentioned earlier. Naive Bayes classifier and DTW results are compared with ground truth. The number of false positives divided by total number gives the error percentage. Note that convolution based elevator detection is used in the classifier cases. DTW with generic templates performed better than the generally trained classifier for most users. The user specific trained classifier performed the best on all users. Computationally, DTW is much more expensive and time consuming than the naive Bayes classifier. Since a mobile device must use its resources efficiently and since there was not a clear advantage for using DTW, naive Bayes classification is our method of choice.

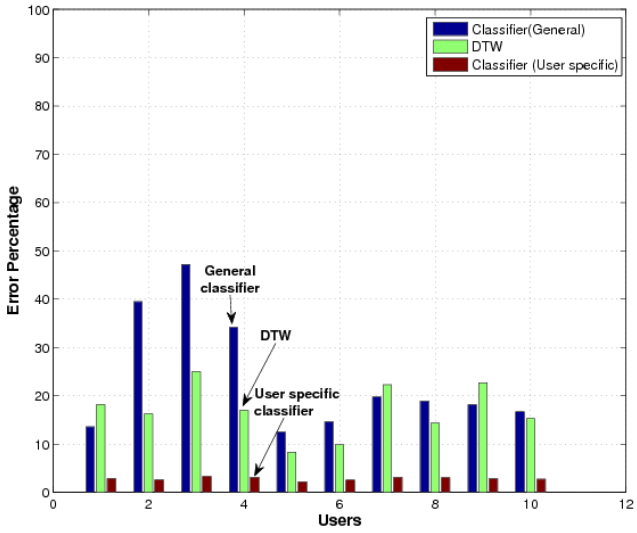


Fig. 10. Percentage of misclassification between Classifier and DTW

6 Conclusion and Future Work

This paper presents an indoor localization system that demonstrates the ability to perform floor level detection using mobile phones. Specifically, we use the accelerometers of the phone to detect activities like *stair_up*, *stair_down*, *elevator_up* and *elevator_down* and further investigate these activities to detect movement between floors. We propose two algorithms (DTW and naive Bayes classifier) to perform this activity detection. These algorithms offer a trade off between detection accuracy and computational requirement (energy efficiency).

Our initial results for floor-level detection are encouraging. We have manually computed the average times needed to traverse up/down a floor for one building but this may vary significantly across different users and different buildings. Our next step is to implement a learning algorithm to adapt this (with user feedback) for the buildings that the particular user visits. We would also like to implement a coordinate frame transformation on the phone so that the application becomes independent of where the phone is on the body.

References

1. Google Android, `http://www.android.com/`
2. Aalto, L., Göthlin, N., Korhonen, J., Ojala, T.: Bluetooth and wap push based location-aware mobile advertising system. In: MobiSys 2004: Proceedings of the 2nd international conference on Mobile systems, applications, and services, pp. 49–58. ACM, New York (2004)
3. Baek, J., Lee, G., Park, W., Yun, B.-J.: Accelerometer signal processing for user activity detection, Berlin, Germany, vol. 3, pp. 610–617 (2004)
4. Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. In: International Conference on Computer Communications (INFOCOM), pp. 775–784 (2000)

5. Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., Hightower, J., Klasnja, P., Koscher, K., Lamarca, A., Landay, J.A., Legrand, L., Lester, J., Rahimi, A., Rea, A., Wyatt, D.: The mobile sensing platform: An embedded activity recognition system. IEEE Pervasive Computing 7(2), 32–41 (2008)
6. Jeon, A., Kim, J., Kim, I., Jung, J., Ye, S., Ro, J., Yoon, S., Son, J., Kim, B., Shin, B., Jeon, G.: Implementation of the personal emergency response system using a 3-axial accelerometer. In: 6th International Special Topic Conference on Information Technology Applications in Biomedicine, ITAB 2007, vol. X, pp. 223–226 (November 2007)
7. Jeon, A., Kim, J., Kim, I., Jung, J., Ye, S., Ro, J., Yoon, S., Son, J., Kim, B., Shin, B., Jeon, G.: Implementation of the personal emergency response system using a 3-axial accelerometer, Tokyo, Japan, pp. 223–226 (2008)
8. Krause, A., Ihmig, M., Rankin, E., Leong, D., Gupta, S., Siewiorek, D., Smailagic, A., Deisher, M., Sengupta, U.: Trading off prediction accuracy and power consumption for context-aware wearable computing. In: ISWC 2005: Proceedings of the Ninth IEEE International Symposium on Wearable Computers, Washington, DC, USA, pp. 20–26. IEEE Computer Society, Los Alamitos (2005)
9. Mathie, M., Coster, A., Lovell, N., Celler, B.: Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. Physiological Measurement 25(2), 1–20 (2004)
10. Miluzzo, E., Lane, N.D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S.B., Zheng, X., Campbell, A.T.: Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In: SenSys 2008: Proceedings of the 6th ACM conference on Embedded network sensor systems, pp. 337–350. ACM, New York (2008)
11. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
12. Muscillo, R., Conforto, S., Schmid, M., Caselli, P., D'Alessio, T.: Classification of motor activities through derivative dynamic time warping applied on accelerometer data, August 2007, pp. 4930–4933 (2007)
13. Otsason, V., Varshavsky, A., LaMarca, A., de Lara, E.: Accurate gsm indoor localization, Berlin, Germany, pp. 141–58 (2005)
14. Preece, S., Goulermas, J., Kenney, L., Howard, D., Meijer, K., Crompton, R.: Activity identification using body-mounted sensors-a review of classification techniques. Physiological Measurement 30(4), R1–R33 (2009)
15. Ravi, N., Dandekar, N., Mysore, P., Littman, M.L.: Activity recognition from accelerometer data, Pittsburgh, PA, United states, vol. 3, pp. 1541–1546 (2005)
16. Savvides, A., Han, C.-C., Srivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. In: International Conference on Mobile Computing and Networking (MOBICOM), pp. 166–179 (2001)
17. Varshavsky, A., de Lara, E., Hightower, J., LaMarca, A., Otsason, V.: GSM indoor localization. Pervasive and Mobile Computing 3(6), 698–720 (2007)
18. Want, R., Hopper, A., Falcao, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems 10(1), 91–102 (1992)
19. Ward, A., Jones, A., Hopper, A.: A new location technique for the active office. IEEE Personal Communications 4(5), 42–47 (1997)
20. Woodman, O., Harle, R.: Pedestrian localisation for indoor environments. In: UbiComp 2008: Proceedings of the 10th international conference on Ubiquitous computing, pp. 114–123. ACM, New York (2008)