Irina Diaconita, Andreas Reinhardt, Delphine Christin, Christoph Rensing: *Inferring Smartphone Positions Based on Collecting the Environment's Response to Vibration Motor Actuation*. In: Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet), pp. 99-106, November 2015.

Inferring Smartphone Positions Based on Collecting the Environment's Response to Vibration Motor Actuation

Irina Diaconita
KOM
TU Darmstadt
Darmstadt, Germany
diaconita@kom.tu-darmstadt.de

Andreas Reinhardt
Department of Informatics
TU Clausthal
Clausthal-Zellerfeld, Germany
reinhardt@ieee.org

Delphine Christin
University of Bonn
and Fraunhofer FKIE
Bonn, Germany
christin@cs.uni-bonn.de

Christoph Rensing
KOM
TU Darmstadt
Darmstadt, Germany
rensing@kom.tu-darmstadt.de

ABSTRACT

Smartphones have become ubiquitous in recent years and offer many useful services to their users, such as notifications about incoming calls and messages, or news updates in real-time. These notifications however do not consider the current user's and phone's context. As a result, they can disturb users in important meetings or remain unnoticed in noisy environment. In this paper, we therefore propose an approach to infer the phone's context based on its vibration motor. To this end, we trigger the phone's vibration motor for short time periods and measure the response of its environments using the built-in microphone and/or accelerometers. Our evaluation shows that leveraging accelerometers allows to recognize the current phone's context with an accuracy of more than 99%. As a result, our proposed solution outperforms our previous work based on played and recorded ringtones in terms of classification performance, user annoyance, as well as potential privacy threats.

Keywords

context detection; phone position classification; active probing; vibration motor actuation

1. INTRODUCTION

Despite multiple embedded sensors and powerful resources, current smartphones continue to ring in inappropriate situations or calls are still missed because users do not hear them. While the screen brightness is adapted to the environmental light conditions or targeted advertising is sent using Bluetooth neighborhood information, adapting the volume and mode of the notification to the phone's context has not been implemented in state-of-the-art phones yet. Research on inferring the current phone's context has however been conducted in the last decade. Preliminary work have focused on collecting mobile contextual data [27], before shifting towards their on-board processing with the introduction of sensing frameworks like FUNF [1].

Within the scope of this paper, we hence follow this trend and investigate a novel approach to determine the phone's current position based on its vibration motor and accelerometers. In more details, we trigger the phone's vibration motor for short periods of time in order to stimulate a deterministic signal whose distortion and attenuation are captured by the accelerometers. The collected acceleration readings are processed and analyzed on the phone to identify the current position between (1) in a pocket, (2) in the user's hand, (3) in a bag, e.g., a purse or a backpack, (4) on a desk facing the ceiling, and (5) on a desk facing the surface of the desk. By means of a comprehensive study, we analyze how acceleration signatures allow for the inference of the phone's location in real-time. Additionally, we compare the performance of our proposed approach against an existing audio-based approach [6], and demonstrate its higher position determination accuracy. Moreover, by using the vibration motor and the accelerometers to determine the phone's context, no sound samples need to be either recorded (potentially endangering users' privacy) nor played (potentially disturbing the users).

We discuss related work in Section 2. We then introduce our system concept and design considerations in Section 3. We provide details on our prototype implementation in Section 4, before assessing its performance in Section 5. We finally make concluding remarks in Section 6.

2. RELATED WORK

Different applications make use of the phone's embedded sensors to detect user context. Examples include user mobility prediction, [32, 25], indoor localization [7], healthcare [13], and gaming [9]. In the following, we especially focus on solutions based on sound samples and accelerometer data, as we use both sensor modalities in our solution. Audio-based approaches aim at recognizing speech [28] and speaker [17], emotions or stress levels from human voices [18]. Music is also a topic of interest, ranging from genre recognition [31] to modeling [16]. In contrast, medical applications [12], user activity recognition [24], and general purpose frameworks [22] usually analyze environmental sounds to detect the user's current context. In comparison, accelerometers can be used to determine a person's physical activity [3, 30] and monitor their exercising habits [11]. Additionally, Li et al. [14] determine when a person is falling and derive posture information, while Mazilu et al. [20] help Parkinson patients train their gait. Further applications include gesture recognition [15] and transportation means identification [10].

When especially considering phone's position, Schmidt et al. [26] aim at distinguishing between five phone modes, two referring to the phone position (hand and table), two to the phone state (silent and general) and one to the environment (outside). To this end, they use a combination of external sensors including two accelerometers, a photodiode, temperature and pressure sensors, a CO gas sensor and a passive IR sensor. Later, Miluzzo et al. [21] used microphone, camera, accelerometer, and gyroscope readings to distinguish between two states of the phone: inside (a bag, a pocket, etc.) and outside (in the hand, on a desk, etc.). Several approaches [8, 33] use accelerometers but rely only on passive probing of the environment. In addition to accelerometers, Wiese et al [33] use the light and proximity sensors extended by an external two-dimensional capacitive array and a multi-spectral sensor. Cho et al [5] rely on the vibration motor and accelerometer readings in order to determine the type of surface the phone is on. They use 1.3 seconds windows, which can cause a slight delay in the phone mode adaption and use SVM as classifier, achieving an average accuracy of 85%.

Our work differs from related work for the following reasons. We compare the following different methods to identify five phone positions: 1) using audio recordings of the notification sounds introduced in our previous work [6], 2) using audio recordings of the sounds produced by the vibration motor, and 3) using accelerometer readings while the vibration motor is triggered. To the best of our knowledge, we are the first to make use of the sound produced by the vibration motor as an indicator of the phone environment. To further adapt the phone mode, we also consider the surrounding noise level and user mobility. We share more similarities with [5] but we consider a significantly smaller window size and combine the sounds of the vibration motor with the resulting acceleration. As a result, we achieve an average accuracy of 99%, outperforming the 85% in [5]. Furthermore, we opportunistically determine the phone position when a new notification is received. By doing so, we avoid duty-cycling and thus reduce the energy consumption.

3. CONCEPT

The phone position influences the user's interaction with her device. A user holding the phone in her hand can receive visual notifications and can readily interact with it. A phone on, e.g., a desk may suggest that the user is involved in another activity or might have even left the phone behind. When a phone is in a pocket or bag, a sound-based notification is necessary to get her attention, but can be inappropriate depending on the user's current activity, such as attending a meeting or driving. To adapt the notification mode to the phone position and the user's context, a first method could be to rely on the users. However, our daily experiences show that users usually tend to forget it. Since most of users' phone usage is based on interpersonal communication [4], this can result in many daily disturbances or missed, e.g., emails and calls.

We have therefore designed three approaches to determine the phone position, all of them actively probing the phone's environment. Moreover, we consider five usual phone positions: pocket, hand, bag, desk facing the ceiling, and desk facing the surface of the desk. We hence present the particularities of these approaches below, before detailing data processing and classification common to all three approaches.

3.1 Vibration Motor & Accelerometers – PCD-VR

The first approach simultaneously relies on the vibration motor and the accelerometer readings. We assume that phone movements are modified depending on the surrounding environment. For example, it can be muffled in an enclosed space, like a bag or a pocket, or amplified on a hard surface. Instead of periodically applying this method, we start it when alerts on, e.g., incoming calls are sent. This makes our approach more energy-efficient, since we do not need to duty cycle and only collect accelerometer readings and classify them when a notification is received. As compared to audio-based methods, classifying the phone position based on the accelerometer data consumes less energy and is not affected by surrounding noise. Moreover, no sound samples are collected, potentially endangering users' privacy.

3.2 Vibration Motor & Audio – PCD-VA

Instead of the acceleration caused by the vibration motor, we consider here the resulting noise and especially, its variation due to the phone environment. A potential challenge is to distinguish the sounds produced by the vibration motor itself, and the sounds produced by the phone's movements against its environment. As above, our approach relies on piggybacking incoming calls and notification alerts, when the phone is in vibration mode. Even if signals, like Gaussian noise, can yield better classification results and be more robust in noisy environments, this approach is more energy friendly and reduces the user disturbance to the minimum.

3.3 Audio & Audio – PCD-AA

The last method relies on audio recordings of specific sound clips being played and presented in detail in [6]. This method serves us as a benchmark for the evaluation of the other two methods, PCD-VR and PCD-AA, as well as for designing hybrid methods.

3.4 Hybrid Approaches

In order to improve classification results, we considered combinations of the above-presented approaches. First, if the phone is on a loud mode, with both notification sounds and vibrations being used, we can use both PCD-AA and PCD-VA. In the second case, the phone is on vibration mode, thus enabling us to combine PCD-VA and PCD-VR.

The classification results can be improved by using the sensor readings to determine not only the phone position, but also surrounding noise level and whether the user is sta-

Table 1: Deciding on the most reliable sensor readings

Environment	User	Most reliable readings
Silent	Stationary	Undecided, both reliable
Silent	Moving	Audio recordings
Noisy	Stationary	Accelerometer readings
Noisy	Moving	Undecided, both slightly less reliable

tionary or moving. Based on this extra information, we could decide whether one sensor is affected less by environment conditions and hence more trustworthy than the other, as shown in Table 1. For instance, if the user is moving through a silent environment and the classification results for audio and accelerometer readings diverge, we would choose the results based on the audio recording. Similarly, in a loud environment where the user itself is stationary, the accelerometer sensor would be chosen since contrary to the the microphone, its signal quality is not affected by the noise.

3.5 Phone Position Classification

We present two architectures developed for our system. In both cases, a smartphone app gathers the relevant data: a chunk of a notification sound, a chunk of the sound made by the vibration motor, or accelerometer readings.

The first system carries out the data preprocessing and classification on the phone. The second one, depicted in Figure 1, sends the data to a server, where it is preprocessed and classified. The server then sends back to the phone the determined phone position and noise level, based on which the phone could adapt its volume accordingly.

For the worst case scenarios, namely the audio-based classification approaches, or experiments showed an average delay for sending the data to the server and receiving back the classification result of 97 ms for a 100 ms recording. For transferring and classifying accelerometer readings, the delay was, as expected, much lower. This proves also the second solution to be feasible for adapting the volume for the very notification sound we collected for the recording. Given that the time to process or transfer accelerometer readings over the network is significantly lower, PCD-VR is even more appropriate for our scenario. In Section 5, we present a comparison of the processing times and battery consumption for the two architectures.

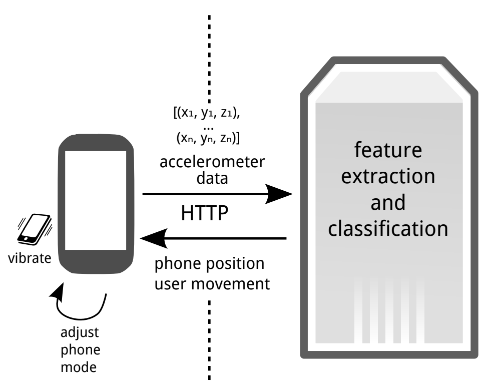

Figure 1: Architecture of the system classifying the data on an external server

We use the same classification pipeline for both systems. We first preprocess the samples, eliminating the silent windows and converting the time domain windows to frequency domain. Then, we leverage machine learning technologies. For dimension reduction, we extract well-established features from the field of audio signal processing. Afterwards, we classify the samples using tree and cluster-based approaches. Afterwards, tree and cluster-based approaches are used for the classification of the samples.

4. IMPLEMENTATION DETAILS

For each of the three considered approaches, the samples were collected on Android phones, while the classification was carried out in two different ways: either on the phone or on an external server. On-board classification uses Weka for Android [19]. On the server, the application was implemented in Python and used the scikit-learn library [23].

4.1 PCD-VR

For this approach, an Android app triggered the phone's vibration motor and gathered the accelerometer readings at the same time. The pattern we used for the vibration motor is the same as the one used by phones when receiving a call, namely blocks of 1 500 ms of vibrations followed by by 500 ms breaks. For the sample collection, we used the `TYPE_ACCELEROMETER` from the Android sensor API, which provides the acceleration force, measured in m/s^2 applied to the phone on the three physical axes, x, y, and z, including gravity. We specified the sensor delay to be 0 ms (`SENSOR_DELAY_FASTEST`) in order to get the maximum sampling frequency supported by the phone. Most of the accelerometer readings were recorded with a Google Nexus 5 smartphone, which features a MPU-6515 acceleration sensor from InvenSense. This sensor offers different settings with a sensitivity between 2048 and 16384 LSB/g and a range between 2 and 16g. It also provides very low sensitivity change depending on the ambient temperature as well as high-speed delivery of accelerometer data and low power usage at with a normal operating current of $450\mu A$. We were able to gather an average of 49 accelerometer readings in a given 100 ms window. Our processing pipeline is shown in Figure 2.

After collecting the samples, we extracted the feature vector for each window, using a combination of features that have been successfully used in various combinations to classify movement [2, 3]. The structure of our feature vector is presented in Equation 1.

$$\begin{aligned} \mathit{features(data)} = (&min_{x,y,z}(data), max_{x,y,z}(data), \\ &median_{x,y,z}(data), rms_{x,y,z}(data), \\ &pcorrelation_{x,y,z}(data), sd_{x,y,z}(data)) \end{aligned} \quad (1)$$

Where $min_{x,y,z}(data)$, $max_{x,y,z}(data)$, and $min_{x,y,z}(data)$ are the minimal, maximum, and median of the x, y, and z components of the acceleration data read for one window. $rms_{x,y,z}(data)$, $pcorrelation_{x,y,z}(data)$, and $sd_{x,y,z}(data)$ are, respectively, the root mean square, Pearson correlation, and standard deviation of the data along the three axes. This gives us a total of 18 coefficients for our feature vector. Figure 3 shows 6 of these coefficients, extracted for the x-axis component of one acceleration window.

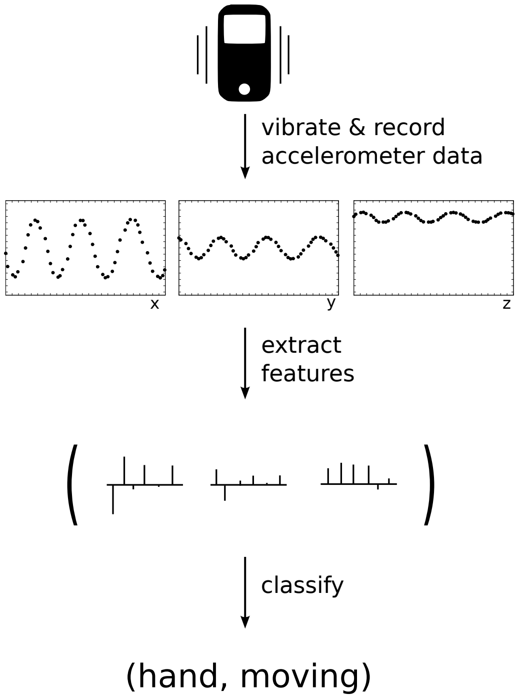

Figure 2: Processing pipeline of the accelerometer readings

The last step is the classification. We have mainly used tree-based algorithms, like Decision Trees (DT) and Random Forest (RF), and cluster-based algorithms, like K-Nearest Neighbors (KNN). This is motivated by the way the feature vectors form clusters in the 18-dimensional feature space, these values being unique and almost constant for each phone position. For the comparison, we also used Gaussian Mixture Model (GMM).

4.2 PCD-VA

The data collection was carried out by an Android app that triggers the phone's vibration motor and records the sounds generated by it, together with the environment sounds at a sampling rate of 44.1 kHz with a 16 bit depth. The pattern we used for triggering the vibration motor was the same as the for PCD-VR (1500ms of vibration, 500ms of silence). Given the volume of the sounds produced by the vibration motor, we did not have any concerns about clipping.

We use a processing pipeline comprised of windowing, silence removal, Fourier transformation, feature extraction and classification. In the first step, we apply a rectangular windowing function which uses a fixed-length window size. Based on the results of experimentation with different window sizes, we chose a window size of 4096 samples, corresponding to about 100 ms. Since silent windows provide no value for the classification and lower the accuracy, they are removed in the next step. We calculate the signal energy for the corresponding window and discard it if its signal energy falls below a certain threshold. In order to convert the signal from the time to the frequency domain, we then apply a Fourier transform in the next step.

The following step is extracting the feature vector of a window in order to reduce the dimensions of the data. We

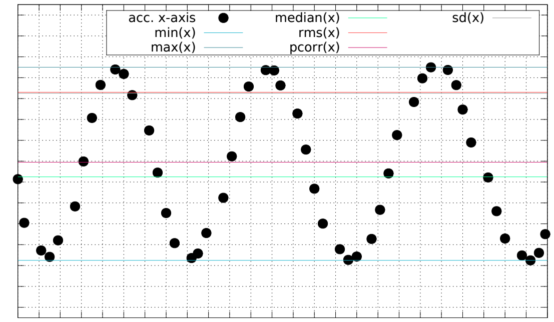

Figure 3: Features for the x-axis component of one acceleration window

evaluated four different types of features: Mel Frequency Cepstral Coefficients (MFCC), Delta Mel Frequency Cepstral Coefficients (Delta MFCC), the Band Energy (BE) and the Powerpectrum (PS). MFCC emulate the behavior of the human ear [29] and are "perceptually motivated" [31]. MFCC are popular tools for speech and speaker recognition [29], as well as music modelling [16]. Furthermore, they have been successfully used to classify environment sounds [22, 24]. We calculate the first N=13 MFCC coefficients, since existing work has shown this number to be sufficient for high quality results [22, 29]. However, MFCC features do not take the temporal aspect into account, being calculated only for the individual windows. For this reason, we also experimented with the Delta MFCC coefficients, which add to the N MFCC coefficients their first order derivatives with respect to time. The final classification is then the same as above.

4.3 PCD-AA

For collecting the samples, we used an Android app that can play back ringtones, alarm and notification sounds, and record them at the same time, at a sampling rate of 44.1 kHz with a 16 bit depth. Unlike PCD-VA, clipping was a concern for this approach. Since the optimal volume is decided based on the classification result, we had the app set the volume for the sample collection. Thus we avoided both clipping and too silent recordings. This did not create any disturbances for the user, since the smallest recording length was only 100 ms. We looked at all ringtones, alarms and notification sounds of Samsung Galaxy Nexus, and all ringtones of Nexus 5 and Samsung Galaxy S3. The preprocessing and classification process are again the same.

5. EVALUATION

We first introduce the evaluation setup, before comparing both classification architectures. We then present and compare the evaluation results of our three proposed approaches.

5.1 Evaluation setup

We used three types of phones, Samsung Galaxy Nexus, Samsung Galaxy S3, and Nexus 5. They all support the 44.1 kHz sampling rate and 16 bit audio depth required for an optimal audio sample classification. We have collected over 550 000 audio samples and over 1 388 624 accelerometer readings and use a 10-fold cross validation approach for the classification. We gathered the audio samples we considered in both silent and noisy environments, while for the accelerometer readings we also considered both stationary and mobile users and environments. We aim at classifying the current phone position, together with either environment noise level for the audio-based scenario or user mobility for the accelerometer-based scenario. The user mobility can be classified either as stationary or mobile. We evaluate and compare the precision and recall achieved for the various types of features and classifiers, as well compare the overall performances of our three approaches.

5.2 Classification Architectures

Both approaches to classifying the data, namely on the phone and on the server, used the same methods, so we based our decision on the processing time and battery consumption. We considered for our test the most computationally intensive scenario, audio recordings classification. Since classifying only individual samples would have given us a limited picture, we used all our recordings of “Over the Horizon”, one of the most common Samsung recordings, which totaled 880 minutes. For a 10-fold cross-validation approach, this means the system was trained with roughly nine hours' worth of recordings. The phone we used for these tests was a Samsung Galaxy Nexus. Thus, for classifying all recordings the smartphone app required 457.6 minutes, whereas the server application needed only 5.28 minutes. This means, for a 100 ms window, an average processing time of 52 ms on the phone, versus 0.6 ms on the server.

We measured the drop in battery over one hour, since even the most talkative users are unlikely to get close to one hour's worth of notifications in one day, which would translate into 3,600 samples, or notifications. The server-based approach caused 4% drop in battery percentage and required 20 minutes to send the data to the server and receive the answers, plus 0.416 minutes to classify the data. The smartphone app was responsible for a 36% drop in battery percentage and needed 238 minutes to finish the classification.

5.3 PCD-AA Evaluation

We evaluated this approach for all Galaxy Nexus, Galaxy S3, and Nexus 5 ringtones. With some exceptions, most of the ringtones differed between phones. We also looked in more detail into the most notable exception, “Over the Horizon”, which is one of the most popular Samsung ringtones. Furthermore, we evaluated our approach for at all Samsung Galaxy Nexus alarms and notification sounds. Due to space constraints, we will restrict this subsection to a succinct discussion of the ringtones-based classification, the rest of the results being available in [6].

Our recordings were taken both in silent and noisy environments. In what follows, we will summarize the phone position classification results obtained for the 25 Galaxy Nexus ringtones. We use MFCC and Random Forest, which have proven to be the best feature array and classifier combination. PCD-AA yields an average precision of 94.4%. Precision values vary between 90% for “Zeta” to 99% for “Aquila”. These variations stem from the different durations and repetition patterns. The silent intervals between the repetitions will be eliminated in the silence removal step, if the environment is silent. If the environment is noisy, these samples will be kept, but classifying them will be analogous to classifying recordings of environment sounds alone, which has a precision of 77%. Therefore, the shorter the audio clips and the more numerous the repetition cycles, the less accurate the classification results will be.

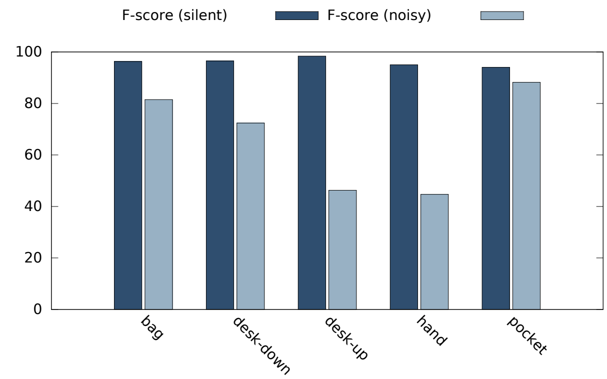

Figure 4: Classification precision for PCD-VA in silent and noisy environments

“Over the Horizon”, one of the most popular Samsung ringtones, has a 93% precision for the phone position classification, as depicted. When classifying both phone position and environment noise level, its precision drops to 89%. In the remainder of this section we will be using it as benchmark for PCD-VA and PCD-VR, due to both its popularity and the similarity between its results and the average results of the other ringtones.

5.4 PCD-VA Evaluation

We examine the results achieved when classifying audio recordings of environment sounds, together with the sounds generated by the phone's vibration motor. While we do not differentiate between the specific situations, the settings in which the recordings were gathered include offices and homes as silent environments and outdoors, public transportation means, shops, cafés and parties as noisy environments. MFCC yields the best results, as far as features are concerned, followed closely by Delta MFCC. Despite the advantage brought by taking into account the temporal aspect, Delta MFCC seems to suffer slightly from overfitting, hence the somewhat less accurate results. As far as classifiers are concerned, Random Forest and K-Nearest Neighbors provide the best results, outperforming Decision Trees and Gaussian Mixture Models. Still, Random Forest outperforms K-Nearest Neighbors. Since that MFCC and Delta MFCC are the best features, our data has N=13 or N=26 dimensions. Given the structure of our data, building multiple trees is more efficient and accurate than clustering the data on N dimensions.

Figure 4 compares the F-scores for phone positions classes in both silent and noisy environments. One can notice that PCD-VA is able to outperform PCD-AA in a silent environment (96% vs. 94%), while in a noisy environment due to the low volume of the vibration motor sounds, the environment noise dominates the recordings and thus produces significantly worse results. The fact that in a silent environment, the result is better than for PCD-AA is a particularly relevant finding, as the PCD-VA method can be applied on all phones since the sounds produced by the vibration motor are very similar. On the other hand, asking users to

True class \ Predicted class	noisy bag	noisy desk-down	noisy desk-up	noisy hand	noisy pocket	silent bag	silent desk-down	silent desk-up	silent hand	silent pocket
nb	82.3	3.3	1.9	4.5	5.5	1.9	0.2	0.0	0.2	0.0
ndd	2.2	76.2	12.1	5.2	1.6	0.4	0.9	0.2	0.4	0.7
ndu	3.3	38.7	41.0	9.4	1.9	0.9	0.5	0.5	0.9	2.8
nh	1.9	8.2	9.7	69.6	5.0	1.9	0.3	0.0	2.5	0.9
np	2.9	1.8	1.4	2.0	88.1	1.8	0.0	0.0	0.2	1.8
sb	3.6	1.3	0.4	1.7	1.9	90.4	0.0	0.0	0.2	0.4
sdd	0.0	0.0	0.0	0.0	0.0	1.4	96.5	0.7	0.7	0.7
sdu	0.0	0.0	1.2	0.0	2.4	1.2	3.6	91.6	0.0	0.0
sh	3.5	6.1	3.5	3.5	2.6	1.3	0.0	0.0	79.7	0.0
sp	2.5	6.7	3.1	3.1	8.0	3.1	2.5	0.0	0.0	71.2

Figure 5: Confusion matrix for position and noise level classification using PCD-VA (in %)

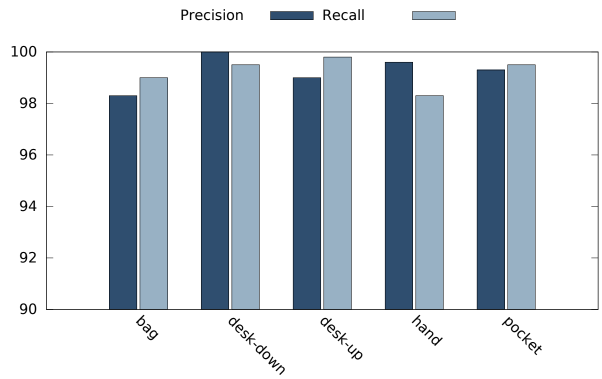

Figure 6: Classification precision and recall scores for PCD-VR

True class \ Predicted class	moving bag	moving hand	moving pocket	stationary bag	stationary desk-down	stationary desk-up	stationary hand	stationary pocket
mb	96.6	1.2	0.3	0.9	0.0	0.0	0.6	0.3
mh	0.3	96.3	0.0	1.4	0.0	0.3	1.7	0.0
mp	0.5	0.0	95.9	0.0	0.0	0.0	0.0	3.6
sb	0.1	0.7	0.0	98.7	0.0	0.3	0.1	0.0
sdd	0.2	0.2	0.2	0.0	99.5	0.0	0.0	0.0
sdu	0.0	0.0	0.0	0.9	0.0	99.1	0.0	0.0
sh	0.5	1.2	0.0	0.0	0.0	0.0	98.3	0.0
sp	0.2	0.2	2.3	0.4	0.0	0.6	0.2	96.0

Figure 7: Confusion matrix for position and user movement classification using PCD-VR (in %)

switch to a ringtone with a better classification precision is completely unfeasible.

Next, we aim to determine both the phone position and the noise level of the environment. Figure 5 shows the corresponding confusion matrix when using MFCC and Random Forest. The method proves to be effective in determining the noise level of the environment, few confusions occurring between noisy and silent situations, despite the impact of noise on classification results. Overall, there are significantly less misclassifications of the noise level, compared to those of the phone position. This is a quite relevant finding, given the importance of noise level when deciding the optimal phone mode. Most mislabeling occurs between the situations when the phone is lying on a desk, facing up or down, in a noisy environment. Some confusions also occur between hand and desk facing upwards.

5.5 PCD-VR Evaluation

We analyze the results of phone position classification using PCD-VR. Data was collected in the same variety of environments as for the previous approaches. While noise level of the environment had no impact on the classification results, user movement was an important extra factor we needed to take into account when evaluating this method.

Classification of phone position using the afore-mentioned features extracted from the accelerometer data and Random Forest shows very good results with an overall precision of 99.2%. At this point we did not distinguish whether the user was moving or not, but samples were collected in both stationary and moving situations. Figure 6 shows the precision and recall values for a classification over all of the collected accelerometer samples, i.e., more than 1 300 000 samples. As the plot shows, the results were almost perfect, except that for less than 1% of the situations the user's hand was classified as a bag. The combination of vibration motor and accelerometer can also be used to classify the combination of user movement and phone location with an outstanding precision of 97.8%. It is worth noting that we added a whole new dimension to our data by determining whether the user is stationary or moving, which caused a drop in precision of only 1.4%. Figure 7 shows the corresponding confusion matrix, when using Random Forest as classifier.

Next, we compare the result of the main classifiers we experimented with, as shown in Figure 8. For this comparison, we considered phone positions alone. As for the previous approaches, Random Forest obtains the best results, being very closely followed by K-Nearest Neighbors, which is, in turn closely followed by Decision Trees. Gaussian Mixture Models perform significantly worse, with a precision of 81.5%, compared to the 99.2% achieved by Random Forest. This proves our initial assumption regarding the fitness of tree-based and cluster-based approaches for our scenario. This stands true regardless of the data source we use, be it audio or accelerometer readings, the structure of our data being essentially determined by environment conditions that stay the same. Basically, phone position will have analogous effects on the different sensor readings. In a pocket, both sound and phone vibrations will be muffled.

These findings are particularly relevant. Not only does PCD-VR largely outperform PCD-AA and PCD-VA, but also given the smaller amounts of data to be processed or sent over the network, this allows us to save on battery life or network traffic. Also, the accelerometer is not affected by environment noise and is thus able to perform well in situations where audio-based approaches would provide significantly worse results. Last but not least, while there is a chance that a user might perceive a 100 ms window of a ringtone in a very silent environment, this is extremely unlikely to happen when triggering the vibration motor.

5.6 Overall Comparison

We compare the overall performance of PCD-AA, PCD-VA, and PCD-VR for phone position classification. Figure 9 shows their respective F-scores in three types of situations: silent, noisy, and a combination of these. As expected, PCD-AA and PCD-VA perform best in silent situations, suffering

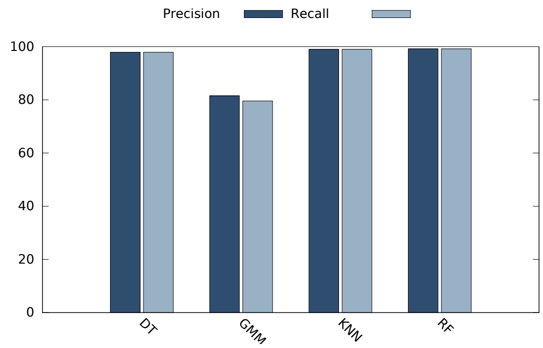

Figure 8: Comparison of overall precision and recall for all classifiers used with PCD-VR

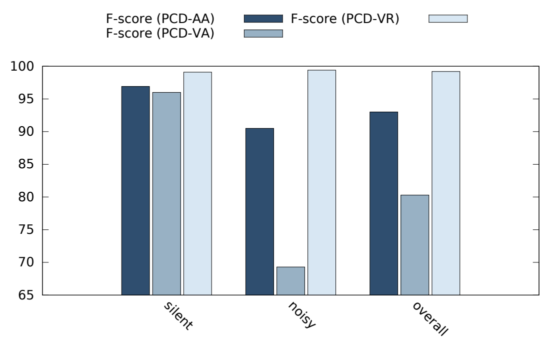

Figure 9: Comparison of overall F-scores for PCD-AA, PCD-VA, and PCD-VR in silent, noisy, and combined situations

a drop in F-score due to noise. This is more obvious in noisy situations alone compared to the combination of situations. PCD-VA is more affected by noise than PCD-VA due to the low volume of the vibration motor sounds, that get more easily covered by environment sounds. PCD-VR achieves noticeably better results in all situations, particularly in the noisy and combined situations. Thus, its F-score in noisy environments is 9.8% better than that of PCD-AA and 43.4% better than that of PCD-VA. In combined situations, it obtains 6.6% and 23.5% better F-scores than PCD-AA and PCD-VA respectively. Obviously, accelerometer readings are not affected by acoustic noise. This, together with the lower energy consumption or network traffic, as well as the practically zero potential for user disturbance make this method the most fit one for real life deployments.

6. CONCLUSIONS

More and more sensing capabilities are integrated into state-of-the-art smartphones, allowing them to monitor and adapt to their environment. With user notifications representing one of the core smartphone functionalities, a phone's ability to capture its current context can be used to improve the way of relaying these notifications to the user. For example, when the phone is located in the user's pocket, a vibration alert can increase the user's awareness of the notification, whereas a increase in the ringtone volume can be used when the phone is in the user's bag. In this paper, we have presented a novel way of determining a phone's location based on acceleration sampling. It involves triggering the phone's integrated vibration motor for a short amount of time, and is thus of little to no disturbance to the user while probing its environment. Our comprehensive evaluation has shown that our approach is able to determine a phone's location from the five mostly used settings with more than 99% accuracy. In summary, we believe that our vibration sampling method is a promising means to help current smartphones identify their current situational context better, and thus to make these *smart*phones eventually *smart*.

Acknowledgements

This work has been co-funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 – MAKI.

7. REFERENCES

[1] N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland. Social fMRI: Investigating and Shaping Social Mechanisms in the Real World. *Pervasive and Mobile Computing*, 2011.

[2] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, 2009.

[3] L. Bao and S. S. Intille. Activity Recognition from User-Annotated Acceleration Data. In *Pervasive Computing*. 2004.

[4] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, 2011.

[5] J. Cho, I. Hwang, and S. Oh. Vibration-Based Surface Recognition for Smartphones. In *IEEE 18th International Conference on Embedded and Real-Time Computing Systems and Applications*, 2012.

[6] I. Diaconita, A. Reinhardt, D. Christin, and C. Rensing. Bleep bleep! Determining Smartphone Locations by Opportunistically Recording Notification Sounds. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2014.

[7] R. Elias and A. Elnahas. An Accurate Indoor Localization Technique Using Image Matching. In *Proceedings of the International Conference on Intelligent Environments*, 2007.

[8] K. Fujinami and S. Kouchi. Recognizing a Mobile Phone's Storing Position as a Context of a Device and a User. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. 2013.

[9] S. Hardy, A. El Saddik, S. Göbel, and R. Steinmetz. Context Aware Serious Games Framework for Sport and Health. In *IEEE International Workshop on Medical Measurements and Applications Proceedings*, 2011.

[10] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based Transportation Mode Detection on Smartphones. In *Proceedings of the 11th ACM*

Conference on Embedded Networked Sensor Systems, 2013.

[11] M. Kranz, A. Möller, N. Hammerla, S. Diewald, T. Plötz, P. Olivier, and L. Roalter. The Mobile Fitness Coach: Towards Individualized Skill Assessment Using Personalized Mobile Devices. *Pervasive and Mobile Computing*, 2013.

[12] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel. Accurate and Privacy Preserving Cough Sensing Using a Low-cost Microphone. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.

[13] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Proceedings of the 6th International Workshop on Wearable and Implantable Body Sensor Networks*, 2009.

[14] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou. Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information. In *6th International Workshop on Wearable and Implantable Body Sensor Networks*, 2009.

[15] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based Personalized Gesture Recognition and its Applications. *Pervasive and Mobile Computing*, 2009.

[16] B. Logan et al. Mel Frequency Cepstral Coefficients for Music Modeling. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2000.

[17] H. Lu, A. J. Bernheim Brush, B. Priyantha, A. K. Karlson, and J. Liu. SpeakerSense: Energy Efficient Unobtrusive Speaker Identification on Mobile Phones. In *Pervasive Computing*. 2011.

[18] H. Lu, D. Frauendorfer, M. Rabbi, M. S. Mast, G. T. Chittaranjan, A. T. Campbell, D. Gatica-Perez, and T. Choudhury. StressSense: Detecting Stress in Uconstrained Acoustic Environments Using Smartphones. In *Proceedings of the ACM Conference on Ubiquitous Computing*, 2012.

[19] R. J. Marsan. Weka for Android. *GitHub Repository*, 2011.

[20] S. Mazilu, U. Blanke, M. Hardegger, G. Troster, E. Gazit, M. Dorfman, and J. M. Hausdorff. GaitAssist: A Wearable Assistant for Gait Training and Rehabilitation in Parkinson's Disease. In *Proceedings of the International Conference on Pervasive Computing and Communications Workshops*, 2014.

[21] E. Miluzzo, M. Papandrea, N. D. Lane, H. Lu, and A. T. Campbell. Pocket, Bag, Hand, etc.-Automatically Detecting Phone Context Through Discovery. In *Proceedings of the ACM International Workshop on Sensing Applications on Mobile Phones*, 2010.

[22] S. Nirjon, R. F. Dickerson, P. Asare, Q. Li, D. Hong, J. A. Stankovic, P. Hu, G. Shen, and X. Jiang. Auditeur: A Mobile-Cloud Service Platform for Acoustic Event Detection on Smartphones. In *Proceedings of the The 11th International Conference on Mobile Systems, Applications, and Services*, 2013.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2011.

[24] M. Rossi, S. Feese, O. Amft, N. Braune, S. Martis, and G. Troster. AmbientSense: A Real-Time Ambient Sound Recognition System for Smartphones. In *Proceedings of the International Conference on Pervasive Computing and Communications*, 2013.

[25] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: A spatio-temporal prediction framework for pervasive systems. In *Pervasive Computing*. 2011.

[26] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. Advanced Interaction in Context. In *Handheld and Ubiquitous Computing*, 1999.

[27] A. Schmidt, M. Beigl, and H. Gellersen. There is More to Context than Location. *Computers and Graphics*, 1998.

[28] R. V. Shannon, F.-G. Zeng, V. Kamath, J. Wygonski, and M. Ekelid. Speech Recognition With Primarily Temporal Cues. *Science*, 1995.

[29] S. Srivastava, S. Bhardwaj, A. Bhandari, K. Gupta, H. Bahl, and J. Gupta. Wavelet Packet Based Mel Frequency Cepstral Features for Text Independent Speaker Identification. In *Intelligent Informatics*. 2013.

[30] R. P. Troiano, D. Berrigan, K. W. Dodd, L. C. Mâsse, T. Tilert, and M. McDowell. Physical Activity in the United States Measured by Accelerometer. *Medicine and Science in Sports and Exercise*, 2008.

[31] G. Tzanetakis and P. Cook. Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 2002.

[32] L. Vu, Q. Do, and K. Nahrstedt. Jyotish: Constructive Approach for Context Predictions of People Movement from Joint Wifi/Bluetooth Trace. *Pervasive and Mobile Computing*, 2011.

[33] J. Wiese, T. S. Saponas, and A. B. Brush. Phoneprioception: Enabling Mobile Phones to Infer Where They Are Kept. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013.