University of Colorado, Boulder
CU Scholar

Computer Science Technical Reports | Computer Science

Spring 5-1-2013

Impact of Smartphone Position on Sensor Values and Context Discovery

Khaled Alanezi
University of Colorado Boulder

Shivakant Mishra
University of Colorado Boulder

Follow this and additional works at: http://scholar.colorado.edu/csci_techreports

Recommended Citation

Alanezi, Khaled and Mishra, Shivakant, "Impact of Smartphone Position on Sensor Values and Context Discovery" (2013). *Computer Science Technical Reports.* Paper 1030.
http://scholar.colorado.edu/csci_techreports/1030

This Technical Report is brought to you for free and open access by the Computer Science at CU Scholar. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of CU Scholar. For more information, please contact cuscholaradmin@colorado.edu.

Impact of Smartphone Position on Sensor Values and Context Discovery

Khaled Alanezi
Department of Computer Science
University of Colorado
Boulder, CO, 80309-0430
+1 7202573718
Khaled.Alanezi@colorado.edu

Shivakant Mishra
Department of Computer Science
University of Colorado
Boulder, CO, 80309-0430
+1 3034924686
mishras@cs.colorado.edu

ABSTRACT

With near-ubiquitous availability of smartphones equipped with a wide variety of sensors, research in building context-aware services has been growing. However, despite a large number of services proposed and developed as research prototypes, the number of truly context-aware applications available in smartphone applications market is quite limited. A major barrier for the large-scale proliferation of context aware applications is poor accuracy. This paper addresses one of the key reasons for this poor accuracy, which is the impact of smartphone positions. Users carry their smartphones in different positions such as holding in their hand, keeping inside their pants or jacket pocket, keeping in their purse, and so on. This paper addresses the issue of poor application accuracy due to varying smartphone positions. It first shows that smartphone positions significantly affect the values of the sensor data being collected by a context aware application, and this in turn has a significant impact on the accuracy of the application. Next, it describes the design and prototype development of a smartphone position discovery service that accurately detects a smartphone position. This service is based on the sensor data collected from carefully chosen sensors and runs orthogonal to any other context aware application or service. Finally, the paper demonstrates that the accuracy of an existing context aware service or application is significantly enhanced when run in conjunction with the proposed smartphone position discovery service.

Keywords

Smartphone, position discovery, sensing

1. INTRODUCTION

Modern smartphones embody a large set of sensors that can be utilized to learn a wealth of information about a user's surrounding environment. Researchers view the availability of such sensors as an opportunity for developing context-aware applications that can provide services tailored for each user's context. Context-aware mobile computing is not a new research topic, for example, a survey paper [7] covering advances in this field was published more than a decade ago. Despite the concept being there for a while, a breakthrough for the number of context-aware applications offered in smartphones application markets (e.g., App Store for Apple iOS or Google Play for Android OS) is yet to happen. For the most part, the current context-aware applications do not meet users' high expectations from technology.

A key problem with current context aware applications is that they typically provide low level of accuracy, particularly when used in an environment different from what was conceived at the application development stage. The issue of achieving high accuracy in context-aware applications is complex by nature. It can be attributed to the large number of factors such as the complexity of the context being discovered, number of diverse sensors involved, differences in the analysis techniques used for each type of sensor, and the required level of accuracy imposed by the context-aware application.

A major factor leading to low accuracy in mobile context-aware applications is the wide variety of ways a user may carry his/her smartphone, henceforth referred to as *smartphone position*. Different users may carry their smartphones in hand, in purse, in pants pocket, in shirt pocket, etc. Sometimes, their smartphones are in covered positions such as in purse or pockets while uncovered at other times such as while watching a video or talking on the phone. Sensor values of different sensors naturally vary based on smartphone position, which in turn impacts the accuracy of the context derived from these values. In this paper, we focus on developing a generic solution to address this issue.

Current context-aware applications can be divided into three categories in regards to the way they address the smartphone position problem. The first category ignores the presence of the problem, assuming a fixed position [22]. The second category requires the users to place the smartphone in a certain position to achieve the proposed level of accuracy [20, 31]. For example, a solution that automatically determines a user's current physical activity such as walking, jogging, standing, etc. [20], asked the participants in the experiment to place the smartphone in the front pocket of their pants. We believe that such assumption can limit the practicality of the developed solution. It is unrealistic to require users to refrain from certain smartphone positions to get good results. The third category of applications is designed to produce good results regardless of the smartphone position [17]. However, we believe that the techniques used in these applications are application-specific and can't be generalized.

We take a bottom-up approach to tackle the smartphone position problem. First, to motivate the problem, we examined a fall detection application proposed in [3] in two different ways, first without knowing the smartphone position and second with the smartphone position known. We show that the accuracy of the application is significantly improved in the latter case. Section 2 provides the details of this motivating example. Next, we analyze the impact of the smartphone position on raw sensor data. This is important since context derivation algorithms start from raw sensor data collected from smartphone sensors. To do this, we conducted a range of experiments that involved collecting sensor data from several different users carrying their smartphones in several different positions. Analysis of the raw sensor values collected from different smartphone position shows that the level of smartphone position impact on raw sensors data ranges from no impact at all for the case of GPS to a considerably high impact as in the case of gyroscope and accelerometer. The details of the analysis can be found in Section 3 of the paper.

Based on the analysis of the impact of smartphone position on raw sensor data, we have designed, implemented and evaluated a smartphone position discovery service. This service utilizes the sensor values collected from some carefully chosen sensors and detects the smartphone position with very high accuracy. It runs orthogonal to any other context aware service or application. Sections 4 and 5 describe the design, implementation and a detailed evaluation of this service. The key motivation for the smartphone position discovery service is that it can aid in improving the accuracy of any other context-aware application by informing the other application about the smartphone position so that it may process its sensor information more accurately. In general, we anticipate the following benefits from offering the smartphone position service to context-aware applications:

- Context-aware applications can provide more accurate results by processing their sensor data more accurately, or avoiding taking decision in disadvantageous smartphone positions.

- Context-aware applications that utilize energy demanding sensors or multiple sensors can take a proactive role that helps in saving energy by not triggering these sensors in case the smartphone position is found to be inappropriate.

- Collaborative sensing applications, where the context discovered by a single smartphone can be shared among a group, can enhance their accuracy by eliminating participants with noisy data due to disadvantageous smartphone positions.

Section 6 demonstrates that the accuracy of a context aware application is improved in this way. There are three important contributions of this paper:

1. A thorough study of the impact of smartphone position on all sensors that are currently prevalent in commercial smartphones is provided.

2. Based on the results of this study, a new smartphone position discovery service is proposed. This service detects the smartphone position very accurately and runs orthogonal to any other context-aware application.

3. Finally, the paper demonstrates that the accuracy of an existing context aware application is improved when run in conjunction with the proposed smartphone position discovery service.

2. MOTIVATION

Consider three different context-aware applications each utilizing different sensor(s). SurroundSense [4] performs logical localization such as detecting if the user is currently having coffee at Starbucks, partying, shopping at Wal-Mart, etc. The application is able to achieve logical localization by harnessing online sensor data from camera, microphone, and accelerometer sensors and comparing them with previous knowledge about the place. Smartphone position is a major obstacle for SurroundSense. For instance, if the smartphone is in a covered position, e.g. Hip Pocket, Pants Pocket, or Jacket Pocket, the system will not be able to take the required image to perform the color fingerprinting for the location. We believe that applications like SurroundSense can consult the smartphone position discovery service in order to take the required image when the phone is in reliable positions. We also believe that the sensor data from microphone and accelerometer would benefit from smartphone position by excluding the disadvantageous positions for the corresponding measured context.

Next, lets consider a an application that uses the microphone sensor. Researchers in [21] have developed an audio-based cough counting system running on a smartphone to monitor the health of patients with respiratory diseases. The application requires the patient to place the smartphone either in the shirt pocket or attached to a neck strap. In fact, the authors acknowledge that the chosen two positions do not represent an optimum choice in terms of patient comfort. However, the application needed to stick to these positions to achieve acceptable accuracy. We believe that smartphone position discovery service can play an important role if integrated with the cough counting system. Users can be given the chance to carry their smartphones freely and coughs will only be counted in suitable positions, i.e. when the smartphone is in the upper body region. In the data usage statistics of smartphones reported in [11], it was shown that user-smartphone interaction durations can be as high as 500 minutes a day. We believe that with such long interaction durations between the user and the smartphone, the cough system would still have enough opportunity to capture audio in favored positions without restricting the users to place their smartphones in specific positions.

Finally, we take an example for a context-aware application that is based on the accelerometer sensor. Here the benefit of smartphone position knowledge is not just bound to taking a go/no-go decision to capture contexts as in the previous two examples. Some applications provide better accuracy if trained for a single position. The fall classification application in [3] detects the type of fall from four different fall categories namely forward trips, backward slips, left lateral falls, and right laterals. The output of this application can be used by experts in the field of elderly care to develop fall prevention mechanisms and to assist first responders in providing more customized emergency procedures. In order to detect the type of fall, the application uses supervised machine-learning classifier with training data collected beforehand. Despite the fact that the experiment used a smartphone for data collection, users were not given the chance to carry the smartphone freely. Rather, the used smartphone was attached to the backside of a belt and users were asked to wear this belt and simulate the different categories of falls. Restricting the smartphone position in the experiment surely results in higher accuracy since the unification of position in both training and test data reduces the variability in the data, thereby, putting fewer burdens on the classifier. Nevertheless, we believe that such restriction in terms of smartphone position limits the practicality of the application.

To overcome the problem caused by arbitrary smartphone positions, we propose a two steps approach for such context-aware applications. First, the offline training for the classifier can be done with different smartphone positions to generate a classifier trained for each position. Second, with the presence of the smartphone position discovery service, the application will know in advance the current smartphone position and choose the classifier corresponding to that specific position during the classification process. To demonstrate the potential improvement in accuracy that this approach can achieve, we have conducted two experiments to detect the above-mentioned four types of falls (similar to [3]). In the first experiment, both the training data and the test data were collected at arbitrary smartphone positions by allowing the user to put the smartphone either in pants pocket, hip pocket or jacket pocket. The confusion matrix for this experiment is shown in Table 1. In the second experiment, three training files, each containing the four types of falls, were collected for three mentioned smartphone positions. Afterwards, the user was asked to simulate the required four types of falls and the classifier was pointed to the training file corresponding to the smartphone position, *assuming the smartphone position is known in advance*. The confusion matrix of the second experiment is shown in Table 2.

We notice that with arbitrary position (Table 1), the classifier fails significantly to distinguish the left lateral fall from the right lateral fall. Also, slips are being confused with right lateral falls in many occasions. The overall accuracy for the arbitrary positions experiment is 72.22%. We now turn into evaluating the ideal solution of assuming a smartphone position known in advance (Table 2). The results show a significant improvement for all fall categories with an overall accuracy of 94.8%. We conclude from these two experiments that with the knowledge of the smartphone position, the accuracy of fall classification improves dramatically, and in general the accuracy of any context aware application is likely to improve. However, the assumption of a complete knowledge of smartphone position is not a valid one. Any service that provides smartphone position information to a context aware application will most likely be not 100% accurate. So, given that a smartphone position discovery service is not 100% accurate, the question we hope to answer is whether the accuracy of a context aware application based on that position discovery service would still be higher than when that application does not use the position discovery service. We address this for the same experiment in section 6 after presenting the implementation details for the smartphone position discovery service.

		Prediction Percentage			
		1	2	3	4
Fall Type	1	**77.8**	0	0	22.2
	2	11.1	**88.9**	0	0
	3	0	0	**33.3**	66.7
	4	0	0	11.1	**88.9**

Table 1: Fall classification accuracy with arbitrary smartphone position using SMO classifier.
(1) Slips, (2) Trips, (3) Left Lateral, (4) Right Lateral.

		Prediction Percentage			
		1	2	3	4
Fall Type	1	**93.3**	0	6.7	0
	2	0	**100**	0	0
	3	6.7	0	**93.3**	0
	4	7.1	0	0	**92.2**

Table 2: Fall classification accuracy with arbitrary smartphone position using SMO classifier.
(1) Slips, (2) Trips, (3) Left Lateral, (4) Right Lateral.

3. IMPACT OF SMARTPHONE POSITION

We have conducted a series of experiments to study the impact of smartphone positions on raw sensor values. The objective of the study is to answer two questions. First, what sensors are most influenced by smartphone positions? Second, for those sensors, what are the features that can best reveal the differences in the raw data corresponding to the smartphone position? These features can then be utilized by our proposed smartphone position discovery service. We fo-

Figure 1: Inspected positions from left to right, Hand Holding, Talking on Phone, Watching a Video, Pants Pocket, Hip Pocket, Jacket Pocket and On-table.

cused on the sensors that are commonplace in current smartphones: accelerometer, gyroscope, light sensor, microphone, GPS and magnetometer. We collected data for each type of sensor at six different smartphone positions [9]. Since our goal is to cover the common scenarios in the daily life of a user, we added an extra position covering the typical situation of a smartphone placed on a table. Figure 1 illustrates these smartphone positions. Overall, ten users participated in these experiments. The analysis covered the physical contexts of idle, walking, and running. Following sections demonstrates the results of the experiments of walking users. A seperate section is devoted for the other physical contexts due to the challenge they impose on the position discovery service. The on table position is also analyzed separately due to its specific nature of being the only off-body position from the positions under analysis.

3.1 Accelerometer

Accelerometer, whether embedded in a smartphone or as a wearable sensor, has been widely used to analyze the physical activities of human beings [28, 20, 5, 10, 30, 22] as well as to detect other contextual information linked to the physical activities in the surrounding environment [34]. A single accelerometer reading provides three values in x-axis, y-axis, and z-axis. It is typical to use the magnitude as a single value to reflect on the three values. We logged the data for the accelerometer at the rate of 10Hz (i.e. 10 readings per second), which appeared to be sufficient to capture potential repetitive behavior. Figure 2 compares accelerometer magnitude values for two positions, hand holding and pants pocket. The plot reveals that the pants pocket position magnitude values exceed 15 m/s^2 very frequently and reaches 20 m/s^2 in many data points. Whereas, for the case of the hand holding position, the accelerometer magnitude spikes hardly reach the value of 15 m/s^2. The reason for this difference is that if the smartphone is in the lower body part region (i.e. pants pocket and hip pocket), it is going to be subject to more vibrations than if it is in the upper body part region. This observation is in harmony with many accelerometer analysis techniques in other research that required the smartphone to be placed near the pelvic region to better detect user physical activity [28, 20].

Next, we are interested in comparing two positions from the lower body part namely the pants pocket and the hip pocket positions. Unlike the previous case, we did not observe any big gap in the peak values for the two positions. Hence, we resorted to statistical analysis to reveal the potential differences. The mean, variance and standard deviation have been widely used in prior work with the accelerometer sensor [28, 20, 5]. In order to calculate these statistical features, we begin by dividing the data into time frames. Our chosen time frame size is five seconds, as this period of time is enough to capture any likely repetitive behavior influenced by the smartphone position. We then calculated the mean, variance, and standard deviation for each time frame. Figure 3 provides plots for the standard deviation of four participants for the pants pocket and hip pocket positions. Notice that the standard deviation of the pants pocket position is higher than the standard deviation for hip pocket positions for the first three users. However, for user no. 4 the opposite is true for most of the time frames except time frame numbers 4 and 6. We attribute such differences to the diversity in body movements' styles, different clothing (e.g. pocket size or design) and other measures related to the experimental environment. Nonetheless, we believe that the similarities found in first three users represent an opportunity that can be exploited by the smartphone position discovery service.

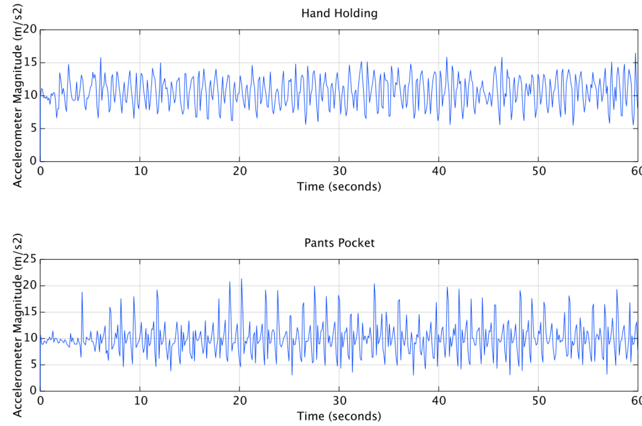


Figure 2: Accelerometer readings magnitude for hand holding and pants pocket positions.

3.2 Microphone

In the microphone experiment, our goal is to identify any differences in the sound recorded that can be linked to the smartphone position. The focus of a prior work with the microphone [25] was limited to determining whether the smartphone is inside or outside the user's pocket. Analyzing the friction noise generated as a result of the smartphone being inside a pocket can be used to unveil such information. We requested users to record a sound file using the smartphone with different smartphone positions under analysis. The recording was made in a quiet room to avoid any background noise interference. We recorded each audio clip at a rate of 44.1 kHz. The audio clips were split into five seconds frame segments to detect potential repetitive behaviors. We then carried out amplitude analysis, nearly similar to the

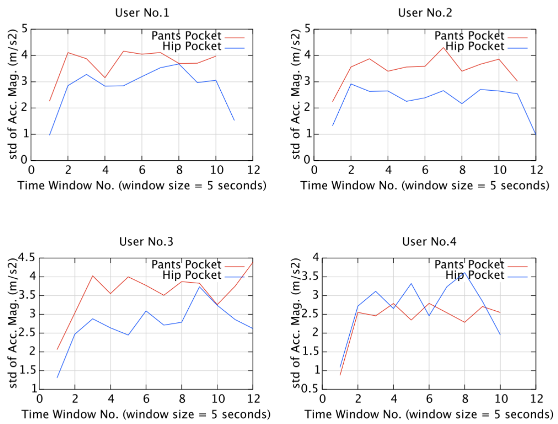


Figure 3: Accelerometer values standard deviation for pants pocket and hip pocket positions.

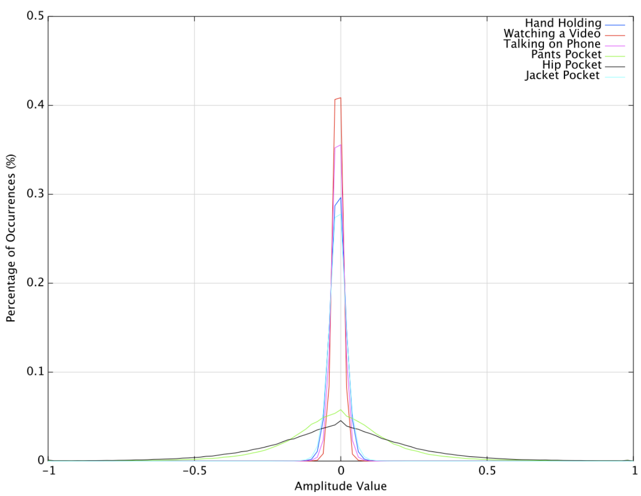


Figure 4: Sound signatures for different smartphone positions.

analysis done in [4], on the sound frame segments to generate sound fingerprints related to each smartphone position. The fingerprints are generated as follows. First, the amplitude range between -1 and 1 is divided into 200 equal intervals with one hundred intervals in the negative range and another one hundred in the positive range. Second, by counting the number of occurrences of amplitude values within each range we develop a histogram. Finally, the values corresponding to each interval in the histogram are divided by the total number of samples, which generates the percentage of amplitude occurrences within each interval. The acoustic fingerprints for different smartphone positions for a single user are illustrated in Figure 4.

By looking at the resultant fingerprints, one can notice that the upper body positions of hand holding, watching a video, talking on phone and inside jacket pocket have almost similar fingerprints with the biggest percentage of amplitudes concentrated near the zero amplitude value. Since the recordings were done in a quiet room, this result indicates that the friction noise for these positions was minimal. In contrast, the sound fingerprints for the pants pocket and hip pocket positions have amplitude values scattered in the range of -0.5 and 0.5. Therefore, we can conclude that the sound signatures divide the positions into two groups that are distinguishable from each other. However, the signatures within each group share common theme making the task of differentiating among them quite hard.

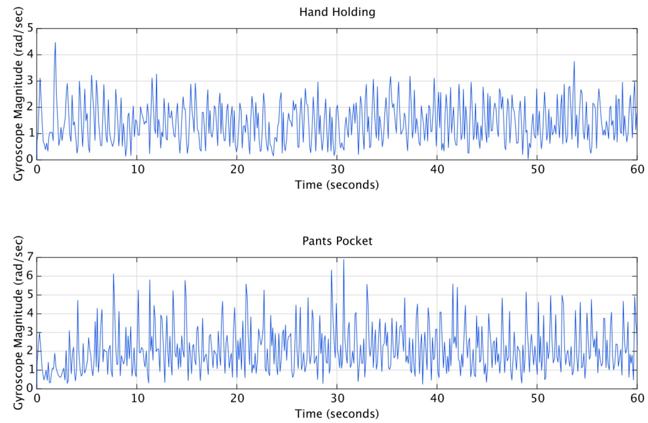


Figure 5: Gyroscope readings magnitude for hand holding and pants pocket positions.

3.3 Gyroscope

Similar to accelerometer, gyroscope has sensitivity to motion, and has been used in many context aware applications [16, 18, 15]. Gyroscope reading also consists of three components in the x-axis, y-axis, and z-axis. The three values represent instantaneous angular velocity of the smartphone in each direction. Our application for logging accelerometer values was also used to log gyroscope values at the same rate as that of accelerometer, 10 Hz. We conducted the same analysis that we did with accelerometer data to uncover any potential repetitive behavior of gyroscope raw data that can be attributed to the smartphone position. Figure 5 depicts the magnitude of gyroscope raw values for a user for two different smartphone positions, hand holding and pants pocket for one minute.

Notice that for the hand holding position, gyroscope magnitude values can hardly reach the 3 rad/sec, whereas, for the pants pocket position the magnitude values pass the 3 rad/sec a lot of times and reach 5 rad/sec on some occasions. This clearly indicates that the gyroscope sensor is analogous to accelerometer in having sensitivity to the smartphone position. In addition, Figure 5 reveals the difference between the upper body part positions and the lower body part positions. We now turn into another experiment aiming at identifying the potential differences between smartphone positions

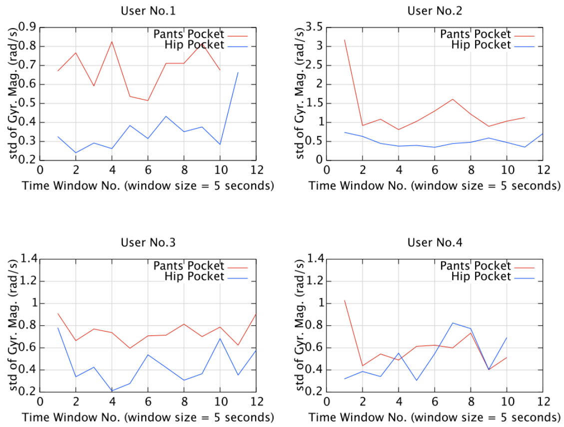


Figure 6: Gyroscope values standard deviation for pants pocket and hip pocket positions.

within the same body part. Figure 6 plots the standard deviation of the gyroscope data for four users in two different smartphone positions, pants pocket and hip pocket.

It is clear that similar to the accelerometer, the standard deviation values of pants pocket position are always higher than the hip pocket position. This is true for all users except user 4 who had the same behavior in the accelerometer standard deviation values.

3.4 GPS

The GPS is well known of not being able to work indoors due to blocking of microwave signals by buildings. However, our outdoor experiments have shown that different smartphone positions have no impact on the GPS location readings of latitude, longitude and altitude. Actually, this observation conforms to the results of [27].

3.5 Magnetometer

The magnetic field sensor measures the strength of earth magnetic field in the three directions of x, y, and z. It has been used in indoor localization techniques in the absence of GPS signals [18, 16, 8]. Figure 7 displays two plots of magnetometer readings magnitudes related to the positions of hand holding and pants pocket. We have used the magnitude to fuse the three directions values into a single reading for simpler comparison.

We would like to pinpoint two facts from the plot. First, the magnitude values for both positions are experiencing the similar cycle that starts with the peak before 20 seconds and ends with the peak before 40 seconds. Second, the hand holding position curve is smoother than the pants pocket curve. This can be noticed from the more frequent small spikes in the pants pocket curve. The first point can be attributed to the fact that while performing the experiment, our users were repeating the same circular path. Therefore, we can see the same impact of direction changes on the two

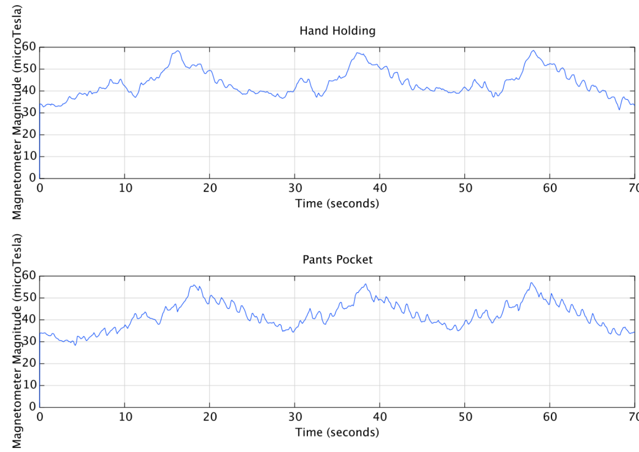


Figure 7: Magnetometer readings magnitude for hand holding and pants pocket positions.

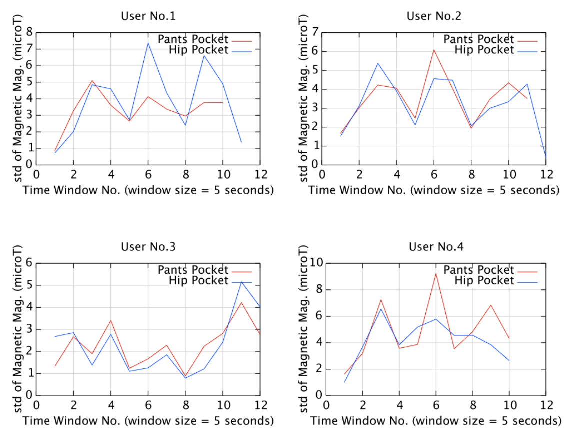


Figure 8: Magnetometer values standard deviation for pants pocket and hip pocket positions.

curves. The second point is actually related to the smartphone position. In the hand holding case, smartphone experienced fewer vibrations due to body movements making the curve smoother. In contrast, for the pants pocket position, the vibrations were high resulting in the spikes. We also look at the standard deviation of four of the users in Figure 8 to decide on the statistical features that can be used by the smartphone position discovery service.

One can notice from the figures that the standard deviations for the two body positions of pants pocket and hip pocket don't experience any pattern for the four users. This is due to the fact that the influence of direction changes is much higher than the influence of the smartphone positions vibrations. Since direction changes can be done arbitrarily, the smartphone position discovery service can't depend on magnetometer to perform the required distinction.

3.6 Light Sensor

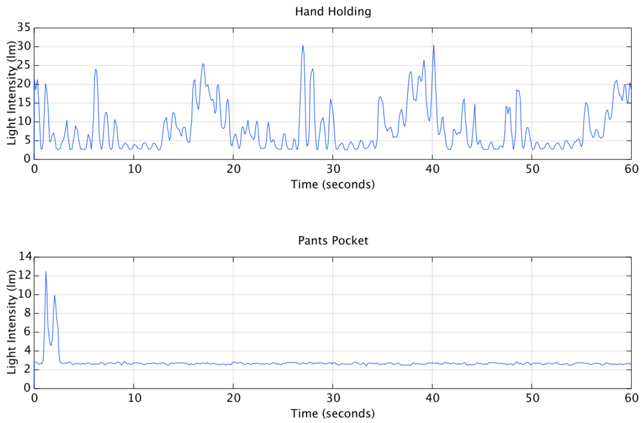


Figure 9: Light sensor readings for hand holding and pants pocket positions.

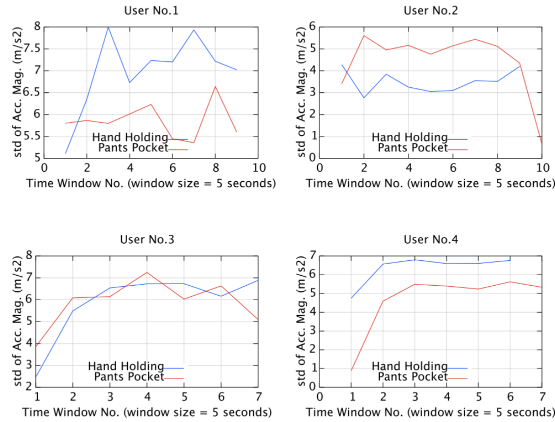


Figure 10: Accelerometer values standard deviation for hand holding and pants pocket positions for running users.

Light sensor provides a single reading in lumen (lm) representing the measured luminous flux. As expected, throughout our experiments, we have seen that the measured light intensity values are influenced by whether the smartphone is in covered or uncovered position. However, no patterns have been noticed that can be attributed to the specific smartphone position. Covered positions have average values nearly equal to the lowest light intensity value the smartphone can give. In contrast, uncovered positions produce much higher values that are dependent on the light intensity of the environment. This observation can be clearly spotted from Figure 9, which depicts the light intensity raw data for the two smartphone positions of hand holding and pants pocket.

3.7 Physical Contexts

All our experiments so far have been done for a walking user. Though walking context is popular, a user could be in other physical contexts such as idle and running. Notice that in the idle context, sensor values of accelerometer, gyroscope, magnetometer and GPS are unaffected by the smartphone position. On the other hand, all the sensors that are impacted by smartphone positions in walking context will also be impacted by smartphone positions in running context, although the nature and magnitude of the impact may be different. Since accelerometer demonstrated sensitivity to smartphone positions, we recorded accelerometer values for a running user. Figure 10 illustrates accelerometer standard deviation of four running users for hand holding and pants pocket positions.

Unlike the walking context, the raw data for the running context doesn't reveal any patterns that can be exploited by the smartphone position service to distinguish between the smartphone positions. By looking at Figure 10, we see that for user nos.1 and 4, the hand holding position has higher standard deviations for all frames when compared to the pants pocket position, whereas, the opposite is true for user no. 2. Also, for user no. 3, the standard deviation doesn't follow any pattern. We attribute this chaotic behavior to the range of different paces a running context might exhibit. When the user is asked to run, on some occasions the user would run very fast, while in others the same user would run relatively slower. Therefore, running context represents a challenging environment for the smartphone position discovery service.

3.8 On-table Position

A distinguishing factor between a smartphone on the table and another smartphone in an on-body position in the case of a non-moving user is the orientation. As can be seen in Figure 11, the gravity effect of 9.8 m/s^2 will appear in the z-axis for a smartphone placed on the table. This value will appear as positive if the smartphone is placed normally and as negative if the smartphone is flipped upside down. The x-axis and y-axis will have a value of nearly 0 for the same situation. Figure 11 also shows light intensity which can be used to discover if the smartphone is in a drawer.

3.9 Discussion

It is clear that the sensor values of accelerometer, gyroscope, microphone, magnetometer and light sensor are affected by smartphone position. Thus, context aware applications that depend on one or more of these sensors have the potential to benefit from a smartphone position discovery service. On the other hand, GPS sensor values remain largely unaffected by smartphone positions. Sensor values of accelerometer, gyroscope and magnetometer are affected by the differences in vibrations at different smartphone positions while microphone values are affected by friction noise and light sensor values are affected by whether the phone is covered or uncovered. So, a context aware application that is based on accelerometer, gyroscope and/or magnetometer sensor values is likely to benefit from the knowledge of ac-

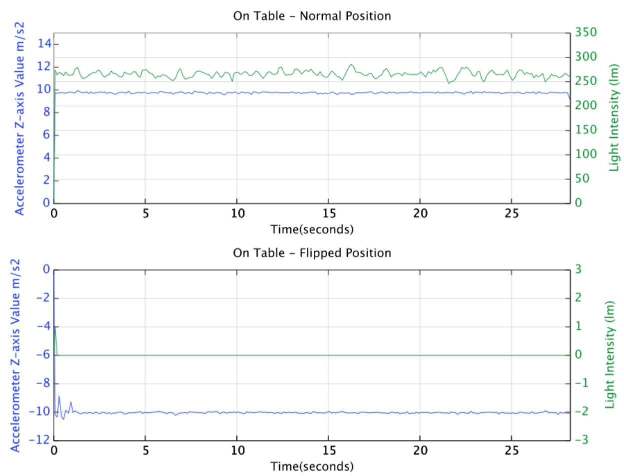


Figure 11: Accelerometer z-axis and light intesnity for a smartphone placed on a table with normal and flipped scenarios.

tual smartphone position such as hand holding, pants pocket, jacket pocket, hip pocket, etc. On the other hand, a context aware application that is based on light sensor values is likely to benefit from the knowledge of whether the phone is covered or uncovered. Finally, a context aware application that is based on microphone sensor values is likely to benefit from the knowledge of whether the phone is in the upper body position (hand holding, jacket pocket, talking on the phone, or watching video) or lower body position (hip pocket or pants pocket).

4. SMARTPHONE POSITION DISCOVERY

Based on our observations, we have designed, implemented and evaluated a smartphone position discovery service that provides four types of information: (1) Is the user idle, walking or running? (2) Is the phone covered or uncovered? (3) Is the phone placed in upper body or lower body? (4) What is the actual smartphone position? This service is designed to be configurable, so that an application can choose to receive only one or two or all types of information. The challenge in building the proposed service is that it utilizes sensor data from specific sensors (e.g. accelerometer and gyroscope), whose values are dependent on the physical contexts of the user. It is possible that the data from a particular sensor under one smartphone position and user activity is indistinguishable from the data from the same sensor under a different smartphone position and user activity. We address this challenge by detecting user's physical context (idle, walking or running) and utilizing data from multiple sensors. The key idea is that different sensors are affected differently by various user contexts, and we exploit these differences to accurately detect smartphone positions.

To detect whether the smartphone is in covered or uncovered position, the service compares the online captured light intensity data with a predefined threshold. The situation is more complex when it comes to the other finer granularity information. For both the upper-body/lower-body and the exact smartphone position decisions, the service uses machine-learning libraries to compare knowledge obtained from online sensor data with knowledge from labeled training data prepared offline. This classification process involves accelerometer or gyroscope or both sensors based on the preference of the serviced context-aware application. Figure 13 illustrates this design. It is worth noting that the complete solution runs on the smartphone. The smartphone position service can be utilized locally by other applications running on the same smartphone or remotely by collaborative sensing applications running on other smartphones.

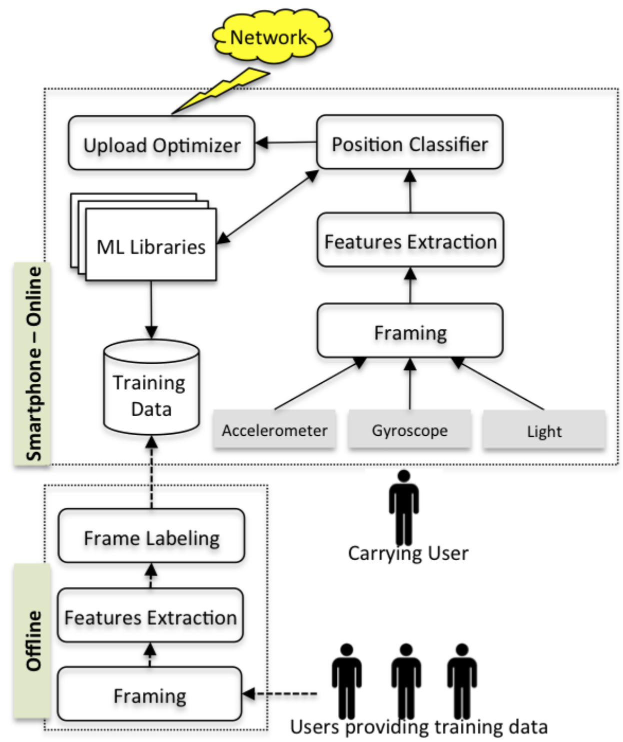


Figure 12: System Design for the Smartphone Position Discovery Service.

4.1 Offline Components

There are three offline components: Framing, Feature extraction and Frame labeling. The Framing component aims at capturing the repetitive patterns in the raw sensor data by dividing the data stream, from accelerometer and gyroscope, into five-second frames. Our choice of five-second frame size is based on analysis presented in [9] on the effect of frame size on step detection accuracy. Their analysis revealed that a frame size larger than 3 seconds is sufficient enough to provide good step detection accuracy and favored the five-second frame as it gives more accurate results.

The features extraction component calculates statistical features for each frame. Frame features must be chosen smartly to reveal the different patterns induced by each smartphone position. Our frame features are subset from the features presented in [20]. Based on our observations in Section 3, we have chosen the mean, variance, and standard deviation over 50 data points (10 data points per second) for each frame to capture the variations in accelerometer and gyroscope data. We have also included two other features (average for each axis and average absolute difference of each axis) related to each axis so as to capture the different orientations a smartphone can take for each position. This is because we noticed that each position has a common orientation that occurs quite frequently. Average of each axis captures the variation in the data due to body motion at the axis level. In addition, it reveals the nature of the orientation the smartphone is experiencing for each body position. Average absolute difference of each axis is the sum of the differences between each axis data point and the mean of that axis divided by the number of data points. We include the average absolute difference to enhance the solution accuracy in capturing the information revealed by axis data points.

Finally, the frame labeling component labels each frame with the corresponding smartphone position before loading the data to the training database. The labeling process was done manually. We asked our users to capture the data for the different smartphone positions while walking and labeled resultant frame segments with the practiced smartphone position during the experiment. Each frame record carries two labels: upper-body/lower-body position and the exact smartphone position.

4.2 Online Components

There are six online components: Training data, Machine learning libraries, Sensor values from accelerometer, gyroscope and/or light sensor, Framing, Feature extraction, Position classifier and Upload optimizer. The training data is the output from the offline components. Sensor data from ten users performing the same experiment for different smartphone positions was collected offline. After performing the (offline) framing and feature extraction processes, the resultant frame records constitute the knowledge database to be utilized for automatic discovery of smartphone position. Once ready, the training data is placed on the external memory card of the smartphone to be utilized by the smartphone position service.

We utilized Java language machine-learning libraries provided by the WEKA tool [1]. Specifically, for the implementation of the smartphone position discovery service, we used WEKA for Android [2], which is a version from WEKA libraries ported to Android platform. The correctness of used classifiers was tested by performing a test experiment with the same training and test data on a desktop by normal WEKA and on Android device by WEKA for Android. Same results were obtained for the two experiments.

We chose three sensors (accelerometer, gyroscope and light sensor) based on the analysis done in Section 3. Accelerometer is used for detecting physical context, accelerometer and/or gyroscope are used for detecting the actual phone position, and light sensor is used for detecting whether the phone is covered or uncovered. The use of two sensors for detecting actual phone position is subject to a tradeoff between energy consumption and smartphone position detection accuracy. The framing and features extraction components have the same functionalities as in the offline case. The only addition is the capture of average light intensity per frame, which is not required for training the classifier.

The position classifier component receives the gathered online frame data and uses it in three ways. First, it compares the standard deviation of accelerometer magnitude with predefined thresholds to determine idle/walking/running contexts.Second, it consults the machine learning classifier to detect smartphone position information. Third, it compares the light intensity average of the frame with a predefined threshold to determine covered/uncovered position. In the case of idle or running contexts, the position service provides the latest smartphone position discovered under proper physical context along with a timestamp and leave it to the consuming context-aware application to use this cached smartphone position based on its accuracy preferences. Our goal here is to exploit the fact that, in some situations the user might change their physical context but maintain the same smartphone position.

Finally, the upload optimizer is utilized only in case the position is required to be relayed over the network as part of a collaborative sensing solution. We developed this component because we envision the smartphone position discovery service to be an important part of collaborative sensing applications. The upload optimizer logic is based on optimization techniques discussed in [26]. The optimizer implements three alternative techniques for upload optimization: (1) Upload whenever a position change occurs; (2) Upload when a position change persists for some period of time; and (3) Upload the position with the highest number of occurrences within a window of given size. While the first technique is simple and provides most accurate results, it is subject to noise due to momentary smartphone position changes. The second technique eliminates this noise and reports only more permanent position changes. Finally, the third technique is suitable when there are frequent smartphone position changes. This technique tries to report the most commonly occurring smartphone position.

4.3 Use Case Scenario

Assume a context-aware application that is interested in the exact position. The flow diagram in Figure 13 demonstrates the flow of execution for the position service to provide this information. In the beginning, the service will use the variance of the accelerometer for the captured window to detect the physical context. If the smartphone is idle, the service will detect if it is placed on table or any other position. In case the smartphone is idle and not on table, the exact position is difficult to provide and a cached recent value along with the activity is returned instead. Also, a running context means that position can't be detected and a cached value is returned. If the user is walking, the service will utilize the online features extracted from the accelerometer and gyroscope to provide the exact position.

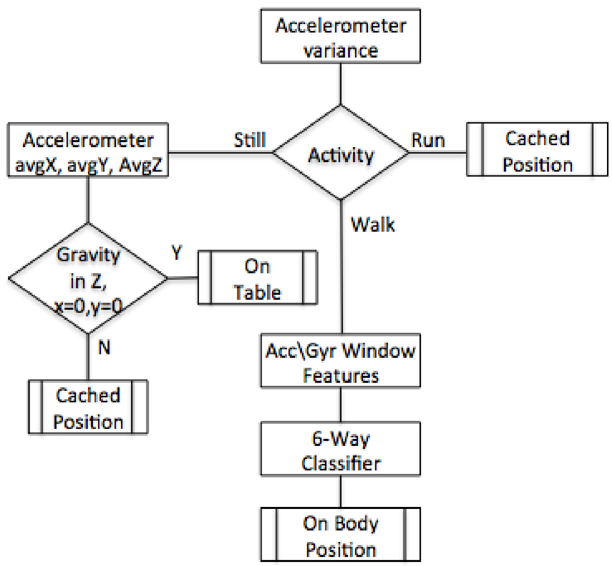


Figure 13: Position service execution flow for a request for an exact position.

5. IMPLEMENTATION AND EVALUATION

We implemented the smartphone position discovery service on Samsung Galaxy Note device running Android version 4.0.3 (Ice Cream Sandwich). The device has a Dual-core 1.4 GHz ARM Cortex-A9 processor and 1GB of RAM and is equipped with the accelerometer, gyroscope and light sensors required for the service. We collected data from ten different participants to train the smartphone position discovery classifier. Before conducting the experiment, an approval was obtained from the Institutional Review Board at CU Boulder. We asked each participant to carry the smartphone in the six smartphone positions. The experiment setup was kept as natural as possible. Participants were free to move at their own pace and place their smartphones at any orientation they liked.

5.1 Physical Contexts

The physical contexts of idle or running is quite straight forward to detect as these contexts represent the two extremes of no movements (for idle) and drastic movements (for running), which make them easily distinguishable from the normal walking context.The two physical contexts can be distinguished by looking at the accelerometer standard deviation of the frame and comparing it to predefined threshold values. The threshold values we used were 0.5 m/s^2 and 5 m/s^2 to detect idle and running contexts respectively. In both cases, the position service responds with the physical context detected and the latest cached smartphone position. In our experiments, these thresholds achieved near-perfect accuracy.

	Percentage Accuracy of Position Discovery		
	Naive Bayes	Simple Logistic Regression	J48
Upper-Body	89.2	91.2	92.8
Lower-Body	77.2	90.5	83.9
Total Accuracy	84.6	90.9	89.4

Table 3: Upper-body vs lower-body detection accuracy based on accelerometer features using different classifiers.

5.2 Covered vs. Uncovered

This information is typically important for context-aware applications consuming the camera input, e.g. [4, 6]. Such applications can save energy by avoiding the cost of taking a picture in case the smartphone is in covered position. We noticed that the least light sensor reading from our smartphone is around 2.5 lm. This reading is produced when there is no light in the surrounding environment or when the smartphone is in covered position. Therefore, we used the value of 3 lm as a threshold to distinguish between a covered and uncovered position. If the average light intensity for the online captured frame is less than 3 lm, we conclude that the smartphone is in covered position, and vice versa. This choice worked almost perfectly except in two rare cases. First, in a dark environment, the reading would always indicate covered position. Second, if the smartphone is in covered position, however, the user clothing happened to have a degree of transparency.

5.3 Smartphone Position: Accuracy

5.3.1 *Upper-Body vs Lower-Body*

Smartphone positions covered in this research can be divided into two groups: upper-body group, including hand holding, talking on phone, watching a video, and jacket pocket; and lower-body group, including pants pocket and hip pocket. To detect the group that a smartphone is in, we trained the classifier with accelerometer data from 10 users and carried a 10-folds cross-validation test. The results of the classification process with different classifiers are shown in Table 3. The achieved accuracy using accelerometer is fairly high for the simple logistic regression and J48 classifiers. Therefore, we conclude that the accelerometer is the best candidate to perform this classification task and exclude gyroscope from our analysis.

5.3.2 *Exact Smartphone Position*

Both accelerometer and gyroscope have shown sensitivity to smartphone positions. Our goal here is to compare be-

	Percentage Accuracy of Position Discovery			
	Naive Bayes	Multilayer Perceptron	Logistic Regression	J48
Handholding	75.9	83.8	94.5	97.8
Watching a Video	91.8	93.2	96.0	97.0
Talking on Phone	79.4	89.7	91.7	91.3
Pants Pocket	65.4	67.9	58.0	78.2
Hip Pocket	57.6	78.8	71.6	90.7
Jacket Pocket	27.6	67.1	73.2	75.3
Total Accuracy	66.7	80	80	88.5

Table 4: Smartphone position detection accuracy based on accelerometer features using different classifiers.

tween the two sensors. In the beginning, we conducted a test experiment with the six smartphone positions and collected the data for both the accelerometer and gyroscope. Then, to evaluate a single sensor, we kept the data for that sensor and deleted the data for the other sensor. By doing so, we ensure fair comparison since the three results we show next are basically for the same experiment, but, with different sensors included. Here we also used data from 10 users and performed a 10-folds cross-validation test. Table 4 illustrates the results of smartphone position classification using only accelerometer. We note that the J48 decision tree classifier achieves good accuracy of 88.5% with the accelerometer as the only input. On the other hand, the Naive Bayes classifier had the lowest accuracy of 66.7%. We also note that the source of confusion varies from one classifier to another for the same experiment. For example, in the multilayer perceptron experiment, the jacket-pocket position produced the lowest accuracy. On the other hand, with the Logistic Regression in use, the pants pocket position was the hardest position to classify.

Next, we evaluate the service when gyroscope is in use. Table 4 illustrates the results of smartphone position classification using only gyroscope. We note that all classifiers achieved lower total accuracy when compared to the use of accelerometer. This shows that the gyroscope is less sensitive to smartphone positions than accelerometer. Nevertheless, the accuracy achieved by the gyroscope is still at acceptable levels making the sensor worth considering for some cases. For example, some positions achieved accuracy level of above 80% for some classifiers. However, the overall accuracy remains less than the values achieved when the accelerometer is in use.

Now, let's move on to the situation where we use both accelerometer and gyroscope to detect the smartphone position. Table 6 provides the position discovery results for both sensors. As expected, all the classifiers achieved a gain in accuracy when compared to the previous two single-sensor configurations. We also note that three out of the four classifiers achieved very high accuracy levels (above 80%). However, this improved accuracy comes at the cost of increased energy demands by continuously sensing two sensors and performing the features extraction calculations twice.

	Percentage Accuracy of Position Discovery			
	Naive Bayes	Multilayer Perceptron	Logistic Regression	J48
Handholding	72.2	78.4	85.4	63.6
Watching a Video	64.7	81.8	84.8	84.2
Talking on Phone	48.4	64.8	72.3	75.6
Pants Pocket	60.6	82.0	75.5	55.2
Hip Pocket	52.4	72.6	64.1	52.3
Jacket Pocket	46.7	62.0	52.9	47.6
Total Accuracy	57.6	74.0	72.7	62.9

Table 5: Smartphone position detection accuracy based on gyroscope features using different classifiers.

	Percentage Accuracy of Position Discovery			
	Naive Bayes	Multilayer Perceptron	Logistic Regression	J48
Handholding	76.6	86.4	98.9	92.6
Watching a Video	92.4	91.3	98.0	95.8
Talking on Phone	76.9	91.9	96.8	93.1
Pants Pocket	73.2	85.8	78.8	75.9
Hip Pocket	66.2	93.5	88.3	83.5
Jacket Pocket	51.4	80.8	75.6	68.6
Total Accuracy	73.0	88.6	89.3	84.9

Table 6: Smartphone position detection accuracy based on both accelerometer and gyroscope using different classifiers.

5.3.3 *Smartphone Position: On-Table Position*

Due to its special nature, the on table position is handled separately. We directly use the window features of average x-axis, average y-axis, and average z-axis to detect this position. The z-axis value will be nearly equal to the gravity pull value of 9.8 m/s^2 whereas the x-axis and y-axis will have the value of near zero. We also use the light to detect if the smartphone is placed on a table or is actually inside a drawer. By detecting this position separately we avoid the cost of the classification while still achieving a good detection accuracy.

5.3.4 *Smartphone Position: Energy Consumption*

To analyze the energy demands of the proposed smartphone position deiscovery service, we measured the battery drain when each of the three configurations is employed. The results are shown in Figure 14.

Battery consumption measurements suggest that the use of the gyroscope sensor adds a considerable energy consumption overhead. With ten hours of operation based on accelerometer only, the smartphone position service drained less than 10% of the smartphone battery. However, when both sensors are employed, the battery drain for ten hours surpasses 50% of battery lifetime. On the other hand, using light sensor only drained 40% over the same period. This clearly indicates that there is a tradeoff between accuracy and energy consumption. We believe that there is still space to enhance the energy consumption of our solution by employing efficient continuous sensing techniques proposed in [23, 33]. In addition, an important point that we would like to highlight is the possibility of sharing sensor data between the smartphone position service and the context-aware ap-

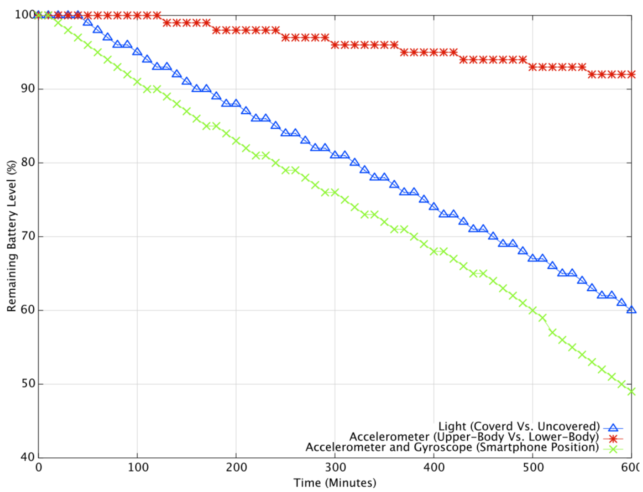


Figure 14: Battery consumption for the three sensor configurations.

	Percentage Accuracy of Position Discovery			
	Naive Bayes	Multilayer Perceptron	Logistic Regression	J48
Handholding	100	100	100	66.67
Watching a Video	92.30	100	100	92.31
Talking on Phone	100	100	100	100
Pants Pocket	100	94.44	100	83.33
Hip Pocket	100	100	93.33	100
Jacket Pocket	100	100	100	100
Total Accuracy	98.71	99.07	98.88	90.38

Table 7: Smartphone position accuracy using different classifiers and custom-trained data.

plication, which occurs when there is a match between the sensors required for the two. In such scenarios, the smartphone position service would just intercept the sensor data that is already being gathered by the application, thereby, eliminating the need for extra sensing.

5.3.5 *Group Training vs Custom Training*

A smartphone is typically a personal device owned by a single user. Therefore the idea of each user (custom) training his/her position discovery classifier is worthwhile. We experimented with this idea, where a user trained his smartphone by performing the above-mentioned classifier training experiments. Next day, we collected sensor data from the same user and ran our smartphone position discovery service using the custom-trained classifier from the previous day. Table 7 shows the accuracy of smartphone position detection when both accelerometer and gyroscope data were used.

We can see that the total accuracy for each classification algorithm has improved dramatically (compare the results with Table 6). One point to not is that the user wore similar clothing on both days in this experiment. We expect that the detection accuracy may be slightly lower for different style of clothing. One way to address this is to train the classifier with different clothing styles. The idea of training a classifier on smartphone by the user before application use has been used in [24]. However, the authors tried to keep the training period as minimum as possible as they believed that users might refrain from using applications requiring training beforehand. We share the same concern and believe that our application can be installed with multiple users training data, which has shown acceptable accuracy levels, and the user is then given a choice for custom training.

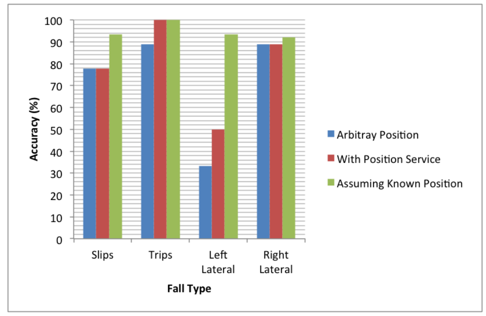
6. APPLICATIONS

In Section 2, we observed that there is a significant overall improvement in the accuracy of fall detection if the smartphone position is known (72.2% to 94.7%). However, we noted that the assumption of a complete knowledge of smartphone position is not a valid one. Results in Sections 5 showed that our smartphone position service does not provide 100% accuracy. To understand the impact of a smartphone position discovery service that is not 100% accurate, we implemented the fall classification application [3] and integrated it with our smartphone position discovery service to provide the smartphone position information to the application. We used the two-steps approach of first performing online detection for the current smartphone position, and then choosing a classifier trained for the same smartphone position to detect the type of fall.

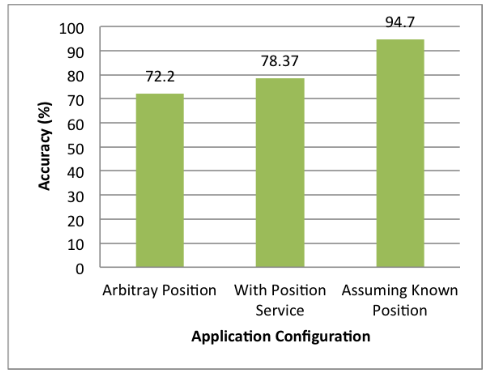
Figure 15(a) and (b) show the “per-fall“ and “overall accuracies“ for this case. We also included the results from the other two earlier experiments (Tables 1 and 2) of arbitrary smartphone position and complete smartphone position knowledge to make it easier for the reader to grasp the effect of introducing the smartphone position discovery service. Notice from Fig. 14(a) that the accuracies of trips and left lateral falls detection have been improved. For the other two types of falls, introducing the position service to the scene didn’t improve the results but didn’t negatively impact them. We would like to stress on two facts. First, we used a three-way classifier in this experiment to detect the positions of pants pocket, hip pocket and jacket pocket. This choice has made the job of the classifier easier. However, these choices are the ones relevant to the fall classification experiment since the other three positions include the user carrying the smartphone in hand. During a fall, a hand can wave chaotically and it is difficult to capture this behavior in a classifier. The second fact is that we anticipate that each context-aware application will be interested in its own set of positions just as the fall classification is only interested on those covered body-positions, which justifies our choice of positions. For example, a wide range of applications depending on the camera sensor will be interested in positions where the smartphone is not covered.

Fig.14(b) reflects the overall accuracy improvement. The improvement in the case of “known-position“ proves the fact that by introducing the position service, context-aware applications will achieve better results. We also saw an improvement for the case of with-the-position service. However, the improvement was not as significant as in the optimal situation. We noticed that our position service provided the correct position in most situations, but it was the fall classification that is difficult to achieve due to arbitrary after fall behaviors.

The accuracy gain from using the position service introduces three sources of cost. First, the addition of the gyroscope sensor placing more demands on the smartphone battery as shown in Section 5.2. We would like to point here that since the fall classification application is already collecting accelerometer data, the user might choose to operate only on the accelerometer data, thereby, eliminating the extra sensing cost, though with lower accuracy gain. Second, the additional memory required for training five classifiers - one for position and four for fall types - compared to a single classifier. In our experiment, memory consumption shows an increase of less than 1MB of RAM usage for the case of using the two-steps approach with the additional training files, one for each smartphone position, loaded in memory. Current smartphones - similar to the galaxy note used in the experiment - come with a 1GB of RAM, making this additional cost acceptable. The third cost is related to the more CPU overhead incurred by performing two classification operations - one for position and another for fall classification-rather than a single classification for the fall detection. However, we didn't see any noticeable increase by introducing the second classification operation.



(a) Per-Fall Accuracy



(b) Overall Accuracy

Figure 15: Fall detection accuracy for the three situations of arbitrary smartphone position, with smartphone position service, and assuming known smartphone position.

7. RELATED WORK

Many researchers have studied the problem of smartphone body-position. We focus on studies aimed at providing generic solution for the problem. The work in [19] anticipated the importance of body-position knowledge even before popularity of sensor equipped smartphones. The provided analysis utilized wearable accelerometer sensor to differentiate between four body-positions. However, a valid note raised by the authors was that, as opposed to the built-in smartphone accelerometers, the wearable sensor has the advantage of being firmly attached to the body position, which will certainly lead to better accuracy in position detection.

[13] is another early work addressing the smartphone position problem in general. However, the covered positions were limited to on-table, in-hand, and inside-pocket. The authors implemented a prototype to automatically adjust the current ringing profile based on the discovered position. The solution augmented the smartphone with a 2D accelerometer and demonstrated an accuracy of 87% in discriminating between the mentioned positions. The authors in [25] presented some preliminary work to distinguish between the in-pocket and out-of-pocket body-positions for a smartphone based on the microphone sensor. Good accuracy level of 80% was acheived. However, the positions covered in this solution remain limited and we believe that more precise body-position solution is required.

The work in [12] utilizes accelerometer to distinguish between five body-positions. According to the offline evaluation in the paper, 72.3% accuracy is achieved for data collected from 10 users with the use of multilayer perceptron classification algorithm. The work in [29] suggested the use of a rotation-based approach to recognize four body-positions. The presented solution is based on accelerometer and gyroscope. Achieved accuracy using SVM classification was 91.69%. Our solution achieves comparable accuracy with a fully working solution on a smartphone. The work in [32] targets body-position localization of wearable sensors used for health monitoring applications. Authors used SVM to achieve a localization accuracy of 87% when distinguishing between 6 body-positions. The studied positions are typical for health sensors and are not applicable to smartphones.

Finally, authors in [14] have taken a completely different approach by utilizing multispectral material sensor - external to the smartphone - to distinguish between the different smartphone positions (including body-positions). The solution is based on the idea that smartphone positions are typically correlated with surrounding material with specific features, which the sensor they used is able to detect.

8. CONCLUSION AND FUTURE WORK

This paper presented an automatic solution for smartphone position discovery implemented entirely on a smartphone. The proposed solution can act as a service provider to context-aware applications running on the same smartphone by providing them with smartphone position information. The service can answer the following four questions. (1) Is the user idle, walking or running: (2) Is the smartphone covered or uncovered? (3) Is the smartphone attached to upper-body or lower-body? (4) What is the actual position of the smartphone? In order to answer each question, the service utilizes specific sensor(s) chosen carefully based on a thorough analysis presented in the paper. The paper also presented analysis for the accuracy and energy consumption of the solution. Context-aware application developers can use the smartphone position service to enhance the accuracy of their applications where each application can choose among the different information offered based on their accuracy requirements and energy constraints.

Our solution requires continuous sensing since a transition from one smartphone position to another can occur at an arbitrary time, and we need to discover this transition spontaneously. In our future work, we are planning to overcome this problem by employing techniques similar to the hierarchical sensing technique presented in [33], which perform discovery only if a position state change occurred. We propose to use the accelerometer at a minimum sampling rate needed for state transition detection. Once a state transition is found, the more expensive sensors of gyroscope and light, will be called as needed to detect the new smartphone position.

One popular position that we did not include is inside bag position. This position is difficult to detect due to arbitrary orientations a smartphone can take and the chaotic movement patterns it can experience inside a bag. We believe that the sound and light sensors can help detect this position. From the light sensor we can conclude that the position is a covered position. Then, in order to distinguish among the set of covered position (i.e. inside pocket and inside bag), we can utilize the microphone. This is another area of future work.

9. REFERENCES

[1] Weka 3: Data mining software in java.

[2] Weka for android.

[3] Mark V Albert, Konrad Kording, Megan Herrmann, and Arun Jayaraman. Fall classification by machine learning using mobile phones. *PloS one*, 7(5):e36556, 2012.

[4] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *MobiCom*, 2009.

[5] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive Computing*, pages 1–17, 2004.

[6] X. Bao and R. Roy Choudhury. Movi: mobile phone based video highlights via collaborative sensing. In *MobiSys*, 2010.

[7] G. Chen, D. Kotz, et al. A survey of context-aware mobile computing research. Technical report, TR2000-381, Dept. of CS, Dartmouth College, 2000.

[8] J. Chon and H. Cha. Lifemap: A smartphone-based context provider for location-based services. *Pervasive Computing, IEEE*, 10(2):58–67, 2011.

[9] Y. Chon, E. Talipov, and H. Cha. Autonomous management of everyday places for a personalized location provider. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(4):518–531, 2012.

[10] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, et al. The mobile sensing platform: An embedded activity recognition system. *Pervasive Computing, IEEE*, 7(2):32–41, 2008.

[11] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *MobiSys*, 2010.

[12] K. Fujinami, S. Kouchi, and Y. Xue. Design and implementation of an on-body placement-aware smartphone. In *ICDCSW*. IEEE, 2012.

[13] H.W. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications*, 7(5):341–351, 2002.

[14] C. Harrison and S.E. Hudson. Lightweight material detection for placement-aware mobile computing. In *UIST*, 2008.

[15] D.A. Johnson and M.M. Trivedi. Driving style recognition using a smartphone as a sensor platform. In *ITSC*. IEEE, 2011.

[16] W. Kang, S. Nam, Y. Han, and S. Lee. Improved heading estimation for smartphone-based indoor positioning systems.

[17] A.M. Khan, Y.K. Lee, SY Lee, and T.S. Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *FutureTech)*. IEEE, 2010.

[18] N. Kothari, B. Kannan, E.D. Glasgwow, and M.B. Dias. Robust indoor localization on a commercial smart phone. *Procedia Computer Science*, 10:1114–1120, 2012.

[19] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster. Where am i: Recognizing on-body positions of wearable sensors. *Location-and Context-Awareness*, pages 257–268, 2005.

[20] J.R. Kwapisz, G.M. Weiss, and S.A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74–82, 2011.

[21] E.C. Larson, T.J. Lee, S. Liu, M. Rosenfeld, and S.N. Patel. Accurate and privacy preserving cough sensing using a low-cost microphone. In *UbiComp*, 2011.

[22] Y.S. Lee and S.B. Cho. Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer. *HAIS*, pages 460–467, 2011.

[23] H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, and A.T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys*, 2010.

[24] E. Miluzzo, C.T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A.T. Campbell. Darwin phones: the evolution of sensing and inference on mobile phones. In *MobiSys*, 2010.

[25] E. Miluzzo, M. Papandrea, N.D. Lane, H. Lu, and A.T. Campbell. Pocket, bag, hand, etc.-automatically detecting phone context through discovery. In *PhoneSense*, 2010.

[26] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A. Campbell. Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones. *Pervasive Computing*, pages 355–372, 2010.

[27] Y. Odaka, Y. In, M. Kitazume, K. Hirano, M. Higuchi, S. Kawasaki, and H. Murakami. Error analysis of the mobile phone gps and its application to the error reduction. *Seikei University engineering research report*, 47(1):75–82, 2010.

[28] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman. Activity recognition from accelerometer data. In *AAAI*, 2005.

[29] Y. Shi, Y. Shi, and J. Liu. A rotation based method for detecting on-body positions of mobile devices. In *UbiComp*, 2011.

[30] F. Sposaro and G. Tyson. ifall: An android application for fall monitoring and response. In *EMBC*. IEEE, 2009.

[31] L. Sun, D. Zhang, and N. Li. Physical activity monitoring with mobile phones. *Toward Useful Services for Elderly and People with Disabilities*, pages 104–111, 2011.

[32] A. Vahdatpour, N. Amini, W. Xu, and M. Sarrafzadeh. Accelerometer-based on-body sensor localization for health and medical monitoring applications.

[33] Y. Wang, J. Lin, M. Annavaram, Q.A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *MobiSys*, 2009.

[34] P. Zhou, Y. Zheng, and M. Li. How long to wait?: Predicting bus

arrival time with mobile phone based participatory sensing. In *MobiSys*, 2012.