Vibration-Based Surface Recognition for Smartphones

Jungchan Cho, Inhwan Hwang, and Songhwai Oh
CPSLAB, ASRI
School of Electrical Engineering and Computer Science
Seoul National University, Seoul, Korea
Email: {jungchan.cho, inhwan.hwang, songhwai.oh}@cpslab.snu.ac.kr

***Abstract*—With various sensors in a smartphone, it is now possible to obtain information about a user and her surroundings, such as the location of a smartphone and the activity of the smartphone user, and the obtained context information is being used to provide new services to the users. In this paper, we propose *VibePhone*, which uses a built-in vibrator and accelerometer, for recognizing the type of surfaces contacted by a smartphone, enabling the sense of touch in smartphones. For humans and animals, the sense of touch is fundamental for both recognizing and learning the properties of objects. The sense of touch is obtained from the texture of an object and humans recognize the type of an object by scrubbing the surface with fingers. Since a smartphone cannot physically scrub the contacting surface, we emulate the touch by generating vibrations using a smartphone and propose a method to recognize the type of contacting objects. The recognition of the object type by vibration alone is an extremely difficult task, even for a human. However, we demonstrate that it is possible to distinguish object types into broad categories where a phone is usually placed, e.g., sofas, plastic tables, wooden tables, hands, backpacks, and pants pockets. The proposed *VibePhone* system achieves an accuracy over 85% on average. We have prototyped *VibePhone* on an Android-based smartphone which changes its background display based on the contacting surface. We envision that the haptic perception in future smartphones will enable new experiences to the users.**

***Index Terms*—Context-awareness, Smartphones, Accelerometers, Vibration, Surface Recognition**

I. Introduction

The knowledge of the user's location can be useful for a number of applications, including location based services (LBSs). The localization problem can be divided into outdoor localization and indoor localization problems. Outdoor localization is well solved by GPS, but indoor localization has many challenging issues. In [1], the indoor localization problem is solved by instrumenting the environment with sensors but it requires high installation and management costs. Although an existing wireless network infrastructure, such as a cellular network and Wi-Fi, has been used for localization [2], [3], it is difficult to distinguish between neighboring rooms (coarse localization). Recently, for fine localization, some approaches [4], [5] used a combination of sensors, such as a microphone, camera, and accelerometer on a phone.

All the approaches explained above assume that the user carries a smartphone at all times. Therefore, it is important to check whether the user carries a smartphone. This problem can be solved if a smartphone can recognize the type of a contacting object. In addition, the information about the contacting object can be used to schedule embedded sensors to save energy. For example, if a smartphone recognizes that the contacting object is a hand, the smartphone can turn on sensors for sensing the user context. The present paper is concerned about a method for recognizing the type of surfaces contacted by a smartphone, enabling the sense of touch to smartphones. For humans and animals, the sense of touch is fundamental for both recognizing and learning about the properties of everyday objects. The sense of touch is obtained from the texture of an object and humans recognize the type of an object by scrubbing the surface with fingers. Since a smartphone cannot physically scrub the contacting surface, some researchers tried to emulate the touch by generating vibrations using the built-in vibrator in a smartphone.

Shafer [6] used a vibrator and accelerometer to classify seven pre-defined locations using a support vector machine with time-series accelerometer readings and achieved a recognition rate of 70%. Kunze et al. [7] proposed a symbolic localization method through active sampling of acceleration and sound signatures. Their method used vibration and sound (short and narrow frequency beeps) to sample the response of the environment. Two potential applications are proposed in [7]. One application is to identify a previously encountered location from historical data while the other application is to recognize the surface types. Although their approach achieves recognition rate of up to 81% for identifying one of 35 trained locations and 86% for recognizing 12 surface types, the recognition rates using vibration alone are only 25% for the specific location identification and 65% for recognizing 12 surface types. It can be seen from their experiments, recognizing the contacting surface object or its type by vibration alone is an extremely difficult task. In this paper, we focus on recognizing the surface type and improving the recognition accuracy using only vibration through a study of various feature sets from accelerometer readings.

We introduce a method to recognize an object type using a smartphone and analyze acceleration responses to different surfaces in frequency and time domain. The proposed method is called *VibePhone* and it uses the built-in vibrator and accelerometer of a smartphone. When a smartphone vibrates, we record the corresponding accelerometer reading. By analyzing the vibration pattern from different types of surfaces, we can classify the type of the contacting surface. To improve the

Fig. 1. Examples of different surface classes.

classification accuracy, we have developed and tested a number of different features that are constructed from accelerometer readings. We have also analyzed the effect of the sampling rates of an accelerometer and found that a higher sampling rate achieves a better performance.

We have evaluated the performance of *VibePhone* on broad categories of surfaces where a phone is usually placed, i.e., sofas, plastic tables, wooden tables, hands, backpacks, and pants pockets (see Figure 1). The experimental results show the average precision is over 85% and the time domain information is useful when the sampling rate is low. We have implemented *VibePhone* on an Android-based smartphone and we evaluated its performance by placing the phone on new objects which are similar to the surface classes known to *VibePhone* and *VibePhone* recognized the correct surface classes with an accuracy of 82%. The experimental results show that it is possible to recognize the type of surfaces using the built-in vibrator and accelerometer of a smartphone.

The remainder of this paper is organized as follows. Section II describes the design of *VibePhone*. Experimental results are shown in Section III. The implementation of *VibePhone* on a smartphone is discussed in Section IV and the smartphone implementation is evaluated in Section V.

II. Design

A. Design Considerations

We are surrounded by objects with different surfaces as shown in Figure 1. From experience, we know that a smartphone vibrates differently on different surface types. However, the recognition of the type of surface by vibration alone is difficult, even for human. Hence, instead of identifying a specific object by vibration alone, we group objects into broad categories (surface classes) where a phone is usually placed, i.e., sofas, plastic tables, wooden tables, hands, backpacks, and pants pockets. Then we measured vibrations repeatedly on the same surface class such that we collect enough measurements representing the characteristic of the class.

B. System Overview

An overview of *VibePhone* is shown in Figure 2. In the calibration stage, raw accelerometer readings are normalized into standard G unit (g-force) using device-specific parameters to compensate hardware variations of accelerometers in different smartphones. The normalized accelerometer readings are divided into frames and *VibePhone* finds global vertical and horizontal directions. With two global directions for each

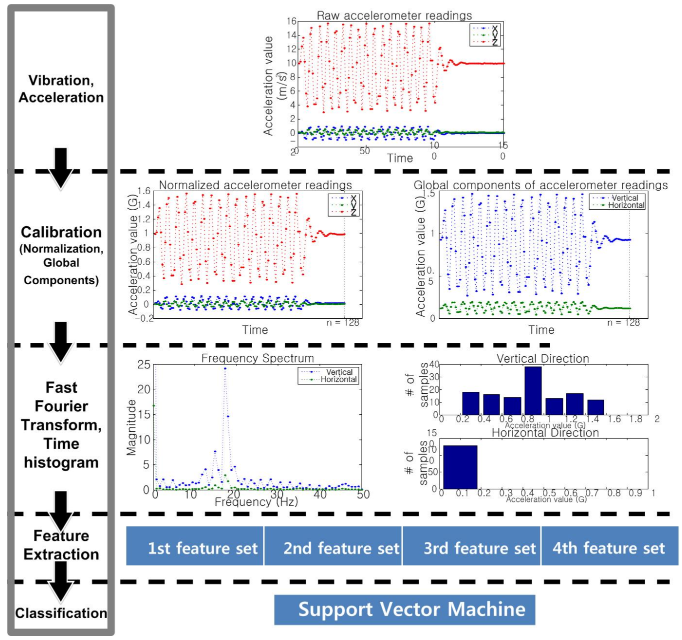


Fig. 2. An overview of *VibePhone*. A smartphone vibrates on a surface for one second and measures accelerations for 1.5 second. An acceleration sample consists of 128 accelerometer readings. These raw accelerometer readings are calibrated to compensate hardware variations. Then features are extracted and the smartphone recognizes the type of surface using a classification algorithm. Four different feature sets have been tested for *VibePhone*.

frame, *VibePhone* extracts features in each direction and performs classification. We have tested four feature sets for *VibePhone* and they are described in Section II-D.

C. Calibration

To compensate the hardware variation of an accelerometer in each smartphone, we need to convert raw accelerometer readings into standard G unit using device-specific parameters. We use a simple normalization scheme proposed in [8] as follows.

Let $\mathbf{a} = [a_x, a_y, a_z]^T$ be a raw accelerometer reading and $\mathbf{g} = [g_x, g_y, g_z]^T$ be the actual acceleration along each axis in G unit. The target function is defined as follows:

$$f(K_x, K_y, K_z, b_x, b_y, b_z) = \sqrt{g_x^2 + g_y^2 + g_z^2},$$
$$g_{axis} = K_{axis} a_{axis} + b_{axis},$$

where K_x, K_y, K_z and b_x, b_y, b_z are the respective scaling factors and offsets of the accelerometer, respectively. When the phone is stationary, the target function is assumed to return one. To solve this parameter estimation problem, we use a least square estimator based on the linear approximation of function f [8]. In our experiment, we collect 2,000 accelerometer readings in $\pm x$, $\pm y$, and $\pm z$ directions when the phone is stationary. The normalization parameters are computed as $\hat{K}_x = 0.0983, \hat{K}_y = 0.1024, \hat{K}_z = 0.1001, \hat{b}_x = 0.0222, \hat{b}_y = -0.0057$, and $\hat{b}_z = -0.0093$.

After raw accelerometer readings are normalized with estimated offsets and scaling factors, we find the global vertical and horizontal directions. The use of global directions is shown to be more robust under different orientations of a smartphone [9]. We call a set of accelerometer readings over a predefined period of time a *frame*, and the global vertical (the direction of gravity) and horizontal (perpendicular to the direction of gravity) components in a *frame* can be obtained as follows.

Let the normalized accelerometer readings in a *frame* be $\tilde{\mathbf{a}}_i = [\tilde{a}_x(i), \tilde{a}_y(i), \tilde{a}_z(i)]^T, i = 1, \ldots, n$, where $\tilde{a}_x(i)$, $\tilde{a}_y(i)$, and $\tilde{a}_z(i)$ are the normalized accelerometer readings in each axis and n is the number of accelerometer readings in a *frame*. We denote its gravity estimation by $\tilde{\mathbf{g}} = [m_{\mathbf{x}}, m_{\mathbf{y}}, m_{\mathbf{z}}]^T$, where $m_{\mathbf{x}}, m_{\mathbf{y}}$, and $m_{\mathbf{z}}$ are the average values in each axis, i.e., $m_{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \tilde{a}_x(i)$. Also, let $\mathbf{v}_i$ and $\mathbf{h}_i$ denote the components in the global vertical and horizontal directions. Then we calculate the length of the $\mathbf{v}_i$ using a dot product, $v_i = \tilde{\mathbf{a}}_i \cdot \tilde{\mathbf{g}}$ with the sign of v_i indicating its direction. Also, we compute the component in the global horizontal direction $\mathbf{h}_i$ using v_i and $\tilde{\mathbf{g}}$, as follows: $\mathbf{v}_i = v_i\tilde{\mathbf{g}}$ and $\mathbf{h}_i = \tilde{\mathbf{a}}_i - \mathbf{v}_i$. Although we know $\mathbf{h}_i$ lies on the horizontal plane which is orthogonal to gravity, $\mathbf{h}_i$ does not mean the absolute direction of $\mathbf{h}_i$ on the horizontal plane. It is impossible to get the absolute direction of $\mathbf{h}_i$ using the accelerometer alone without a compass. Hence, we use its magnitude $\|\mathbf{h}_i\|$ as the measure of horizontal movement in our final coordinate space.

We obtain a set of features, i.e., $\{(v_i, \|\mathbf{h}_i\|), i = 1, \ldots, n\} \in \mathbb{R}^{2\times n}$, by projecting the normalized accelerometer readings onto the global vertical ($\mathbf{v}$) and global horizontal ($\mathbf{h}$) directions. The projected features are then passed to the feature extraction stage.

D. Features

While there is a prior work on recognizing the type of surface contacted by a smartphone [6], [7], the performance by vibration alone has been poor. Hence, we attempt to analyze and improve the performance by using various feature sets. When generating feature sets, we referred to the method described in [8], which shows a high mobility recognition accuracy with 24 features. *VibePhone* is similar to the mobility recognition in that we analyze the accelerometer readings to recognize the contacting surface. Hence, we used this feature set for surface recognition and call this feature set the Jigsaw feature set. We have also developed a time-domain histogram feature set to include additional information in the time domain.

a) Jigsaw feature set: We used the Jigsaw feature set [8] which consists of 24 time-domain and frequency-domain features. The time domain features consist of mean, variance, and mean-crossing rate. (Three features for each global direction. A total of six time domain features.) The mean crossing rate is the number of crossings over the average value.

In the frequency domain, all features are based on spectrum analysis. The frequency domain features consist of a spectrum peak, sub-band energies, sub-band energy ratios, and spectral entropy. (Nine features for each global direction. A total of 18 frequency domain features.) The peak frequency is used to indicate the dominant frequency. The spectrum is evenly divided into four sub-bands, B_1, B_2, B_3, B_4, and the energies in four sub-bands form four additional features. There are also three sub-band energy ratios, $\frac{B_1}{B_2}, \frac{B_3}{B_4}, \frac{B1+B2}{B3+B4}$, and they are used to summarize the energy distribution in low and high frequency ranges. The spectral entropy is a rough description of the frequency distribution.

b) Time histogram feature set: Since the Jigsaw features are focused more on the frequency domain, we have designed a time histogram feature set to capture acceleration information in the time domain. We have analyzed the collected accelerometer readings in the global coordinates and found that the accelerometer readings are distributed within a certain range depending on the surface class. To represent the distribution of accelerometer readings in the time domain, we compute a histogram with a fixed bin size. The number of accelerometer readings for each bin becomes a feature. The bin size needs to be determined carefully. If the size of a bin is too small, the histogram becomes spiky and, if the size is too large, it becomes too smooth. In both cases, the histogram cannot represent the characteristics of the distribution. After a number of trials with different bin sizes, we have found that the following bin size works the best. In the vertical direction, we set the bins of length 0.2 from 0.1 to 1.5 and, in the horizontal direction, we set the bins of length 0.2 from 0.1 to 0.7. Figure 4 shows time histograms of various surface classes, showing that the distributions of accelerometer readings are different depending on the surface classes. In the time domain, the maximum and minimum values in a certain range are generally representative values and so we added them to the time histogram feature set. Time histogram features are extracted for each direction. The performance with time histogram features is discussed in Section III-B.

E. Classifiers

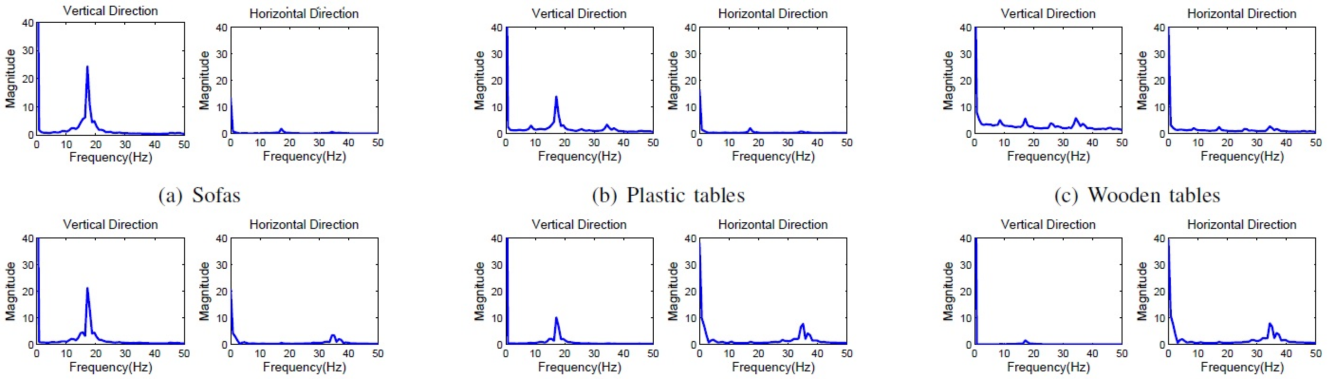
A support vector machine (SVM) is used to classify the surface classes. A SVM is originally designed for a binary classification problem and extended to multi-class classification problems. There are two approach to construct a multi-class SVM using two-class SVMs: one-versus-all and one-versus-one. In this paper, the one-versus-one approach from [10] is used. The SVM algorithm has two parameters: the choice of a kernel function and an upper bound C which is used in optimization. We used the Gaussian radial basis kernel function and set $C = 1000$.

III. Experiments

In this section, we discuss experimental results on surface class recognition using various feature sets. For each surface, we vibrated a smartphone for one second and measured acceleration for 1.5 second with 100Hz sampling rate. An acceleration sample consists of 128 accelerometer readings (a *frame*) as shown Figure 2. For all experiments, we used a Samsung Galaxy S2 Android phone.

A. Acceleration Data Collection

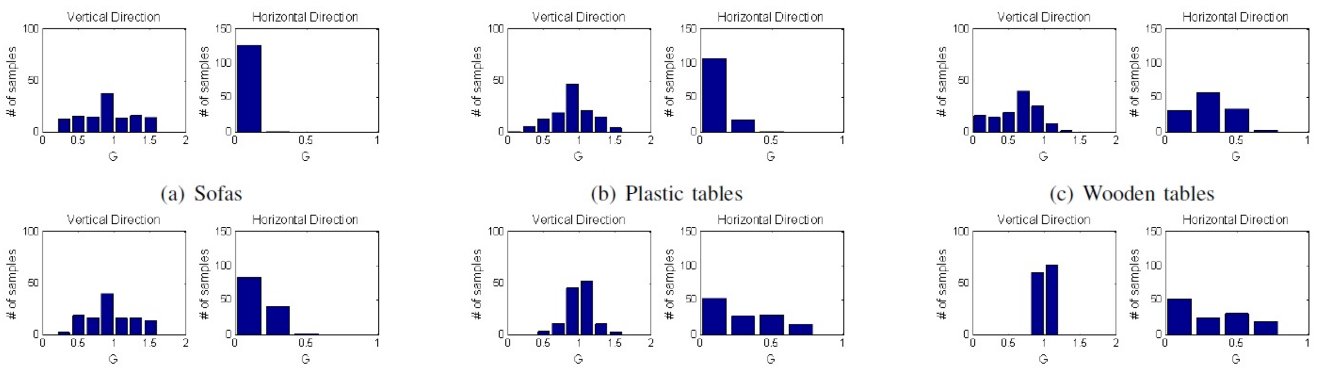
We broadly categorized surfaces into six surface classes where a smartphone is usually placed, i.e., sofas, plastic tables, wooden tables, hands, backpacks, and pants pockets. For each surface class, we collected measurements from ten different



(a) Sofas (b) Plastic tables (c) Wooden tables

(d) Hands (e) Backpacks (f) Pants pockets

Fig. 3. Frequency responses of various surface classes. The plots show the means of frequency distributions. Notice the differences in distribution between different surface classes.



(a) Sofas (b) Plastic tables (c) Wooden tables

(d) Hands (e) Backpacks (f) Pants pockets

Fig. 4. Time histograms of various surface classes. The plots show the mean histograms. There are differences depending on the surface class and we can use histograms as features to distinguish surface classes.

objects so that the collected acceleration samples can well represent the characteristics of the surface class. Figure 1 shows examples of surface classes. As the surface of an object may give different accelerations depending on the location where a smartphone is placed, we collected acceleration samples as follows:

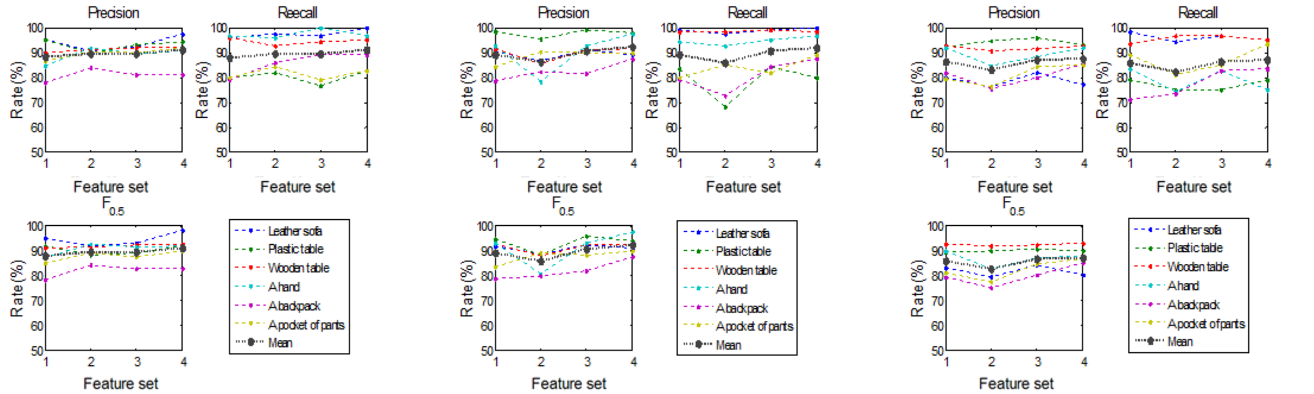
1) On flat surfaced objects, such as sofas, plastic tables, and wooden tables, we randomly selected four different locations.
2) For hands, we chose four different positions, i.e., left and right hand, and both hands with two positions (vertical and horizontal).
3) Because pants pockets have two different positions (right and left pockets), we measured accelerations two times for each pocket.
4) For all cases, we measured acceleration six times for each position.

The total number of acceleration samples we collected are 1,440 (6 types $\times$ 10 objects $\times$ 4 positions $\times$ 6 times).

B. Classification Results

We preprocessed the collected acceleration samples as explained Section II and split them into ten subsets. Each subset contains samples from one object for each surface class, so the number of acceleration samples in each subset is 144 (6 types $\times$ 1 objects $\times$ 4 positions $\times$ 6 times). We randomly selected five subsets as the training set and the rest five subsets are used as a test set to measure the performance. For classification, we tested four different feature sets and they are (1) a Jigsaw feature set, (2) a time histogram feature set, (3) a combination of a Jigsaw feature set and time histogram feature set, and (4) a combination of a Jigsaw feature set, a time histogram feature set, and coefficient magnitudes from the fast Fourier transform (FFT). We index each feature set by a number from 1 through 4 for easy identification and used a support vector machine (SVM) as a classifier. For performance comparison, we used precision, recall, and $F_{0.5}$ measures. In pattern recognition and information retrieval, precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. A measure that combines precision and recall is the F-measure which is a harmonic mean of precision and recall. The F-measure has a parameter β to adjust the effect between precision and recall. We set a parameter β to 0.5 and the $F_{0.5}$ measure puts more emphasis on precision than recall.

We conducted three experiments by randomly selecting five



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

Fig. 5. Results from three experiments. The numbers on the x-axis denote features sets (1 for Jigsaw feature set, 2 for time histogram feature set, 3 for a combination of 1 and 2, and 4 for 3 with FFT coefficient magnitudes.

subsets which are used as a training data set and the remaining subsets are used as a test set to verify consistency of the algorithm. Figure 5 shows precision, recall and $F_{0.5}$ measures according to different feature sets. As shown in Figure 5, the average precision is over 85%, regardless of feature sets we used. Although the time histogram feature set alone shows the lowest performance among four features, it is useful for low sampling rates as discussed in Section III-C.

C. *Effects of Sampling Rates*

Up to this point, we demonstrated that it is possible to recognize the surface class by analyzing accelerations obtained from a smartphone. In this section, we examine the effect of the sampling rate of an accelerometer. The sampling rate of the accelerometer in Samsung Galaxy S2 can be adjusted and possible sampling rates are 20Hz, 50Hz, and 100Hz. With 50 Hz and 20 Hz sampling rate, a frame has 64 and 32 accelerometer readings, respectively. We evaluated the classification performance at 20Hz, 50Hz, and 100Hz. Figure 6 shows that the performance is worse when the sampling rate is 20Hz. It can be explained from the Nyquist sampling theorem, which states that a signal must be sampled at least twice as the bandwidth of the signal to accurately reconstruct the waveform. Therefore, the accelerometer with 50Hz or 20Hz sampling rate can capture acceleration information up to 25Hz or 10Hz, respectively. As can be seen from Figure 3, there is useful information in ranges 10 $\sim$ 20Hz and 30 $\sim$ 40Hz. Hence, the desirable sample rate for classifying a surface class is larger than 80Hz.

However, another interesting observation can be made from Figure 6. While the results from Experiment 1 shows a little difference among different feature sets, for Experiment 2 and 3, the feature set 2 gives the best performance at the sample rate of 20Hz. It shows that time histograms are useful for low sampling rates.

IV. Implementation

Based on the *VibePhone* algorithm described in Section II, we developed an Android app. The *VibePhone* app has two

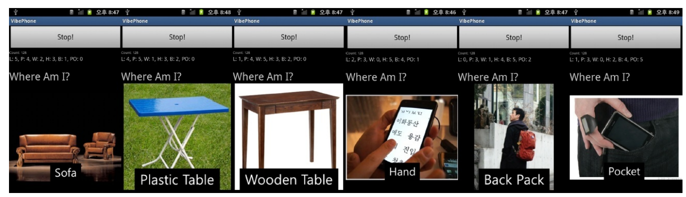
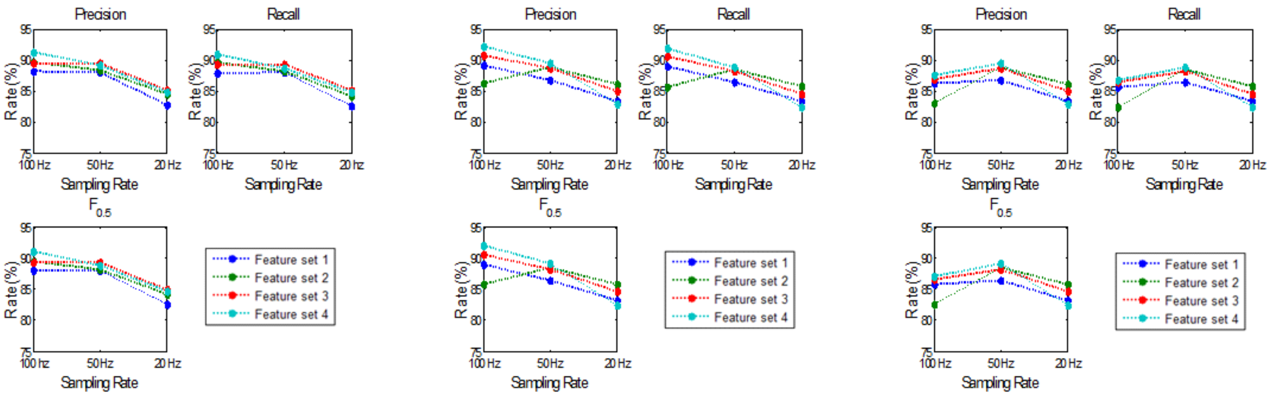


Fig. 7. Six possible output displays from the *VibePhone* app for different surface classes. When there is an incoming call, the *VibePhone* app gets activated and displays the recognized surface class.

modes of operations. It can be run as a stand-alone application to recognize the contacting surface class or it can become active when there is an incoming call and changes its display according to the contacting surface class. The latter approach is ideal for recognizing surface classes as it opportunistically collects surface information from the vibration of an incoming call and saves battery life. When the *VibePhone* app gets activated, it collects the accelerometer data for about 1.3 seconds to collect 128 samples at the sampling rate of 100Hz. It then processes the raw accelerometer readings and extracts features. Then the trained SVM classifier is used to determine the class of surface. Once the surface class is determined, it displays the classification result using a photo of the detected surface class as shown in Figure 7. In our implementation, the Jigsaw feature set is used for its compactness.

V. Evaluation of the *VibePhone* App

We tested the *VibePhone* app on 71 different objects, which consist of sofas (13), plastic tables (13), wooden tables (14), hands (11), bags (10), pants pockets (10). (The numbers in parenthesis are the number of instances of each surface class collected for this experiment.) The average accuracy of the *VibePhone* app is 82% and the result is shown in Figure 8. A surface class which shows the best performance is wooden tables. An object which we have tested under the category of wooden tables usually has an extremely hard surface and it shows distinct characteristics as discussed earlier. The *VibePhone* app recognizes a plastic surface well for most of the



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

Fig. 6. Results of three experiments at different sampling rates. The performance degrades as the sampling rate decreases. The result from Experiment 2 and 3 show that the time histogram feature set works well under low sampling rates.

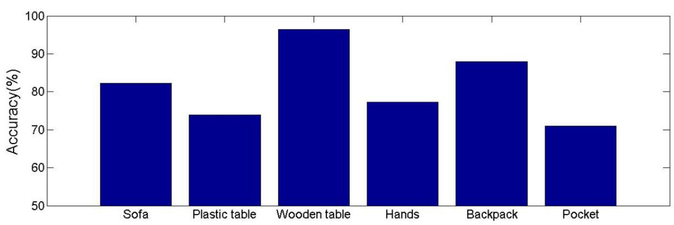


Fig. 8. Surface recognition results. Note that the accuracy axis does not start at 0.

time but sometimes confuses it with a wooden table or sofa. The tested plastic objects also have hard surfaces but not as stiff as wooden tables. In a sense, objects in the class of plastic tables have a characteristic which is between a wooden table and a sofa. As expected, the surface class of sofas includes many of soft surfaces. It was difficult to distinguish hands from backpacks. We believe that the confusion is due the tilt of the smartphone when the training set was collected. The algorithm sometimes confuses a hand with a plastic table or sofa and we find that this is due to the holding patterns used in the experiment, which was different from training. When we hold a phone tightly or when a phone is placed on an open palm, we lose charateristics of hands found in the training set.

The procedure that *VibePhone* recognizes the surface type is similar to the tapping behavior. Giving a vibration to the surface of a specific object and recognizing the surface with the accelerometer data is similar to how a human recognizes an object by tapping on the surface with both eyes closed. Even though *VibePhone* currently provides a recognition into one of six surface classes, we find that the results are promising.

VI. Conclusions

This paper introduced and analyzed a method to recognize the type of surfaces contactecd by a smartphone. The proposed *VibePhone* uses a built-in vibrator and accelerometer of a smartphone, for recognizing the type of contacting surfaces, enabling the sense of touch to smartphones. To improve the classification accuracy, we have developed and tested a number of different features that are constructed from accelerometer readings. The experimental results show that *VibePhone* can recognize the correct surface class with an accuracy of 85%, demonstrating the possibility of the haptic perception in future smartphones.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0027155).

References

[1] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. on Information Systems (TOIS)*, 10:91–102, January 1992.

[2] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. Intl. Conf. on Mobile Computing and Networking (MobiCom) (2004)*, pages 70–84, 2004.

[3] J. Hightower, S. Consolvo, A. Lamarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. In *Proc. Intl. Conf. on Ubiquitous Computing (UbiComp) (2008)*, page 903, 2005.

[4] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: mobile phone localization via ambience fingerprinting. In *Proc. Intl. Conf. on Mobile Computing and Networking (MobiCom) (2009)*, pages 261–272, 2009.

[5] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *Proc. Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys) (2009)*, pages 165–178, 2009.

[6] I. Shafer. Learning location from vibration. In *http://www.mrcaps.com/proj/LocationVibration/files/vibration-ishafer-paper.pdf*, 2007.

[7] K. Kunze and P. Lukowicz. Symbolic object localization through active sampling of acceleration and sound signatures. In *Proc. Intl. Conf. on Ubiquitous Computing*, pages 163–180, 2007.

[8] 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 *Proc. Intl. Conf. on Embedded Networked Sersor Systems (SenSys) (2010)*, pages 71–84, 2010.

[9] D. Mizell. Using gravity to estimate accelerometer orientation. In *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 252 – 253, 2003.

[10] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox, 2005.