AccelWord: Energy Efficient Hotword Detection through Accelerometer

Li Zhang, Parth H. Pathak, Muchen Wu, Yixin Zhao and Prasant Mohapatra
Computer Science Department, University of California, Davis, CA, 95616, USA
Email: {jxzhang, phpathak, muwu, yxzhao, pmohapatra}@ucdavis.edu

ABSTRACT

Voice control has emerged as a popular method for interacting with smart-devices such as smartphones, smartwatches etc. Popular voice control applications like Siri and Google Now are already used by a large number of smartphone and tablet users. A major challenge in designing a voice control application is that it requires continuous monitoring of user's voice input through the microphone. Such applications utilize *hotwords* such as "Okay Google" or "Hi Galaxy" allowing them to distinguish user's voice command and her other conversations. A voice control application has to continuously listen for hotwords which significantly increases the energy consumption of the smart-devices.

To address this energy efficiency problem of voice control, we present AccelWord in this paper. AccelWord is based on the empirical evidence that accelerometer sensors found in today's mobile devices are sensitive to user's voice. We also demonstrate that the effect of user's voice on accelerometer data is rich enough so that it can be used to detect the hotwords spoken by the user. To achieve the goal of low energy cost but high detection accuracy, we combat multiple challenges, e.g. how to extract unique signatures of user's speaking hotwords only from accelerometer data and how to reduce the interference caused by user's mobility.

We finally implement AccelWord as a standalone application running on Android devices. Comprehensive tests show AccelWord has hotword detection accuracy of 85% in static scenarios and 80% in mobile scenarios. Compared to the microphone based hotword detection applications such as Google Now and Samsung S Voice, AccelWord is 2 times more energy efficient while achieving the accuracy of 98% and 92% in static and mobile scenarios respectively.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Voice I/O; I.5.4 [**Pattern recognition**]: Applications—*Signal processing*

MobiSys'15, May 18–22, 2015, Florence, Italy.

http://dx.doi.org/10.1145/2742647.2742658.

General Terms

Mobile, System, Energy, Efficiency

Keywords

AccelWord, hotword detection, accelerometer, energy, measurement

1. INTRODUCTION

With remarkable advancement in smartphone technology and increasing popularity of upcoming wearable devices, voice control is emerging as an attractive method of interaction with smart-devices. Voice control applications like Siri [1] on iOS devices and Google Now [2] on Android devices are already used by many smartphone and tablet users. Voice control is especially an attractive choice for wearable devices like smartglass and smartwatch. Such devices have a very small touch-enabled screen which restricts the applicability of touch-based control beyond a few primitive touch gestures. For this reason, voice control is commonly used in current commercial smartwatches [3] and smartglasses [4]. It also holds tremendous potential as objects surrounding us (in homes, offices and elsewhere) become more and more intelligent, and can provide various capabilities like electronic assistance. Such devices are already becoming commercially available (e.g. voice controlled intelligent speaker [5] that also acts as electronic assistant).

Although voice control enables an intuitive way for users to interact, one major challenge is that it requires continuous sensing of audio signals. This means that a device should turn on the microphone to continuously monitor user's voice commands. This results in significant energy consumption which is a major challenge for battery-powered mobile devices such as smartphones, smartwatches and smartglasses. Voice controlled devices implement *hotwords* (e.g. "Okay Google", "Hi Galaxy") in order to distinguish between user's voice commands to the device and her other conversations. This requires the device to continuously perform hotword detection by recording audio through microphone and checking whether the spoken words are the hotwords. Reducing the energy consumption of the hotword detection is an extremely challenging problem. To reduce the energy consumption, some devices utilize other low power sensors like accelerometer. Here, voice control applications monitor certain movements or gestures performed by the user (like double tap on screen [3] or tilting head up [4]) before turning on the microphone to listen for voice commands. However, such solutions are often not user-friendly (only work when user can

touch/wear the device) and require user to get accustomed to different wake-up patterns for different devices. In some latest smartphones (e.g. Nexus 6 [6]), a dedicated low-power processor is used for audio sensing. However, this incurs additional cost which is not suitable for low-cost devices for pervasive Internet-Of-Things (IoT) applications. Moreover, there are a number of new smart devices (such as fitness bands and smartwatches) that do not have a microphone embedded in them. Enabling voice commands on such devices still remains a difficult challenge to solve.

In this paper, we propose AccelWord - an energy efficient solution for hotword detection using the accelerometer sensor. AccelWord is based on the observation that the MEMS (MicroElectroMechanic System) accelerometer sensors available in smartphones, smartwatches and nearly all smart devices are sensitive to user's spoken voice. When the user speaks, the generated audio signal causes variations in the observed acceleration in the accelerometer sensor. In fact, we show that these variations represent user's spoken words surprisingly well, and it is possible to extract unique signatures of user's speaking the hotwords simply from accelerometer data. Based on this, we build the AccelWord system which performs the hotword detection purely using the acceleromter data and turns on the microphone once the accelerometer data matches the extracted signature of the hotword. We show that AccelWord has the hotword detection accuracy of 85% in static scenarios with less than 5% of false positive rate. Compared to the microphone-based hotword detection, AccelWord is 2 times more energy efficient while achieving the accuracy of 98%. Since low-power low-cost accelerometer sensor is available in majority of the devices for motion recognition, we think AccelWord will enable accurate yet low-energy and low-cost implementation of voice control.

In recent research such as [7], [8], it has been observed that MEMS accelerometer/gyroscope sensors are sensitive to user's speech and nearby keystrokes, posing severe privacy risks of information leakage. However, in this paper, we are primarily concerned with how this sensitivity can be exploited for energy-efficient hotword detection. AccelWord addresses multiple challenges towards creating an accurate and energy-efficient hotword detection. First, since the impact on accelerometer due to user's voice can be considered as user's voice signals modulated at a lower frequency (200 Hz in case of current accelerometers), it is not clear which features can be used to extract hotword signatures. For higher energy efficiency, it is essential that the computational cost of calculating features is not very high. To address this challenge, AccelWord utilizes low complexity features that are often used in activity recognition (e.g. walking, running etc.) through accelerometer. Our study reveals that these features can accurately distinguish hotwords from other spoken words of the user.

The other important challenge in using accelerometer for hotword detection is to separate the accelerometer variations due to user's movement from that due to user's voice. This is especially important because mobile devices like smartphones and smartwatches consistently move when carried or worn by the user. In such cases, the accuracy of hotword detection should be still high even in the presence of mobility. By applying a suitable high-pass filter on the accelerometer data, AccelWord can achieve a similar level (94.5%) of accuracy as in static cases.

The contribution of this paper breaks down into the following aspects:

- We provide measurement-based evidence that accelerometers used in today's mobile devices are sensitive to user's voice. It is also demonstrated that the variations in accelerometer data when user speaks different words are sufficiently different which allows us to extract unique signatures of hotwords.

- We design and implement AccelWord framework which detects user's speaking of hotword purely by monitoring the accelerometer sensor data. It utilizes statistical pattern and frequency analysis to create signatures of the hotwords using the accelerometer readings. The extracted signatures are then used to train a classifier that can detect the hotword in real-time. We show that AccelWord can perform accurate hotword detection even in the presence of user mobility and high audio noise.

- We implement AccelWord on Android smartphone and evaluate it using experiments with 10 users. It is shown that AccelWord can detect the hotword with an average accuracy of 85% in static scenarios and average false-positive rate of 4.7%. When the user is mobile, the accuracy and false positive rate are observed to be 80% and 5.6% respectively. Compared to microphone-based hotword detection applications - Google Now and Samsung S Voice - AccelWord can achieve 98%, 92% and 93% of accuracy in static, mobile and noisy scenarios respectively.

- We show that AccelWord performs accurate hotword detection while consuming comparatively very low energy. Measurement results on two different phones show that AccelWord consumes 50% and 57% less power than Google Now and Samsung S Voice respectively.

The rest of the paper is organized as follows. We give a brief overview of AccelWord in Section 2. The feasibility of AccelWord is verified in Section 3. In Section 4, we present how the voice signature is extracted and how the training is performed. The implementation and the performance evaluation of AccelWord are presented in Section 5 and Section 6. We discuss the future explorations and the related work in Section 7 and Section 8 respectively. Section 9 concludes the paper.

2. OVERVIEW OF ACCELWORD

2.1 Motivation: Energy Expensive Voice Control

In this section, we first take a look at how current voice control applications operate and their energy efficiency. Most current voice control applications use "hotwords" detection to enable complete speech recognition. This is shown in Fig. 1. When a voice control application is running, it constantly listens for the hotwords spoken by the user. Examples of such hotwords include "Okay Google" or "Hi Galaxy" for Google Now [2] and Samsung S Voice [9] applications respectively. When the hotwords are detected, any following spoken words by the user are recognized using speech recognition. The purpose of using hotword detection instead of

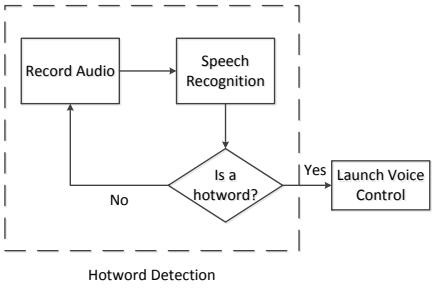


Figure 1: Flow chart of microphone based hotword detection

continuously recognizing every word user speaks is that it is more computationally efficient. This is because hotword detection merely classifies the spoken words into two classes - the hotwords and the other words - with light-weight speech signature matching.

Although hotword detection requires lesser computation than complete speech recognition, both of them require the device microphone to be on all the time. Constantly listening on the microphone makes the current voice control applications very energy inefficient. To demonstrate this, we measure and compare the power consumption of 2 voice control applications - Google Now and Samsung S Voice. We use the Monsoon Power Monitor [10] to record power consumption on two smartphones - Samsung Galaxy S4 and Google Nexus S. For understanding the baseline power consumption, we create an android app (called "Microphone") that simply turns on the microphone but does not perform any speech recognition. The example traces of power consumption for all three apps are presented in Fig. 2. In order to isolate the power consumption of the apps, we disable all network interfaces using airplane mode (except for Samsung S Voice which requires active Internet connection to operate) and restrict the number of background processes to 0. After ensuring that only the desired app is running, we measure the power consumption of app's Graphical User Interface (GUI) before starting the hotword detection. This power consumption is deducted from the total power consumption of the app when it is running to obtain the power consumption of listening, hotword detection and speech recognition. The average values of 30 minutes are reported in Fig. 3. Since Samsung S Voice is exclusive for Samsung phones and is not available in Android app store, the power consumption of S Voice on Nexus is not applicable.

It is observed from Fig. 3 that the power consumption of the 2 voice control apps is higher than the Microphone app due to their additional computational requirement of hotword detection and speech recognition. Depending on the hotword detection and speech recognition algorithms, the power consumption increases slightly when the user is speaking. However, in any case, the major factor on average power consumption in all the apps is when the app is listening for the hotword. Because such apps are designed to listen for user's commands at all times, keeping the microphone on and detecting hotword consumes substantial energy.

Continuous listening using the microphone and hotword detection in current voice control apps are energy inefficient. This motivates us to investigate an alternative way of continuous voice sensing that is both accurate and energy efficient.

2.2 Design Goals and Challenges

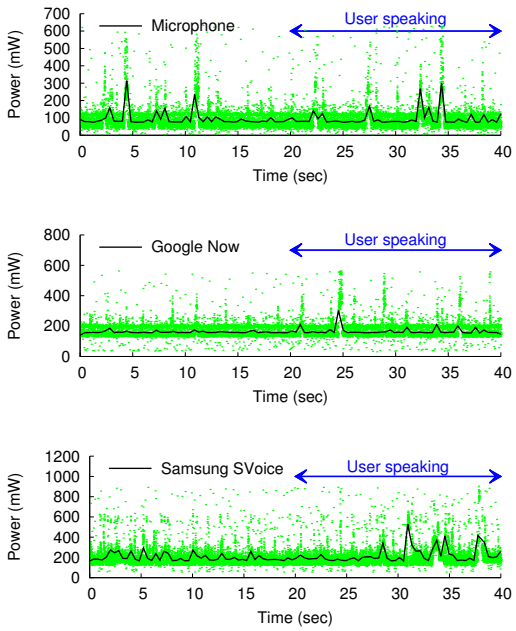


Figure 2: Example: the power trace of three apps

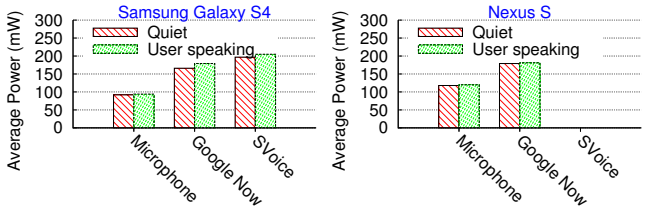


Figure 3: The Power Consumption of Current Hotwords Detection Apps.

A hotword detection scheme should meet the following design goals in order to be truly pervasive.

Accuracy: We define the accuracy of a hotword detection scheme to be the ratio of the number of times user spoken hotwords are correctly detected to the total number of times the user speaks the hotword. Accurately detecting the hotword is essential to any voice control application. Even though recent voice control applications such as Google Now have shown to achieve high accuracy in hotword detection, frequent failures to detect the hotword is one of the dominant factors preventing pervasive use of voice control in smartphones and wearable devices. Note that the other dominant factor in slow adaptation of voice control application is inaccuracy in speech recognition after the hotword is detected. However, since there is plethora of research [11–15] already done on this topic, we do not consider complete speech recognition in this work and simply focus on the hotword detection.

Robustness: Another important design goal is that a voice control application should be robust to its dynamic operating environment. This means that it should be robust in hotword detection in the following three scenarios:

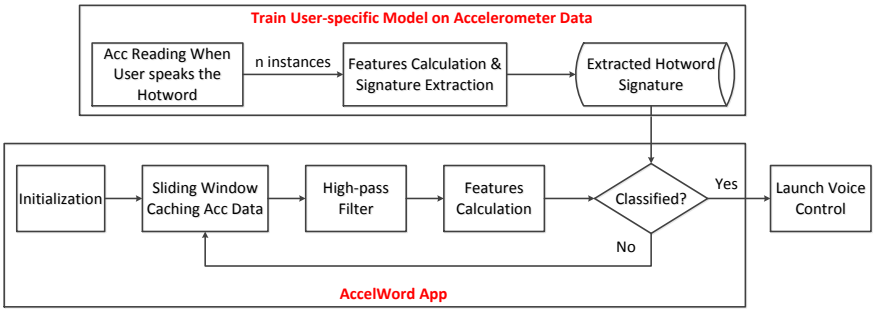


Figure 4: The System Architecture of AccelWord

(1) User mobility: It is necessary that the hotword detection accuracy is high even when the device is in constant motion. For example it is necessary that a smartwatch detects user's hotword even when the user is walking.

(2) Different voice frequency (female or male): It is essential that the voice control application detects the hotwords for both female and male users. Because female voice exhibits higher frequency [16] than the male voice, accuracy should be least affected by the input voice frequency.

(3) Noisy surroundings: The noise of the surrounding environment can affect the voice input recognition especially when the user is in noisy outdoor places such as malls, cafes, etc. The hotword detection accuracy should not be affected by the surrounding noise.

Energy Efficiency: As we showed in the previous section, the current voice control applications are expensive in terms of their energy consumption. For ubiquitous deployment of voice control in all battery-operated smart devices, it is necessary that it operates with a smaller energy footprint. This requires that both - sensing of voice input and hotword detection using signature matching - are energy efficient.

2.3 System Architecture

To this end, we design and implement *AccelWord* which achieves high accuracy and energy efficient hotword detection. AccelWord utilizes accelerometer instead of microphone to listen the sound signal of the input voice. Specific signatures are then extracted from the accelerometer data and inserted into the AccelWord app for hotword detection. Fig. 4 illustrates the architecture of the system.

- **Hotword signature extraction:** Due to the low power consumption property of accelerometer, we try to extract the signatures of hotwords from the accelerometer readings instead of microphone samples. The signature is constructed by comparing the set of accelerometer readings of hearing of hotwords and the set of accelerometer readings of hearing other random sentences. For energy efficiency purpose, the training is done offline.

- **AccelWord app:** AccelWord is a standalone app running on Android devices. During the initialization stage, AccelWord will load the extracted signature of the hotword. AccelWord dynamically buffers a certain number of accelerometer samples and periodically calculates the features of the samples. The calculated features are compared with the extracted signature loaded in the initialization stage. If a hotword is detected, AccelWord will send an intent to the Android OS to launch the voice control, otherwise the process will be repeated.

3. FEASIBILITY OF ACCELWORD

3.1 Accelerometer Design

Current accelerometer sensors found in smartphones and other smart devices like smartwatches and smartglasses are Micro Electro Mechanical Systems (MEMS). Such MEMS accelerometers have three main components - an inertial mass, spring legs and stationary fingers. This is shown in Fig. 5. The inertial mass is anchored to the substrate using two pairs of flexible spring legs. When an acceleration is applied, the inertial mass moves which causes a change in the capacitance between the stationary fingers. This change is recorded to accurately measure the acceleration. In a 3-axis accelerometer, 3 separate sets of components are employed to measure the accelerations separately.

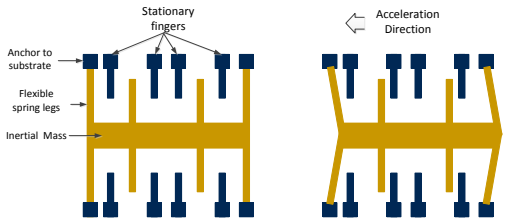


Figure 5: A sketch of a MEMS accelerometer

3.2 Impact of Voice Signal on Accelerometer

When a user speaks, the resultant acoustic signals strike the inertial mass of the accelerometer, causing it to move and report very small changes in acceleration. From the perspective of the accelerometer, such variations are considered undesirable noise, and [17–19] have studied its effects and proposed ways to combat the noise. Depending on the sampling frequency of the accelerometer, the acceleration changes can reflect a part of the characteristics of the user's voice and the spoken words. The typical maximum sampling frequency of today's MEMS accelerometers is in the range of a few thousand Hz. For example, Invensense MPU-65xx accelerometer found in Apple iPhone 6, Google Nexus 5 and Samsung Galaxy S5 has the highest sampling frequency of 4000 Hz (referred as "output data rate" in [20]). However,

our experiments with Android 4.4 OS shows that the operating system restricts the maximum sampling frequency of an accelerometer to 200 Hz in order to reduce power consumption (similar restriction was also observed for gyroscope [7]). This sampling frequency has important implications on how voice signal affects the accelerometer readings.

A human ear can perceive any sound that is within the range of 20 Hz to 20 KHz [21]. This is why a typical microphone has a sampling frequency over 40 KHz since Nyquist sampling theorem states that the sampling frequency should be at least twice ($\geq$ 40 Hz) the highest frequency in the signal (20 Hz) for reconstruction. This implies that with 200 Hz of sampling frequency of the accelerometer, we can not perfectly reconstruct the sound. In this work, we are not interested in the complete reconstruction of the voice using accelerometer. Such reconstruction requires a very high sampling rate which can result in very high energy cost. Instead, we are interested in generating signatures of different hotwords spoken by the user through the analysis of accelerometer readings available at a lower sampling frequency.

AccelWord is feasible because of the fact that typical fundamental frequency of a male's speaking voice is between 85 Hz and 155 Hz, and female's speaking voice is between 165 Hz and 255 Hz [22]. This means that accelerometer data even at the sampling frequency of 200 Hz, can reflect some parts of human voice. We first demonstrate using an experiment that human voice has a measurable effect on accelerometer data even when sampled at 200 Hz.

Experiment Setup: To validate the impact of voice on smartphone's accelerometer, we use the experiment setup as shown in Fig. 6. The goal of the setup is to emulate a scenario where a user is speaking to her smartphone in her hand or on a desk, or to a smartwatch on her wrist. For repeatability, user's voice is recorded by a professional sound recording software (Audacity) at sampling frequency of 384000 Hz and played on a phone (iPhone 4S) repeatedly as needed. Another smartphone (Samsung Galaxy S4 running Android 4.4.2) acts as a receiver of the voice. The receiver phone collects the accelerometer data at the highest sampling rate (measured to be 199 Hz). The speaker and receiver phones are fixed at a distance of 12 inch (typical distance between user's mouth and her phone or watch). To avoid any effects of direct surface vibrations, we place both the phones on separate desks that are not in contact with each other. This first set of experiments were carried out in a silent room inside a university building. To avoid the acoustic interference from human presence, we remotely control the speaker iPhone wirelessly from a different room using a MacBook Air.

The speaker phone's output volume is varied to generate different Sound Pressure Levels (SPL) at the receiver. The SPL is measured using an Android app (Sound Meter [23]) on the receiver phone (Samsung Galaxy S4). Table 1 show the measured SPL at the receiver and example scenarios where the SPL is observed [16, 24].

Impact on Accelerometer: Fig. 7 shows the variation of accelerometer reading when the speaker is playing vowel "A" spoken by two of the authors. The spectrum analysis of the two users and the background noise are shown in Fig. 7a. The average SPL of the background noise measured on the receiver is 25 dB. The receiver's accelerometer readings under different SPLs are shown in Fig. 7b and Fig. 7c. Since the voice comes from right above side of the receiver,

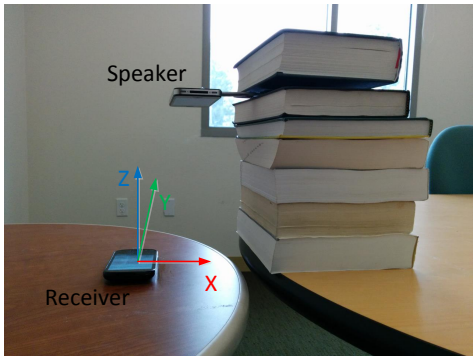


Figure 6: Experiment Setup

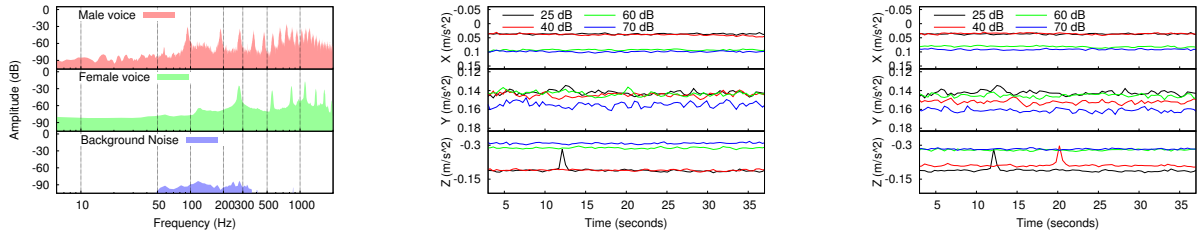
Measured SPL (dB)	Typical Scenario [16, 24]
70	Human to phone conversation. (distance: 12 inch)
60	Human to human conversation. (distance: 1 meter)
50	Gentle keystroke.
40	Quiet university libraries.
30	Quiet bedroom at night.
20	Calm breathing.

Table 1: Example Scenarios of SPL Levels

the accelerometer reading on the Y axis does not vary much ($< 0.02 m/s^2$). However, on the X axis and Z axis, we can observe considerable amount ($0.06 m/s^2$ - $0.15 m/s^2$) of difference on the accelerometer reading when the male SPL is increased from 25dB to 70dB. The similar phenomenon is also observed on the female voice input. Although the variations on X axis and Z axis caused by the female voice is slightly lesser than the male voice, they are still significantly higher than the variation on the Y axis. This indicates that the human voice at high enough SPLs will have a detectable amount of impact on the smartphone accelerometers.

3.3 Accelerometer vs. Gyroscope - Energy Comparison

Accelerometer is sensitive to acoustic signals mainly because it is an MEMS sensor. Another MEMS sensor - gyroscope - is also widely used in smartphones and other smart devices. The gyro sensor is also shown to be affected by the voice signals in [7]. Since our objective is to use the acoustic sensitivity of accelerometer to perform energy-efficient hotword detection and not to reconstruct the complete sound, it is necessary to compare the energy efficiency of accelerometer and gyroscope. Due to the design differences in MEMS, it is known that gyroscope sensors consume more energy than the accelerometer sensors even at the same sampling frequency [17, 20]. Comparing the specifications of accelerometer and gyroscope sensors used in all major smartphones, it is found that normal operating current of only operating gyroscope is on an average 6 times higher than the that of operating only accelerometer [17, 20]. However, the actual power consumption when collecting these sensors' data depends on many other factors such as data collection application, OS, other hardware components like CPU and memory. We measure this total power consumption on Nexus S and Samsung Galaxy S4. Here the sensor data is collected by our Android app at 200 Hz, and the power is measured



(a) The Normalized FFT of The Input Voice (b) Accelerometer Readings When User Speaks Vowel "A" (Male) (c) Accelerometer Readings When User Speaks Vowel "A" (Female)

Figure 7: The Impact of Speaking Vowel "A" on Accelerometer

using the Monsoon power monitor. We use the exact same implementation for collecting the data from both sensors in our app. The power consumption results are shown in Fig. 8. It is observed that collecting gyroscope data consumes 55.8% more power than the accelerometer, and as expected, both acclerometer and gyroscope consume significantly lower energy compared to the microphone (as shown in Fig. 8d).

Based on the observations, it can be concluded that (i) accelerometer is sensitive to the human voice, and (ii) it is also energy efficient. Therefore, we make use of the accelerometer sensor to implement AccelWord, an app using accelerometer to detect specific voice signals (hotwords).

4. HOTWORD DETECTION USING ACCELWORD

From the previous section, we know that accelerometer sensor is affected by user's voice. In this section, we demonstrate that the effect on the accelerometer data due to the user's voice is rich enough so that it can also be used to detect the hotwords spoken by the user. For this, we first show what features of accelerometer data can be used to create signatures of the hotword. Based on the signature, we build a machine learning classifier that performs the hotword detection.

While creating the signature of hotwords using the accelerometer data, we focus on two goals:

(1) We are only interested in *distinguishing* the hotword from other spoken words of users. This way, our hotword detection is a binary classification problem in terms of machine learning and not a speech recognition problem where all spoken words are reconstructed. Once the hotword is detected, the microphone can be turned on to record user's voice and existing methods of speech recognition can be applied.

(2) Such hotword detection should be online and energy efficient. This means that the process of accelerometer data collection, analysis and matching with hotword signatures should be computationally efficient in order for the hotword detection to be energy efficient. We already know from Fig. 8 that accelerometer data collection consumes less power than recording via microphone. However, it is necessary to design efficient ways of analyzing and matching the accelerometer data.

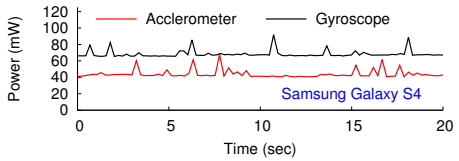
One of the most difficult challenges in accurate hotword detection is that user's mobility causes significant changes in accelerometer data. It is necessary that the hotwords are detected even when user is mobile. For this, we need to filter the mobility *interference* from the accelerometer signals to distill the effect of user's voice before performing the hotword classification. We first show how to extract hotword signature from the accelerometer data in a stationary case and then extend our analysis to user's mobility.

4.1 Extracting Hotword Signature

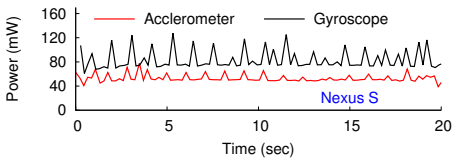
One possible approach of identifying hotword is to upsample the accelerometer data collected at 200 Hz to 40 KHz, and then reproduce some parts of user's spoken words from the resultant audio file. However, this can incur huge energy cost due to the computational complexity of upsampling as well as analyzing the additionally generated data. Also, since we are not interested in reproducing the voice, such additional processing is unsuitable for our application. Instead, we take a different approach in analyzing the accelerometer data as described next.

Candidate Features: We propose to use activity recognition related features to analyze the accelerometer data. Table 2 lists a set of features that are found to be highly correlated [25] to physical activity of humans such as walking, running, sitting, standing, etc. The main advantage of using these features over the audio analysis related features (used in speech recognition [26, 27]) is their lower computational complexity. Majority of features in Table 2 are time series analysis of data which can be efficiently calculated for fast online processing.

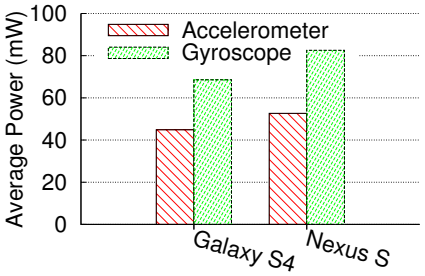
Feature Selection: Because the candidate set of features we want to use are primarily studied in terms of activity recognition, it is not clear how well they can be used for hotword detection. To evaluate their usefulness, we calculate the values of the features when user speaks the hotword and other sentences or randomly chosen text. We use the experiment setup discussed in Section 3. Two separate recordings of user's spoken words are played through the speaker phone at 100% volume level (70 dB SPL at the receiver phone). In the first recording, the user speaks the hotword "Okay Google" once which is repeated 200 times. In the second recording, the user speaks commonly used sentences ("Good morning", "How are you", "Fine, thank you" etc.) which are then repeated 400 times in random order. After playing the recordings through the speaker phone, the accelerometer data from the receiver device is used to calculate the candidate set of features. We set the time window for feature calculation to be 2 seconds based on the observation that most user could complete speaking the hotword within that time. Note that an online hotword detection would require considering many practical issues such as using a sliding window for continuous evaluation, and we have addressed these



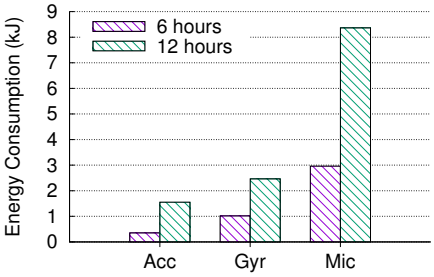
(a) Example: The Power Monitor Trace of Galaxy S4



(b) Example: The Power Monitor Trace of Nexus S



(c) The Average Energy Consumption of a 30 minutes Trace



(d) The Long Term Total Energy Consumption of Three Sensors

Figure 8: The Energy Consumption of Accelerometer, Gyroscope and Microphone

issues in our AccelWord app design in Section 5. Here, we first seek to understand how the presented features can be used to distinguish the hotword from the other words.

To determine how well a given feature can distinguish the hotword from other spoken words, we use Information Gain based feature selection. Information gain [28] is a commonly used feature evaluation method where entropy of classification is compared in the presence and the absence of a given feature. Let G be the set of instances in which H are hotword instances and N are instances of other spoken words. Let $E(G)$ be the entropy of G. If $p(H)$ and $p(N)$ are the fraction of hotword and non-hotword instances then $E(G)$ can be calculated as

$$E(G) = -p_H \cdot \log_2 p_H - p_N \cdot \log_2 p_N \quad (6)$$

Let $I(F)$ be the information gain of the feature F. $I(F)$ can be calculated as

$$I(F) = E(G) - \sum_{f \in V(F)} \frac{|G_f|}{|G|} E(G_f) \quad (7)$$

where $V(F)$ is the set of values the feature F can take and G_f is the subset of G where the feature $F = f$. This way, $I(F)$ can be considered as a measure of additional information available due to the presence of feature F in classifying the hotword and other words. The information gain values are between 0 and 1 where a higher value indicates a feature being more useful in classification.

Fig. 9 shows the information gain of candidate features with respect to two classes - hotword and not hotword. It is observed that most features in the candidate set exhibit high information gain which shows that they can be used for hotword classification. Some features (Kurtosis, Skewness and MCR) have zero information gain which means that they do not have any useful value in classifying the hotword. We use the rest of the features to build the AccelWord classifier.

4.2 Combating Mobility Interference

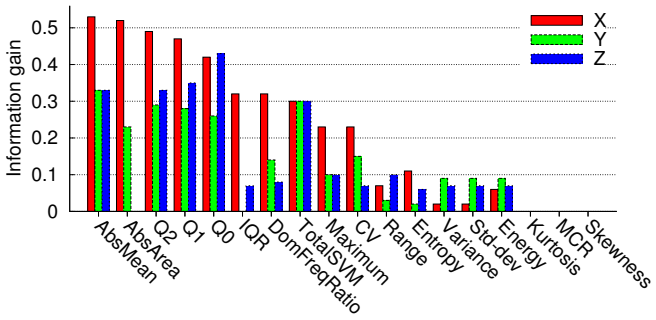


Figure 9: Information gain of candidate features

To combat the noise caused by user's mobility, we first conduct a series of mobility experiments to understand the interference of user's mobility to our problem. Based on the observations and analyzing the numerical results of the mobile scenarios, we are able to design proper techniques to detect hotwords even when the users are moving.

Mobility Experiment Setup: For the mobility experiments, we use the same phones as in the static experiments (Section 3.2). As shown in Fig. 10, the receiver phone is wrapped to the left wrist of the user, while the speaker is held closely to the user's mouth. The volume of the speaker is adjusted to ensure that the SPL at the receiver is 70 dB when the distance is 12 inch. The user walks in approximately 1 m/s speed in a $4m \times 9m$ room along an elliptic trajectory. For repeatability of the experiments, we will only focus on the walking and speaking mobility pattern, since the other mobility patterns, e.g. running and speaking, jumping and speaking, are quite hard for our experimenters and volunteers to repeat for a large number of times. Therefore,

The following features are calculated for accelerometer signal of X, Y and Z axis over time window of t seconds

• **Time domain:**
Calculated separately for each X, Y and Z axis:
- Minimum, maximum, median, variance, standard deviation
- Range: difference between maximum and minimum, measure of extreme changes in acceleration
- Absolute Mean (AbsMean): average of absolute values of acceleration
- CV: ratio of standard deviation and mean times 100; measure of signal dispersion
- Skewness (3rd moment): measure of asymmetry in distribution of signal samples
- Kurtosis (4th moment): measure of peakedness in distribution of signal samples
- Q1, Q2, Q3: first, second and third quartiles; measures the overall distribution of accelerometer magnitude over the window
- Inter Quartile Range (ICR): difference between the Q3 and Q1; also measures the dispersion of the signal
- Mean Crossing Rate (MCR): measures the number of times the signal crosses the mean value; captures how often the signal varies during the time window
- Absolute Area (AbsArea): the area under the absolute values of accelerometer signal. It is the sum of absolute values of accelerometer samples in the window. Let a_{s_i} denote the i^{th} sample of accelerometer's $s \in \{X, Y, Z\}$ axis, then

$$AbsArea_s = \sum_{i=1}^{window\ length} |a_{s_i}| \quad (1)$$

Calculated across X, Y and Z axis:
- TotalAbsArea: sum of AbsArea of all three axis.

$$AbsArea = \sum_{i=1}^{window\ length} |a_{x_i}| + |a_{y_i}| + |a_{z_i}| \quad (2)$$

- TotalSVM: the signal magnitude of all accelerometer signal of three axis averaged over the time window.

$$TotalSVM = \frac{\left[\sum_{i=1}^{window\ length} \sqrt{\sum_{s\in\{X,Y,Z\}} {a_{s_i}}^2}\right]}{window\ length} \quad (3)$$

• **Frequency domain:**
Calculated separately for each X, Y and Z axis:
- Energy: it is a measure of total energy in all frequencies. Let m_i be the magnitude of FFT coefficients.

$$Energy = \sum_{i=1}^{window\ length/2} m_i^2 \quad (4)$$

- Entropy: captures the inpurity in the measured accelerometer data. Let n_i be the normalized value of FFT coefficient magnitude.

$$Entropy = -\sum_{i=1}^{window\ length} n_i\ log_s(n_i) \quad (5)$$

- DomFreqRatio: it is calculated as the ratio of highest magnitude FFT coefficient to sum of magnitude of all FFT coefficients.

Table 2: Candidate features

we will leave the study of other mobility patterns for future exploration.

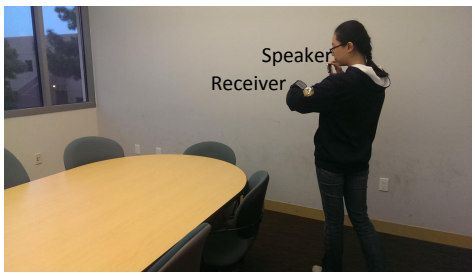


Figure 10: Mobility Experiment Setup

Impact of Mobility Interference: The results of mobile experiments are shown in Figs. 11 and 12. Fig.11 shows an example 1 second window of the accelerometer readings when the user is walking and speaking. Comparing Fig. 11 with Fig. 7, we can observe that the readings of the accelerometer in mobile scenarios are at least one order of magnitude higher than the readings in static scenarios, which indicates extremely low signal-to-interference ratio. In other words, the data collected on accelerometer must be pre-processed before being used to generate the signatures of hotwords.

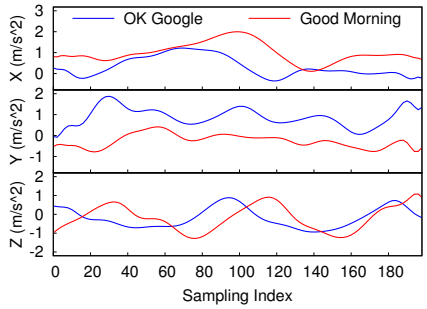
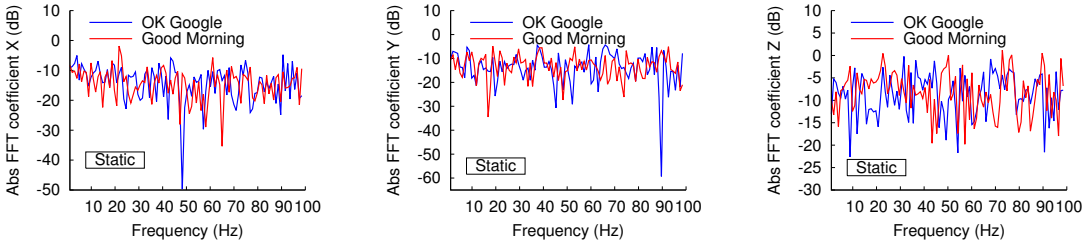
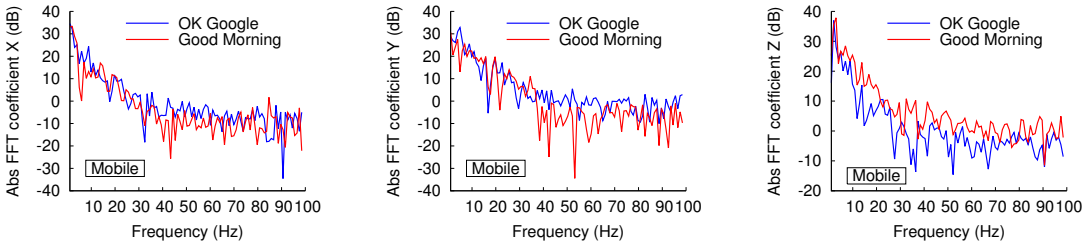


Figure 11: Example: Accelerometer Readings when User Moves and Speaks Short Sentenses

There has been a considerable amount of research in recognizing human activities through accelerometer data. From previous works [25,29,30], it is known that the most human activities (such as walking, changing postures etc.) exhibit lower frequency (0.1-2 Hz). Fig. 12 compares the frequency domain of the accelerometer reading of the static and the mobile scenario. It is observed that even when user is mobile and performs high intensity activities, the energy is mainly concentrated in the lower frequency band (<= 30 Hz). This is confirmed in Fig. 12 which compares the FFT coefficients of the accelerometer signals for static and mobile scenarios. It is observed that the energy in frequency band lower than 30 Hz is much higher for the mobile case. We also analyze another mobility scenario where user is sitting on a chair performing routine activities at workplace. Compared to walking, such an activity is of lower intensity, however, it forms an important use-case for AccelWord where user sitting at home or workplace provides voice commands to her phone. Fig. 13 shows the FFT coefficients of such sitting activity and compares it with a typical waking activity. In



(a) Static: X Axis (b) Static: Y Axis (c) Static: Z Axis)



(d) Mobile: X Axis (e) Mobile: Y Axis (f) Mobile: Z Axis

Figure 12: The FFT of the Static and Mobile Scenarios

both cases, the user is assumed to be not speaking anything. We observe that sitting results in even less energy at lower frequencies (<=20 Hz) compared to walking activities.

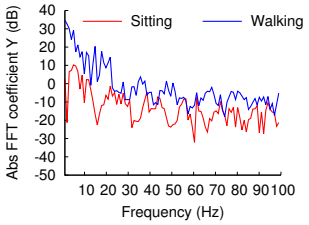


Figure 13: FFT of User's Sitting and Walking Activity

This means that a high-pass filter can be used to filter out the mobility interference from the accelerometer signal before calculating the features we discussed in Section 4.1.

The problem, however, is to choose the correct cut-off frequency for the high-pass filter since attenuating signals more than necessary at the lower frequencies may also remove the effect of user's voice. Since in our case, the human voice is "received" by the accelerometer, high-pass filtering with 30 Hz can cause severe reduction in the accuracy of hotword detection. We rely on the empirical data to find the suitable cut-off frequency that can accurately remove mobility interference while preserving the effect of user's voice on accelerometer signal. We observed in Fig. 9 that in stationary case, all three frequency domain features - DomFreqRatio, Entropy and Energy - have high information gain. This means that they are useful in classifying the hotwords from the other spoken words. We evaluate the information gain of these three features while applying a high-pass filter with different cut-off frequency. Fig. 14 shows the change in information gain as the cut-off frequency increases from 1 Hz

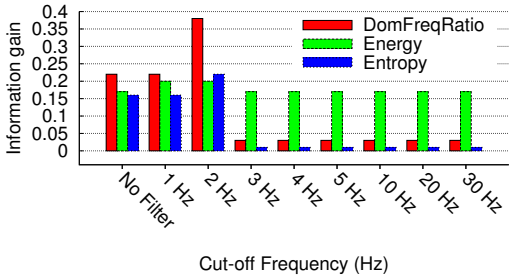


Figure 14: Impact on information gain with varying values of cut-off frequency of high-pass filter; For each feature, we report the information gain value which is the maximum across all three axis

to 30 Hz. When the information gain reduces, it can be inferred that the frequency domain features which were previously important in classification are no longer useful and the overall classification accuracy will also reduce. We observe that the information gain for the three features first increases until the cut-off frequency of 2 Hz. This means that until this point, the high-pass filter works well in removing the mobility interference and improving the classification. However, the information gain values drop sharply (for DomFreqRatio and Entropy) after 2 Hz which indicates that filtering beyond 2 Hz removes information that is useful in classification. Based on this empirical observation, we choose the cut-off frequency of 2 Hz for the high-pass filter.

4.3 Training AccelWord Classifier

For training the AccelWord classifier, a user is required to speak the hotword a certain number of times while the accelerometer data is collected. The user is also required to utter any other randomly chosen words or sentences. Once

the accelerometer data is collected, the AccelWord classifier can be trained. Additional details of how many times the hotword is spoken etc. are provided in Section 5. Once the training instances are provided, the features are calculated and the machine learning classifier is built. This process of calculating features and building the classifier can be done on the smartphone or it can be offloaded to a cloud for energy savings. Note the this process is only performed once and is not required to be repeated after the training. Also, we do not build separate classifiers for stationary and mobile cases as doing so would require to first detect if the user is mobile or stationary. In all cases, we simply use one classifier where any mobility in training instances is filtered using the high-pass filter. Once the classifier is built, it can perform the hotword detection in real time by monitoring the accelerometer data.

Decision Tree Classifier: For real-time classification, we propose to use a simple sliding window based approach where at any time instance, last t seconds of accelerometer signals are buffered. After every certain period, the features are calculated for the buffered data and signature matching is performed using the classifier to check if the hotword was spoken in the last t seconds or not. Because both the feature calculation and model checking need to be performed periodically, it is necessary to choose a computationally efficient machine learning classifier. We use simple decision tree to build our AccelWord classifier. Because a decision-tree based classifier can be implemented using simple if-else conditions, it can perform the classification with very low computational complexity. This is crucial to meet our low energy consumption goal of AccelWord.

We note that using more complex machine learning methods (such as decision trees with bagging or boosting [31]) can improve the hotword detection accuracy, they might also increase the computational cost and energy for hotword detection. We leave this exploration of optimizing accuracy and energy of AccelWord to future work.

5. IMPLEMENTATION

We implemented *AccelWord* as a standalone app running on Android 4.4.2 (API Level 19) devices. To avoid any GUI related power consumption variations, we design the AccelWord's front-end to be simple, as shown in Fig. 15. For efficient calculation of the features, we rely on the data structures defined in widely used Java library "commons math". Since typical hotwords are usually quite short in length and most users can speak them in less than 2 seconds, AccelWord buffers 2 seconds of accelerometer data (400 samples) in a FIFO queue. Note that this can be adjusted based on the typical time taken to speak the hotword. In each run of the feature calculation, AccelWord first filters the data using a high-pass filter with cut-off frequency of 2 Hz. Then the calculated features are compared with the extracted hotword signature. We set the time interval between each feature calculation to be a variable and test with different interval lengths.

We train the AccelWord classifier off-line on a workstation and import the model to the app. This is similar to other voice control applications like Google Now where pre-trained model of user speaking the hotword is incorporated in the app. This allows a fair comparison in terms of the energy consumption since there is no extra energy consumed for training during the run-time.

Once the hotword is detected by the AccelWord app, it initiates the "Google Voice Search" using "SEARCH_ACTION" Android intent. Here, the microphone is turned on and user's voice commands are recognized by Google voice search engine. For better repeatability, we implement two modes in the app. In the first mode (referred as AccelWord Energy mode), we simply log the result of hotword detection algorithm and do not initiate a Google search even when the hotword is detected. This allows us to measure the energy in a more controlled way where there is no additional energy consumed for Android intent access and other relevant processes. In the second mode (referred as AccelWord Performance mode), the app will not only perform hotword detection, but will also switch to Google Voice Search GUI if the hotword is detected.

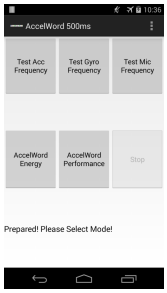


Figure 15: AccelWord Android App

6. PERFORMANCE EVALUATION

To evaluate the performance of AccelWord, we conduct hotword detection tests with 10 volunteers (5 females and 5 males). Two other voice control applications (Google Now and Samsung S Voice) are used to provide the performance comparison. The experiments are conducted on two phones: Samsung Galaxy S4 and Google Nexus S. Since the Samsung S Voice is exclusive to Galaxy phones, its data is not reported (marked N/A) in a few (less than two) scenarios.

In the experiments, we choose "Okay Google" to be our hotword - the same as Google Now. The Samsung S Voice uses "Hi Galaxy" to be its hotword. For training the AccelWord classifier, each volunteer speaks the hotword 10 *vaild* times. Here, *valid* means that the hotword speaking instance is used in the training only if it can be successfully recognized by Google Now or Samsung S Voice. Each volunteer also speaks 20 other randomly chosen short sentences (<= 2 seconds) of their liking to generate non-hotword test instances. Once the hotwords and random sentences are recorded, each sentence is repeatedly played 10 times (5 static and 5 mobile) in the experiments (100 times "Okay Google", 100 times "Hi Galaxy" and 200 times other random sentences) to evaluate in presence of other randomness (background noise etc.).

The performance of AccelWord is evaluated in two aspects - accuracy and energy consumption.

Accuracy: Accuracy is evaluated with two metrics:

- **True Positive (TP) Rate:** It is defined as the percentage of instances where speaking of the hotword is correctly recognized as speaking of the hotword.
- **False Positive (FP) Rate:** It is defined as the percentage of instances where speaking of other sentences is recognized as speaking of the hotword.

It is worth noting that AccelWord is a user-specific classifier which means that a separate classifier is built for each user. This is because the accelerometer-based hotword detection has an added advantage that it can distinguish the user for which the classifier was trained from the other users. This loose form of user authentication is especially beneficial for voice control applications since it is not only possible to detect the hotword but it is also possible to recognize if it was the owner user who spoke the hotword. We will evaluate this claims of speaker recognition in Section 6.3. Because the frequency of male and female voice is different, we present the accuracy results for both male and female users separately. The results with label "female" are the average values of the 5 female volunteers, and the same for the results of the 5 male volunteers.

Energy: For comparing the energy consumption, we first measure the GUI power consumption of each of the AccelWord, Google Now and Samsung S Voice applications when the app is in the foreground (screen on) but it is not running the hotword detection. This GUI power consumption is then removed from the subsequent measurements when the app is performing the hotword detection. This allows a fair comparison since the GUI power consumption can be significantly different depending on the front-end design. The energy comparison is provided for both the devices separately.

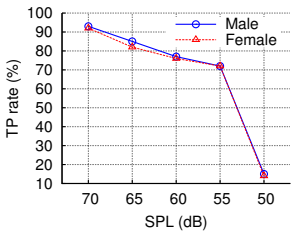
Our experimental results show AccelWord can achieve similar accuracy of hotword detection as Google Now and Samsung S Voice applications while consuming only 50% of the energy compared to both the apps. Sections 6.1, 6.2 and 6.3 show the hotword detection accuracy, energy efficiency and speaker recognition results respectively. For better presentation, we show all the TP rate in figures and all the FP rate in tables consistently.

6.1 Accuracy

We study the hotword detection accuracy in terms of three factors: (1) SPL at the receiver phone, (2) background noise and (3) user's mobility.

Sound Pressure Level (SPL): Intuitively, higher value of SPL on the receiving phone should result in better detection of hotword. We evaluate this using two cases - one where both training and testing instances have the same SPL and the other where they have multiple different SPLs. To achieve a desired SPL on the receiving phone, we play the recorded audio of hotword and non-hotword sentences on the iPhone 4S used in Fig. 6 and Fig. 10 and adjust the iPhone's volume without changing the distance between the iPhone and the receiving phone.

Trained and Tested with the Same SPL: We use 5 different values of SPL (70, 65, 60, 55, 50 dB) and train and test separate classifiers for each. In each case, all the instances of training and testing are of the same SPL value. 10-fold cross-validation is used to evaluate the TP and FP rates. Fig. 16 shows the TP and FP rate values. It is observed that the TP rate decreases monotonically as the SPL decreases while the FP rate increases. This indicates that the signatures generated at higher SPLs are better which allows improved classification. We can also observe both the TP rate and the FP rate drop to almost 0 when the SPL becomes 50 dB. The reason is that very low sound input at 50 dB SPL fails to cause any noticeable variation in the accelerometer data. As we will show next while comparing with other applications,

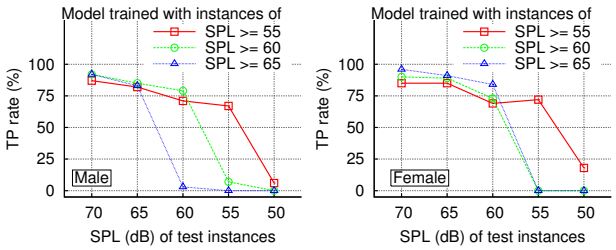


SPL (dB)	FP Rate (%) Male	Female
70	4.8	4.1
65	5.5	5.1
60	7.8	7.5
55	8.5	8.0
50	1.8	2.1

Figure 16: TP and FP rates when AccelWord is trained and tested with instances of the same SPL

at 50 dB SPL, both Google Now and S Voice also fail to recognize any human voice.

Trained and Tested with Multiple SPL: In reality, when user speaks the hotword, the reported SPL at the receiving phone is likely to be different at different times. To test this realistic case, we train the classifier using instances of multiple different SPLs and then test it with instances of a given SPL. For example, the classifier can be trained with instances of 60, 65 and 70 dB SPLs, and tested with instances of 60 dB SPL. The results are presented in Fig. 17 and Table. 3. It is observed that when the classifier is trained with instances of SPL >= x, the TP rate is high for all cases when testing instances have the SPL >= x. For example, for training with SPL >= 60 dB, the TP rates of 60, 65 and 70 dB testing instances are above 80% in male users. Compared to training and testing with the same SPL, we observe that the accuracy drops a little when trained with multiple SPLs. This is expected since training and testing with the same SPL instances is likely to produce a model that fits better. However, since training with instances of multiple SPLs is more realistic, we will use the model trained with instances of SPL >= 55 dB in the rest of the paper for comparing with other apps and evaluation in noisy environments.



(a) Male (b) Female

Figure 17: TP rate when the classifier is trained with instances of multiple SPLs and tested with instances of a given SPL

From the figures, we can also observe that the TP rates of male volunteer scenarios are relatively higher than the female volunteer scenarios. If only consider 55dB and above scenarios, AccelWord achieves 4.1% higher TP rates on male volunteers than on female volunteers on average. This is because the female vocal range is slightly higher than males, while the sampling frequency of the accelerometer is limited at 200Hz. Therefore signature generated by male voice

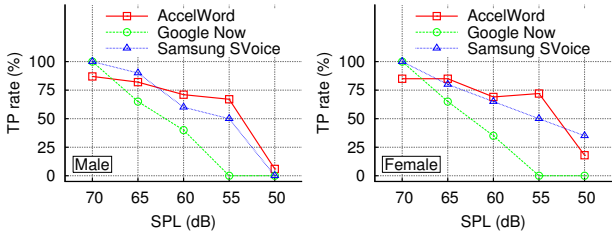
SPL Tested (dB)	FP Rate (%) SPL Trained (dB) >=55	>=60	>=65
70	4.8	3.3	1.3
65	6.5	5.8	2.0
60	6.9	8.3	1.3
55	7.4	4.3	0.3
50	2.0	2.1	1.5

(a) Male

SPL Tested (dB)	FP Rate (%) SPL Trained (dB) >=55	>=60	>=65
70	5.7	3.5	0.3
65	3.0	5.1	0.8
60	4.3	1.8	0.5
55	2.7	0.0	0.0
50	3.0	0.0	1.5

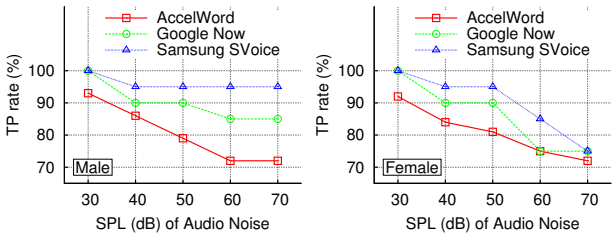
(b) Female

Table 3: FP rates when classifier trained and tested with multiple SPLs



(a) Male (b) Female

Figure 18: Accuracy Comparison with Google Now and Samsung S Voice



(a) Male (b) Female

Figure 19: True Positive Rate vs. Audio Noise SPL

instances is relatively more significant than the signature generated by female voice instances.

AccelWord vs. Other Apps: We now compare the hotword detection accuracy of AccelWord with Google Now and S Voice with varying SPL. The results are presented in Fig. 18. Since all the instances we consider for AccelWord training are the ones which are *valid* in Google Now and S Voice, both the apps achieve 100% of TP rate at 70 dB. In comparison, AccelWord achieves an accuracy of 86% at 70 dB SPL. For 65 and 60 dB, AccelWord achieves higher TP rate than Google Now and comparable TP rate to S Voice. Starting from 60 dB, we observe that Google Now does not react to majority of the hotwords. Since its internal implementation details are unknown, it is unclear why the hotword detection drops sharply after 60 dB. One possible reason is that the FP rate increases significantly at lower SPL. The same phenomenon is also observed with Samsung S Voice starting from 55 dB. Therefore, we only compare the TP rates in 65dB and above. On average, the TP rate of AccelWord is 99.1% and 97.6% of the TP rates of Google Now and S Voice in 65dB and above SPLs.

Accuracy in Noisy Environment: Another important aspect to study is the impact of audio noise on AccelWord. When a user is in a noisy environment (such as public places, malls, cafes, etc.), the surrounding audio noise can be sufficiently high to cause variations in accelerometer. To evaluate how AccelWord works in presence of the audio noise, we generate two kinds of background noise - white noise using Audacity and the background noise recorded at a local cafeteria. We vary the audio noise level from 30 to 70 dB. Note that 70 dB of audio noise is already considered upper limit of acceptable noise pollution by most countries e.g. USA (65 dBA) [32], China (70 dB) [33], Japan (65 dB) [34]. Fig. 19 and Table 4 show the TP rates and the FP rates in different background noise SPLs. The presented results are average values when two kinds of noise are applied separately. The hotword SPL is fixed to 70dB for all the three apps in the noisy environment experiments.

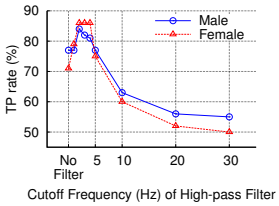
Noise SPL (dB)	FP Rate (%) Male	Female
30	4.8	4.1
40	6.9	5.4
50	6.8	4.2
60	6.3	7.8
70	7.9	8.3

Table 4: False Positive Rate vs. Audio Noise SPL

From the figure, we can observe AccelWord can achieve comparatively lower TP rate than Google Now and S Voice in noisy environment. On average, the TP rate of AccelWord is only 9% and 13% percent less than Google Now and S Voice respectively. From Table. 4, we can observe the FP rate of the AccelWord increases as the noise SPL increases. However, the FP rate of AccelWord is still less than 10% in environment as noisy as 70 dB SPL. We consider this as one of the limitations of AccelWord in its current form. Today's microphones employ advanced techniques for noise canceling that can filter out the background noise. However, in its current form, AccelWord does not have any mechanism to cancel the effect on accelerometer caused by the audio noise. This is further discussed in Section 7.

Accuracy in Mobile Scenarios: We also tested the performance of AccelWord in mobile scenarios. The setup described in Fig. 10 is used for the experiments. Here, each volunteer walks in circle for 5 minutes and this is repeated 3 times. We fix the distance between the speaker phone and the receiving phone to be 30 inches, and the speaker phone volume is adjusted such that received SPL is 70 dB.

As discussed in Section 4.2, the mobility causes interference to the AccelWord hotword detection and it can be removed using a high-pass filter. It was shown that a high-pass filter with 2 Hz cut-off frequency can be used as it allows us to remove the mobility interference without removing the effect of audio signal. We verify this by varying the cut-off frequency of the high-pass filter and observing the resultant TP rate of AccelWord. The results are shown in Fig. 20. Same as Fig. 14, we observe that hotword detection TP rate first increases from applying no filter to cut-off frequency of 2 Hz. This is because the high-pass filter removes the low-frequency noise of user's walking movement which improves the extracted hotword signatures and in turn increases the classification accuracy. Further increasing the cut-off fre-



Cutoff Frequency (Hz)	FP Rate (%) Male	Female
0	1.5	2.1
1	5.3	6.2
2	5.3	6.2
3	7	6.8
4	7	6.8
5	7	6.8
10	8	7.2
20	7.5	6.8
30	6.9	7.1

Figure 20: Change in TP and FP Rates for Different Cutoff Frequency

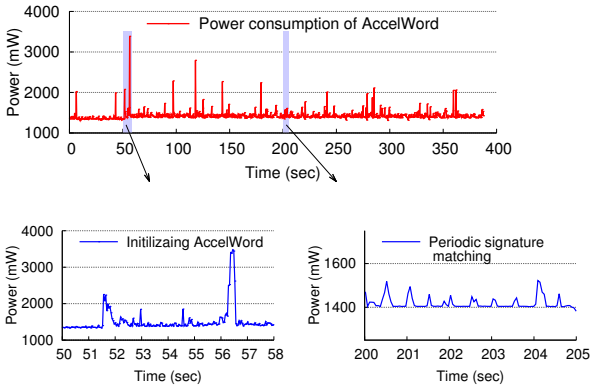


Figure 21: Example: Energy Trace of AccelWord

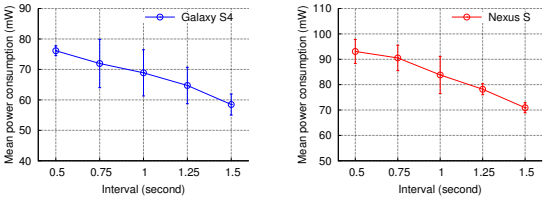
quency causes a drop in TP rate because more and more "useful" signal is filtered out while removing mobility interference, generating weaker signatures of hotword.

Although the accuracy of AccelWord is slightly lower than in the static scenarios, the TP rate (84% with 2Hz high-pass filter) is still high enough for accurate hotword detection. Further improvements can be achieved by designing more advanced methods of mobility interference cancellation.

6.2 Energy Efficiency

In this section, we will compare the power consumption of the three apps: AccelWord, Google Now and S Voice. The power consumption is measured using the Monsoon Power Monitor at sampling frequency 5 kHz. An example of the power trace of AccelWord running on Galaxy S4 is shown in Fig. 21. Between 0s and 51s, the AccelWord app is in the foreground but not running. This shows the GUI power consumption of the AccelWord app. The AccelWord app is started at 51s and it finishes its initialization at 57s, as shown in Fig. 21b. The initialization process includes: initialize the queue cache, register the accelerometer listener to the OS and load the pre-trained classifier model. After 57 seconds, AccelWord begins to monitor the hotwords. In this example, we set the interval between each feature calculation to be 500 ms, and hence, we can observe approximately 10 pulses each 5 second, as shown in the enlarged Fig. 21c.

The power consumption of AccelWord is measured on a Galaxy S4 and a Nexus S. We vary the time interval between each feature calculation from 500ms to 1500ms. For each time interval setting, 5 runs of measurements are conducted on each phone, where each run lasts for 30 minutes. The average values are shown in Fig. 22.



(a) Galaxy S4 (b) Nexus S

Figure 22: The Power Consumption of AccelWord

From the figure, we observe the expected that the computation energy can be reduced if the interval between each feature calculation is longer. However, the energy consumption of accelerator sampling will not be affected. Comparing the power consumption at 500ms interval (the highest) with the power consumption of Google Now and S Voice shown in Fig. 3, we can observe that AccelWord only consumes half of the power consumed by Google Now and Samsung S Voice, as shown in Table. 5. It is worth noting that current speech recognition employs much more sophisticated machine learning techniques such as early rejection (compute low-cost features first to quickly reject the non-hotwords and calculate the high-cost features only if necessary). It also worth noting that in terms of processing and implementation, AccelWord may be less efficient compared with its counterparts with large development and maintenance teams, e.g. Google and Samsung. It is necessary to point out that the energy savings of AccelWord is primarily due to energy-inexpensive sensing via accelerometer instead of using the microphone. Given that our current implementation of AccelWord is not optimized (compared to well-designed solutions like Google Now), even more energy savings are likely with AccelWord when more sophisticated machine learning methods are used along with better implementation.

	Energy Saving (%) Galaxy S4	Nexus S
Google Now	46.19%	53.85%
Samsung S Voice	57.14%	N/A

Table 5: The percentage of energy saved

6.3 Speaker Identification

Although our objective of designing AccelWord is to enable energy-efficient hotword detection, we observe that it can also distinguish a user's voice from other users. We claim that the AccelWord hotword signatures are user-specific. Because every user's voice frequency is different, the extracted hotword signature of accelerometer also reflects the user who speaks the hotword. Such speaker recognition can be especially useful in scenarios where there are multiple users and each of them wants to only interact with their own smart-device without any cross-talk.

To evaluate that AccelWord users can be distinguished from each other, we perform a multi-class classification. Here, we use all the instances of the 10 users speaking the hotword for training and testing with 10-fold cross-validation. The classification results are presented in the form of a confusion matrix in Fig. 23.

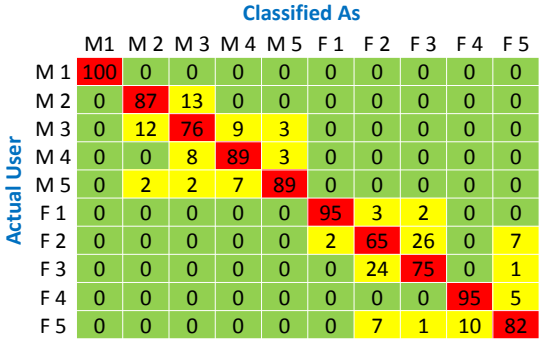


Figure 23: Confusion Matrix of Speaker Identification (M: Male F: Female)

It is observed from Fig. 23 that AccelWord can identify the speakers with very high accuracy (female - 82% and male - 88%). It is interesting to observe that none of the female users are ever classified as male users (and vice versa). This means that classifying whether user is male or female using accelerometer data can be done with very high accuracy. The feasibility of speaker and its gender identification can be leveraged to provide user authentication along with hotword detection using AccelWord.

7. DISCUSSION

In this section, we discuss the limitations of AccelWord and suggest directions for further exploitation of using accelerometer for specific word detection.

Higher Sampling Rate: One possible strategy to improve the accuracy of AccelWord is to increase the sampling frequency of accelerometer. On the other hand, increasing sampling frequency may increase the energy spent on sampling. So there is an optimization problem in the trade-off of accuracy and energy efficiency.

Low-power Processor: Latest smartphones such as Moto X [35] and Nexus 6 [6] use a dedicated low-power processor for continuous sensing related tasks (e.g. audio sensing). When AccelWord is executed via such a processor, the energy savings are still likely to be proportionally lower than audio sensing on such a processor.

Audio Noise Cancellation: In microphone based speech recognition research area, there exists many high performance and efficient noise cancellation techniques, e.g. Least Mean Squares (LMS), Normalized Least Mean Squares (NLMS) and Affine Projection (AP) [36]. However, the noise cancellation in the accelerometer based hotword detection scenario is more complex, since the noise comes from two aspects: background audio noise and the mobility inference. Deeper exploration on how the voice is modulated on accelerometer will help to design noise cancellation algorithms for accelerometer based hotword detection mechanisms.

Other Mobility Patterns: In this work, we only considered the walking and speaking mobility pattern. In reality, the mobility of the users may be more complex, e.g. running, driving or taking flights, which will result in more severe interference. To resolve these possible problems, new methods should be developed to remove or reduce the interference cased by different mobility patterns.

Privacy Implications. The confusion matrix shown in Fig. 23 reveals the potential risk of using AccelWord to identify some of the user's privacy information, e.g. user's gender. Worse still, if the attacker is able to apply a patch to the Android system to tune up the sampling frequency of accelerometer, it is also possible to use accelerometer readings to reconstruct portions of the human speech. If this is possible, applications with access permissions to accelerometer can also hear user's speech creating a huge privacy risk. In our ongoing work, we are exploring the feasibility and severity of such privacy leakage, and possible protection mechanisms.

8. RELATED WORK

Speech Recognition by Microphones: In [11], the authors introduced *JustSpeak*, a universal voice control solution which enables Google speech recognition on any screen of the Android system. In [12], Ignacio*et al.* used Deep Neural Networks (DNN), a state-of-art machine learning technique, to identify the language spoken by smartphone users. There are also a number of theoretical papers focusing on improve the computation efficiency of the training process of speech recognition [13–15]. These works are quite different from ours, since AccelWord relies on accelerometer to monitor voice signals and targets only on hotword detection.

Identify User Activities by Accelerometer: There are a number of works focusing on identify or detect user actives by analyzing the accelerometer data. [25,29,30,37,38] talk about general machine learning algorithms on identifying mobility actives, e.g. speaking, walking, running, dancing, stairs-up, stairs down. In [39], the authors designed a wearable ring platform to detect user's gestures up to 2Hz frequency. In [8], the authors presented (sp)iPhone which uses the accelerometer on a smartphone to detect human keystrokes through the vibration disseminated by the desk surface. This is different from Accelword where the accelerometer variations are due to sound instead of direct vibrations of a physical surface. Accelword does not require the speaker and the receiver to be placed on the surface of the same solid object.

Audio Reconstruction: Davis *et al.* presented the *Visual Microphone* system which can passively reconstruct the sound in a room by camcording the vibration of a potato chip bag in the room [40]. In [41], the authors presented *AVASR* which reconstructs human speech by processing the speaker's lip motion captured by Microsoft Kinect. *Gyrophone* [7] is a system which upsamples the gyroscope samples received at 4000 Hz to reconstruct the audio. The upsampling and reconstruction procedure is computationally expensive and unsuitable for our objective of energy efficient hotword detection. We have demonstrated that the accelerometer variations sampled at 200 Hz is enough for hotword detection at low energy cost.

9. CONCLUSION

In this paper, we introduced AccelWord, an accelerometer based hotword detection system for smart devices. AccelWord proves the feasibility of using accelerometer to detect the signatures of voice signals. Comprehensive experiments show AccelWord performs accurate hotword detection while consuming comparatively very low energy. Future exploration directions are also provided to further improve the performance of AccelWord.

10. REFERENCES

[1] "Apple siri, https://www.apple.com/ios/siri/."

[2] "Google now, http://www.google.com/landing/now."

[3] "Android wear." http://www.android.com/wear/.

[4] "Google glass." https://www.google.com/glass/start/.

[5] "Amazon echo." http://www.amazon.com/oc/echo.

[6] "Nexus 6, https://www.google.com/nexus/6/."

[7] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *USENIX'2014*.

[8] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS'2011.

[9] "Samsung s voice." http://www.samsung.com/global/galaxys3/svoice.html.

[10] "Monsoon power monitor." https://www.msoon.com/LabEquipment/PowerMonitor/.

[11] Y. Zhong, T. V. Raman, C. Burkhardt, F. Biadsy, and J. P. Bigham, "Justspeak: Enabling universal voice control on android," in *W4A 2014*, 2014.

[12] I. Lopez-Moreno, J. Gonzalez-Dominguez, and O. Plchot, "Automatic language identification using deep neural networks," in *ICASSP'2014*.

[13] W. Zhang and P. Fung, "Discriminatively trained sparse inverse covariance matrices for speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, pp. 873–882, May 2014.

[14] C. Chelba, P. Xu, F. Pereira, and T. Richardson, "Distributed acoustic modeling with back-off n-grams," in *ICASSP'2012*.

[15] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 10, 2014.

[16] Wikipedia. Examples of Sound Pressure, `http://en.wikipedia.org/wiki/Sound_pressure#Examples_of_sound_pressure`.

[17] STMicroelectronics. Everything about STMicroelectronics 3-axis digital MEMS andoscopes, `http://www.st.com/web/en/resource/technical/document/technical_article/DM00034730.pdf`.

[18] Ceramic capacitors feature reduced acoustic noise, `http://www.electronics-eetimes.com/en/ceramic-capacitors-feature-reduced\\-acoustic-noise.html`.

[19] G. Roth, "Simulation of the effects of acoustic noise on mems gyroscopes," *Thesis, Auburn Univeristy*, 2009.

[20] "Inven sense inc. mpu-6000 and mpu 6050 product speficication." http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf.

[21] Wikipedia. Human Hearing Range, `http://en.wikipedia.org/wiki/Hearing_range`.

[22] Wikipedia. Voice Frequency, `http://en.wikipedia.org/wiki/Voice_frequency`.

[23] S. Meter. Google Play Store, `https://play.google.com/store/apps/details?id=kr.sira.sound`.

[24] EngineeringToolnbox. Sound Pressure Levels of Common Sources, `http://www.engineeringtoolbox.com/sound-pressure-d_711.html`.

[25] E. Munguia Tapia, *Using machine learning for real-time activity recognition and estimation of energy expenditure*. PhD thesis, Massachusetts Institute of Technology, 2008.

[26] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld, "The sphinx-ii speech recognition system: an overview," *Computer Speech & Language*, vol. 7, no. 2, pp. 137–148, 1993.

[27] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *IEEE ICASSP'2000*.

[28] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 3rd ed., 2011.

[29] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," in *SIGKDD'2010*.

[30] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science*, vol. 34, pp. 450–457, August 2014.

[31] T. K. Ho, "The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 8, pp. 832–844, 1998.

[32] United States Environmental Protection Agency, Summary of the Noise Control Act, 1972.

[33] Enviroment Projection Agency of the State Council of China, The quality standared of noisy enviroment, 2008.

[34] Ministry of the Environment of Japan, Current Framework of Vehicle Noise Regulation in Japan, September 2012.

[35] "Moto x (2rd generation), https://www.motorola.com/us/motomaker?pid=flexr2."

[36] S. A. Hadei and M. Lotfizad, "A family of adapative filter algorithms in noise cancellation for speech enhancement," *International Journal of Computer and Electrical Engineering*, vol. 2, April 2010.

[37] A. Matic, V. Osmani, and O. Mayora, "Speech activity detection using accelerometer," in *IEEE EMBC'2012*.

[38] S. V. Dusan, E. B. Andersen, A. Lindahl, and A. P. Bright, "System and method of detecting a user's voice activity using an acceleromter." US Patent No. 20140093093 A1.

[39] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization," MobiSys'14.

[40] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: Passive recovery of sound from video," *ACM Trans. Graph.*, July 2014.

[41] G. Galatas, G. Potamianos, and F. Makedon, "Audio-visual speech recognition incorporating facial depth information captured by the kinect," in *EUSIPCO'2012*.