BodyBeat: A Mobile System for Sensing Non-Speech Body Sounds

Tauhidur Rahman[1], Alexander T Adams[2], Mi Zhang[1], Erin Cherry[3], Bobby Zhou[1], Huaishu Peng[1], Tanzeem Choudhury[1]
[1]Cornell University, [2]University of North Carolina at Charlotte, [3]University of Rochester
tr266@cornell.edu, aadams85@uncc.edu, mizhang@cornell.edu, erinc@cs.rochester.edu, bz88@cornell.edu, hp356@cornell.edu, tkc28@cornell.edu

ABSTRACT

In this paper, we propose BodyBeat, a novel mobile sensing system for capturing and recognizing a diverse range of non-speech body sounds in real-life scenarios. Non-speech body sounds, such as sounds of food intake, breath, laughter, and cough contain invaluable information about our dietary behavior, respiratory physiology, and affect. The BodyBeat mobile sensing system consists of a custom-built piezoelectric microphone and a distributed computational framework that utilizes an ARM microcontroller and an Android smartphone. The custom-built microphone is designed to capture subtle body vibrations directly from the body surface without being perturbed by external sounds. The microphone is attached to a 3D printed neckpiece with a suspension mechanism. The ARM embedded system and the Android smartphone process the acoustic signal from the microphone and identify non-speech body sounds. We have extensively evaluated the BodyBeat mobile sensing system. Our results show that BodyBeat outperforms other existing solutions in capturing and recognizing different types of important non-speech body sounds.

Keywords

Mobile Sensing, Non-Speech Body Sound, Acoustic Signal Processing, Embedded Systems

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Signal Processing Systems

General Terms

Algorithm, Design, Experimentation, Measurement

1. INTRODUCTION

Human speech processing has been studied extensively over the last few decades. The emergence of Apple Siri, the speech recognition software on iPhones, in many ways, is a mark of success for speech recognition technology. However, there is very little research on using sensing and computing technologies for recognizing and interpreting non-speech body sounds. Non-Speech body sounds contain invaluable information about human physiological and psychological conditions. With regard to food and beverage consumption, body sounds enable us to discriminate characteristics of food and drinks [12]. Longer term tracking of eating sounds could be very useful in dietary monitoring applications. Breathing sounds, generated by the friction caused by the air flow from our lungs through the vocal organs (e.g. trachea, larynx, etc.) to the mouth or nasal cavity [26], are highly indicative of the conditions of our lungs. Body sounds such as laughter and yawn are good indicators of affect. Therefore, automatic tracking these non-speech body sounds can help in early detection of negative health indicators by performing regular dietary monitoring, pulmonary function testing, and affect sensing.

During the past few years, a number of mobile and wearable sensing systems have been developed to detect non-speech body sounds. For example, Larson *et al.* [17] used the smartphone's microphone to detect cough. Hao *et al.* [14] also used the smartphone's microphone to capture both snoring and coughing sounds for assessing the quality of sleep. In [30], Yatani and Khai developed a wearable system that used a condenser microphone with a stethoscope head to capture a variety of non-speech sounds. Nirjon *et al.* [23] integrated a standalone microphone into an earphone to extract heartbeat information. All these existing work used condenser microphones that capture sounds via air pressure variations. However, we argue that the condenser microphone is not the most appropriate microphone to capture non-speech body sounds. One reason is that some non-speech body sounds such as eating and drinking sounds are very subtle and thus generate very weak air pressure variations. This makes them very difficult to be captured by condenser microphones. Second, the condenser microphone is very susceptible to external sounds and ambient noises. As a result, the quality of body sounds captured by condenser microphones decreases significantly in real-world settings.

In this paper, we present the design, implementation, and evaluation of BodyBeat: a mobile sensing system that is capable of capturing a diverse set of non-speech body sounds and recognizing physiological reactions that generate these sounds. BodyBeat is built on top of a novel piezoelectric sensor-based microphone that captures body sounds conducted through the body surface. This custom-made microphone is designed to be highly sensitive to subtle body sounds and less sensitive to external ambient sounds or external noise. To recognize these non-speech body sounds, we carefully selected a set of discriminative acoustic features and developed a body sound classification algorithm. Given the computational complexity of this algorithm and the resource limitation of the smartphone, we partitioned the whole com-

MobiSys'14, June 16–19, 2014, Bretton Woods, New Hampshire, USA.

http://dx.doi.org/10.1145/2594368.2594386.

putational framework and implemented a distributed computing system that consists of an ARM micro-controller and an Android smartphone. To evaluate the effectiveness of BodyBeat, we tested the custom-made microphone, the classification algorithm, and the distributed computing system using non-speech body sounds collected from 14 participants. Specifically, the main contributions of this paper are: (1) we design and implement a custom-made piezoelectric sensor-based microphone that is able to capture a diverse set of body sounds whiling dampening external sounds and ambient noises; (2) we develop a body sound classification algorithm based on a set of discriminative acoustic features; (3) we implement the signal processing and machine learning algorithm on an ARM micro-controller and an Android smartphone; and finally (4) we benchmark the performance of our custom-made microphone against other state-of-the-art microphones, evaluate the performance of the body sound classification algorithm, and profile the system performance in terms of CPU and memory usage and power consumption.

The paper is organized as follows. Section 2 outlines the challenges and design considerations of the development of the body sound sensing system. Section 3 presents the design and test results of our custom-made piezoelectric sensor-based microphone. In Section 4, we describe our feature selection and classification algorithms for recognizing a diverse set of body sounds. In Section 5, we explain in details the implementation of the computational framework on the ARM micro-controller and the Android smartphone. We discuss the potential applications of BodyBeat in Section 6. Finally, we give a brief review on some of the existing work in Section 7 and conclude this paper in Section 8.

2. DESIGN CONSIDERATIONS

In this section, we discuss the challenges of capturing and recognizing non-speech body sounds. We also describe how we tackled these challenges in the design of BodyBeat. The detailed design is described in Section 3 and 4.

2.1 Capturing Non-Speech Body Sounds

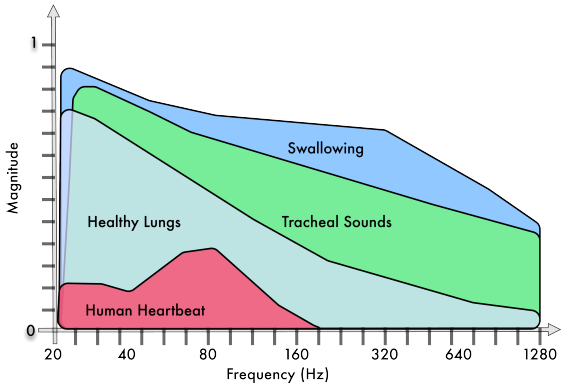


Figure 1: Illustrates approximate frequency range and relative loudness of selected body sounds

In the context of mobile sensing, the built-in microphone is the most widely used sensor for detecting acoustic events [22, 20, 21]. However, the mobile phone microphone (typically an electret or condenser microphone) is specifically designed for the purpose of voice communication and thus the frequency band is optimized for speech. Non-speech body sounds are generated by complex physiological processes inside the human body. After body sounds is produced inside our body, the energy of the body sounds decreases significantly by the time they reach the body surface. Therefore, non-speech body sounds are in general barely audible. Based on the frequency differences between voice and body sounds, the mobile phone microphone is not the best acoustic sensor for capturing non-speech body sounds. In building the BodyBeat microphone, we considered the following design requirements:

1. The microphone should capture a wide array of subtle body sounds lying in different portion of the frequency spectra.
2. The microphone should be robust against any external sound or ambient noise.
3. The microphone should have mechanisms compensating friction noise due to user's body movement.

The first two requirements are essential for continuous capture of different body sounds with a high signal-to-noise ratio. In the third requirement, the mechanical movement of the body may generate noise due to the friction between body surface and the microphone, which may render captured body sounds uninterpretable. Therefore, we should have mechanism with the microphone to avoid the generation of the friction noise due to users' body movement.

We propose a new microphone, BodyBeat, that captures a wide range of non-speech body sounds. Specifically, BodyBeat adopts a custom-built piezo-electric sensor to capture these sounds. Since its worn around the user's throat, the bone conduction sensor is very sensitive to the vibration caused by non-speech body sounds in the frequency spectrum of 20Hz to 1300Hz. In addition, BodyBeat is also customized to dampen any external sound or noise from the ambient environment. In this manner, most of the features of non-speech body sounds are preserved and captured without being skewed by external sounds. In Section 3, we describe our custom-built microphone and demonstrate its superior performance in capturing non-speech body sounds, compared to a range of other state-of-the art microphones.

2.2 Recognizing Non-Speech Body Sounds

Compared to speech sounds, non-speech body sounds have distinct frequency spectrum. Specifically, the frequencies of speech sounds range from 300Hz to 3500Hz. In comparison, non-speech body sounds are located within the lower region of the frequency spectrum, ranging from 20Hz to 1300Hz. As an example, Figure 1 illustrates the frequency spectrum of four non-speech body sounds. As shown, the human heartbeat is one of the more subtle body sounds with a low magnitude from 20Hz to 200Hz. Breathing sounds (ranging from 20Hz to 1300Hz) are much louder in the 20Hz to 200Hz range but have a large loss in magnitude as the frequency range increases [29]. The unique nature of the body sound's power spectra suggests that spectral features such as power in different filter banks or spectral centroid, spectral variance, spectral entropy might contain valuable information to discriminate among body sounds. Moreover, the concentration of the body sound in the low frequencies warrants higher attention to the minute changes in the low frequencies, in other words higher frequency resolution in the low frequencies. In a previous exploration, Yatani and Khai [30] also used logarithmic filter banks (having center frequencies and bandwidth increase logarithmically).

We begin our exploration by designing and extracting a variety of acoustic and statistical features with the objective of comprehensively describing the characteristics of body sounds. We critically examine the performance of the feature pool and selected a subset of them, which are the best in modeling body sounds. Lastly, we train our inference algorithm and optimize for different parameters.

Sensor ID	Origin	Type of Mic Sensor	Diaphragm Material	Using Stethoscope Head	Reference
M1	Custom-made	brass piezo	latex	no	-
M2	Custom-made	brass piezo	silicon	no	-
M3	Custom-made	film piezo	latex	no	-
M4	Custom-made	brass piezo	latex	yes	-
M5	Custom-made	condenser	plastic	yes	BodyScope [30]
M6	Off-the-shelf	unknown	unknown	no	Invisio [3]
M7	Off-the-shelf	unknown	unknown	no	Temcom [7]

Table 1: Introducing all the microphones considered for recording subtle body sounds

2.3 Resource Limitations and Privacy Issues

While designing BodyBeat, we considered the resource requirements of various computational frameworks and opted for techniques that were capable of running analog to digital conversion of the audio signal; acoustic feature extraction; and classification of body sounds in real-time. Implementing the algorithm entirely in the Android smartphone would be very computationally expensive, and it would cause an unnecessary battery drain. In contrast, another extreme implementation approach would be transferring all the data to a web-based service that classifies the raw (or semi-processed audio signal) to different body sounds. This approach requires good internet connectivity to transfer large amounts of data. Therefore, we optimized our approach by implementing our algorithm in two different platforms: an ARM micro-controller and an Android smartphone.

The audio codec and portions of the feature extraction were implemented on the ARM micro-controller. The ARM unit also employed a frame admission control using some acoustic features, which filtered unnecessary frames that contained no body sounds of interest. If the ARM unit finds a frame containing a specified body sound, it sends the frequency spectrum of the current frame to the Android phone via Bluetooth. We employed a fast and computationally efficient fix-point signal processing algorithm in the ARM unit. Unlike a web-based implementation, this distributed implementation infers body sounds in real-time, which will allow for real-time intervention applications in the future.

We also take the privacy issues into consideration in the design of BodyBeat. To safeguard privacy, BodyBeat filters out the user's raw speech data via an admission control mechanism. In addition, the BodyBeat microphone is specifically designed to be robust against external sounds and thus any speech from other conversation partners is not captured.

3. MICROPHONE DESIGN AND EVALUATION

In this section, we present the design of our BodyBeat microphone for capturing non-speech body sounds. We compare the performance of a set of seven microphones based on the design requirements presented in Section 2.1.

3.1 Microphone Design

Figure 2 illustrates the architecture of our custom-built piezoelectric sensor-based microphone. The microphone was built around a piezoelectric sensor and a 3D printed capsule. This capsule is made with a 3D printer using Polyactic Acid (PLA) filament. The capsule was then filled with a soft silicone (shore hardness of 10) as internal acoustic isolation material. The piezoelectric sensor was then placed in the capsule with the back of the sensor lying on top of the soft silicone filling to capture the subtle body sound vibrations. After the silicone filling cured, the exposed

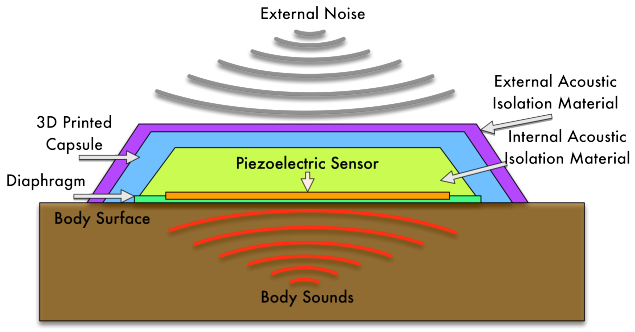


Figure 2: Diagram of piezoelectric sensor-based microphone

front of the piezoelectric sensor was covered with a thin diaphragm (~.001mm), made of either silicone or a piece of latex. Lastly, the exterior of the capsule was covered using external acoustic isolation material, which is a hard, dense, brushable silicone (shore hardness of 50). The internal and external acoustic isolation material (respectively the soft silicone layers inside the capsule and hard silicone layer outside the capsule) act as acoustic isolators, which helps to reduce external noise. In addition, the soft silicone inside the capsule helps the piezoelectric sensor to absorb the surface vibrations without damping the piezoelectric transduction too much. For this design, selecting the right diaphragm material is crucial. A material that has very similar acoustic properties of muscle and skin will maximize the signal transfer to the microphone. Moreover, as the diaphragm is placed on users' skin, we considered inert materials so as to not irritate users' skin.

3.2 Performance Benchmarking

In this work, we built four different types of microphones (M1, M2, M3, and M4) based on the same architecture shown in Figure 2. We varied two variables (type of piezo and diaphragm material) to build these four microphones (M1, M2, M3, and M4). In addition, we duplicated the microphone proposed in [30]. It (M5) is made with a small condenser microphone attached to a stethoscope head. We also considered two additional state of the art commercial bone conduction microphones: M6 [3] and M7 [7]. Instead of capturing sound directly from the air, both M6 and M7 are designed to pick up sound conducted through bone from direct body contact. They also have been extensively used for speech communication under highly noisy environment for army, law enforcement agencies, fire rescuers etc. We ran two tests using the seven microphones listed in table 1. Firstly, a frequency response test is ran to compare the sensitivity of different microphones. Then we run an external noise test to compare the susceptibility of different microphones. Based on these two test, we select a microphone that is highly sensitive to the body sound and less

susceptible to external sound. Lastly, we run a microphone position test to select the optimal head location to attach the BodyBeat microphone to capture a wide range of body sounds.

3.2.1 *Frequency Response Test*

We ran a series of frequency response tests from 20Hz to 16,000Hz with all microphones (M1 to M7). The frequency response test allowed us to measure the inherent characteristics of each microphone. This is a common test to help engineering build microphones according to certain specifications and to classify them based on what they are best at recording. For our requirements, a higher and relatively flat and unaltered response in the low frequency range allows us to detect subtle sounds and indicates that no anomalies were introduced during the recording.

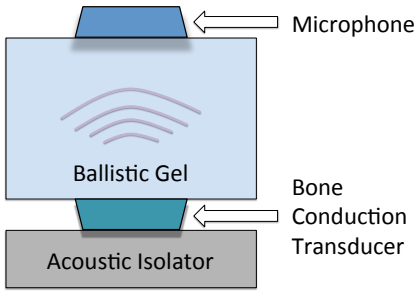


Figure 3: Frequency Response test setup, used to establish the sensitivity of each microphone from 20Hz to 16kHz

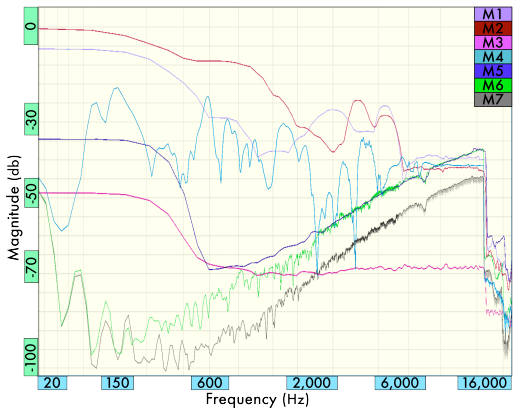


Figure 4: Frequency Response comparison of different microphones from 20Hz to 16kHz

We used a bone conducting transducer as our output device and created a sweeping tone that changed its frequency from 20 Hz to 16,000 Hz. An 8 x 8 x 5.5 centimeter block of ballistic gel was placed on top of the bone conducting transducer. The ballistic gel block is a standard proxy of human flesh or muscle because of its similarity in acoustic properties (e.g. speech of sound, density, etc.). We firmly attached different microphones to the other side of the ballistic gel block. Figure 3 shows the setup of the frequency response test. We ran this experiment for all the seven different microphones listed in table 1.

Figure 4 shows the frequency responses of different microphones. Our results indicate that with a constant gain, M1, M2, and M3 are the most sensitive below 700Hz. M3 maintained the flattest response, but lower than that of M1's and M2's. The most inconsistent response pattern was found in M4, which showed significant peaks and drop-offs at seemingly random intervals along the frequency axis. M6 and M7 have similar response patterns. M5's response was mostly flat under 600Hz, but it showed similar trends to M6 and M7 above 600Hz. Above 700Hz, M1-M5 had similar response patterns though the magnitude of M5's response was significantly lower. Unlike other microphones, we found a very irregular oscillating frequency response for M6 and M7, which is also considerably lower in the lower part of the frequency range (below 7000 Hz). One explanation of this phenomenon is that most of the off-the-shelf microphones (M6 and M7) are designed for recording speech; thus, they are not optimized for body sounds that lie in relatively lower part of the frequency spectrum. As most of our targeted non-speech body sounds are in a lower part of the frequency range, the frequency response of M2 suggests that it is the most appropriate microphone for capturing subtle body sounds.

3.2.2 *External Noise Test*

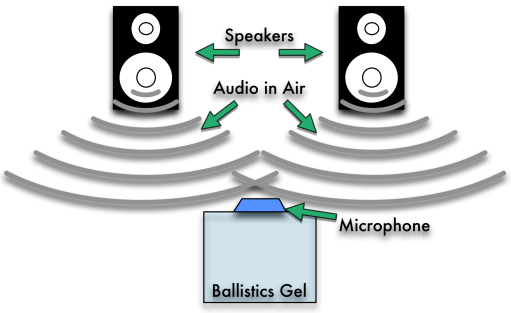


Figure 5: External noise test setup

The external noise test was performed to compare the microphone's robustness against any external or ambient noise. Four prerecorded external noises were played through two speakers to recreate the scenarios in this experiment. These sounds included: white noise, social noise (recorded in a restaurant), traffic noise (recorded in an intersection of a highway), and conversational noise (recorded while another person was talking). For this test, each microphone was positioned over the ballistic gel so that the element was facing the gel and the speakers were facing the back of the microphone. The different recordings were played through the speakers (i.e., audio in air), approximately one meter above the microphone. Figure 5 illustrates the setup of the external noise test.

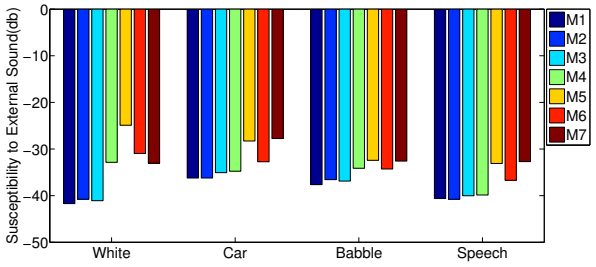


Figure 6: Comparison of different microphone's susceptibility to different type of external sounds or ambient noises

We measure susceptibility (in db) using equation 1, where $Power_{mic}$ is the power of the signal recorded by the microphone and $Power_{speaker}$ is the power emitted from the speaker. We

used the standard Root Mean Square (RMS) metric to measure the power. Figure 6 illustrates the susceptibility of different microphones under different types of external sounds. The smaller value of the susceptibility metric of the custom-built M1, M2 and M3 shows that they are more sound proof against external sounds. M5 turned out to be the least robust against external noise. The two off-the-shelf microphones (M6 and M7) were less robust against external sound than M1-M3.

$$Susceptibility = 10 * \log(\frac{Power_{mic}}{Power_{speaker}}) \quad (1)$$

Based on the frequency response test and the external noise test, we found our custom-built microphone, M2, to be the optimal microphone. While the external noise test was better for M1 than M2, the overall frequency response of M2 was consistently higher in magnitude, up to approximately 2000Hz. The difference in external noise was much less significant than the difference in frequency response between M1 and M2. The construction of these two microphones was identical except for one feature: the diaphragm of M1 was covered with a thin piece of latex, while the diaphragm of M2 was covered with a thin piece of silicone. This leads us to the conclusion that latex is mildly better at preventing external noise than silicone, but silicone is much better at transferring vibration below 2000Hz than latex. Therefore, we selected M2 for the BodyBeat microphone, as it is very insensitive to external sounds and highly sensitive to any sound generated inside the body (including speech).

3.3 Microphone Position Test

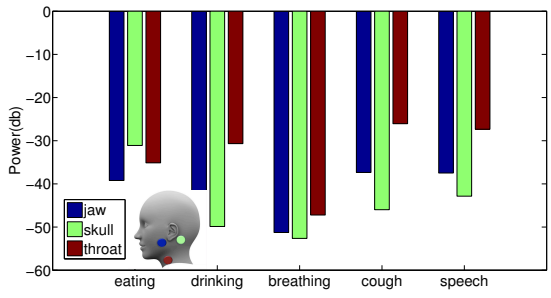


Figure 7: Comparing different body positions (jaw, skull, and throat) for capturing different types of body sounds

We conducted a microphone position test to find the optimal position to place the custom microphone (M2) in order to enable it to capture a wide range of body sounds. This test consisted of two parameters: the first being body position (jaw, skull, & throat) and the second being body sounds (eating, drinking, breathing, coughing, & speech). We recorded the five types of body sounds with M2 in each of the three body positions, and we then compared the power of the captured signals across different body positions. Figure 7 illustrates the power ($10log(P)$ in decibel unit) of the signals captured at different body positions.

Among the three locations, the throat gives us the maximum power (db) for all types of non-vocal body sounds, except eating. The power of the captured eating sounds was similar in all three locations. However, the eating sound captured in the skull contained slightly higher power than that captured in other positions. This is likely because the eating sound can very easily propagate through the teeth and then through the jaw to the skull. Considering our goal of capturing the wide range of body sound classes, the throat is the right location for the BodyBeat microphone.

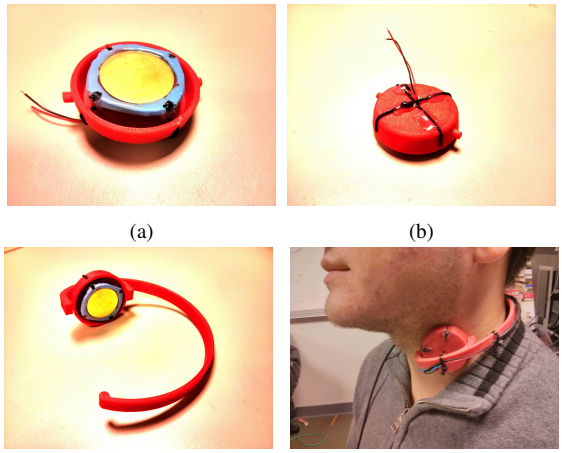


Figure 8: (a) Microphone attached to suspension mechanism (front view), (b) Microphone attached to suspension (back view), (c) 3D printed neck band, (d) Neck band, suspension capsule, and microphone fully assembled, (e) User wearing the fully assembled system

3.4 Neckpiece Design

To capture a wide range of non-speech body sounds from the throat area, we designed a neckpiece to securely attach the custom-made microphone to the throat area. In order to handle users' daily interactions and maintain performance, we also considered friction noise when designing the neckpiece. Human body movements generate noise due to the friction between the silicone diaphragm and the skin. We maintained usability by adopting a suspension mechanism, which allows the microphone's position to be partially independent of the neckpiece. In other words, the microphone remains in place and firmly attached to the neckpiece even when moving, thus minimizing friction noise. Figures 8a and 8b illustrate the top and bottom view of the microphone attached to the suspension capsule.

The microphone is attached to the suspension capsule with four elastic strings (approximately 1mm in diameter). The suspension allows for approximately four millimeters of movement on all sides and four millimeters of vertical movement (for a total of eight millimeters of movement on all three axes). Figure 8c shows the 3D printed neck band. The suspension capsule is attached to the neck band by placing the two cylindrical knobs into the corresponding holes on the two small, inward pointing extensions on the neck band. The band is flexible, which allows for the capsule to be easily placed in (or taken out) and still be tightly attached to the neck band (Figure 9). This design also allows the suspension capsule and microphone to pivot on the horizontal axis, allowing users to adjust for comfort. In figure 8, the current BodyBeat wearable system is still relatively big in size, which may cause some wearability issues. we will iteratively improve the design of *BodyBeat*. We will also look for opportunities to integrate *BodyBeat* into promising wearable systems (such as Google Glass) to enhance wearability.

4. CLASSIFICATION ALGORITHM

4.1 Data Collection

We recruited 14 participants (5 females) with different heights and weights to collect a wide range of body sounds. The partici-

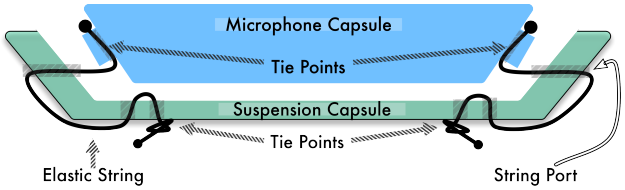


Figure 9: Microphone and Suspension Capsules

Index	Non-Speech Body Sound	Description
1	Eating	Eat a crunchy cookie
2	Eating	Eat an apple
3	Eating	Eat a piece of bread
4	Eating	Eat a banana
5	Drinking	Drink water
6	Deep Breathing	Deeply breath
7	Clearing Throat	Clear your throat
8	Coughing	Cough
9	Sniffling	Sniffle
10	Laugh	Laugh aloud
Index	**Other Sounds**	**Description**
11	Silence	Take a moment to relax
12	Speech	Tell us about yourself

Table 2: The list of non-speech body sounds and other sounds collected in this work

pants are asked to wear the BodyBeat neckpiece and to adjust the position of the microphone so that it is placed beside the vocal chord. The types of body sounds and a short description of each task are listed in table 2. We also collected silence and human speech sounds. Since our primary focus is detecting non-speech body sounds, we treat silence and human speech sounds as sounds that our classification algorithm should be able to recognize them and filter them out. During data collection, all body sounds were recorded with a sampling rate of 8kHz and a resolution of 16-bit. In total, each of our participants contributed approximately 15 minutes of recordings.

To examine the acoustic characteristics of the collected body sounds, we plot their corresponding spectrograms in Figure 10. Spectrogram illustrates a visual representation of the frequency spectrum in a sound as it varies with time. As a comparison, the spectrograms of both silence and speech are also incorporated. As expected, silence spectrogram contains almost no energy throughout the duration of the recording. On the other hand, the spectrogram of speech contains significantly more energy due to the vibration of vocal fold during speech utterances. Among non-speech body sounds, the swallowing sound during drinking generates a distinct frequency pattern. Coughing sound generates two harmonic frequencies following a particular time varying pattern in the spectrogram. When eating crispy hard foods (like chips), chewing is much more pronounced and visible in the spectrogram than that of soft food like bread. The frequency response of deep breathing is much more powerful than that of normal breathing, although both of the breathing variants follow similar trend (in terms of changes of frequency distributino over time). Lastly, the two spectrograms of eating soft food (bread) and normal breathing (in Figure 10) follow a very similar trend.

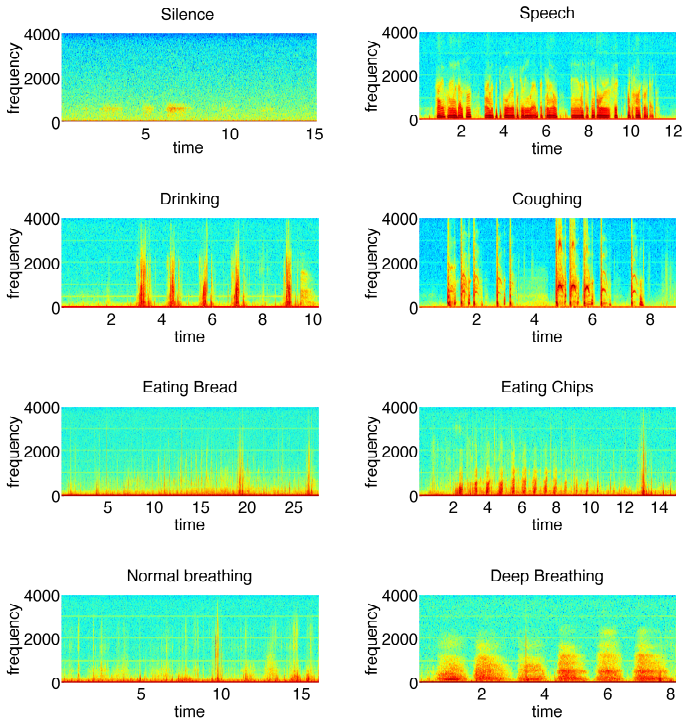


Figure 10: The spectrograms of silence, speech, and non-speech body sounds

4.2 Feature Extraction

The raw audio data sampled from the microphone was first segmented into frames with uniform length and 50% overlap. The length of the segmented frame is critical for the classification procedure that follows. In this work, we considered the frame length in the range from 16ms to 256ms. The optimal frame length is determined empirically based on the classification performance.

To characterize body sounds, we employed a two-step feature extraction procedure. In the first step, we extract a number of acoustic features from each frame to construct frame-level features. Acoustic features for analyzing human speech have been studied extensively in the past decades. However, limited research has been done to interpret non-speech body sounds. Therefore, in this work, we include a standard set of acoustic features used in human speech analysis and a number of other features that have been demonstrated to perform well in capturing paralinguistic features of vocal sounds. Table 3 lists all the frame-level features and their corresponding acronyms. Specifically, the frame-level features include 8 sub-band power features, RMS energy, zero crossing rate (ZCR), 9 spectral features, 12 Mel Frequency Cepstral Coefficients (MFCCs). Let us consider that the sampling frequency is f_s (8000 hz). Now for extracting the 8 log subband power features, we divide the spectrum into 8 subbands respectively having the following frequency ranges (0, f_s/256), (f_s/256, f_s/128), (f_s/128, f_s/64), (f_s/64, f_s/32), (f_s/32, f_s/16), (f_s/16, f_s/8), (f_s/8, f_s/4), (f_s/4, f_s/2). The first sub-band power represents the total power in a very small frequency region from 0 to 31.25 Hz. From the second sub-band, the bandwidth of each sub-band is twice as much as that of the former sub-band. The logarithm (base 10) is applied to represent the power of each sub-band in a bel scale. The spectral features are used to characterize different

Group	Frame level descriptors	Acronym
Energy	log power of 8 subbands	LogSubband[i]
	Total RMS Energy	RMSenergy
Spectral	Spectral Centroid	SpectCent
	Spectral Flux	SpectFlux
	Spectral Variance	SpectVar
	Spectral Skewness	SpectSkew
	Spectral Kurtosis	SpectKurt
	Spectral Slope	SpectSlope
	Spectral Rolloff 25%	SpectROff25
	Spectral Rolloff 50%	SpectROff50
	Spectral Rolloff 75%	SpectROff75
	Spectral Rolloff 90%	SpectROff90
Crossing Rate	Zero Crossing Rate	ZCR
MFCC	12 Mel Frequency Cepstral Coefficients	mfcc[i]

Table 3: The list of frame-level features

Type	Statistical Functions	Acronym
Extremes	Minimum	min
	Maximum	max
Average	Mean	mean
	Root Mean Square	RMS
	Median	median
Quartiles	1st and 3rd Quartile	qrtl1, qrtl3
	Interquartile Range	iqrl
Moments	Standard Deviation	std
	Skewness	skew
	Kurtosis	kurt
Peaks	Number of peaks	numOfPeaks
	Mean Distance of Peaks	meanDistPeaks
	Mean Amplitude of Peaks	meanAmpPeaks
Rate of Change	Mean Crossing Rate	mcr
Shape	Linear Regression Slope	slope

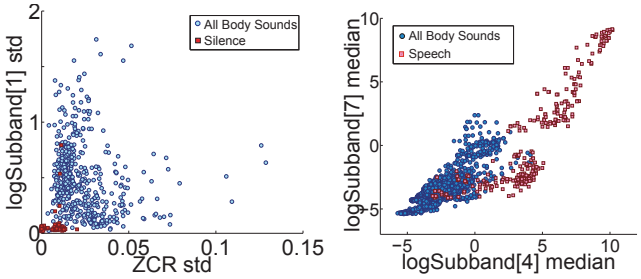
Table 4: The list of statistical functions applied to the frame-level features for extracting window-level features

aspects of spectra including the 'center of mass' (spectral centroid), 'change of spectra' (spectral flux), 'variance of the frequency' (spectral variance), 'skewness of the spectral distribution' (spectral skewness), 'the shape of spectra' (spectral slope, spectral rolloffs) etc. Lastly, MFCC coefficients capture the Cepstral coefficients using the source vocal tract model in speech signal processing.

Based on those extracted frame-level features, we grouped frames into windows with much longer duration and extract features at the window-level. We considered the window length in the range of 1s–5s, also determined empirically based on the classification performance. To extract window-level features, we applied a set of statistical functions across all the frame-level features within each window. Table 4 lists all the statistical functions applied to the frame-level features within the window to capture different aspect of the frame-level features. Specifically, the window-level features capture the averages, extremes, rate of change, and shape of the frame-level features within each window. For example, one window-level feature is the mean value of the zero crossing rates (ZCR) in frames, which is measured by at first estimating the ZCR of individual frames and then calculating the arithmetic mean value across all the ZCRs in a particular window. In total, we extracted 512 window-level features.

4.3 Feature Selection

The two-step feature extraction in the last section generates a total of 512 features. Since we are going to implement the overall feature extraction and classification framework on resource limited smartphone and wearable platform, it is not computationally efficient to include all these features. Therefore, the goal is to select a minimum number of features that achieve reasonably good



(a) All Body Sounds vs. Silence (b) All Body Sounds vs. Speech

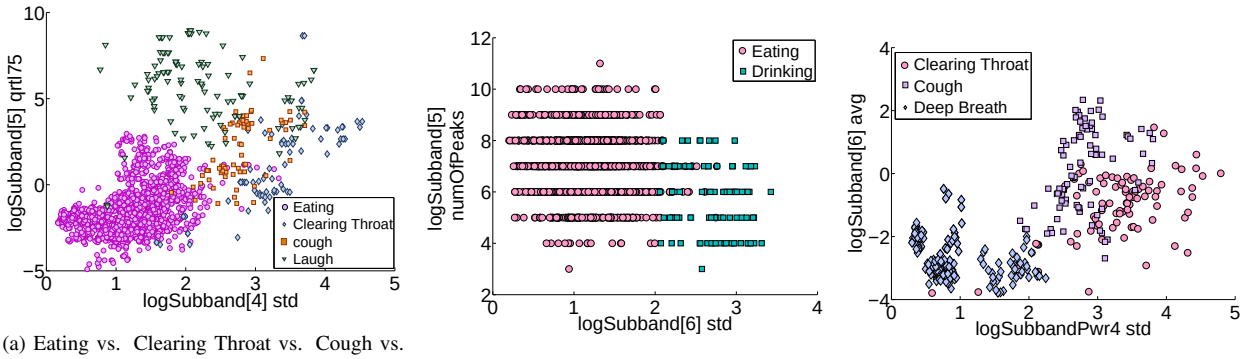
Figure 11: Scatter plots in 2D feature spaces

classification performance. In our work, we use the *correlation feature selection* (CFS) algorithm to select the subset of features [13]. In general, the CFS algorithm evaluates the goodness of features based on two criteria. First, the feature is highly indicative of the target class. Second, the new feature select must be highly uncorrelated with the features already selected. We used CFS algorithm to select a set of 30 features.

From these 30 features we further select the most optimized feature set for the target classifier. To do this, we run a sequential forward feature selection algorithm with the classifier's performance as criteria to select the top N best features. As a classifier we used Linear Discriminative Classifier which will be explained in further detail in Section 4.4. The best features selected includes logSubband[1] std, logSubband[4] median, specQrtl25 min, logSubband[4] std, logSubband[5] qrtl75, logSubband[5] numOfPeaks, ZCR std, logSubband[6] std, logSubband[6] mean, specRoff50 meanCrossingRate, and logSubband[7] median.

To show the performance of these selected features, a series of scatter plots in 2D feature space are shown in Figure 11 and 12. First, Figure 11a shows the scatter plot of silence and all the body sound classes with respect to the two features: the standard deviation of zero crossing rate and the log of first sub-band power (logSubband[1] std). Silence typically consisted of low energy random signal. The signal's zero crossing rate and the logSubband[1] does not vary much across frames. Thus, using these two features, we can discriminate all body sounds from silence. Figure 11b shows speech and all the body sounds in the feature space of logSubband[4] median and logSubband[7] median. As illustrated, speech signals contain much higher power in both of the sub-bands. Thus, using these two features, we can discriminate speech from all the body sounds considered for this study.

Figure 12 shows the difference among different body sounds in different pairs of selected features. Figure 12a indicates that eating sounds are fairly different from cough, laughter, and clearing the throat in the 2 dimensional feature space of logSubband[4] std and logSubband[5] qrtl75. Both of the features have the low values for eating sounds. The 5th sub-band laughter contains slightly higher energy than others. Figure 12b shows that the most discriminative feature for separating eating from drinking is logSubband[6] std. It means that the 6th sub-band's log energy varies more for the drinking sound than that of eating sounds. Figure 12c shows that deep breathing sounds contain lower energy in 6th sub-band. The standard deviation of the 4th sub-band's log energy also varies much less for deep breathing sounds compared to cough and clearing throat.



(a) Eating vs. Clearing Throat vs. Cough vs. Laugh

(b) Eating vs. Drinking

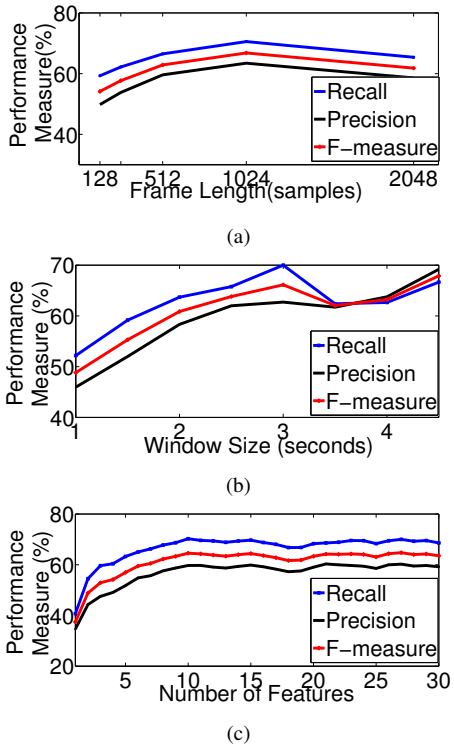
(c) Clearing Throat vs. Cough vs. Deep Breath

Figure 12: Scatter plots in 2D feature spaces

4.4 Classification

We use Linear Discriminant Classifier (LDC) as the classification algorithm. We chose LDC over other classification algorithms such as Support Vector Machine (SVM), Gaussian Mixture Model (GMM) and Adaboost because LDC is also computationally efficient and lightweight enough to be implemented in resource-limited smartphone. Table 5 shows the results of different classifiers with different feature sets. We used two different cross-validation experiments: a Leave-One-Person-Out (LOPO) and a Leave-One-Sample-Out (LOSO) cross-validation experiment. The LOPO cross-validation results are the most unbiased estimate of our classifier's performance, when the classifier is asked to detect the body sounds of a new person that the classifier has not seen before. In contrast, the LOSO cross-validation assumes that the classifier is trained on the data collected from the target user. The performance results from the LOSO cross-validation can be thought as the ceiling performance of the system. The best performance is achieved using LDC and energy, spectral features, and MFCC is used to extract the initial set of window-level features for selecting the top window-level features. The performance reaches to 72.5 (average recall) and 63.4 (precision). Table 5 also shows that with only energy and spectral features as frame-level features, the LDC classifier can get a nice performance, which is 71.2 (recall), 61.5 (precision), and 66.5 (accuracy) from the LOPO experiment. Moreover, if a user contributes some training data towards making the classifier, the performance measure reaches to 88.1% recall (from LOSO experiment). Notice that losing MFCC from our frame-level feature set does not affect the classifier's performance much (absolute reduction in terms of recall is 1.3 %), but if we don't have to extract MFCC features, that indicates that we could save a lot of system's resource in terms of power [22], speed, and memory. Considering this factor, we decided to use just energy and spectral features as frame-level features with LDC as classifier for the rest of our analysis and system implementation. Lastly, we also build the classification algorithm used by a recent study [30] to compare with our proposed BodyBeat classification algorithm. We find that our system outperforms BodyScope [30]. Lastly, table 6 shows the class level recall and precision from the LOPO experiment this classifier.

The choice of both the frame and window size length used to extract features significantly impacts classification performance. A coarse frame or window size may not capture the local dynamic (time variant) properties of the body sounds. On the other hand a very fine frame or window may be prone to noise and thus may decrease the discriminative properties of the features. We run this analysis to find the optimal frame and window size. Figure 13a and 13b shows the impact of the frame and window size on the classifier's performance. The frame size of 1024 samples (125 milliseconds) and the window size of 3 seconds maximize the classifier's performance. The number of features selected using the feature selection also plays a very important role on the performance measure of the classifier. Figure 13c shows that the performance measures in terms of recall, precision and F-measure saturates when we use 10 window-level features.



(a)

(b)

(c)

Figure 13: shows the impact of (a) frame size (b) window size (c) total number of features on the performance of classifier

5. SYSTEM IMPLEMENTATION

The BodyBeat non-speech body sound sensing mobile system is implemented using an embedded system unit and an Android application unit. The custom-made microphone of BodyBeat system is

Frame-level Features	LOPO			LOSO		
	R	P	F	R	P	F
Energy & Spectral	71.2	61.5	66.5	88.1	81.9	86.5
MFCC	66.3	52.8	57.8	75.0	71.5	73.2
Energy & Spectral & MFCC	72.5	63.4	67.6	90.3	82.3	86.6
BodyScope[30]	57.6	55.5	56.5	76.6	71.5	73.8

Table 5: Classification performance in terms of Recall (R), Precision (P) and F-measure (F) based on both Leave-One-Person-Out (LOPO) and a Leave-One-Sample-Out (LOSO) cross-validation

	Eating	Drinking	Deep Breathing	Clearing Throat	Coughing	Sniffling	Laugh	Silence	Speech
Recall	70.35	72.09	64.09	68.75	80.00	75.00	61.90	74.38	81.06
Precision	73.29	57.21	60.95	61.11	62.07	58.00	61.90	61.66	84.69

Table 6: The Recall and Precision for each class from the LOPO experiment using LDC as classifier and energy and spectral features as frame-level features

directly attached to the embedded system. The embedded system unit utilizes an ARM microcontroller unit, an audio codec and a Bluetooth module to implement capture, preprocessing and frame admission control of the raw acoustic data from the microphone. The Android application unit on the other hand implements the two stage feature extraction, and inference algorithm. These two units communicate with each other through Bluetooth. Figure 14 illustrates the system architecture of the overall system. In what follows, we present the system implementation details of both the embedded system unit and the Android application unit.

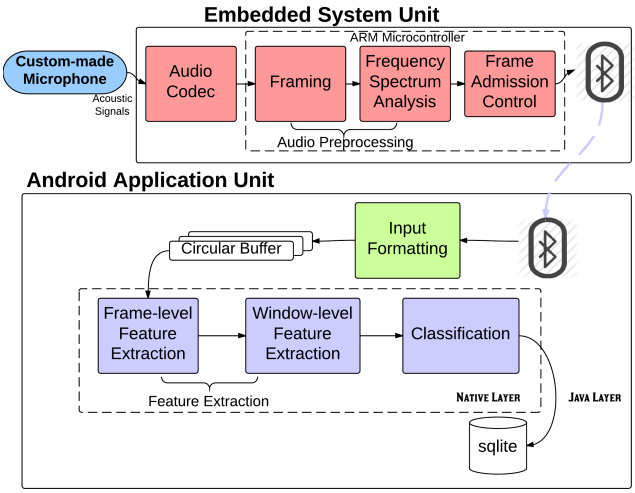


Figure 14: The Block Diagram of BodyBeat System Architecture

5.1 Embedded System Unit

At the center of the embedded system unit, we used a commercially available Maple ARM microcontroller [4]. The board consists of a 72MHz ARM cortex M3 chip with most of the standard peripherals including digital and analog input/output pins, 1 USB, and 3 Universal Asynchronous Receiver/Transmitters (UARTs), and Serial Peripheral Interface (SPI). The clock speed, advanced peripherals, and interrupt capabilities enables us to do some rudimentary real-time audio signal processing and at the same time to drive a Bluetooth codec to communicate with the Android application unit.

As seen in Figure 14, the ARM microcontroller connects to an audio codec via SPI [6]. The audio codec ([6]) contains a Wolfson WM8731 chip. The audio codec receives the analog audio signal using a 1/8 inch input jack and samples the audio with an array sampling frequency up to 88000 Hz and with a resolution up to 24 bit/sample. The ARM unit is also connected with a class 2 Bluetooth ratio modem (commercially called BlueSMiRF Silver [5]). The Bluetooth modem contains the RN-42 chip that receives data from the ARM unit via UART and sends data to the Android application with an SPP profile with a data rate of 115000bps. The Bluetooth modem ensures reliable wireless connectivity with the Android device up to a distance of 18 meters. A rechargeable LiPo battery is used to power the ARM microcontroller, including the audio codec and Bluetooth modem. Figure 15 shows the prototype of embedded unit.

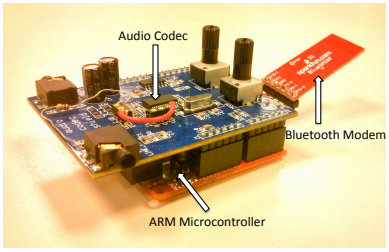


Figure 15: The ARM Micro-Controller Unit

5.1.1 *Audio Preprocessing*

Audio preprocessing is the first step that happens in the ARM microcontroller, which receives the digital samples of the BodyBeat microphone's analog audio stream by the Audio Codec. The sampling frequency and bit resolution are chosen to be 8000Hz and 16 bit, respectively, as it provides us with a detailed picture of the audio and lowers the computational load of the system at the same time. As the Audio Codec samples the analog audio signal and sends a digital signal to the ARM microcontroller unit via SPI, the interrupt in the ARM microcontroller unit collects the data in a circular buffer. The audio data stored in the circular buffer is then segmented with a frame length of 1024 samples (125 milliseconds). While the interrupt fills the circular buffer, the main thread essentially checks continuously if another 1024 samples has filled the circular buffer. Upon detecting the arrival of a new frame, the ARM unit stars a RADIX-4 Complex Fast

Fourier Transformation (FFT) implementation, which is written in C language [2]. The FFT implementation uses fixed-point arithmetic with a sine table for optimizing speed by sacrificing some memory. To prevent an arithmetic overflow, fixed scaling is employed.

5.1.2 Frame Admission Control

The ARM microcontroller also does a frame admission control to filter out audio frames that do not contain any body sounds. After getting the FFT of the Hanning windowed audio frame of 1024 samples, we extracted a few important sub-band power and zero crossing rate features to detect the presence of speech and silence. In Figure 11, we already demonstrated how with a few features we can filter out frames containing silence and speech. We took a few measures to optimize our implementation in this regard. For example, one of the features that we implemented in our ARM microcontroller is logSubband[4] median. Floating point logarithm calculation is heavy in terms of both CPU. We used a log table to lower the CPU requirements by sacrificing some memory. When a certain frame is detected not to contain any silence or speech, the ARM microcontroller transfers the power spectrum of the current frame to the Android unit. To asynchronously transfer different frames, we send a preamble to mark the start of a frame.

5.2 Android Application Unit

The Android application unit, which is approximately 2200 lines of Java, C, and C++ code, includes the input formatting of the data, which is followed by feature extraction and classification. The android unit implements a feature extraction and classification algorithm in the native layer using C and C++ for faster execution. The complete binary package including resource files is approximately 1280 KB.

5.2.1 Input formatting

The Android unit receives data via Bluetooth from the embedded system unit as shown in Figure 14. This module used the Android Bluetooth APIs to scan for other Bluetooth devices around the phone, to fetch the information of the paired (or already authenticated) remote Bluetooth modem in the embedded system unit, and to establish wireless communication channel. The Android application receives each frame asynchronously from the embedded unit. The Android Bluetooth adapter continuously looks for a four byte long preamble, which indicates the start of a new frame is being sent by the embedded system unit. Upon receiving the preamble, the input processing module continuously stores all the received data in a temporary buffer. As soon as the temporary buffer is full (received 513 samples, each 16 bit), the input processing module takes all the data of the current frame from the temporary buffer and updates a two dimensional circular buffer. At the same time, the input processing unit starts to look for another preamble indicating the start of another frame. This preamble helps the Android application unit to receive each frame of data separately. The two dimensional circular buffer is shared by both the producer thread and the consumer thread as data storage and data source. The two dimensional circular buffer stores each frame's data (513 samples) in a row. Thus, consecutive frame data is stored in different rows in the two-dimensional circular buffer. All the work in input processing happens in producer thread. To facilitate the two dimensional circular buffer sharing by the two threads, it includes two separate pointers for the two threads (producer and consumer) at different rows of the two dimensional circular buffer.

5.2.2 Feature Extraction and Classification

Once the two dimensional circular buffer contains 24 frames of data (window length 3 seconds) for the feature extraction and inference, the consumer thread passes the data to the native layer. To ensure 50% overlap between two consecutive windows, the consumer thread's pointer moves to 16 rows to point to the new frame. The entire feature extraction and classification algorithm is implemented in the native layer considering the speed requirements for real-time passive body sound sensing. Section 4 gives the detailed description of the discriminative features for body sound classification. The frame-level features are first extracted from frame-level data. We used various statistical functions to extract window-level features at this stage. The window-level features are then used to infer the body sound. While implementing the feature extraction and classification, we took several measures to optimize power, CPU, and memory usage. We used additional memory for lowering CPU load. All the memory blocks are pre-allocated during the initialization of the Android application unit and are shared across multiple native layer calls.

5.3 System Evaluation

In this section, we present the system evaluation of the BodyBeat system. We first discuss CPU and memory benchmarks, which is then followed by the detailed time and power benchmarks, including both the embedded system unit and the Android application unit. All the measurement of the Android application unit is done with Google Nexus 4.

5.3.1 CPU and Memory Benchmarks

Status	CPU Usage	Memory Usage
Silence or speech	8-12%	45MB
Body Sound	15-22%	47MB

Table 7: CPU and Memory Benchmarks of the Android Application Unit

Table 7 shows the CPU and memory benchmarks of our system. When the BodyBeat microphone captures either silence or speech, the Android application unit consumes less than 12% of the CPU and 45 MB of memory, because of embedded system's frame admission control. During the presence of body sounds, the CPU and memory usage increases and reaches up to 22% and 47 MB.

5.3.2 Time and Power Benchmarks

Figure 8 shows the average running time of different routines in both the embedded system unit and Android application unit for processing 3 seconds of audio from the BodyBeat microphone that contains some body sound. In the embedded unit, the first routine forms a frame of 1024 samples and multiplies it with the Hanning windowing function to compensate Gibbs phenomena. The framing only takes 5 milliseconds where the next process Fast Fourier Transformation (FFT) takes 80 milliseconds. The frame admission control takes up to 20 milliseconds.

The input processing in the Android application unit takes the most of the time, as it includes the delay due to Bluetooth communication. The feature extraction passes each frame (power spectra received via Bluetooth, length 513 data) in the window to the native layer to extract frame-level features. The frame-level feature extraction takes a moderate amount of time, as this is one of the most heavy routine in Android application unit. Lastly, the window-level feature and classification takes only 5 and 1.5 milliseconds to run.

Unit	Routine	Time (ms)
Embedded	Framing	5
	FFT	80
	Frame admission control	20
Android	Input Processing	2448
	Frame-level feature extraction	84
	Window-level feature extraction	5
	Classification	1.5

Table 8: Average running time of different routines in the ARM microcontroller unit and the Android application unit to process 3 seconds (one window) of audio data containing some body sound

Routine	Average Power (milliWatt)
Input Processing (IP)	343.74
IP & Feature Extraction (FE)	362.84
IP & FE & Classification	374.49

Table 9: The Power benchmarking of Android app unit

The embedded system unit consumes 256.64 milliwatt(mW) when the system is waiting to be paired and connected with an Android system. The embedded system unit consumes about 333.3 mW power while the raw audio data contains valuable body sounds and the frame admission control allows the data to be transferred to the Android system unit. On the other hand, when frame admission control detects either silence or speech in the signal and stops transmission of the data to Android unit, the embedded system unit's power consumption decreases to 289.971mW. Table 9 illustrates the average power (in milliwatt unit) consumed by different routines of the Android application unit. The average power consumption by the Android application unit is about 374.49 mW, when the application unit runs all the routines (input processing, frame- and window-level feature extraction and classification).

6. POTENTIAL APPLICATIONS

An increasing number of mobile systems are bringing health sensing to the masses. In this regard, we were inspired to build a mobile system for sensing a wide range of non-speech body sounds. By listening to the internal sounds that our bodies naturally produce, we can continuously sense many medical and behavioral problems in a wearable form factor. Here, we highlight some future applications that can be developed with our BodyBeat sensor:

6.1 Food Journaling

Since BodyBeat can recognize eating and drinking sounds, it has the potential to be used in food journaling applications. Despite technological advancements, developing automatic (or semi-automatic) systems for food journaling is very challenging. For example, the PlateMate [24] system demonstrated the feasibility of using Amazon Mechanical Turk to label photographs of users' meals with caloric information. However, this system required that users actually remember to take a photo of what they eat. With the invention of BodyBeat, you can imagine a future system that detects when a user is eating. The system then either automatically takes a picture of their food with a life-logging camera (e.g. Microsoft SenseCam, Google Glass), or simply reminds the user to take a photo of their food. Lastly, it uploads the image to Mechanical Turk for caloric labeling.

6.2 Illness Detection

The BodyBeat system allows us to detect coughing, deep or heavy breathing, which can be indicative of many pulmonary diseases. While a few previous studies have illustrated success detecting these body sounds indicative of illness (e.g. [17, 30]), BodyBeat mobile system can be used in an application which will detect the onset, frequency, and the location of coughing, heavy breathing or any other kind of pulmonary sounds. As sensing devices become more ubiquitous, cough detection could allow us to track the spread of illnesses, with similar motivation to TwitterHealth research [27]. In future work, we plan to work with medical doctors to expand the BodyBeat system to detect other body sounds of interest, such as sneezing and specific types of coughing (e.g., wheezing, dry cough, productive cough).

7. RELATED WORK

The microphone is a rich sensor stream that contains information about our surroundings and us. Many studies proposed mobile systems that leverage the smartphone's built-in microphone to infer the surroundings of a person [21], physiological state (sleep [14], cough [17]), and psychological state (stress [20]). A recent study proposed Aditeur, a mobile system that detects audio events in real-time using the phone's built-in microphone, backed by a cloud service[22]. Such mobile systems are becoming very popular in the mobile health domain, especially among clinicians and patients for detection of disease, monitoring of health variables, etc. [16]. However, the smartphone's built-in microphone (typically a condenser) is actually optimized for capturing speech. It is difficult to capture subtle non-speech body sounds because of the placement of the smartphone and external noise levels. A few recent studies proposed contact microphone designs. Hirahara et al. proposed a customized microphone with Eurathane elastomer designed to capture non-audible murmur, which is a very weak whispered voice [15]. Various contact microphones are also used in music industry that are designed to directly capture vibrations from musical instruments [1]. None of these contact microphones are not designed and optimized to capture subtle body sounds.

Yatani and Khai [30] proposed BodyScope, a wearable neckpiece with a standard condenser microphone augmented with a stethoscope head, in order to capture an array of body sounds to predict activity. In our study, we designed and implemented a customized microphone based on piezo-electric sensor that are optimized for subtle body sounds. We also propose a neckpiece that is designed with a consideration on the microphone's longer-term wearability and users' comfort. The neckpiece also employ a suspension mechanism to compensate friction noise due to user's body movement. Body sounds are a fundamental source of health information and are being used by physicians since almost the beginning of modern medical science [18]. Due to the subtle nature of body sounds, it is difficult to reliably and passively capture body sound signals with a built-in smartphone microphone. As a result, some studied have explored the feasibility of a customized-wearable microphone for recognizing eating behaviors, breathing patterns, etc. Amft and Troster [10] used a combination of sensors to model eating behavior. They modeled hand gestures with inertial sensors in users' hands; chewing with ear attached microphones; and swallowing with rubber elongation sensors in the throat areas. Recent studies also tried to detect different eating sounds collected from a tri-axial accelerometer implanted inside teeth and using textile neckband [19, 11]. The scalability of this approach might be significantly constricted because of the sensor placement, which is close to oral activity. Another study used the built-in microphone of a hearing aid to recognize chewing events [25]. Some studies have used off-the-shelf bone conduction and condenser microphones to capture audio of eating and then classify the type of food consume and mastication counts [9, 8, 28]. Body vibration captured by

inertial sensors have also been used to estimate heartbeat [23]. The recent studies in the literature are mostly confined to offline exploration of discriminative acoustic features and machine learning. In addition, most studies did not provide information about the implementation of the signal processing and machine learning algorithms in resource-limited hardware. In our work, we presented the implementation of our signal processing and machine learning algorithms in the context of a distributed system, consisting of an ARM micro-controller and an Android phone. We compared our algorithms to baseline algorithms, and we also presented the results from the CPU, memory, and power benchmarking experiments.

8. CONCLUSION AND FUTURE WORK

In this paper, we presented the design, implementation and evaluation of BodyBeat, a wearable sensing system that captures and recognizes non-speech body sounds. We have described the design of our custom-built piezoelectric sensor-based microphone and showed that our microphone outperforms other existing solutions in capturing non-speech body sounds. In addition, we have developed a classification algorithm based on a set of carefully selected features and achieved an average classification average recall of 71.2%. Finally, we have implemented BodyBeat and benchmarked its performance.

In the future, we plan to reduce the form factor of BodyBeat to improve its wearability and minimize its obtrusiveness to users. We also want to keep exploring the potential of BodyBeat on detecting other non-speech body sounds. We also aim to process non-speech body sounds at a lower sampling rate and run the end-to-end evaluation of the system. Finally, based on the promising results reported in this paper, we plan to bring BodyBeat to life by deploying it in the aforementioned applications.

9. ACKNOWLEDGMENTS

This work was partially supported by NSF IIS#1202141 and Intel Science and Technology Center for Pervasive computing.

10. REFERENCES

[1] http://contactmicrophones.com/index.html.

[2] http://coolarduino.wordpress.com/2012/03/24/radix-4-fft-integer-math/.

[3] http://invisio.com/products/headsets/invisio-m3.aspx.

[4] http://leaflabs.com.

[5] https://www.sparkfun.com/products/10269.

[6] http://www.openmusiclabs.com/projects/codec-shield/.

[7] http://www.temco-j.co.jp.

[8] Amft, O., Kusserow, M., and Troster, G. Bite weight prediction from acoustic recognition of chewing. *IEEE Transactions on Biomedical Engineering 56*, 6 (June 2009), 1663–1672.

[9] Amft, O., StÃd'ger, M., and TrÃűster, G. Analysis of chewing sounds for dietary monitoring. In *UbiComp '05* (2005), 56–72.

[10] Amft, O., and Troster, G. On-body sensing solutions for automatic dietary monitoring. *IEEE Pervasive Computing 8*, 2 (2009), 62–70.

[11] Cheng, J., Zhou, B., Kunze, K., Rheinländer, C., Wille, S., Wehn, N., Weppner, J., and Lukowicz, P. Activity recognition and nutrition monitoring in every day situations with a textile capacitive neckband. UbiComp '13 Adjunct (2013), 155–158.

[12] Dacremont, C. Spectral composition of eating sounds generated by crispy, crunchy and crackly foods. *Journal of Texture Studies 26*, 1 (1995), 27–43.

[13] Hall, M. A. Correlation-based Feature Selection for Machine Learning. *PhD Thesis* (April 1999).

[14] Hao, T., Xing, G., and Zhou, G. isleep: Unobtrusive sleep quality monitoring using smartphones. SenSys '13 (2013), 1–14.

[15] Hirahara, T., Shimizu, S., and Otani, M. Acoustic characteristics of non-audible murmur. *The Japan China Joint Conference of Acoustics 100* (2007), 4000.

[16] Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. A survey of mobile phone sensing. *Communications Magazine, IEEE 48*, 9 (2010), 140–150.

[17] Larson, E., Lee, T., Liu, S., Rosenfeld, M., and Patel, S. Accurate and privacy preserving cough sensing using a low-cost microphone. UbiComp '11 (2011), 375–384.

[18] Levine, S., and Harvey, W. *Clinical auscultation of the heart*. W. B. Saunders, 1959.

[19] Li, C.-Y., Chen, Y.-C., Chen, W.-J., Huang, P., and Chu, H.-H. Sensor-embedded teeth for oral activity recognition. ISWC '13 (2013), 41–44.

[20] Lu, H., Frauendorfer, D., Rabbi, M., Mast, M., Chittaranjan, G., Campbell, A., Gatica-Perez, D., and Choudhury, T. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. UbiComp '12 (2012), 351–360.

[21] Lu, H., Pan, W., Lane, N. D., Choudhury, T., and Campbell, A. T. Soundsense: Scalable sound sensing for people-centric applications on mobile phones. MobiSys '09 (2009), 165–178.

[22] Nirjon, S., Dickerson, R. F., Asare, P., Li, Q., Hong, D., Stankovic, J., Hu, P., Shen, G., and Jiang, X. Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones. MobiSys '13 (2013), 403–416.

[23] Nirjon, S., Dickerson, R. F., Li, Q., Asare, P., Stankovic, J., Hong, D., Zhang, B., Jiang, X., Shen, G., and Zhao, F. Musicalheart: A hearty way of listening to music. SenSys '12, ACM (2012), 43–56.

[24] Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Platemate: crowdsourcing nutritional analysis from food photographs. In *UIST '11* (Oct. 2011).

[25] Passler, S., and Fischer, W.-J. Evaluation of algorithms for chew event detection. BodyNets '12 (2012), 20–26.

[26] Reichert, S., Gass, R., Brandt, C., and Andrès, E. Analysis of respiratory sounds: state of the art. *Clinical medicine. Circulatory, respiratory and pulmonary medicine 2* (2008), 45.

[27] Sadilek, A., and Kautz, H. Modeling the impact of lifestyle on health at scale. In *WSDM '13* (2013), 637–646.

[28] Shuzo, M., Lopez, G., Takashima, T., Komori, S., Delaunay, J.-J., Yamada, I., Tatsuta, S., and Yanagimoto, S. Discrimination of eating habits with a wearable bone conduction sound recorder system. In *IEEE Sensors '09* (2009), 1666–1669.

[29] Whittaker, A., Lucas, M., Carter, R., and Anderson, K. Limitations in the use of median frequency for lung sound analysis. *Journal of Engineering in Medicine 214*, 3 (2000), 265–275.

[30] Yatani, K., and Truong, K. Bodyscope: A wearable acoustic sensor for activity recognition. UbiComp '12 (2012), 341–350.