1. INTRODUCTION

Although gestures and movement are a natural, everyday occurrence, it remains to be a complex event to interpret by modern day devices. Research has taken a step forward towards solving this problem by discovering alternative methods in an attempt to capture complex gestures. Usually, we interact with our devices by either physical touch or even voice. As these devices become more sophisticated and smaller in size, we should consider whether we could have these devices potentially do more than just simply react to our touch or voice. In this paper, we consider the possibility of interactive and subtle gesture recognition.

Specifically, we consider the ability to detect the gestures by emitting ultrasonic waves through the air using microphones and speakers from a typical mobile device. Through movement, these waves are altered, providing ways to interpret gestures. In this work, we particularly focus on finger tapping gestures. We ultimately envision a device that provides the user with the ability to wear it on their wrist, such as a watch. As the user wears the device, it would emit an ultrasonic wave (when prompted), which is then interpreted by its own microphones. This challenge is extended further in the case of finger gestures, where the only possible gesture movements that take place are in a close proximity to each other.

In this work, we describe finger gesture recognition using a microphone array. First we try to use the reflection of sound waves by implementing an acoustic pulse radar. After understanding we cannot obtain a fine resolution to detect a subtle gesture, we implement a logistic regression classifier using the features from Doppler shift for each microphone. After testing the classifier on different data sets, we conclude that the classifier can be trained successfully with simply a small number of samples for each person. We also find that a learned classifier from one person, does not generalize well to a different person.

2. RELATED WORK

There exists an extensive amount of work in past research experiments that focus on gesture recognition. We highlight some of the relevant previous work in this section.

Besides building an original device for the sole purposes of recognizing a particular gesture, most work has instead focused on making efficient use of ultrasonic sound waves. Sonar-based equipment can not only read movements, but can also be used to interpret prominent objects accurately. As seen in the work by [8], the ultrasonic software built in this work uses a laptop to detect users with near perfect accuracy.

Some previous attempts at making such a device include FingerMic [2], where they present a trainable wearable device that provides a way to detect finger gestures in situations like ‘hands-full’ computing. Gestures covered in this work include thumb, index and all the fingers at once. Similarly, there is another project that detects one-handed gesture recognition using ultrasonic Doppler sonar [5], this work presents a cheap device that uses ultrasonic sensors to recognize one-handed gestures. These gestures ranged from various movements of one hand, including moving left-to-right or moving the hand in clockwise direction.

Detecting movements has not only existed for entire hand movements, but also for large range body gestures. For example, in the Magic Carpet project [6], the work utilized a pair of Doppler radars to measure up-body kinematics (velocity, direction of motion, amount of motion) over a grid of piezoelectric wires on a carpet to detect foot posture. Using both the carpet and the sensors, they are able to detect movements of the performer. Although this is older work, recent trends generally gear towards detecting movement through smaller devices.

As seen in the work by Scratch Input [4], the project explores an alternative idea that depends on a finger scratch as the sound input. Specifically, they deploy a sensor that reads input when a finger is dragged across wood, fabric, or wall paint. They take advantage of the fact that dragging a fingernail over a particular surface actually produces a high frequency (greater than 3kHz). They utilize a modified stethoscope to detect the sounds, but claim that the sensor is sufficiently small to fit on most mobile devices.

In contrast to these previous approaches, our work mainly focuses on the ability to detect more fine-grained gestures through the air without the need for a surface or an overwhelming and distracting device on the hand. This work was primarily inspired by SoundWave [3], a project that leverages the speaker and microphone in-air gestures. This technique was able to ensure shifts through an inaudible tone at a high frequency of (18-22kHz). Through this tone, signals were captured for two-handed gestures as well as more complex gestures like double-tapping. As the technique was fairly robust, we pursue this idea further in this work.

3. HARDWARE SETTING

In order to accurately capture wave signals in an inaudible frequency range, we needed the right equipment. Our equipment consists of a series of eight electret microphone amplifiers specifically built for Arduino that can cap-
Figure 1: Experiment Setup

ture up to 20kHz frequency \[1\]. Arduino is a single-board microcontroller that provides the ability to build applications that require the need to interpret interactive objects or environments.

As an initial attempt, we used Arduino Duo to read signals from the microphones. However, the bandwidth from an Arduino Duo to a PC was not wide enough to handle real-time data from multiple microphones with a high sampling rate (44.1kHz). Additionally, even with one microphone, Arduino Duo could not capture evenly sampled signals: the device was simply not built for sampling data evenly.

To solve this problem, we used a DAQ to receive synchronized signals from the distributed microphone array. Specifically, we used a NI CB-68LPR DAQ acquired through the Electronic Engineering lab at University of Washington. With this tool, we were able to sample at 60kHz from eight microphones in real time. The DAQ also came with additional software called LabView Signal Express. When configured properly, we were able to acquire and record the signal into a file for further analysis.

As seen in Figure 1, the microphones are placed approximately 2cm away from each other. We used an iPhone 5s’ speaker to emit inaudible sound waves. The DAQ and Arduino Duo are in the right side of the image. We used Arduino Duo as a power supply, since the microphones give the best performance on Arduino devices with the 3.3V supply.

4. GATHERING GESTURE SAMPLES

Through each retrial of the experiment we made sure to find a stable and consistent experiment setting. We positioned the mics at the edge of the table and had each user place their right hand wrist leaning towards the table. In this way, we can simulate the microphones positioned at the wrist, while keeping the relative position between the hand and the microphones fixed. We initially had the user hover their hand above the microphone, but soon discovered that hovering typically resulted in noisy results since the user would more easily shift their hand in slightly different positions above the microphones each time. Figure 2 shows the final setting we used for the experiments.

Before each trial, we configured LabView to read an audio sampling of approximately 5 seconds. During this time frame, the user would naturally move one finger. Not all fingers moved towards the microphones. For example, the thumb, moved in a horizontal direction, nearly in parallel above the microphones. For the classification solution that we discuss in the next section, we increase the recording interval and aim to capture the same gesture many times. To analyze the data, we parse through the file through a Python script in order to interpret the movements.

5. APPROACH

We explore two different approaches to detect tapping gestures with our device setting. The two approaches are: detecting poses using an acoustic pulse radar and detecting movements by tracking down Doppler shifts. We then discuss how we can extract features from Doppler shift information to build a logistic regression classifier in order to detect which finger is tapped.

5.1 Pulse Radar Approach

Radar systems utilize electromagnetic signals reflected from a target in order to estimate the distance of an object. In an ultrasonic setting, a transmitter sends a short sound pulse. This signal is repeated throughout a given time period. Upon hitting an object, this signal is reflected to the receiver, which then interprets the reflections as the distance of nearby objects.

We pursued this idea by having the iPhone emit a short characteristic signal periodically. It emits two full cycles of 18kHz sine wave and pause for the rest, repeating this process 210 times a second. We explored captured signals through three different scenarios: no obstacles, an open palm, and a hand with its index finger bent. The reflected signal is observed after traveling the emitter, the reflector, and the receiver. Since each scenario gives different travel distances over multipaths, we expected the observed signals would differ. If we could differentiate the open palm pose and the finger bent pose, then we could classify tapping gestures by capturing hand pose for each moment.

In Figure 3, we display the pulse signal without a hand in front of the microphones. Keep in mind, the original pulse signal is about 7 samples long with a 60kHz sampling rate, so most of the visible waves in the images are as a result of reflections of the transmitted wave. The middle panel displays the result of the signal with a hand in front of the
Figure 3: Comparison of the pulse signal on ‘no obstacle’ vs ‘open palm’ pose.

Figure 4: Comparison of the pulse signal on ‘open palm’ pose and ‘index finger bent’ pose.

microphone array (positioned as an open palm). Finally, the bottom panel displays the difference between the two signals. The intuition is that the last panel should emphasize the differences between the two reflections. In this case, the differences can be visually seen on the signal.

In contrast, Figure 4 we compare the signals with a hand (as an open palm) in front of the microphones to the signals with a hand with its index finger down towards the microphones. In the last panel, the difference between these two signals is shown. It is not as clear as in the previous experiment. Specifically, it is challenging to differentiate between noise or whether there is a shift in hand position through this difference in signals. Thus, we pursued an alternative route that displayed changes in frequency more clearly.

Another downside of this approach is that the sound is still audible even though the emitted signal is generated as a wave at an inaudible range. We believe this is caused by the limitation of the hardware. Through a phone speaker, it is difficult to create a clean wave without the pre/post vibrations present.

5.2 Doppler Shift

Doppler shift is often used to detect motion through the use of a sound wave. In general, a frequency shift is observed when the source frequency is “shifted” through the velocity of the moving object. While a user moves a finger towards a microphone, the reflected wave from the emitter is observed with a shifted frequency from the original. As the finger is moving towards the microphone, the observed frequency from the microphone is shifted towards higher frequencies. While the finger is being flipped back, lower frequencies are observed on the signal.

For our purposes, the frequency shift can be formalized as follows:

\[
 f_{\text{observed}} = \frac{c - v}{c} f_{\text{emitted}},
\]

where \( c \) is the speed of sound in air (340.29 m/s) and \( v \) is the relative speed between the emitter and the receiver. \( v \) is defined positive when the distance between the two objects is increasing. For this experiment, we placed a speaker emitting a continuous sine wave of 18kHz, representing \( f_{\text{emitted}} \) in front of the microphone array.

Figure 5 provides time-FFT (TFFT) spectrums for an index finger tapping gesture through each of the eight microphones. Microphone indices are named in an increasing order from the left to the right. For each spectrum, the x-axis represents time (a second), while the y-axis represents frequencies from 17,805Hz to 18,195Hz. Each spectrum also has a straight bold frequency band around 18kHz. This represents the signal reached directly to each microphone without any reflection. As seen in the figure, there are also two bumps along this frequency band. The first one (pointing towards higher frequencies) is the Doppler shift caused by the index finger moving towards the microphones, while the second one is the Doppler shift caused by the finger moving back to its original position.

5.2.1 Doppler shift moment detection

Even though we could observe the Doppler shift from each microphone, we need to see differences to figure out which finger is tapped. Our initial thought was that the moment of Doppler shift arrival would differ between each microphone. If one microphone is 10cm closer to the emitter compared to another, the closer one should receive the shift 0.29msec ahead, which is equivalent to 18 sample difference at a 60kHz sampling rate:

\[
34029 : 1 = 10 : x = 60000 : y \quad \Rightarrow x = 0.00029, y = 17.63.
\]

Although this may be the case, we found out it is challenging to detect the time difference between the Doppler in each of the microphones. As we can see from the figure above, it is not clear exactly when the shift happened due to the noise and the smoothness of the TFFT spectrum. Because the graph above contains 60,000 samples, we would need to detect an arrival difference of 1/3400 of the width of the graph.
5.2.2 Frequency shift detection
We also tried to observe the amount of frequency shift. According to Equation 5.2, the frequency shift ($f_{\text{emitted}} - f_{\text{observed}}$) is proportional to the relative velocity between the emitter and the observer. Therefore, each microphone should observe different amount of frequency shifts, with the one facing the direction of the tapping resulting in the highest shift. However, it is hard to see which microphone has the highest shift in Figure 5. We believe the speed of tapping is not fast enough to give enough difference on the graph, while the resolution of the frequency is 15Hz with our 60kHz sampling rate.

5.2.3 Doppler shift volume
Instead of detecting peak values (the moment of shift and the maximum amount of shift), we decided to take the volume of the shift: we compute the integral of amplitude, time, and frequency on the TFFT spectrum. For each finger tap, we compute the volume of the upper band (17,745Hz–17,955Hz) and the volume of the lower band (18,015Hz–18,255Hz). Through this process, we are able to extract 16 values for each finger tap.

The 16 feature values from one of our testers is visualized in Figure 6 in a parallel coordinates graph, which visualizes 16-dimensional space onto a 2-dimensional space, by specifying the $x$th feature value $y$ on 2d coordinate $(x, y)$. The charts are displayed in the order of the index of the finger: the thumb on the top and the pinky on the bottom. Although noise is still prevalent in this chart, we can observe characteristic trends for each of the gestures. To classify a gesture, we run a logistic regression algorithm with $C=100$.

6. LOGISTIC REGRESSION
In this section, we describe the results using the logistic regression classification on the Doppler shift features.

6.1 Confusion Matrix
We used two data set collected from different people: the first data set 0 is recorded from a male and contains 850 samples, the second data set 1 is recorded from a female and contains 544 samples. Each set contains similar number of samples for each finger. We altered the tapping figure every 50 samples so that we could collect samples with more variances. When the classifier is trained and tested on the same data set, we ran 10-fold cross validation. The learning time took around 50 msec to 100 msec on our Python code using scikit-learn library [7].

Figure 7 shows confusion matrices for each test. When tested on the same data set, the classifier performs better especially when detecting a thumb tap. This makes sense because its movement is horizontal, while the other four finger tap give similar vertical movements towards the microphones. Also, we could observe through the confusion matrix that the classifier will often misclassify the tapped finger to a neighbor finger, forming a narrow diagonal band on the confusion matrix. This makes intuitive sense, as signals generated from closer fingers would look similar compared to others.

Upon testing this learned classifier to a different person, results were incomparable: when learned from the male hand and tested on the female hand, the accuracy rate is 0.29, which is very poor considering the random guess classifier would produce 0.20. We believe this poor performance comes from two factors. First, the test environments are different. Even though we tried to constraint the relative position and orientation between the hand and the wrist, there will be a slight difference on the every try. This difference would be magnified with the different hand sizes. Second, gestures are personalized. Even for the same finger tap, the angle of the finger, the speed of tapping, the duration of pause we found is different for every person.

When the train set and the test set uses the same hand, the accuracy rate of the classifier is 86% and 90% for each. For the other case where we learn from one person’s hand and test on another person’s hand, the accuracy rate is 29% and 49% for each.

6.2 Learning Curve for Gestures
Figure 8 shows the accuracy rate according to the size of the training samples, while both the training set and the test set are drawn from the male data set with 10-fold cross validation. The accuracy rate converges quickly: it reaches 95% of its maximum accuracy rate 86% with only 200 sam-
Figure 8: Learning curve of gesture classification using logistic regression

It means that the user would need to move each of the fingers approximately 40 times before the classification algorithm can learn their gesture.

7. CONCLUSION

We present an initial attempt at classifying finger gestures through the use of a microphone array. We primarily focused on detecting poses by detecting reflections of a characteristic signal as well as detecting movements by detecting Doppler shifts. We describe the challenges and the different approaches at interpreting the resulting Doppler effect. We conclude the results of our experiments through a machine learning classification algorithm over the features extracted from its TFFT graph. We achieved high classification success rates of 86% and 90% from two different people. The accuracy rate was low when the trainer and the tester were different, which implies we should train the classifier for each person. This is still feasible considering the relatively small amount of training data required to achieve a good performance.

Future work may include looking at different characteristics of different hand gestures besides the Doppler effect. This includes the ability to distinguish phase differences between the reflected sound angles.

8. REFERENCES

[1] electret microphone amplifier - max4466 with adjustable gain.