Ultrasonic Gesture Recognition with Neural Network

Kun Jin
Supervisor: Arye Nehorai

Department of Electrical and Systems Engineering
Washington University in St. Louis（As a Visiting Student）
Summer 2016

***Abstract*—**This report introduces an approach to recognize hand gestures by utilizing the Doppler effect of ultrasonic soundwaves. I use a microphone array as the receivers and two speakers as the transmitters to generate ultrasonic soundwaves. Gestures are characterized through the Doppler frequency shift they generate in reflections of the soundwaves. I then apply the artificial neural network algorithm for the classification of gestures. The accuracy of the prediction steadily reaches a very high value at 99.6%, which proves the reliability of the approach. With this approach, we can realize non-touch control, which will be of great significance in human-computer interactions in the next few decades, at a very low cost.

I. INTRODUCTION

Gesture is becoming an increasingly popular means of interacting with computers, tablets and smartphones. In situations where direct touch input it is not convenient for the user to use (when fingers are dirty or oily), gesture control becomes necessary. Compared with using optical 3D imagers for gesture recognition, such as Microsoft Kinect, devices using ultrasonic wave are much cheaper and smaller in size. So far, most gesture recognition processes apply machine learning algorithms. Since we don't know if different gestures are linearly separable and it is a multiclass classification problem, we think the neural network algorithm appropriate for this problem.

In recent years, some other researches on gesture recognition using ultrasonic soundwaves have been carried out. Some works focus on building an original device for the purpose of recognizing a particular gesture, many other works have 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.

Some previous attempts at making such a device include SoundSense [1], where four ultrasonic rangefinders are located on four edges of the device to detect hand gestures. Reference [2] presents FingerMic, which is a trainable wearable device that provides a way to detect finger gestures, include the movement of thumb, index and all five fingers at once. Reference [3] detects one-handed gesture using ultrasonic Doppler sonar, this work presents an inexpensive device that uses three microphones for the recognition. These gestures include various movements of one hand, such as moving from left to right, up to down or in clockwise direction. Gesture recognition also include large range body movements. Reference [4] introduces Soundwave, a project focuses more on the software solutions to such topic. It applies the speaker and microphone on a laptop to recognize hand gestures. This technique utilizes Doppler shifts of an inaudible tone at a high frequency of 18KHz and 22KHz (at two times). Frequency shifts were captured for two-handed gestures as well as more complex gestures like double-tapping. Many measurement properties were investigated and the robustness is acceptable.

Inspired but different from aforementioned works , my work mainly focuses on using a microphone array to detect one handed gestures. I also give more detailed information about the signal processing part and describe the machine learning algorithm thoroughly.

With the data collection method in Section II and signal processing method in Section III, this approach reaches a very high

prediction accuracy of 99. 86%, which is higher than that in all the reference works.

II. Data Collection

This section introduces the hardware devices I use and the procedures of data collection. The hardware devices include a microphone array with 16 receivers in a row, two speakers and a laptop. Fig. 1 shows the hardware devices more intuitively. I choose four of the most common bare hand gestures for my work, namely moving towards, moving away, moving left and moving right. To make my approach practical, I think my approach should also recognize the situations without any movement, so there are five classes of movement altogether. The five classes of movement is shown in Fig. 2. For every single movement, a data sample is collected and for each class of movement, 200 samples are collected.

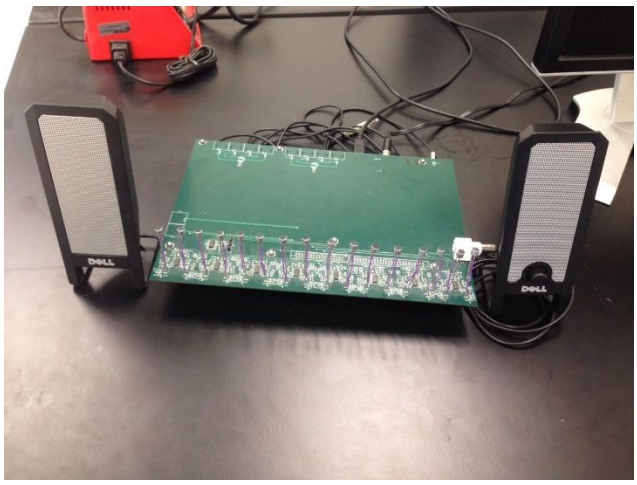
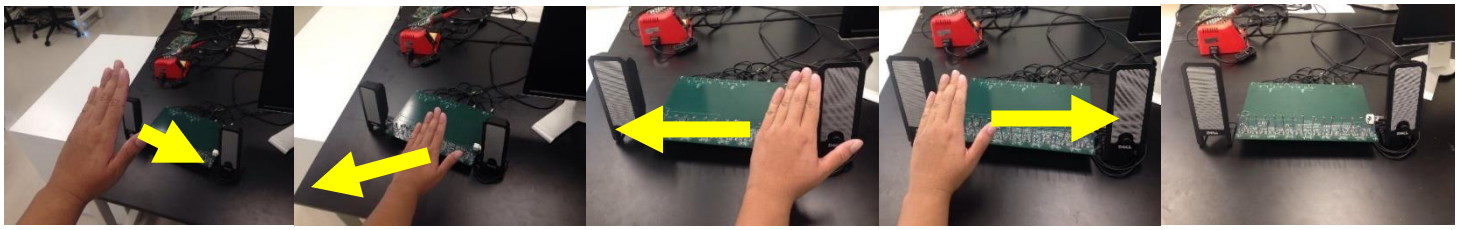


Fig. 1. The hardware devices for my experiment



a) moving towards b) moving away c) moving left d) moving right e) no movement

Fig. 2. Five classes of movements

Before the collection starts, we turn the speakers on and let them emit a 21KHz sinusoidal signal. The sampling rate is 50K and the collection period for each sample is 13x8192/5000 seconds, which is approximately 2 seconds. The data sample is a binary matrix, every element in the matrix is a voltage. For each receiver, 13x8192 voltages it receives are recorded in a sample. As a result, for each data sample, I get a matrix who has 16 rows and 13x8192 columns.

III. Operations and Algorithm

This section introduces my operations in this experiment and gives detailed information about my algorithm. The signal processing and feature extraction is of great significance to every application of machine learning algorithms. In this report, I processed the signal through the procedures shown in the Fig. 3. First, I feed the raw data through a Butterworth band pass filter, whose center is at 21KHz and bandwidth is 1KHz. Then I separate each row of data, whose size is 13x8192 into 12 windows. The windows' size is 2x8192, with an overlap of 50% with the neighboring ones. Finally, for each window, I apply the Fourier transform

and then select the corresponding power of certain frequencies as the features.

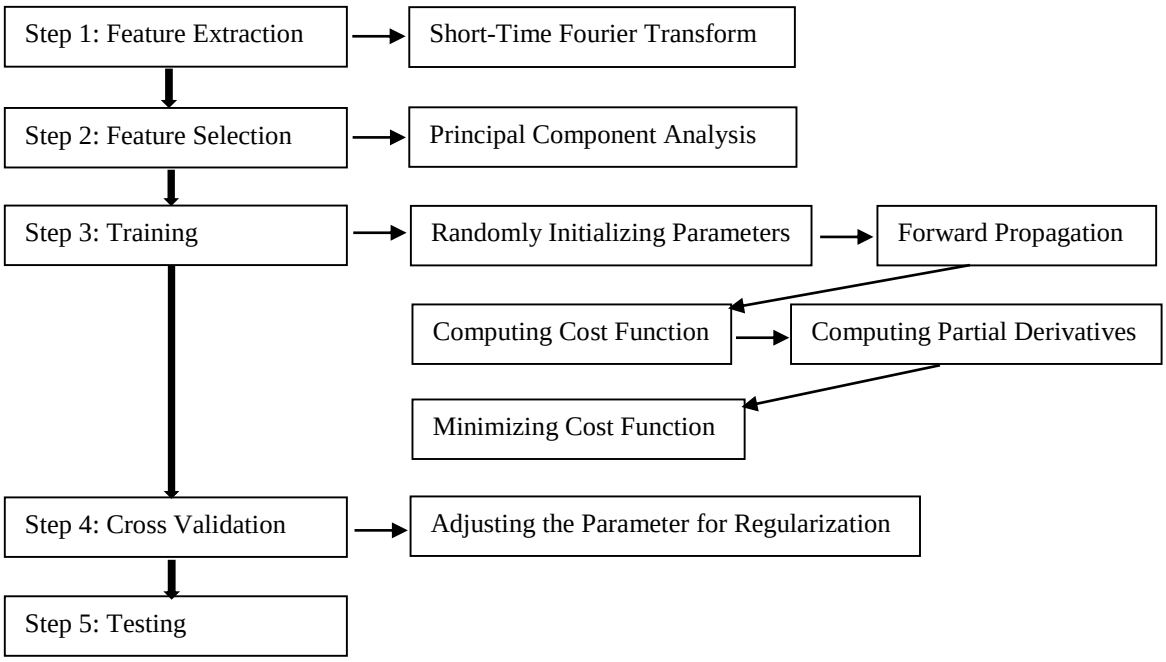


Fig. 3 Procedures of Operations

The problem of my feature extraction method is that there are 12x16x34=6528 (34 stands for the number of frequencies) features, which is much larger than the number of training samples. When the number of features is larger than the number of training samples, the algorithm is very likely to overfit. In order to avoid overfitting problem, I have to cut off some features with smaller impacts on the prediction. I apply the principal component analysis (PCA) to achieve that. To be particular, I use the singular value decomposition (SVD) algorithm in Matlab to get the eigenvalues of each feature. After getting the eigenvalues, I sort them in the descending order and find out the sum of the largest 200 eigenvalues is higher than 99% of the sum of all 6528 features. This shows that the corresponding 200 features are of much greater impact on the prediction than the rest 6328 features.

As mentioned above, I apply the artificial neural network algorithm for the prediction. The network contains three layers, an input layer, a hidden layer and an output layer. The size of input layer n_i=200, is determined by the number of features. The size of the output layer n_o is five, because there are five classes of movements. The size of the hidden layer $n_h = \sqrt{n_i n_o}$ ($n_h = 32$). The activation function is sigmoid function in this algorithm. Fig. 4 is a sketch of the algorithm in my work.

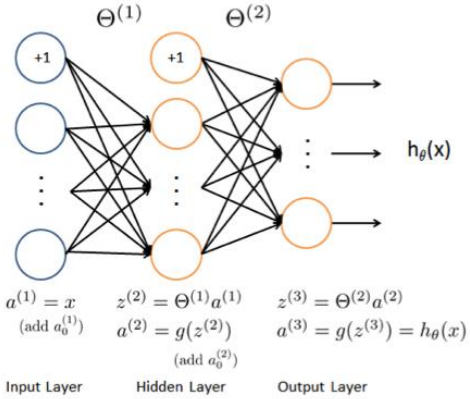


Fig. 4. A diagrammatical sketch for a neural network with one hidden layer

The neural network needs training and cross validation to get the best performance. As a result, I randomly separate the whole data set into three parts, namely the training set (500 samples), the validation set (200 samples) and the testing set (300 samples). The training works in the following procedures. Firstly, it randomly initializes the parameters (weights) in θ . Secondly, it applies forward propagation to get the hypothesis $h_\theta(x^{(i)})$ for any $x^{(i)}$ (the feature vector, i is the index of samples) in the training set. Thirdly, it computes the cost function J(θ), which is defined as:

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log h_\theta(x^{(i)})_k + (1-y_k^{(i)}) \log(1-h_\theta(x^{(i)})_k)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_j^{(l)})^2$$

,

where *K* stands for the length of label vectors (number of gestures), *L* represents the number of layers (including input layer and output layer), and λ is the parameter for regularization. Then it implements back propagation to compute the partial derivatives of the cost function J(θ). Finally, it trains the neural network by minimizing J(θ) as a function of parameters in θ (according to the partial derivatives).

After training comes the cross validation. I feed the features in the validation set into the trained neural network and let it make predictions. According to Fig. 5, the accuracy of prediction varies as I adjust the value of λ , and when the accuracy reaches the maximum value, I get the corresponding λ (λ =0.7), which is the best parameter for regularization in this case.

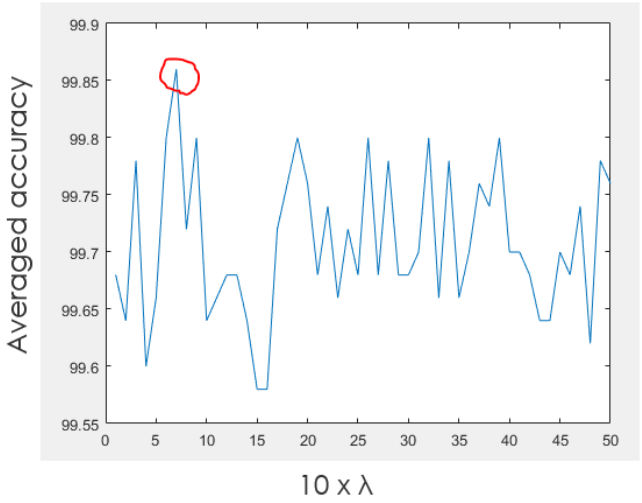


Fig. 5. Results of cross validation

After training and cross validation, I feed the features in the testing set to test the performance of the neural network, the table below demonstrates the confusion matrix for one time. The accuracy is around 99.86%.

	Towards	Away	Left	Right	No Move
Towards	67	0	0	0	0
Away	0	53	0	0	0
Left	0	0	54	0	0
Right	0	0	0	67	0
No Move	0	0	0	0	59

IV. Conclusions

In this report, I present an easy-to-use, economical and practical approach to recognize single hand gestures. It utilized the Doppler effect of ultrasonic sound waves and applied artificial neural network algorithm to match the Doppler frequency shift with corresponding gesture. The prediction accuracy was great and thus proves the usefulness of this approach. The limitation was that I only tried to recognize my own gestures. When the data for others gestures are collected, the usefulness and robustness of this approach can be further improved.

Future works will be focusing on the following points. First of all, collecting data for more gestures from more people. Secondly, making real-time gesture recognition possible. Thirdly, building up a user interface to make it convenient for everyone to use. Last but not least, searching for smaller and cheaper hardware devices so that they can be applied on mobile devices.

From this research program, I not only obtained much theoretical knowledge in the field of machine learning, but also become better at designing a project and working on it step by step. I can also see a significant improvement in my interest and confidence in doing a research.

Acknowledgment

At the point of finishing this report, I' d like to express my sincere thanks to all those who have lent me hands in the course of this project. First of all, I'd like to take this opportunity to show my sincere gratitude to my supervisor, Professor Nehorai, who has given me such an interesting topic to work on, and also given much useful advices on my project. Secondly, I' d like to express my gratitude to the PhD students, Prateek, Mianzhi, Zhen, Yijian, Mengxue, Yiqi, Zhenqi, Hesam, Eric and another undergraduate student here in WUSTL, Zoe, who offered me useful technical guidance on time. Thirdly, I' d like to thank Professor Ed Richter, who lent me those hardware devices and gave me a great environment for data collection. Last but not the least, I'd like to thank Xiangyang, Michael and Kevin who helped me to better understand how to use the hardware devices. Without their help, it would be much harder for me to finish my work.

References

[1] SoundSense: 3D Gesture Sensing using Ultrasound on Mobile Devices Available: http://119.90.25.34/mrorz.github.io/files/soundsense.pdf

[2] Travis Deyle, Szabolcs Palinko, and Erika Shehan. Fingermic: A lightweight bio-acoustic finger gesture interface for hand-full computing.

[3] K. Kalgaonkar and B. Raj. One-handed gesture recognition using ultrasonic doppler sonar. In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pages 1889{1892, April 2009.

[4] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. Soundwave: Using the doppler effect to sense gestures. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, pages 1911{1914, New York, NY, USA, 2012. ACM.