UNIVERSITY OF CALIFORNIA

Los Angeles

Towards Robust and Secure Audio Sensing
Using Wireless Vibrometry and Deep Learning

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical and Computer Engineering

by

Ziqi Wang

2020

ABSTRACT OF THE THESIS

Towards Robust and Secure Audio Sensing
Using Wireless Vibrometry and Deep Learning

by

Ziqi Wang
Master of Science in Electrical and Computer Engineering
University of California, Los Angeles, 2020
Professor Mani B. Srivastava, Chair

The number of audio-sensing-related applications is growing rapidly, such as the voice assistant as an interface between humans and computers, and the automatic-speaker verification system, which involves personal identity. These applications demand reliability and security of the audio sensing system. For example, an audio recognition system can easily get confused by the sound of non-target objects, as everything is fused in the collected audio. Meanwhile, a speaker verification system may fail under spoofing attacks of the computer-generated audio.

In this work, we focus on reinforcing existing audio sensing technologies to make it more robust and secure. This work comes in two parts. In the first part, we explore how we can leverage other modalities to improve the reliability of audio sensing, such as the impulse-radio Ultra-wideband (IR-UWB) radar. Our experiments show that this IR-UWB audio-sensing system can penetrate light-building materials to recover the sound. Meanwhile, the system is capable of measuring the distance between the sound source and the sensor, with which we can easily recover and separate the sound from multiple sources. In the second part, we explore how to defend against state-of-the-art acoustic attacks for critical applications such as voice authentication. We build a deep-learning-based system designed to determine if an audio clip is genuine human speech or, on the other hand, a computer-generated or a replayed one. This system is designed to work along with the automatic speaker verification system to protect it from spoofing attacks. Our results show a significant improvement from the baseline and some generalization abilities on unseen attack types. The work presented in this thesis provides the preliminary steps towards utilizing multiple modalities for robust audio sensing applications across a variety of environments, as well as an extra anti-spoofing protection for these applications using deep learning.

The thesis of Ziqi Wang is approved.

Danijela Cabric

Abeer Alwan

Mani B. Srivastava, Committee Chair

University of California, Los Angeles

2020

To my mother and father...

who raised me up and taught me to love

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Mani B. Srivastava, whose continuous support and far-sighted guidance helped me significantly during my first two years at UCLA, pursuing my master's degree. It is a great pleasure and honor to be one of his students. His passion for exploring new knowledge and tackling challenging problems always inspire me.

Meanwhile, I want to thank other members of my thesis committee, Prof. Abeer Alwan and Prof. Danijela Cabric, for helpful discussions and valuable feedback.

I wish to express my gratitude to my collaborators, Dr. Moustafa Alzantot, Akash Deep Singh, Dr. Zhe Chen, Dr. Luis A. Garcia, and Renju Liu, whose effort is very important in my research work during the last two years. I also thank Tianyue Zheng, Brian Wang, as well as other labmates in Networked and Embedded System Laboratory for helpful discussions. I sincerely appreciate the help of Lu Hu and Yifei Wang to polish my writing.

I also want to thank my family, especially my father and mother, whose love and support stay with me always. I want to thank my grandparents, who set up my dream of building technology that benefits human life at an early age. Thank my friends for always bringing me confidence and listening to me when I feel stressed. I also want to show gratitude to all my teachers and professors, who significantly shaped who I am.

Finally, I would like to show appreciation for the funding agencies that supported the work in this dissertation. This research was supported in part by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001; by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA; and, by the National Science Foundation under award number CNS-1705135. Any findings in this material are those of the author(s) and do not reflect the views of any of the above funding agencies. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

CHAPTER 1

Introduction

Audio is one of the most popular modalities on mobile sensing platforms. In this thesis, we show some preliminary efforts to enable robust and secure audio sensing in complicated environments, where existing audio sensing systems expose vulnerabilities. For example, the audio-based recognition and classification system's performance significantly downgrades in complicated environments due to the fusion of the target audio and the non-target ones. Meanwhile, the computer-generated or the replayed audio post a significant threat to the systems that use the human voice as a biometric for the personal identity.

In this work, we explore the possibility of using Wireless Vibrometry as a novel approach that can perform audio sensing and audio source distance measurement simultaneously, as well as protecting audio sensing systems by a deep-learning-based countermeasure system to defend audio spoofing attacks.

Wireless Vibrometry refers to the research efforts that detect vibrations of objects using wireless signals, which has enabled many applications. Typical examples are acoustic eavesdropping, machine abnormality detection and activity recognition. In the first part of this dissertation, we propose an impulse-based ultra-wideband (IR-UWB) radar system that simultaneously achieves two tasks. Firstly, it is capable of sensing the tiny vibrations caused by sound source activities, e.g. the vibrations of a speaker diaphragm, and recovering the original sound or voice, without requirements of a transmission medium or line-of-sight (LOS) condition. Secondly, this system is capable of obtaining a spatial estimation of the vibrations. We can retrieve audio signals together with distance estimations of those sounds, which enables separations of simultaneous sounds whose sources are located at different distances.

This system employs commercial-off-the-shelves (COTS) IR-UWB radar operating at

sub-10GHz bands, which works by sending and retrieving pulse sequences. We demonstrate mathematically that it is possible to perform audio sensing using impulse-based radio waves. We then build a real-world system together with a learning-free signal processing pipeline. In further testings, we show that this system possesses the ability to obtain target sounds in noisy, non-LOS scenarios. We also showcase some sample applications of such a system, as well as point out the directions for future efforts.

The second part of this work focus on security issues in audio and speech sensing. Over the past decade, voice control has gained popularity as a practical and comfortable interface between users and smart devices. Due to the security and privacy-sensitive nature of many applications (e.g., banking, health, and smart home) running on these devices, automatic speaker verification (ASV) techniques have emerged as a form of biometric identification of the speaker. However, ASV systems are threatened by replay and audio spoofing attacks where an attacker utilizes techniques such as voice conversion or speech synthesis to gain illegitimate control over user devices.

To enhance reliability against attacks, we combine ASV systems with audio spoofing detection systems. In this part, we develop such a system that distinguishes between spoofing attacks and genuine speeches. Our model is inspired by the success of residual Convolutional Neural Networks (CNNs) in many classification tasks. We build three variants of a residual CNN that accept different feature representations (MFCC, Cepstrum, and CQCC) as the input. We compare the performance achieved by our model variants and the baseline models. In the case of computer-generated audio, our model shows 25% improvements from the baseline. While facing replay attacks, our model fusion improves the baseline scores by over 70%.

Part I

Audio Sensing Using Ultra-Wideband Radar

CHAPTER 2

Background

Mechanical vibrations whose frequencies lie in the auditory range create audible sounds. In the first part of this thesis, we introduce an audio sensing system based on Impulse-Radio Ultra-wideband Radar (referred to as *UWB radar* in the rest of this part). The most popular audio sensing system nowadays is the microphone. As a transducer, the microphone works by translating the mechanical sound pressure wave into electrical signals. The sound wave is mechanical, which can only be detected if it propagates through a medium to arrive at the sensor. Meanwhile, the audio sensing system using wireless vibrometry works in an active way. It sends out probe signals to be reflected on the vibrating surface of the sound source, and recovers the sound directly from the minute sound source vibrations. In order words, a wireless vibrometry-based audio sensing system allows allows the sound to 'propagate' even in the vacuum. Moreover, our UWB radar based audio sensing system can simultaneously measure the distance from the sensor to multiple sound sources, making audio sensing distance-aware. In this chapter, we provide background about some concepts used in this part as well as some closely related works that have inspired our research.

Wireless vibrometry refers to the research work trying to sense vibrations using wireless signals, which is a growing field of research. Recent works have focused on recovering information from vibrating objects. For example, [YLL16] uses RFID tags to identify mechanical vibrations periods of spinning targets like a high-speed centrifuge. [ZHC19] employs commercial WiFi signals to detect human breath status. [NGW15] leverages frequency modulated continuous wave in ultrasound frequency to detect chest movement for sleep apnea detection.

As a particular type of mechanical vibration that lies in the human auditory range, sound is ubiquitous in human life, and naturally draws great attention in this field. Several works have emerged in recent years, showing the ability to recover sound from a source of vibration. For instance, [DRW14] has shown that it is possible to recover audio through vibrating objects (such as an empty chip bag) using a high-speed camera. WiFi signals are also used for audio sensing as its channel state information will carry hints of all kind of movements including fine-grained vibrations, due to micro-Doppler effect and multipath. [WWZ15] transmits WiFi signals with software-defined radio to recover sound from loudspeakers. Although some of those systems enable eavesdropping capabilities and raise significant privacy concerns, wireless vibrometry can also benefit our daily life. For example, [ZLH18] employs lasers to detect the vibrations of house appliances, enabling centralized device usage sensing in smart homes. [XLZ19] uses millimeter wave (mmWave) to sense human voice and builds a new interface for controlling voice-assistants in noisy and complicated scenarios.

Prior works in wireless vibrometry show that wireless signals could be used for sound recovery. However, when it comes to wireless signals, there is always a trade-off between range resolution and signal penetration. Signal penetration is the ability for a signal to pass through objects without losing all their energy, while range resolution determines how accurately we can distinguish between two objects placed close to each other. Penetration abilities offers the potential to perform sensing in non-line-of-sight scenarios, while a higher spacial resolution can sense objects more accurately. Typically, signals that have a larger bandwidth have a better range resolution, which can be described as

$$r = \frac{c}{2B}, \tag{2.1}$$

where c is the speed of light and B is the bandwidth employed. Owing to historical reasons like spectrum licensing, wide-band communications schemes can usually be found at a high-frequency range, which employs sub-millimeter or millimeter wavelengths. However, high-frequency signals have low penetration and hence can only sense objects present in the line-of-sight. mmWave, laser, visible light (high-speed camera) are all examples with a high resolution and poor penetration. On the other end of the resolution-penetration spectrum, we have modalities like WiFi sensing that operates at a lower frequency (compared to mmWave), the signals can penetrate solid walls and still be able to recover single-tone audio (i.e., a prolonged musical note), but may experience challenges when sensing fine-grained audio. Also, almost all modalities employed in wireless vibrometry uses continuous waves, making most of them energy-hungry and thus challenging for mobile scenarios.

In this work, we build an audio sensing system on top of Ultra-wideband (UWB) radar. We prefer UWB radar over other wireless sensing modalities partly because we aim to strike a balance between signal penetration ability and range resolution. By definition, a UWB system refers to the radios that occupy more than 500MHz bandwidth. Specifically, we used impulse-radio UWB. Those devices communicate by sending very short impulses that occupy a large frequency range. Working under a sub-10GHz band, UWB possesses the capability of penetrating light building materials. Our experiment shows that UWB radar is capable of not only retrieving audio though wall, but also giving an estimation of how far the sound source is from the sensor. UWB devices are widely used in ranging, tracking, sensing [DGZ18], and health monitoring. Figure 2.1 shows an example of our preliminary experiments using UWB radar to recover human breath.

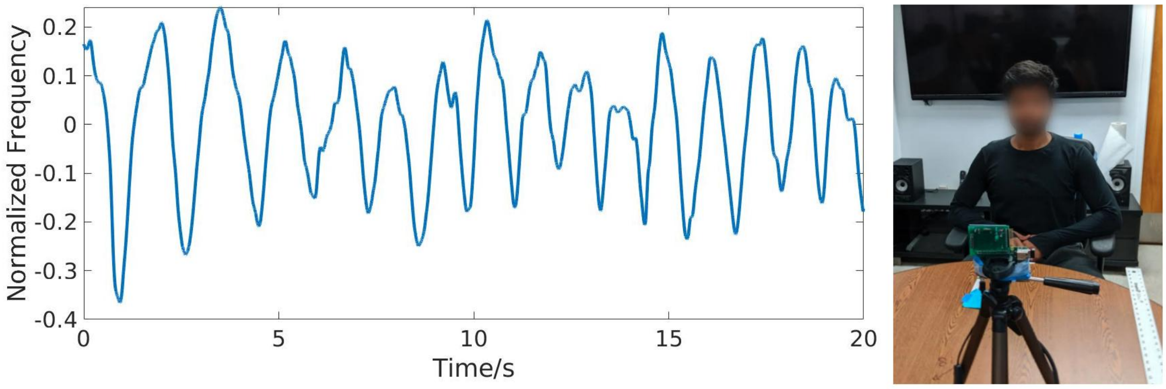


Figure 2.1: *UWB signal changes under human breath*

UWB's transmission power is limited to ensure co-existence with other communication schemes in the same frequency band, such as WiFi and Bluetooth. Unlike WiFi and mmWave, which use a continuous wave, impulse-based UWB is well known for low power consumption. In addition, UWB sensors are entering the mainstream - for instance, the new model of iPhone has already incorporated UWB sensors [Shab]. As a result, we aim to exploit the ubiquity of such sensors to enhance the process of sound recovery in challenging environments. Moreover, with the ranging ability of UWB radar, we expect to make audio sensing "distance-aware", enabling some applications like sound separation.

The rest part of this part is organized as follows. In Chapter 3, we describe the theory of using impulse-based UWB radar for audio sensing mathematically. Then in Chapter 4, we provide details of the UWB audio sensing system design. In Chapter 5, we explore the performance limits of such a system and provide some sample applications. Finally, we summarize some related work in this field in Chapter 6 and conduct some discussions along with a summary in Chapter 7.

Our contribution in this work is as follows:

- We provide theoretical analysis on performing audio sensing using non-continuous, impulse-based wireless signals.
- We build a effective hardware system from commercial-off-the-shelves (COTS) UWB radar sensor with optimal driver settings, as well as a pure statistical signal processing pipeline.
- To the best of the authors' knowledge, we are the first work to investigate the possibility of extracting audio from UWB radar responses.
- We explore a few potential applications of the UWB audio sensing system, and also test its limitations.

CHAPTER 3

Theory of Audio Sensing Using Ultra-wideband

In this section, we formulate this UWB acoustic problem mathematically. Even though UWB provides a higher spatial resolution than other wireless modalities like WiFi, it is still not possible to detect minute millimeter level displacement caused by sound vibration directly with Time-of-Flight estimation. It is non-trivial to show that, the complex baseband equivalent processing in UWB radar enables the sensing of sound-related vibrations.

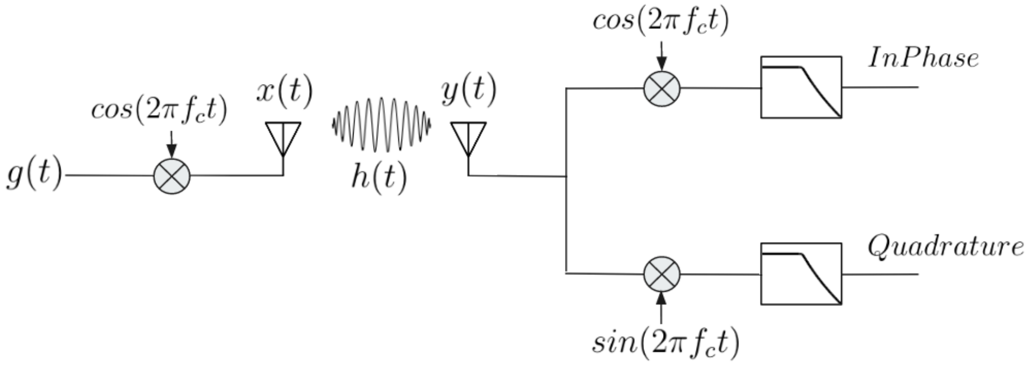


Figure 3.1: *Impulse-radio UWB radar system in equivalent baseband representation*

The equivalent baseband representation of our UWB radar system is shown in Figure 3.1. We define a frame as a period where one pulse is sent out and its responses are collected. The notion of time (t) within one frame corresponds to the time-of-flight of the signal pulse, which is also known as *fast time*. Meanwhile, UWB radars work by sending out probe pulses sequentially with interval T_s. We use *slow time* (t_{slow}) to denote the probe pulse repetition intervals. Usually, *fast time* is fine grained in tens of picoseconds, while *slow time* has the scale of hundreds of microseconds.

The UWB radar sends Gaussian pulses $g(t)$ modulated on a carrier frequency f_c, which can be mathematically represented as

$$x(t) = g(t - kT_s)cos(2\pi f_c(t - kT_s)), \tag{3.1}$$

where T_s is the pulse repetition rate, and the baseband Gaussian pulses $g(t)$ are given in [AGM17] as

$$g(t) = V_{TX}exp(-\frac{t^2}{(\pi f_B\sqrt{2log_{10}(e)})^{-2}}). \tag{3.2}$$

f_B denotes the -10dB bandwidth. The impulse sequence is sent out to interact with the objects in the environment and received by the receiving antenna. Note that in reality, the transmitting antenna and the receiving antenna are co-located. The channel frequency response in an indoor environment can be characterized as a summation of P paths with different time delays and attenuation, i.e.,

$$h(t) = \sum_{p=1}^{P} \alpha_p \delta(t - T_p - T_p^D(t)). \tag{3.3}$$

T_p is the static delay caused by the path length, and for the line-of-sight scenario it is determined by the target distance. $T_p^D(t)$ is the time-varying displacement caused by minute target movement, such as the cone being pushed back and forth by the coil in any speaker. For static objects, $T_p^D(t) = 0$. Our goal in wireless audio sensing is to recover the $T_p^D(t)$, which can be translated into sound. For simplicity, we ignore the Doppler effect cause by large-scale target movements, since it varies very slowly compared to the pulse repetition rate T_s.

The received signal $y(t)$ can be modeled as a convolution of the transmitted signal and the channel frequency response, plus additive noise, i.e.,

$$\begin{aligned} y(t) =& x(t) * h(t) + n(t) \\ =& \sum_{p=1}^{P} \alpha_p g(t - kT_s - T_p - T_p^D(t)) cos(2\pi f_c(t - kT_s - T_p - T_p^D(t))) + n(t). \end{aligned} \tag{3.4}$$

On the receiver side, the received signal $y(t)$ is downconverted. $y(t)$ is multiplied with the carrier frequency in a mixer, and then passed through a loss-pass filter. We take the in-phase branch as an example. Looking at the cosine part only, we will see that,

$$\begin{aligned} m(t) =& cos(2\pi f_c(t - kT_s - T_p - T_p^D(t)))cos(2\pi f_c(t - kT_s)) \\ =& \frac{1}{2}[cos(2\pi 2f_c(t - kT_s - \frac{T_p}{2} - \frac{T_p^D(t)}{2})) + cos(2\pi f_c(T_p + T_p^D(t)))]. \end{aligned} \tag{3.5}$$

The $2f_c$ frequency term is filtered out, leaving $\frac{1}{2}cos(2\pi(T_p + T_p^D(t)))$ term only. Based on this, we can rewrite the in-phase baseband signal after down-conversion and filtering as

$$\begin{aligned} y_{in-phase}(t) &= LPF[y(t) \times cos(2\pi f_c(t - kT_s))] \\ &= \frac{1}{2}\sum_{p=1}^{P} \alpha_p g(t - kT_s - T_p - T_p^D(t)) cos(2\pi f_c(T_p + T_p^D(t))) + \tilde{n}(t). \end{aligned} \tag{3.6}$$

Similarly, the in-phase baseband signal after down-conversion and filtering can be represented as

$$\begin{aligned} y_{quad}(t) &= LPF[y(t) \times sin(2\pi f_c(t - kT_s))] \\ &= \frac{1}{2}\sum_{p=1}^{P} \alpha_p g(t - kT_s - T_p - T_p^D(t)) sin(2\pi f_c(T_p + T_p^D(t))) + \tilde{n}(t). \end{aligned} \tag{3.7}$$

The target time-of-flight T_p can be translated into target distance when multiplied with the speed of light, allowing us to examine different targets at different distances by setting $t = kT_s + T_p$. For those paths without voice-related movement whose $T_p^D(t) = 0$, the response $y(t = kT_s + T_p)$ ideally will not change over slow time. We can filter those static responses out by applying a static clutter suppression algorithm that will be introduced in Chapter 4.

Suppose the sound-related vibration is captured in path p_0, we can isolate the received signal from such a path by setting $t = t_p = kT_s + T_{p0}$, which can be written as

$$y_{in-phase}(t_p) = \frac{1}{2}\alpha_{p0} g(T_{p0}^D(t_p)) cos(2\pi f_c T_{p0} + 2\pi f_c T_{p0}^D(t_p)) + \tilde{n}(t_p). \tag{3.8}$$

$$y_{quad}(t_p) = \frac{1}{2}\alpha_{p0} g(T_{p0}^D(t_p)) sin(2\pi f_c T_{p0} + 2\pi f_c T_{p0}^D(t_p)) + \tilde{n}(t_p). \tag{3.9}$$

We can have an estimation of the scale of $T_{p0}^D(t)$. Suppose the sound-related displacement is 2mm, and the UWB carrier frequency is 7.5GHz, then

$$max(T_{p0}^D) = \frac{d}{c} = \frac{2 \times 10^{-3}}{3 \times 10^{-8}} = 6.67 \times 10^{-12}, \tag{3.10}$$

$$max(2\pi f_c T_{p0}^D) = 6.67 \times 10^{-12} \times 2 \times \pi \times 7.5 \times 10^{9} = 0.314. \tag{3.11}$$

Both are minimal values. Then using Maclaurin series to expand $g(t)$ around $g(0)$ and ignoring second-order and above terms, we have

$$g(t) = g(0) + g'(0)t + o(t^2) = V_{TX} + 0 \cdot t + o(t_2) = V_{TX} + o(t^2). \tag{3.12}$$

Plug Equation 3.12 into 3.8 and 3.9, and ignore high order infinitesimals as well as noise, we get the form of

$$y_{in-phase}(t_p) = \frac{1}{2}\alpha_{p_0} V_{TX} cos(2\pi f_c T_{p_0} + 2\pi f_c T_{p_0}^D(t_p)), \tag{3.13}$$

$$y_{quad}(t_p) = \frac{1}{2}\alpha_{p_0} V_{TX} sin(2\pi f_c T_{p_0} + 2\pi f_c T_{p_0}^D(t_p)). \tag{3.14}$$

By Taylor expansion, $f(t) = \sum_{n=0}^{\infty} \frac{f^{(n)}(t_0)}{n!}(t-t_0)^n$, where $f^{(n)}(t_0)$ is the n^{th} derivatives of $f(t)$ at t_0, we know that

$$\begin{aligned} sin(t) =& sin(0) + cos(0)t + o(t^2) \approx t, |t| < \epsilon \\ cos(t) =& cos(\frac{\pi}{2}) - sin(\frac{\pi}{2})t + o(t^2) \approx -t, |t - \frac{\pi}{2}| < \epsilon \\ sin(t) =& sin(\pi) + cos(\pi)t + o(t^2) \approx -t, |t - \pi| < \epsilon \\ cos(t) =& cos(\frac{3\pi}{2}) - sin(\frac{3\pi}{2})t + o(t^2) \approx t, |t - \frac{3\pi}{2}| < \epsilon \end{aligned} \tag{3.15}$$

In Equation 3.10 we already show that $2\pi f_c T_{p_0}^D$ is a minute number. While $2\pi f_c T_{p_0}$ is very large, $mod(2\pi f_c T_{p_0}, 2\pi)$ is going to put the component inside the sine or cosine within Equation 3.13 and 3.14 near one of the four vicinities above. Without loss of generality, we assume $mod(2\pi f_c T_{p_0}, 2\pi) \approx 0$, then

$$y_{quad}(t_p) = \frac{1}{2}\alpha_{p_0} V_{TX} 2\pi f_c T_{p_0}^D(t_p) = \frac{1}{2}\alpha_{p_0} V_{TX} \frac{2}{c} d_{p_0}^D(t_p) 2\pi f_c = \alpha_{p_0} V_{TX} \frac{2}{c} \pi f_c d_{p_0}^D(t_p), \tag{3.16}$$

where $d_{p_0}^D(t)$ is the distance of sound-related movement, and c is the speed of light. It is clear that the amount of target micro displacement is linearly proportional to the amplitude of the quadratic part of the receiving signal. In other cases, it will be linearly proportional to the amplitude of the in-phase part. Note that $t_p(k) = kT_s + T_{p_0}$, and $d_{p_0}^D$ changes over slow time. For example, if a sine wave single tone (f_{music}) sound is played, then the $d_{p_0}^D$ should be modeled as,

$$d_{p_0}^D(t_{slow}) = max(d_{p_0}^D) sin(2\pi f_{music} t_{slow}). \tag{3.17}$$

We can treat $d_{p_0}^D(t_p) = d_{p_0}^D(kT_s + T_p)$ as the speaker movement $d_{p_0}^D$ being sampled at interval T_s, i.e., sampled at the UWB frame rate. $y_{quad}(t_p)$ or $y_{in-phase}(t_p)$ is proportional to $d_{p_0}^D(t_p)$. Thus we conclude that we can recover the sound-related movement from the amplitude of UWB in-phase or quadrature data, whichever gives a higher signal quality

As a summary, in this chapter, we show mathematically that we can extract the sound-related vibration information by analyzing the amplitude of the In-phase or Quadrature of UWB receiving signal, whichever gives is a higher signal-to-noise-ratio. Our work aims to extract the vibration related information from the UWB radar sensor readings, and this chapter provides a theoretical guarantee for this goal.

CHAPTER 4

System Design and Implementation

4.1 System Overview

In the previous chapter, we gave a mathematical proof of the theory using UWB radar for sound recovery. In this chapter, we build a real-world system from a commercial-off-the-shelf UWB Radar board, and implemented a data processing pipeline to make the theory a reality. Figure 4.1 gives an overview of our UWB audio sensing system.

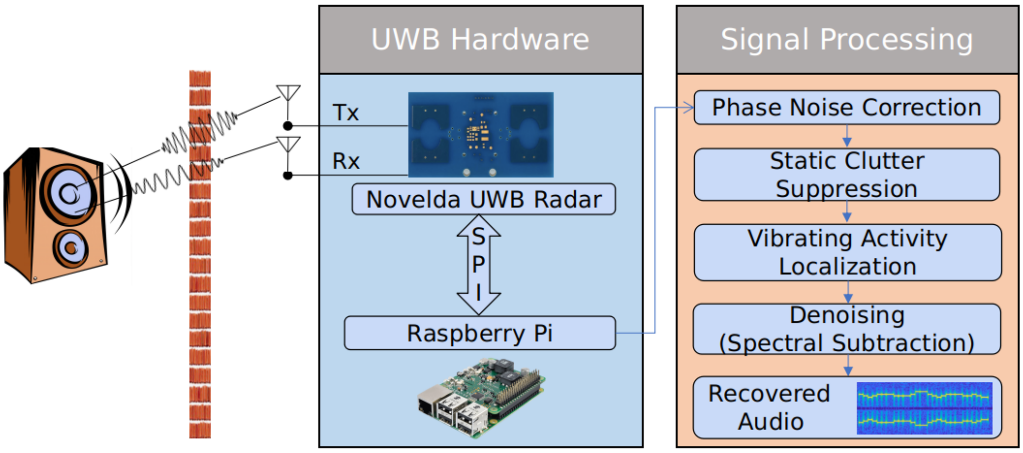


Figure 4.1: *An overview of our UWB audio sensing system*

The system uses a UWB radar that sends out impulses at a constant rate, collects the reflected impulses, and downconverts the radio frequency data to the baseband in-phase and quadrature(I/Q) data. The I/Q data is then analyzed offline with our processing algorithms that consist of a few modules. Firstly, to battle the phase variations caused by sampling clock jitters, we employ the Phase Noise Correction algorithm. Static Clutter Suppression removes the reflections caused by static objects like walls and furniture. As we have analyzed in Chapter 3, the sound-related information will appear on the amplitude of the real or imagery part of the I/Q data. Thus, we juxtapose the in-phase part and the quadrature part. Since the reflected pulses have various time-of-flight which correspond to a wide distance range, it is crucial to ”localize” the distance bins where vibrations happen using the Vibrating Target Localization module. Finally, we can obtain the recovered sound with further denoising, such as a spectral subtraction algorithm. Then we can have a recovered sound for future interfaces, e.g., sound classification or speech recognition.

4.2 Hardware and Drivers

Our system is implemented with Novelda Xethru X4M05 UWB radar[1] board combined with a Raspberry Pi 3B+. Figure 4.2(a) shows the hardware stack of our system, the blue board is the UWB radar transceiver, and it is connected with the Raspberry Pi using a connector board designed by ourselves, whose schematic is shown in Figure 4.2(c). The connection between the Pi and the radar is realized via the SPI interface.

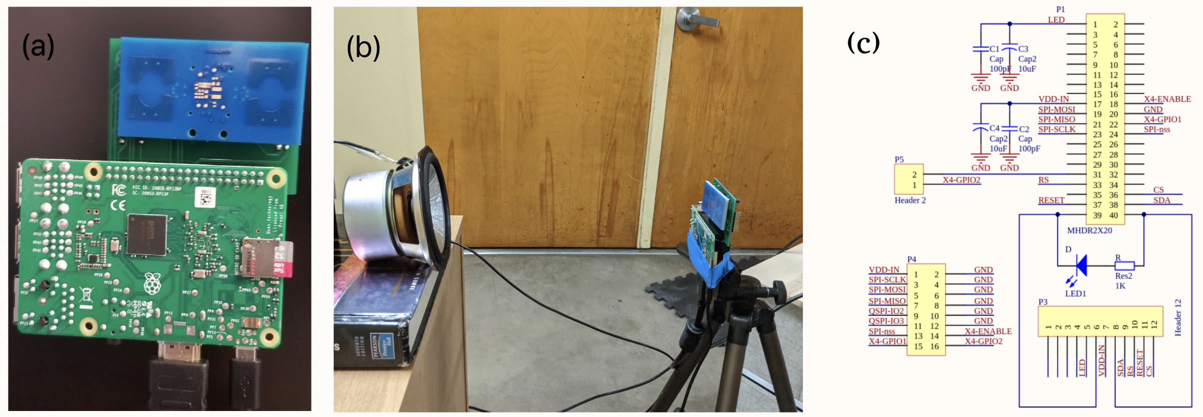


Figure 4.2: *Hardware components overview of the UWB audio sensing system: (a) Hardware appearance (b) Proof-of-concept experiment setup (c) Hardware Connection*

4.2.1 UWB Data Collection

The X4M05 radar board consists of an X4A02 Antenna board and a Novelda X4 impulse radar transceiver System on Chip (SoC). According to its datasheet [XeT], the UWB radar operates at a center frequency of 7.29GHz with a bandwidth of 1.4GHz. Like other

[1] Due to strategic realignment of Novelda, the Xethru community website and some of the datasheets, where many information of the hardware and drivers comes from, is no longer available online. Some driver examples, hardware design files and datasheets are available on their Github archive: https://github.com/novelda

radars, UWB radar works by sending out a probe carrier-frequency modulated Gaussian pulse at a constant rate and collecting responses.

The data collected from the probe pulse to the arrival of the last response is called a *frame*. In practice, we can set the maximum length of the frames to tune the radar range. All the frames are ordered chronologically, and Figure 4.3 shows an example of this data structure. The frames are placed along the Y-axis – the *slow time*. On the X-axis (*fast time*), we have reflective pulse responses with different time delays. Since the fast time denotes the round trip time-of-flight(ToF) of a pulse, we can convert the fast time into *distance bins*.

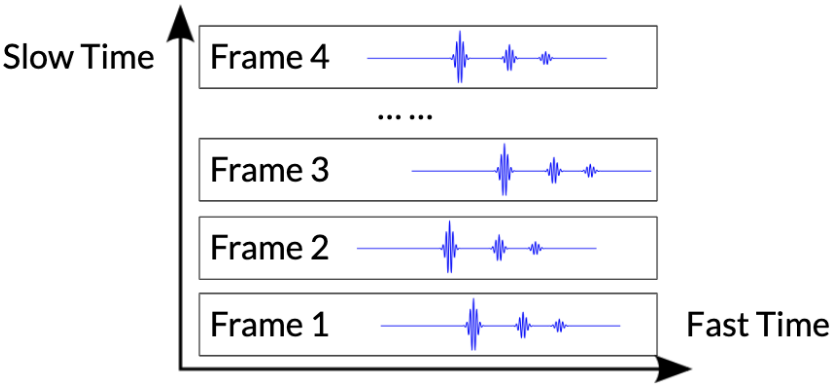


Figure 4.3: *Illustration of the fast time and slow time*

In the UWB radar hardware we use, the Gaussian pulses are modulated on a sub-10 GHz carrier frequency. At the receiver side, a digital down-conversion is performed on the received Radar Frame (RF) data inside the X4 SoC to retrieve the baseband pulses, making each data point a complex double representing in-phase and quadrature (I/Q) baseband data. This down-conversion stage will decimate the RF data by a factor of 8. The sampling rate of the *fast time* is 23.328 GSamples/s. With all the information above, we can calculate the distance between adjacent distance bins in the baseband data as

$$\begin{aligned} bb\_interval &= \frac{LightSpeed}{2 \times SamplingRate} \\ &= \frac{2.998 \times 10^8 m/s \times 8}{2 \times 23.328 \times 10^9 Hz} = 0.0514m. \end{aligned} \tag{4.1}$$

The maximum length of the received Radar Frame (RF) data before the downconversion has 1536 bins. So the maximum range of such a radar system is

$$max\_dist = 1536 \div 8 \times 0.0514 = 9.87m. \tag{4.2}$$

As a summary, the collected data is going to be a complex matrix with dimension $fast\_time \times slow\_time$. The fast time dimension indicates the target distance while the second dimension indicates the elapsed time.

4.2.2 Driver Settings

The Xethru radar driver is implemented based on [XED], with modifications to enable faster data transfer and to strike a balance between sampling rate and signal-to-noise-ratio(SNR). In this section, we describe the major challenges in the driver settings to enable audio sensing.

SPI clock. The X4 radar SoC receives configuration and sends data to Raspberry Pi using Serial Peripheral Interface Bus (SPI). Once the X4 radar SoC finishes a data frame, it raised an SPI interrupt so that the controller (in this case, Raspberry Pi) can read the data. Owing to the fact that the radar SoC only caches the last frame it received, the clock of SPI should be set higher to ensure that the data can be transported in time. We set the clock to 32 MHz.

Transmission Power. The radar transceiver can operate at three different transmitting power settings, which are low(0.48 pJ/pulse), medium(1.47 pJ/pulse), high(2.65 pJ/pulse). In our experiment, we test on both the medium level and the high level, and they are both capable of audio sensing. A higher power level can increase the sensitivity and effective range of the system. However, these settings should be performed carefully to comply with FCC regulations.

Effective Range. As analyzed in Section 4.2.1, the maximum range of the UWB radar can be as far as 9.87m. However, due to the limits of the transmission power and the SNR requirements, we set the effective range to be 0.3m-4.3m. The first few bins are discarded since they are usually overfilled by crosstalks between the transmitting antenna (Tx) and the receiving antenna (Rx).

DAC Settings and Sampling Rate. The X4 uses a swept-threshold sampling method, according to [APP][AGM17]. The pulse duration is so short that a standard

DAC will not be fast enough. A Swept-Threshold Sampling method is used to address this problem. The received signal frame is compared against a threshold to generate one-bit values for all data points in this frame. The threshold will increase by one step before the response of the next repeated pulse comes. Due to the extremely high pulse repetition rate, the vibrating target can be approximated as static in such a short period, which means that the repeated frames can be treated the same as the previous ones. Then after a certain number of frames, we can have a multiple-bits digital representation of the original analog frame. The procedure is denoted as one *iteration*. It is also possible to average multiple *iterations*, or to average multiple pulses during one step (increase *pulse-per-step*) to improve SNR. However, if these two knobs are set too high, the sampling rate will be limited. This relationship can be mathematically described as

$$FPS = \frac{PulseRepetitionFrequency(15.1875MHz)}{Iterations \times PulsePerStep \times (DAC_{max}(1100) - DAC_{min}(949) + 1)} \times DutyCycle. \tag{4.3}$$

Heuristically, we pick $Iterations = 20, PulsePerStep = 2$, and $FPS = 1.5kHz$. Currently, due to the limitations of SPI transfer speed, the sampling rate can not exceed 1.6kHz, otherwise packet loss will be inevitable.

The data was stored locally in the Raspberry Pi and then transferred to a desktop computer with AMD Ryzen 7 2700X processor for processing. Figure 4.2(b) demonstrates a typical setting of our proof-of-concept experiment. The UWB radar system is mounted on a tripod and placed at a distance from the speaker. The speaker is connected with a cell phone to play the test tones.

4.3 Signal Processing Pipeline

The collected data is then analyzed offline with our processing algorithms shown in Figure 4.1 that consist of a few modules: Phase Noise Correction that removes sampling clock jitters, Static Clutter Suppression that suppresses the reflections caused by static objects, and Vibration Activity Localization that determines the distances of the vibrating targets. Finally, we can acquire recovered audio after denoising and normalizing. We will introduce those modules separately in the rest part of this section.

4.3.1 Phase Noise Correction

The basic idea behind this work is to measure the amplitude change over time of the in-phase or quadrature data caused by source vibrations. However, many factors will block us from retrieving the information related to sound vibration, and one of those factors is the phase noise. Phase noise is introduced due to the imperfection of the signal sampling clock. These imperfections may include crystal defects and phase lock loop (PLL) error.

Ideally, if we select out the data from one distance bin and analyze the phase over time, the phase should remain relatively the same supposing there are no vibrations at the current bin. However, with phase noise, you can see a rapid change of phase back and forth, which will then lead to the system mistakenly believe in the existence of a vibration in this bin. Figure 4.4 shows an example of the phase noise between adjacent UWB frames.

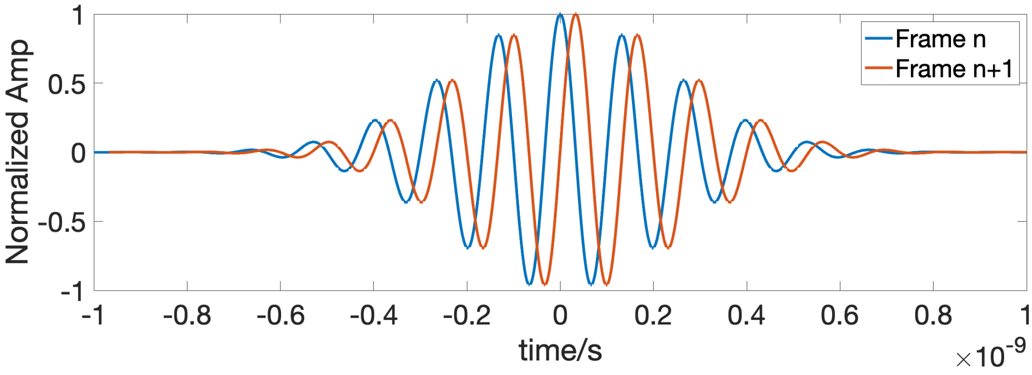


Figure 4.4: *Example of phase noises between adjacent frames*

The insight here is that the signal amplitude in the first few bins is always high, which is due to the "crosstalks" between the transmitting antenna (TX) and the receiving antenna (RX), i.e., direct signal leakage from the Tx to the Rx. Our insight is that this crosstalk can be used as a baseline for phase calibration. Following the method proposed in [PNC], we first calculate the mean phase of bin 1 and use it as a standard reference phase. For each frame, we calculate the difference $\Delta\phi$ between the phase of the first element (i.e., bin 1 data) and the reference phase. Then we multiply all samples from the current frame with $e^{j\Delta\phi}$ to offset the phase error.

4.3.2 Static Clutter Suppression

While vibrations can create a unique pattern on the receiving data, static objects like walls and furniture will also reflect UWB pulses and create strong responses. As shown in Figure 4.5(a), the high peaks around bin 20 and bin 50 are the evidence of static clutters. The static responses are so strong that the useful signal is buried underneath. Luckily, the static clutter is usually time-invariant in a select bin. We apply a Butterworth finite impulse response filter(FIR) on each distance bin, with the stopping frequency at 20Hz and the passing frequency at 70Hz. To ensure zeros phase distortion at the beginning of the sequences, the FIR filtering is applied to input frame data in both the forward and reverse directions. The stop-band attenuation was set at -80dB.

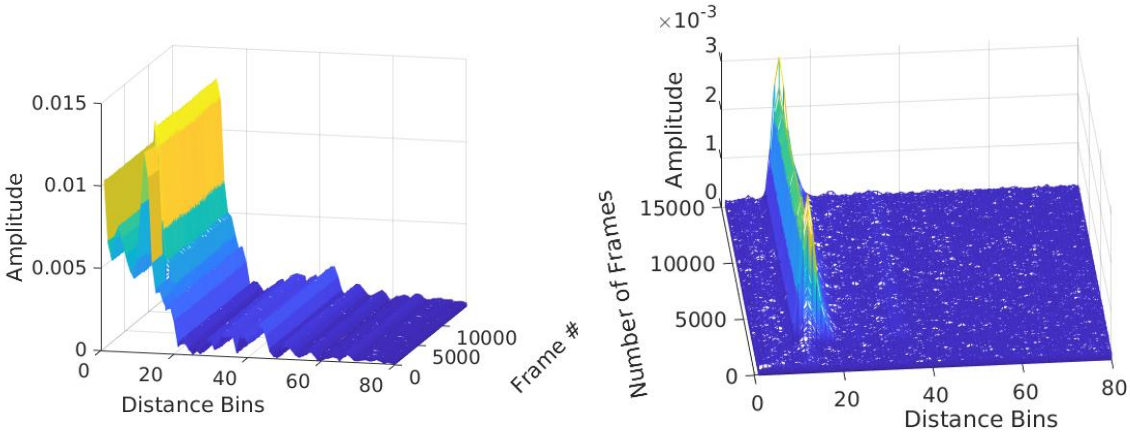


Figure 4.5: *Results of static clutter removal. Left: (a) Raw data after Phase Noise Correction. Right: (b) After Static Clutter Removal.*

Figure 4.5(b) shows the result after static clutter removal. The static peaks in Figure 4.5(a) are filtered out. Also, in the experiment shown in Figure 4.5, the sound lasts for about 12500 frames (8.3s), which is reflected in the peaks colored with green. From the filtered data, we can also see that the speaker is placed about 92.5 cm from the sensor (the ground truth is 100cm) as we see time-varying patterns around bin 18. Our static clutter suppression filter is able to remove all the responses caused by static objects, as well as the low-frequency vibrations caused by human movement or human breathing.

Due to the low-pass nature of UWB audio sensing (to be discussed in future chapters), we also provide an option of doing pre-emphasis at this stage:

$$y(t) = x(t) - \alpha x(t-1), \tag{4.4}$$

where $\alpha \in (0.95, 1)$. This difference equation is equivalent to a high-pass filter that compensates the signal loss in high-frequency ranges.

4.3.3 Vibrating Target Localization

UWB data contains multiple time series(columns) that corresponds to different distance bins. As shown in previous cases, we may visually localize the vibrations in some cases. However, it is vital to select candidate bins with a high signal-to-noise-ratio(SNR), where SNR is defined as the signal power divided by the noise power. Since the signal is still pretty noisy in some channels, only doing thresholding or calculating variance in the time domain will not give satisfying results.

We choose to solve this problem in the frequency domain. Our insight here is that, compared to noise, a channel(frames within a certain distance bin) with sound vibration information has a more concentrated spectrum than a noisy channel. For example, music will have basic notes and their higher order harmonics. While human voice power is more widely distributed in the spectrum, we can still observe basic frequencies F_0 and their harmonics. Thus, we firstly perform a Discrete Fourier Transform (DFT) over all channels to get their spectrums. Then the Herfindahl-Hirschman index (HHI) is used to calculate the concentration level of those spectrums. The Herfindahl-Hirschman index was introduced in economic fields as a measure of market concentration. It is calculated by squaring the "market share" of each frequency and then summing the resulting numbers. Here the "market share" is defined as the power of the current frequency divided by the overall power of the signal time series. The distance bins with the highest HHIs are selected as the candidates of bins containing vibration information.

4.3.4 Denoising and Normalization

After localizing the vibrating target, we can acquire an audio signal estimation by slicing that distance bin from the data. However, the recovered sound, while clearly audible, still contains non-negligible background noise which sounds like winds in microphone recordings. This noise is the $n(t)$ which we ignore in Chapter 3. Our observation is that noise is very close to an Additive White Gaussian Noise(AWGN). This can be demonstrated in

Figure 4.6. Part (a)-(d) of this figure shows that an original speech signal(a) is decimated to our sampling rate to acquire (b). It is then low-pass filtered to simulate the UWB channel response to get (c) and finally added AWGN to get (d). This simulated result is very similar to (e), which is the same signal measure in reality, and thus supports our assumption that the noise is AWGN.

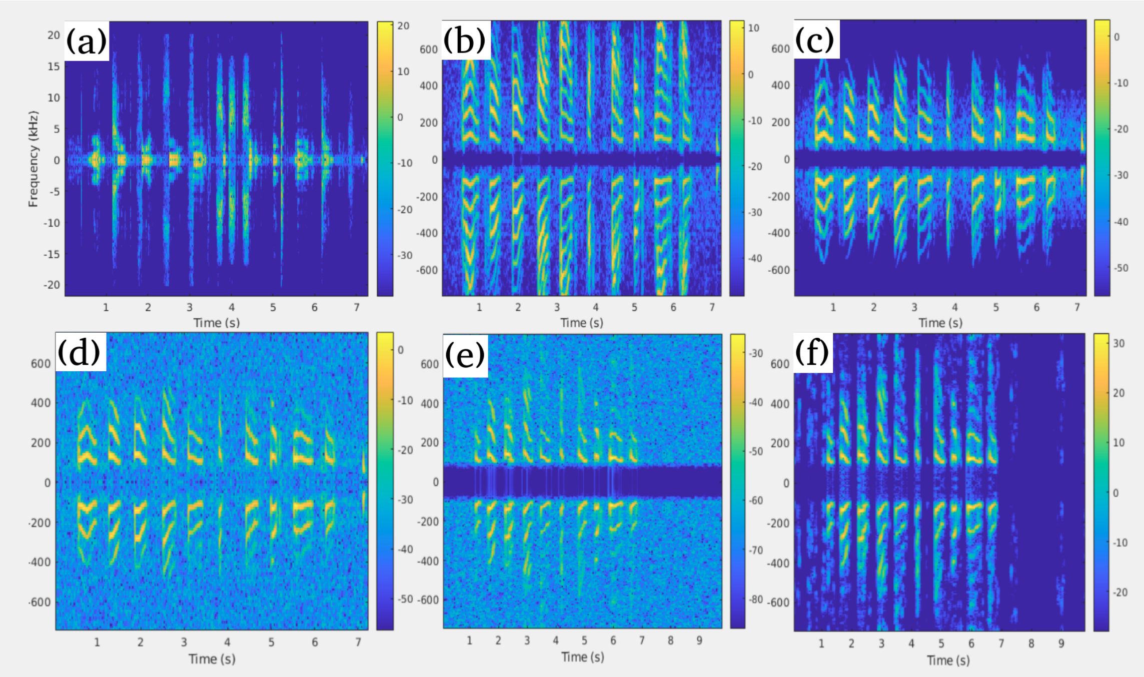


Figure 4.6: *Gaussian white noise simulation and spectral subtraction denoising*

For additive noise, a simple but powerful denoising solution is spectral subtraction (SS). The underlying idea of SS is straightforward. A typical flow chart of SS is illustrated in Figure 4.7. Suppose the signal $x(t) = s(t) + d(t)$, where $s(t)$ is the signal part and $d(t)$ is the noise part. $x(t)$ is divided into overlapping frames. Then after fast Fourier transform, the spectrum of noise $\hat{D}(w)$ can be estimated and updated continuously using pure noise frames. What remains to do is to subtract the noise spectrum amplitude from the noisy signal, i.e.

$$|\hat{X}(w)| = \sqrt{|X(w)|^2 - |\hat{D}(w)|^2}. \tag{4.5}$$

The spectrogram amplitude is then multiplied with the original phase to get an estimation of the clean signal s(t), i.e., $\hat{x}(t)$.

Famous variants of the SS methods are linear SS [Bol79], non-linear SS [BSM79] and multi-band SS [KL02], whose implementations can be found at [Zavb] [Zava] [Zavc]. The

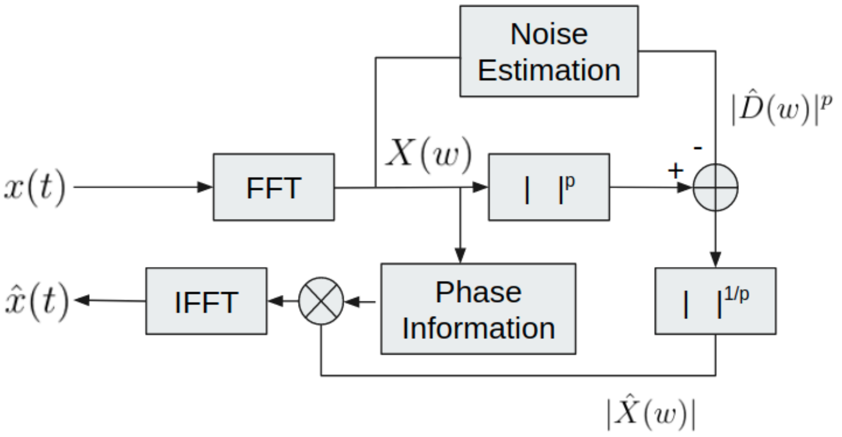


Figure 4.7: *Flow chart of a typical spectral subtraction pipeline*

spectral subtraction algorithm has some inherent problems, for example, music noise introduced by noise residuals. However, even the simplest linear SS works fine on our data. Figure 4.6(f) shows an example of applying linear spectral subtraction on the collected data. The output of such a filter is then normalized and output as a .wav file to generate the recovered sound. Also, we perform a Short Time Fourier Transform (STFT) to visualize the recovered sound. In the output of STFT, the X-axis stands for time while the Y-axis represents frequency.

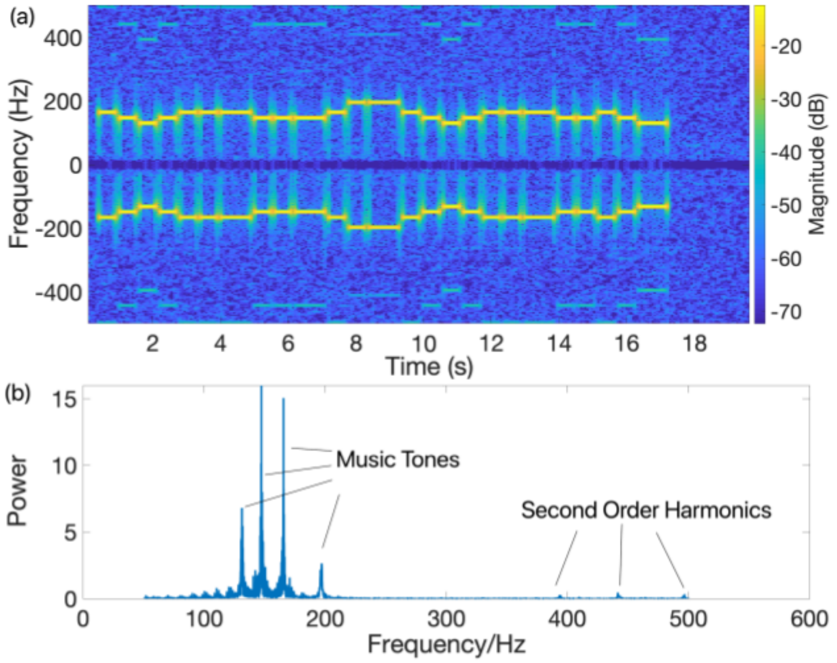


Figure 4.8: *Results of the proof-of-concept experiments*

The results of the proof-of-concept experiments are shown in Figure 4.8, where we play a single tone song *Mary has a little lamb*. From the visualization, we can see that all the notes are recovered clearly.

We also notice that sometimes there is an interference of 60Hz and its multiples. This is probably due to the complicated power frequency electromagnetic field emitted by the circuit regulator or other devices. We assume an IIR comb filter can help us filter out the power frequency components.

In the next chapter, we will explore the limits of our UWB audio sensing system. Factors including distance, target frequency, speaker placement, and the blockage will be studied by field experiments.

CHAPTER 5

Micro-benchmarks and Sample Applications

5.1 Micro-benchmarks

In this section, we aim to test the limits of our UWB audio sensing system. The transmission of wireless signals is affected by a number of factors, including but not limited to distance, angle and blockage. To study how our system performs under real-world conditions, we perform micro benchmark tests to determine the contribution of four different factors: (1) distance, (2) blockage, (3) speaker placement angle, and (4) sound frequency.

Distance. As we have analyzed in Chapter 4, the system suffers from additive noise close to Gaussian white noise. Meanwhile, it is common sense that wireless signal strength will decay in space. In this test, we play a single tone "C4" music tone whose frequency is 261.63Hz using a studio speaker. The speaker volume is tuned to 77.1 dB/SPL at one meter distance measured by a microphone meter. Our system is placed in front of the speaker at a distance starting from 50 cm and increasing by 50 cm each time. The data is send to a PC to be processed offline. Figure 5.1(a) shows a typical setting.

We use signal-to-noise-ratio to measure the quality of the recovered sound. The signal noise ratio (SNR) is defined as

$$SNR = 10log_{10}(\frac{E_s}{E_n}), \tag{5.1}$$

where E_s and E_n are the energy of the signal and the noise, separately. In experiments, we notice that the recovered sound may have a slight frequency drift from the test tone probably due to the sampling clock error. Thus we estimate the power the of signal in frequency domain by firstly localize the peak in the spectrum near the target frequency (261.63Hz) and then sum the energy in nearby frequency bins (within 5 Hz) as an estimation of signal frequency, while using the rest energy as an estimation of noise. Due to the fact that the noise maybe time-varying, we employ a 1.5 second window with a 500 ms overlap to calculate the short-time SNR. Also, in order to authentically measure the noise, we remove the denoising stage from the signal processing pipeline.

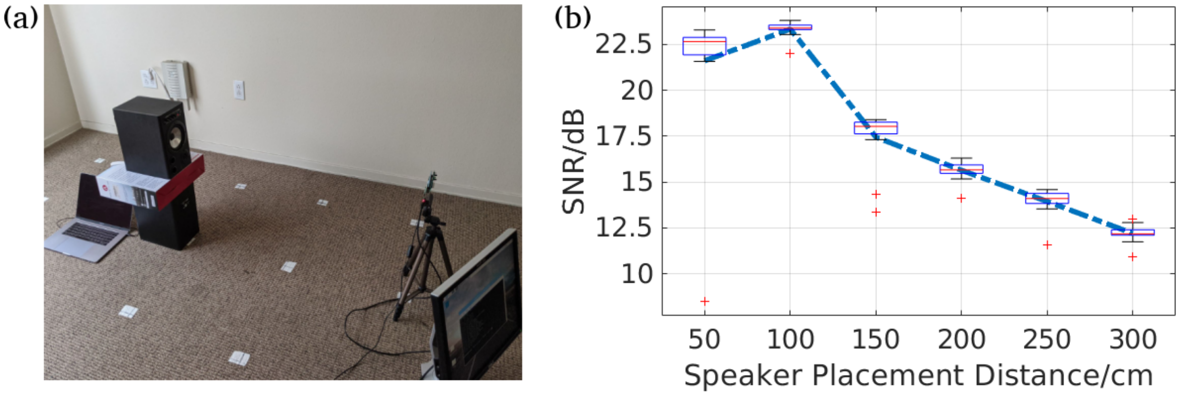


Figure 5.1: *SNR vs speaker placement distance: (a) Experiment Setup (b) SNR plot across different distances*

Figure 5.1 shows the results. We can see that the system can operate smoothly in free space for 3 meters with reasonable SNR. The general trend is that the SNR is decreasing linearly over distances and still have capabilities for a further distance. The reason that 50 cm SNR is worse than that of 100 cm is probably due to the ”current noise” as well as harmonics caused by clipping are captured by the system, and are count into noise energy.

Blockage. At the beginning of this part, we discussed the trade-off between the spatial resolution and the penetration ability. We expect our system can operate in non-line-of-sight (NLOS) scenarios, i.e., can recover the sound behind light building materials. The experiment settings are generally the same as the distance experiments. The difference is that the speaker and the sensor is separated by a hollow wooden wall (a normal wall between the bedroom and the living room) with a thickness of 11.5 cm. The settings are shown in Figure 5.2(a), where the speaker is put inside the bedroom and the sensor is placed outside in the living room.

Figure 5.2(b) displays the results of through-wall sound retrieving experiments. Generally the SNR is still following the linear trend. Compared to that of free space, the through-wall results suffered a one-time loss of around 5dB. Also, the slope of SNR dropping is slightly steeper than that of free space. For human ear, at the distance of 2.92m, even the SNR becomes relatively low, the test tone is still clearly audible. For machine

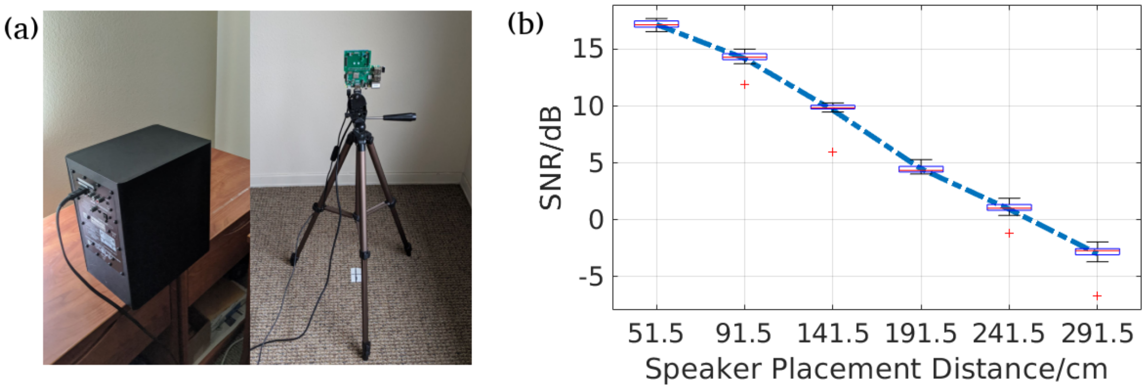


Figure 5.2: *SNR vs through-wall speaker placement distance: (a) Experiment Setup (b) SNR plot across different distances*

processing, however, the effective range may reduce since current models for audio processing are usually not very robust noise. Generally speaking, the system can operate through a wall within a range of 2.5 meters with reasonable performance down-gradation.

Speaker Placement Angle. In reality, it is not practical to require the speaker to be always aligned to the sensor. Thus it is necessary to understand the relative angle between the speaker diaphragm surface and the sensor Tx-Rx surface. Then there are two separate problems. Firstly, if the speaker is aligned to the sensor, but the sensor is pointing at another direction, then the recovered sound quality will be negatively affected. Note that currently our sensor is equipped with a directional antenna whose 5dB main lobe is 50° both in elevation and azimuth. Thus for this problem, we argue that this problem can be solved with a mechanical system that rotates the sensor, and the target angle can be given by searching for the direction that gives the highest SNR.

Secondly, if the sensor beam is in the right direction, but the speaker is placed at different angles, then the performance may vary. Intuitively, the incoming signal beam will experience diffuse reflection on the speaker cone, and then a certain proportion of the signal will be reflected back. We use experiments to measure the effect of speaker placement angle, whose settings is shown in Figure 5.3(a). The distance is fixed at one meter and the speaker is rotated to a few certain angles. The speaker volume is turned down to 74.5 dB/SPL at 1m distance.

From Figure 5.3(b), we can see that the SNR drops quickly after the speaker cone deviates over 10 degrees. Then the SNR fluctuates up and down and hovers at around

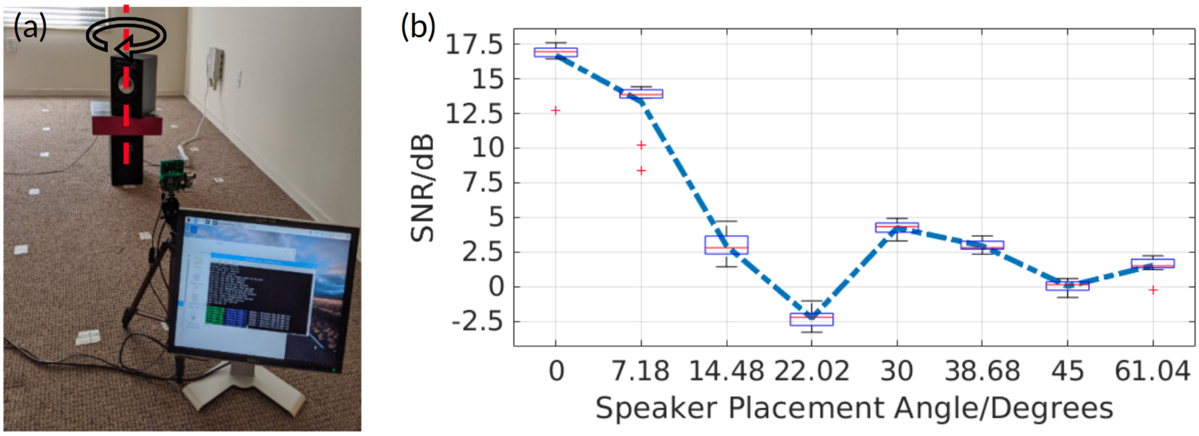


Figure 5.3: *SNR vs speaker placement angle: (a) Experiment Setup (b) SNR plot across different angles*

2.5dB. This observation coincides with our intuition: The fluctuation is owing to the speaker cone geometry. At some certain angles the direct reflection will be stronger than other angles. The diffused reflection signal can still provide clues about the cone vibration.

So far, we have only considered the direct reflections. Our experiments show that it is possible to perform *Reflective Audio Sensing* using UWB devices, which is demonstrated in Figure 5.4. The spectrum in the middle and the spectrogram to the right both prove that sound source vibrations can be recovered from strong reflective paths. This observation bro-dens the application case of our system.

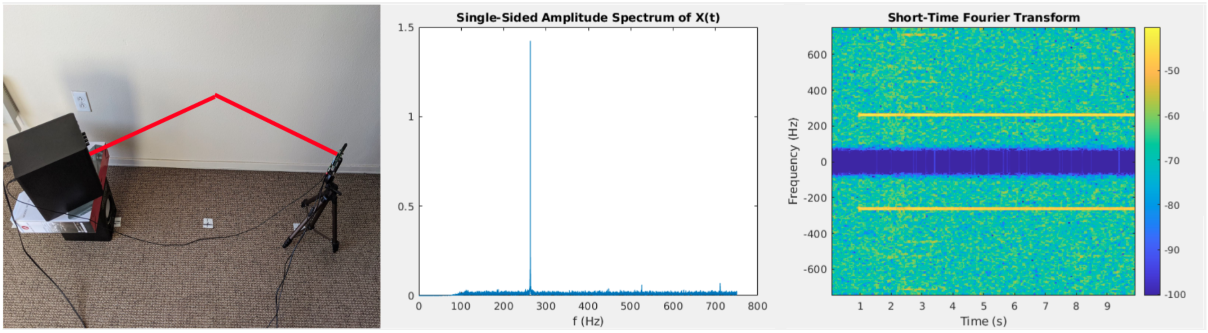


Figure 5.4: *Example of reflective audio sensing*

Sound Frequency. The final characteristic of the system that we want to test is the frequency response. Placing the speaker at 1m distance, we play test tones from 100Hz to 600Hz with increments of 100Hz. Our results in Figure 5.5 generally show a loss-pass trend, which provide a hint that we should use pre-emphasis in signal processing to compensate for this low pass nature.

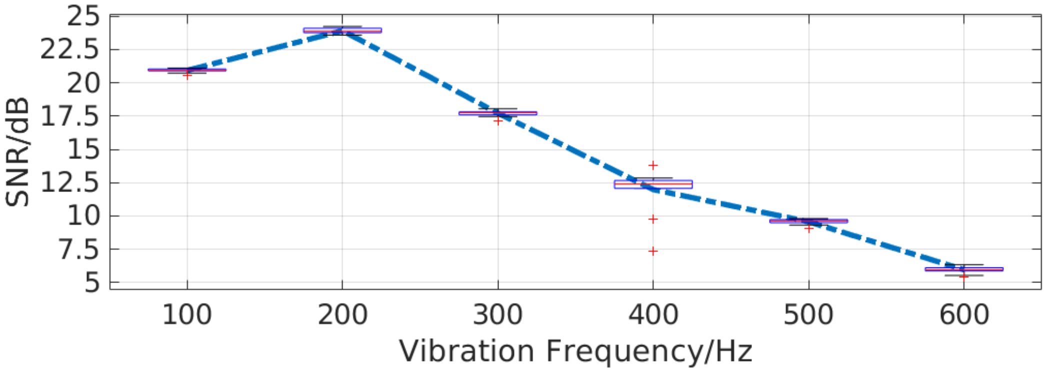


Figure 5.5: *UWB audio sensing system frequency response*

5.2 Sample Applications

In the previous section, we explore the limits of such a UWB audio sensing system. In this section, we will provide some sample use case of our system.

Sound source distance measurement. One of the great advantages of using UWB lies in the fine ranging resolution brought by its ultra-wide signal bandwidth. As we introduced at the beginning of this part, UWB audio sensing is *distance-aware*, which means we can not only recover the sound-related vibrations, but also know how far the vibrating source is from the sensors. We aggregate the data from the first two experiments described in Section 5.1 and estimate the speaker distance from the data. The empirical cumulative distribution function (CDF) plot of estimation error is shown in Figure 5.6.

The mean error is 5.31cm, the median error is 5.24cm, the maximum error is 8.32cm and the standard deviation is 1.63cm. We can see that the our system can give distance estimation within two distance bins, which is pretty accurate.

Through-wall acoustic eavesdropping. People feel that whatever they listen to inside their home is private, that no one can snoop on its contents. However, as a common proverb goes: ”Walls have ears”. It is a saying that can be brought to life with the help of such a UWB audio sensing system. Our experiments in Section 5.1(2) show that our system can operate without line-of-sight, which enables our system to perform through wall eavesdropping just like [WWZ15]. We have also proved that our system is capable of

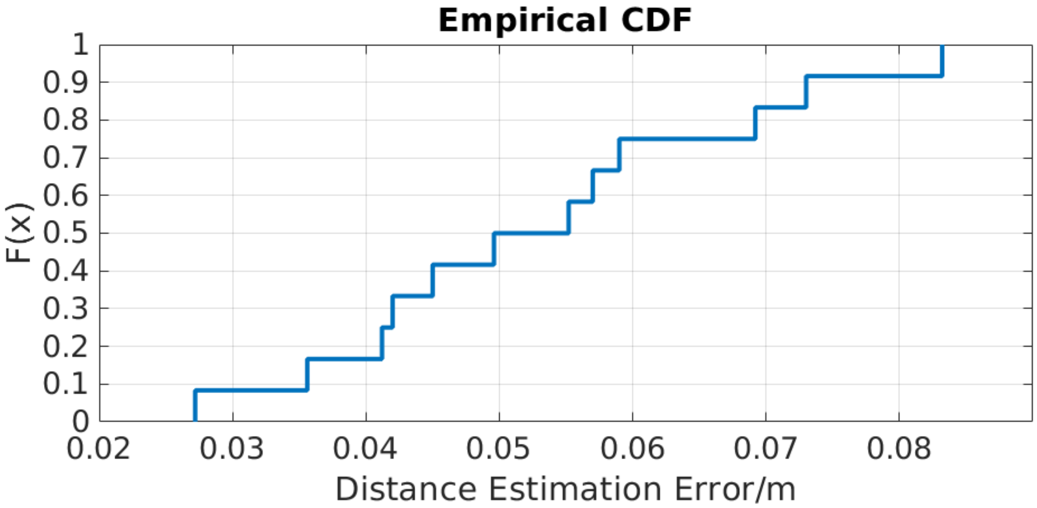


Figure 5.6: *UWB audio sensing system distance estimation error empirical CDF curve*

measuring the distance of the speaker. Suppose we have multiple instances of our system, it may be possible also to localize the sound source by trilateration.

Acoustic feedback in vacuum. The underlying philosophy of acoustic eavesdropping with wireless vibrometry is that, radio frequency wireless signals can penetrate materials, reach the vibrating source directly and come back, without needing mechanical sound waves that need to transmit through mediums. Note that the materials that block the sound waves may not necessarily cause trouble for electromagnetic waves, for instance, sound-absorbing foams or a double-layer vacuum glass wall. Following this lead, we imaging that such a system can be used to provide acoustic feedback in scientific experiments that involves a vacuum chamber, or in space missions where sound cannot propagate.

Sound separation. Sound separation is an active research field. Once voices and speeches are mixed in the microphone recordings, it is difficult to separate them apart as they are entangled both in time and frequency domain. People have been trying to use deep learning-based methods to solve this problem [KWE19]. Our system proposes a new potential solution to this problem - separate the sound in UWB *fast time* domain. Our system is able to deal with multiple simultaneous sounds occurring at different distances, and separate them apart based on the time-of-flight (or *fast time*). Figure 5.7 demonstrates our experiments on sound separation. As shown in part (a), the two speakers are placed at different distances, one at 58cm playing *Mary has a little lamb* and another

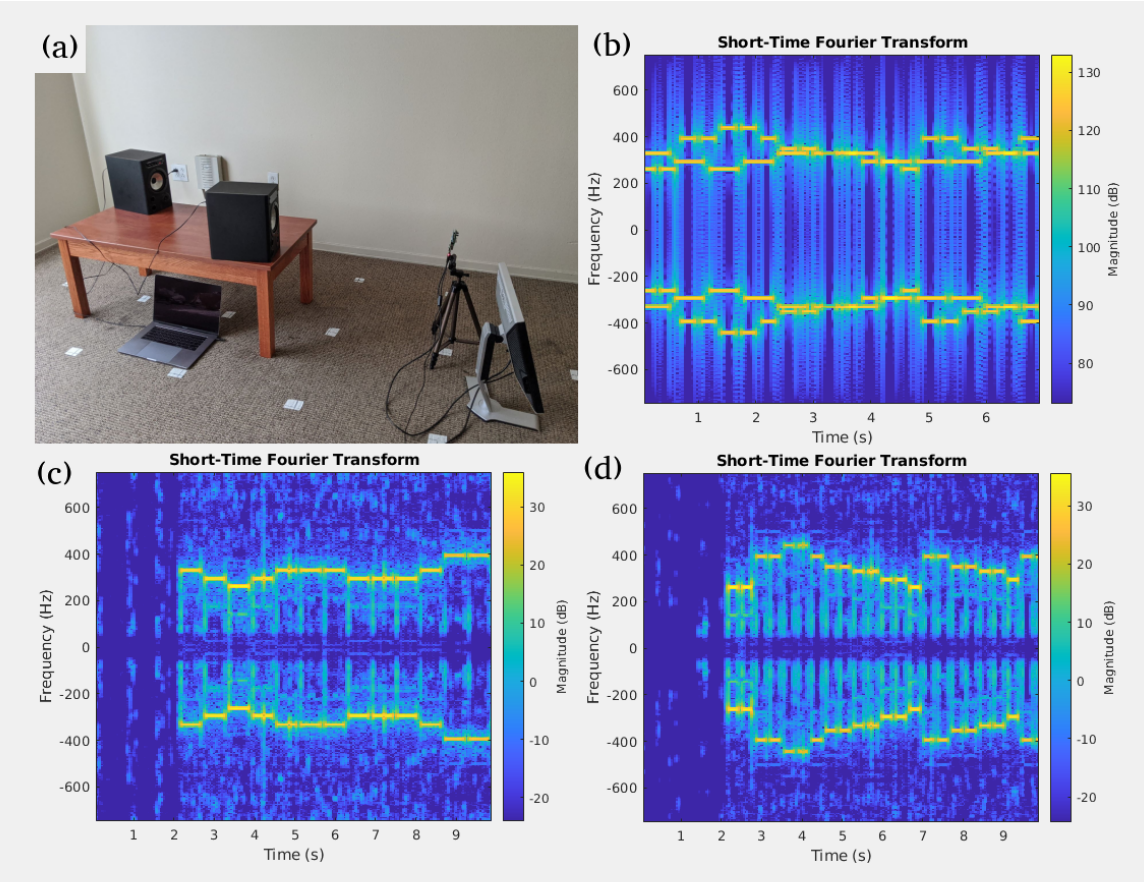


Figure 5.7: *Sound separation using UWB audio sensing system*

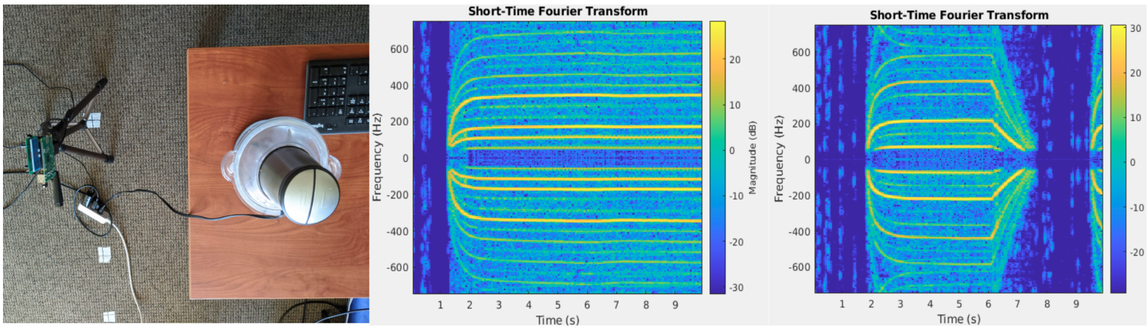


Figure 5.8: *Example on sensing chopper vibrations*

at 122cm playing *Twinkle twinkle little star*. The spectrogram of the round recorded by a microphone is shown in part(b), where the two songs are entangled cannot be easily separated.

Part (c) and (d) show the output of our system. By selecting different distance bins, we can separate the two songs without any residual.

Sound recovery from household tools VibroSight [ZLH18] employs lasers to recover the sound produced by smart home appliances so that their daily usage can be analyzed. Our system also expresses similar potentials in recovering the sound from household tools. Figure 5.8 is an example of sensing the vibration of a food chopper. We turn the chopper to two different speed settings, and the recovered sound clearly indicates the speed, the start, and the end phase of the food chopping spins.

CHAPTER 6

Related Work

In this chapter, we summarize some related work that involves wireless vibrometry and applications of the UWB devices and inspires our work.

Sound recovery using side channels has been a hot field of research in the past decade or so. Most use cases of such techniques are of adversarial nature. As voice-based user interfaces have become more common, so have the attacks on them. Wei et al. [WWZ15] use the acoustic-radio transformation (ART) algorithm which can recover sound produced by a loudspeaker. More specifically, they inspect subtle disturbances in WiFi signals to recover audio.

Vibration inspection has also been used to sense mechanical vibrations to study whether a system is working accurately or not. For example, every building or bridge has a "fundamental frequency" at which it vibrates. However, due to wind or earthquakes, these vibrations may increase and threaten the stability of the structure. Yang et al. [YLL16], use RFID (Radio-frequency Identification) to analyze high-frequency mechanical vibrations in machines and structures using low-frequency RFID solutions. Their solution, Tagbeat, can troubleshoot automobile engines and can even monitor the shaking of blood samples in a high-speed centrifuge.

In WaveEar [XLZ19], the authors create a Voice-User Interface (VUI) using mmWave radar. The authors point the radar at the vocal cords of a person and collect his/her vocal vibrations. The received signal, containing the speech information, is fed to their novel deep neural network for recovering the voice through exhaustive extraction. This technique is better than microphone based sound sensing because it cannot be easily polluted by ambient noise.

There are several other methods to sense vibrations. In Vibrosight [ZLH18], the authors sense physical vibrations at one specific point using long-range laser vibrometry.

This type of vibration sensing technique can perform robustly even in noisy environments and allows the detection of simultaneous activities.

Side channel attacks that take advantage of these vibrations have also been studied in depth. In [KXF], Kwong et al. demonstrate that the mechanical components in magnetic hard disk drives behave as microphones with sufficient precision to extract and parse human speech. This type of technique can be used to record secret conversations because people do not suspect such components to be malicious. This technique is so accurate that Shazam [shaa], a popular mobile app for recognizing music, was able to recognize a song recorded by the hard disk drive.

UWB radars are not new to the world of sensing and have in-fact been used for various sensing tasks. One example is the 'human presence sensor' created by Novelda[nov]. This UWB based device can be used to find out if a user is present in front of a device or not. One additional advantage is that the radar chip has very small dimensions and can be hidden inside the plain looking casing of any objects. Those features ensure that we can achieve the desired functionality while maintaining the aesthetics and avoiding any physical security concerns.

Another example of sensing using UWB radars is called V^2iFi [ZCC20] in which the authors use a COTS UWB radio to reliably detect a driver's vital signs (such as respiratory rate, heart rate, and heart rate variability) under driving conditions in the presence of passengers.

CHAPTER 7

Discussions and Summary

We explore the characteristics and potential applications of our system in Chapter 5. However, this system still has some limitations that will point us to some future directions. We will also provide a short summary of this part of this dissertation in this Chapter.

7.1 Limitations and Future Work

Sampling Rate. A major limitation of our current system is the relatively low sampling rate. While UWB radars have a fast sampling system on the fast time(collecting responses), the vibrations can only be recovered by analysing a series of frames. Thus what matters is the granularity of the *slow time*, i.e., frame rate. Currently, the frame rate cannot go beyond 1.6kHz owing to that fact that the X4 UWB radar chip only caches the last frame it receives, and that the data transmission speed is limited by the SPI interface. In order to perform recovery for human voice, we need a sampling rate of at least 3kHz (the sampling rate of a landline telephone) to ensure understanding of human speech, as the voiceless consonants that are critical in human speech understanding usually have only high-frequency components. In the future, we expect to enable Quad Serial Peripheral Interface (QSPI) or substitute the Raspberry Pi with FPGA to increase the data transmission rate.

Noise. Even though Section 5.1 shows that our system is capable of operating in complicated environments, it still suffers from a drop in SNR under unfavorable conditions. In worst cases, the recovered sound may downgrade below the quality threshold for machine processing or the human auditory system. Here the performance of our system is limited by the radar board and antennas. We expect that the innovative design of the hardware that increases the transmission power may help, as when we increase the power setting from ”low” to ”high”, the effective range of our system increases. Adding a controllable low-noise amplifier and a power amplifier between the X4 SoC and the antenna might help to increase the system performance.

Direction. As discussed before, our system is currently using a directional antenna, which implies that the sensor can only work if it is pointing to the direction of the sound source. In the future, we hope this direction problem can be solved by either a mechanical system that can rotate the sensor, or a system using multiple omnidirectional antennas so that we can use blind beamforming to localize the vibrating targets.

Target Material. Different materials of the target interact with the UWB radar wave differently. It could reflect, absorb, be penetrated by the signal or show a combination of the three in most cases. For example, our system may work perfectly on metal and wooden speaker diaphragms, but its effective range might drop in case of paper cones. Moreover, our tests reveal that the UWB-based system cannot recover voice from a human throat. Those are the intrinsic disadvantages of using wireless vibrometry. One of the future directions can be building a comprehensive wireless vibrometry system that combines modalities like mmWave, impulse UWB, laser, etc. Operating at different frequency ranges, those technologies can compensate for each other and make a more powerful sensing system.

Further application-specific signal processing. Currently, our prototype system only focuses on the audio recovery. In the future, we expect it to become an interface that can be integrated with other technologies in audio processing. For example, it might be possible to couple our system with an end-to-end automatic speech recognition (ASR) system in order to generate a transcript of the recovered audio clip [WHK18].

7.2 Summary

In this part of the dissertation, we propose an audio sensing system using impulse radio Ultra-wideband radar. We mathematically prove the theory of recovering audio using impulse-based wireless vibrometry. We also build a real-world UWB radar system capable of audio sensing, and provide a learning-free signal processing pipeline. Our results show that this system is able to retrieve the sound directly from the vibrating source and also estimate the distance from the source to the sensor. Such characteristics enable many applications like sound separation and through-wall acoustic eavesdropping. We also test the limits of our system and point the directions of future research. We believe that it is a step towards making audio sensing more robust to use UWB radar as an alternative sensing modality.

In the next part, we will focus on the security side of audio sensing by looking at the design of an audio anti-spoofing system that checks the authenticity of collected audio clips.

Part II

Audio Anti-Spoofing Using Deep Neural Networks

CHAPTER 8

Introduction to Audio Spoofing Detection

The state-of-art models for speech synthesis and voice conversion are capable of generating synthetic speech that is perceptually indistinguishable from bona fide human speech. These methods represent a threat to the automatic speaker verification (ASV) systems. Additionally, replay attacks where the attacker uses a speaker to replay a previously recorded genuine human speech are also possible. In this part, we present our deep learning solution in the ASVSpoof2019 challenge [con19] which aims to develop countermeasure systems that distinguish between spoofing attacks and genuine speeches. Our model is inspired by the success of residual convolutional networks in many classification tasks. We build three variants of a residual convolutional neural network that accept different feature representations (MFCC, log-magnitude spectrogram, and CQCC) of the input. We compare the performance achieved by our model variants and the competition baseline models. In the *logical access* scenario, the fusion of our models achieves zero tandem detection cost function (t-DCF) and zero equal error rate (EER), as evaluated on the development set. On the evaluation set, our model fusion improves the t-DCF and EER by 25% compared to the baseline algorithms. Against *physical access* replay attacks, our model fusion improves the baseline algorithms t-DCF and EER scores by 71% and 75% on the evaluation set, respectively.

8.1 Background

Over the past decade, voice control has gained popularity as a practical and comfortable interface between users and smart devices. Due to the security and privacy-sensitive nature of many applications (e.g., banking, health, and smart home) running on these devices, automatic speaker verification (ASV) [EKY13] techniques have emerged as a form of biometric identification of the speaker. ASV system compares a speech sample provided by the user and a sample stored in the database to determine if the two samples come from the same speaker. The number of such voice biometric systems, even under conservative estimations, will easily exceed 600 million around the world, according to a report released by Opus Research in 2016 [Mil16]. A single breach of such an ASV system, e.g., one used for online banking user identification, might lead to a significant loss of property. Therefore, it becomes increasingly urgent to distinguish between the bona fide human speech and spoofed audios.

However, ASV systems are currently threatened by replay [KEY17] and audio spoofing attacks where an attacker utilizes techniques such as voice conversion (VC) or speech synthesis (SS) to gain illegitimate control over user devices. Speech synthesis [ODZ16, WWK16, JBW18] and voice conversion [TCS16, HLH18] have progressed a lot over the past decade reaching the point where it has become very challenging to differentiate between their results and genuine users' speech. To enhance reliability against attacks, we combine ASV systems with audio spoofing detection systems that compute countermeasure scores to distinguish between spoofed and bona fide (genuine) speech. The automatic speaker verification spoofing and countermeasure challenge (ASVSpoof [EKY13, WKE15, KEY17, con19]) competitions have emerged to assess the state-of-art methods for spoofing detection and promote further research in this critical challenge.

The first edition of the competition, ASVSpoof2015 [WKE15], focused on ***logical access*** scenarios where the attacker is using text-to-speech (TTS) and voice conversion (VC) algorithms. The second edition of ASVSpoof competition, ASVSpoof2017 [KEY17], focused on the ***physical access*** scenario where the attacker is performing *replay attack* by recording the genuine speech and then replay it to deceive the ASV system. The new edition of the competition, ASVSpoof2019 [con19], extends the previous versions in several directions. First, it considers all three major forms of attacks: SS, VC, and replay attacks. In addition, the latest and strongest spoof algorithms are used to generate more natural counterexamples for spoof detection systems. Finally, while the previous competitions used the equal error rate (EER) as an evaluation metric, ASVSpoof 2019 adopts a newly proposed tandem decision cost function (t-DCF) as its primary metric and leaves EER as a secondary metric. t-DCF considers the cooperation between the ASV system and the Spoofing Countermeasure system, and defines the cost of different system failure types in a more detailed manner.

In this work, we present our models submitted for the ASVSpoof2019 competition [con19]. Inspired by the success of deep neural networks in many tasks [AAA16, SLJ15, EKN17], we pick a deep neural model as our model family. Among deep neural networks, convolutional networks have been the most successful in image classification [SLJ15], and have been recently applied to other data modalities such as Speech [AMJ14, AAA16], text [ZZL15] and ECG signals [RHH17]. We consider different feature extraction algorithms to convert the input (raw time-domain speech waveform) into a 2D feature representation. That 2D feature representation is fed as an input into our convolutional model. A practical challenge in training very deep (consisting of many layers) convolutional networks is vanishing gradients that makes it hard for lower-layers (closer to input) to receive useful update signals during the training [HZR16]. To overcome this issue, [HZR16] recently proposed an effective solution called *residual networks* which employ skip connections that act as shortcuts allowing training updates to back-propagate faster towards the lower layers during training. Therefore, we also consider adding residual links to improve and stabilize the training of our models. A detailed description of our model architecture is provided in Section 9.3. Finally, we show how the fusion of countermeasure (**CM**) scores produced by models trained on different features help to increase the accuracy of the spoofing detection.

Our contribution in this paper is threefold. First, we design and implement a deep residual convolutional network to perform audio spoofing detection. Our models are released as open source[1]. Second, we provide a comparison between the performance of three different feature extraction algorithms (MFCC, log-magnitude Spectrogram, and CQCC). Third, we evaluate the performance of our residual network with varying choices of input features against the two attack scenarios of ASVSpoof2019 (logical access and physical access) using both the development dataset(including only *known* attacks) and evaluation dataset (including both *known* and *unknown* attacks).

The rest of this paper is organized as follows. Section 8.2 provides a summary of related work. In Chapter 9 we discuss the realization of our countermeasure (**CM**) system, where Section 9.2 describes the feature extraction module of the system, and

[1] https://github.com/nesl/asvspoof2019

Section 9.3 then describes our model architecture design and implementation. Chapter 10 includes our experiment results. Finally, Chapter 11 concludes this part of the dissertation and points the future directions.

8.2 Related Work

While the participants of the previous ASVspoof2015 [WKE15] have built several powerful solutions against audio spoofing, the state-of-the-art of audio spoofing techniques, e.g., TTS [ODZ16, HLH18] and VC [LYT18], has also progressed a lot over the past four years. Likewise, last year's competition ASVSpoof2019 has a more realistic dataset for replay attacks compared to ASVSpoof2017 [KEY17]. Prominent previous approaches against *logical access* attacks include [VMO15], which used spectral-log-filter-bank and relative phase shift features as input to a model combining a deep neural network with support vector machine (SVM) classifier. [CQD15] proposed using a DNN to compute a representative spoofing vector (s-vector). Then it uses normalized Mahalanobis distance between the s-vector and the class representative vector to calculate countermeasure scores. [WYK15] uses relative phase information and group delay feature to train a Gaussian Mixture Model (GMM) for detecting spoofing attacks. Against *replay* attacks, [LNM17] have previously developed a deep learning model combining both CNN and RNN that lead to 6.73% EER on the ASVSpoof2017 evaluation dataset. In ASVSpoof2017, [CXZ17] also used a residual convolutional network, but with a different architecture and input features, to obtain 13.44% EER on the eval set.

CHAPTER 9

Feature Extraction and Model Design

9.1 System Overview

The goal of ASVspoof challenge is to compute a countermeasure (CM) score for each input audio file. A high CM score indicates a bona fide speech, and a low CM score indicates a spoofing attack. In this work, we create a spoof countermeasure (CM) system that works together with any ASV system in a serial manner.

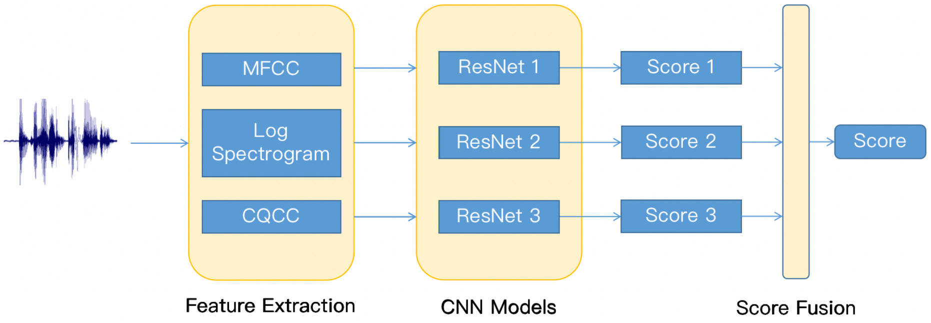


Figure 9.1: *Overall structure of our audio spoofing detction system*

Figure 9.1 shows the overall structure of our audio spoofing detection system, which consists of three major parts. Firstly, various features are extracted from the raw voice waveform. We emphasize on logarithmic power spectrogram as well as cepstral coefficients to capture the time-frequency characteristics of the original sound signal. The features are then used as inputs as residual neural networks (ResNets) to generate log-softmax classification scores, indicating the possibility of whether a voice is bona-fide or not. Finally, we perform a score-level fusion of different combinations of features and network structures to generate a final output score.

The core of this system is deep residual networks that perform binary classification. To prepare the features as the convolutional network inputs, we process the raw audio waveform first a by a feature extraction step, which we will discuss in the next section.

9.2 Feature Extraction

We prepare features from raw audio waveform by one of the following feature extraction algorithms: the Mel-Frequency Cepstral Coefficients (MFCCs), the Constant Q Cepstral Coefficients(CQCCs), and the Logarithmic Magnitude of Short-Time Fourier Transform(log-magnitude STFT).

Mel-frequency Cepstral Coefficients (MFCCs): MFCC is a widely used feature for speech recognition and other applications like music genre classification. Before going to details of MFCC, we first examine the philosophy of using Cepstral Coefficients (CCs) as speech features. The most popular model for human speech generation is the "source-filter model".

In the source-filter model, human speech $y(t)$ is considered to be a convolution of human voice source excitation $s(t)$ (vocal-cord vibration for voiced speech and noise for voiceless speech) and human vocal tract filtering $f(t)$ (e.g., articulation of the lips, the palate and the tongue), i.e.,

$$y(t) = s(t) * f(t). \tag{9.1}$$

In frequency domain, the source and vocal filter are multiplicative, i.e., $Y(w) = S(w) \times F(w)$. Figure 9.2 shows the source filter model in frequency domain. The voiced source is the base frequency of a human speaker and its harmonics. The calculation of CCs is shown in Figure 9.3, where the audio is analyzed through Short-Time Fourier Transform (STFT) to get the spectrogram. Then we take the logarithm on the amplitude to the spectrogram to acquire cepstrum, and finally, use Discrete Cosine Transform (DCT) to convert it to CC's. The logarithm in CC algorithm converts the multiplicative relationship between the source and filter to an additive one, and helps to separate and model the source and filters separately. Different speakers may have different (source) pitch. Meanwhile, the filter transfer functions may carry biometrics of their articulation organs. Intuitively, CCs can help us capture more speaker-identity related information.

Mel-frequency Cepstral Coefficients (MFCC) is a special type of CCs that incorporates

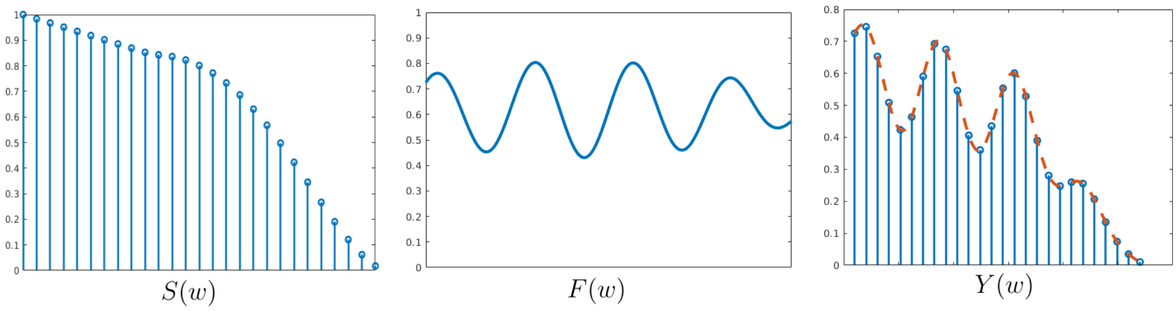


Figure 9.2: *Source-filter model of speech production*

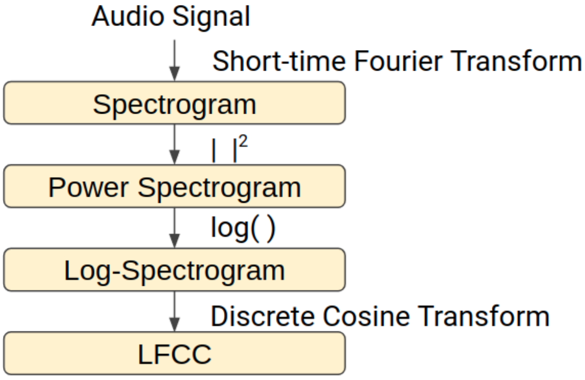


Figure 9.3: *Calculation pipeline of cepstral coefficients*

the characteristics of the human auditory system. Acoustic research reveals that the perceived frequency is different from the natural frequency. The relationship is as follows.

$$mels = 2595log_{10}(1 + \frac{Hz}{700}). \tag{9.2}$$

MFCC is calculated in a way similar to that of vanilla CCs. The difference is that, the spectrogram acquired by computing the short-time-Fourier-transform is passed through a bank of mel-filters shown in 9.4. If we fix the time window, then each filter will output the in-band energy of the original signal. Finally, by cascading the filter band outputs, we get a Mel-Spectrogram that reveals filter bank energies varying over time.

In the MFCC features we use in this system, we pick the first 24 coefficients. We also find the performance can be improved if we concatenate the MFCC with its first-order $\Delta MFCC$ and second derivative $\Delta^2 MFCC$ to produce our feature representation which is a 2D matrix whose x-axis is the time and y-axis is the 72 elements of $(MFCC, \Delta MFCC, \Delta^2 MFCC)$. This improvement is because derivatives of MFCC capture the dynamics in speech, and we believe that the spoofing algorithms may expose

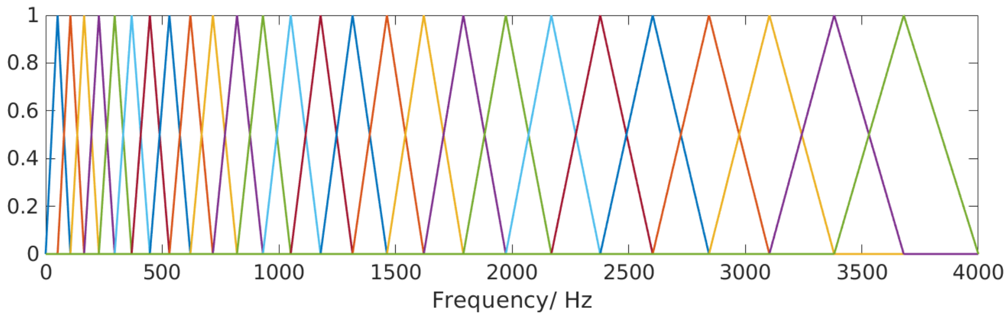


Figure 9.4: *The mel-filter bank in frequency domain*

some unnatural flaws in its dynamics.

Constant Q Cepstral Coefficients(CQCCs): CQCC is another type of CCs proposed especially for speech anti-spoofing. Instead of using STFT, the CQCC uses constant-Q transform(CQT), which was initially proposed for music processing. The intuition behind the usage of CQT is simple. In speech processing, it is a common practice to map the speech into different frequency bands and analyze accordingly. For example, one way to understand the STFT is that it firstly filter the signal into linear frequency bands and then calculate energy in short windows. In CQT, however, the central frequencies of those bands are picked non-linearly. CQT maintains a constant Q-factor, which is defined as the central frequency divided by the bandwidth. As frequency goes higher, the band goes wider to give us geometrically spaced frequency bins.

Figure 9.5 [1] demonstrates a comparison between CQT and STFT. In the first row, each red dot is the intersection of t_s (the center of sampling window in the time domain) and f_s (the central frequency of a filter in the frequency domain). The blue dotted lines indicates the boundary of the filter bands, and the spaces between boundaries are the filter bandwidths. It is clear that, by maintaining a constant Q-factor, we can achieve a higher frequency resolution in low-frequency bands, and a higher time-resolution when frequencies go higher. This might be helpful for computers to perceive speech signal and better extract the identity-related information.

To compute CQCC, after applying CQT, we calculate a power spectrum and take a logarithm like that in a normal CC calculation. Then a *uniform re-sampling* is performed, followed by a DCT to get the CQCCs(which is also a 2D matrix). More details of CQCC

[1] This first row of this figure is directly imported from [TDE17]

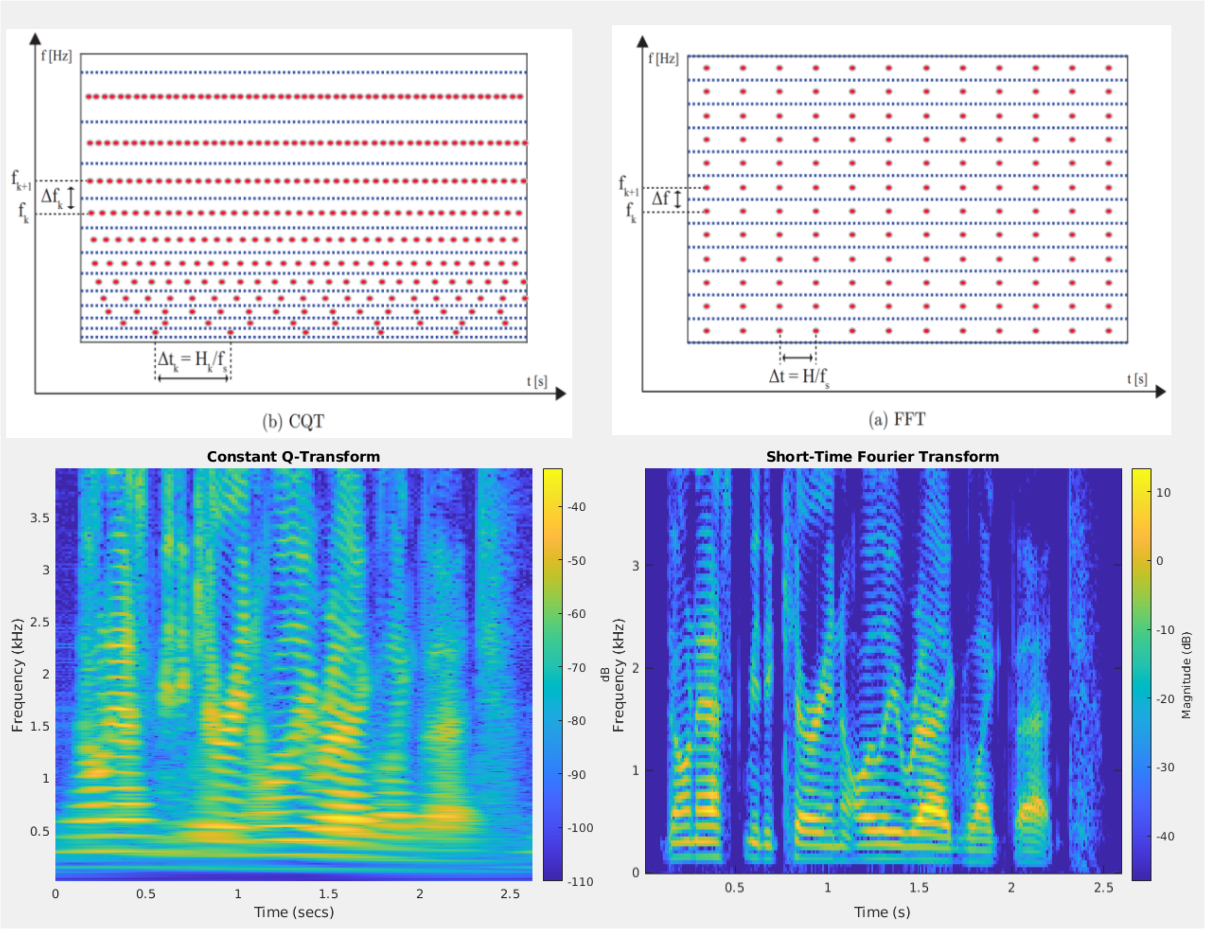


Figure 9.5: *Comparison of STFT and CQT.*

can be found in [TDE17].

Logarithmic Magnitude of STFT: An advantage of deep learning models is their capabilities of representation learning [BCV13, GBC16] by automatically learning high-level features from raw input data. This ability has led to neural models that process raw input images to outperform models dealing with human-engineered features. Inspired by this, we also train models with the log-spectrogram as the input, which is acquired by directly apply logarithm on the magniture of the STFT output (spectrograms).

We first compute the STFT on the raw audio input using hamming windows (window size = 2048 with 25% overlap). Then we calculate the magnitude of each component and convert it to log scale. The output matrix captures the time-frequency characteristics of the input audio waveform and is fed directly as an input to our neural model without any further transformations or conversions. While this input representation is rawer than either MFCC or CQCC, we rely on the representation learning abilities of neural networks to transform this input into higher-level representations within the hidden layers of our model.

9.3 Model Architecture

Convolutional Neural Networks(CNNs) are now the de-facto state-of-the-art in many classification tasks. Our system uses deep residual convolutional networks as a backend to process the feature we proposed in the previous section and generate CM scores.

We build three different models variants `MFCC-ResNet`, `CQCC-ResNet`, and `Spec-ResNet` which process MFCC, CQCC and log-magnitude spectrogram input features, respectively. The three variants have a nearly identical architecture, but they differ from each other in the input shape to accommodate the differences in the dimensions of input features, and consequentially also the number of units in the first fully connected layer which is after the last residual block, as we will explain later.

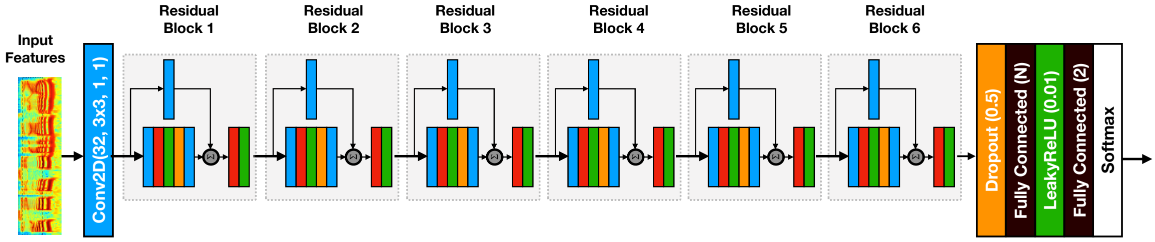


Figure 9.6: *Model architecture for the* `Spec-ResNet` *model. Detailed structure of residual blocks is shown in 9.7.*

Figure 9.6 shows the architecture of the `Spec-ResNet` model that takes the log-magnitude STFT as input features. First, the input is treated as a single channel image and passed through a 2D convolution layer with 32 filters, where filter size $= 3 \times 3$, stride length $= 1$, and padding $= 1$. The output volume of the first convolution layer has 32 channels and is passed through a sequence of 6 *residual blocks*. The output from the last residual block is fed into a dropout layer [SHK14] (with dropout rate $= 50\%$) followed by a hidden fully connected (FC) layer with leaky-ReLU [HZR15] activation function ($\alpha = 0.01$). Outputs from the hidden FC layer are fed into another FC layer with two units that produce classification logits. The logits are finally converted into a probability

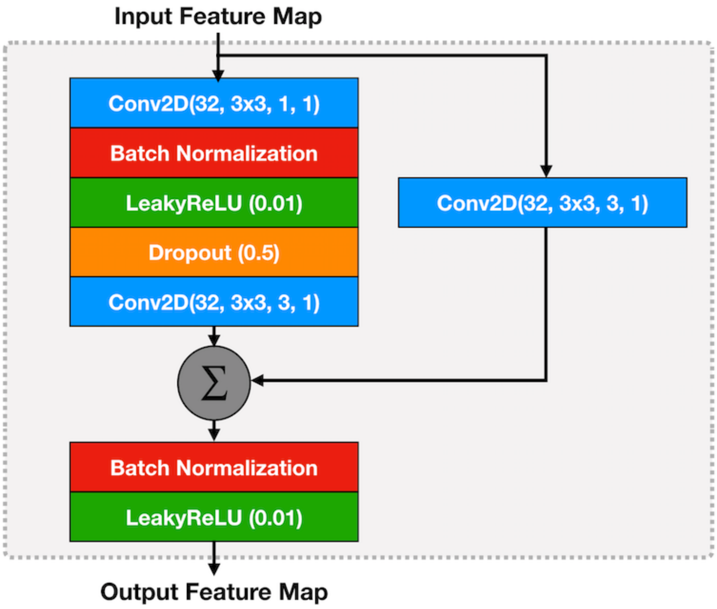


Figure 9.7: *Detailed architecture of the convolution block with residual connection*

distribution using a final `softmax` layer. For specifications of the other two variants of the network, please refer to Figure A.2 in the Appendix.

The structure of a residual block is shown in Figure 9.7. Each residual block has a `Conv2D` layer (32 filters, filter size $= 3 \times 3$, stride $= 1$, padding $= 1$) followed by a batch normalization layer [IS15], a leaky-ReLU activation layer [HZR15], a dropout (with dropout probability $= 0.5$) [SHK14], and another final `Conv2D` layer (also 32 filters and filter size $= 3 \times 3$, but with stride $= 3$ and padding $= 1$). Dropout is used as a regularizer to reduce the model overfitting, and batch normalization [IS15] accelerates the network training progress. A skip-through connection is established by directly add the inputs to the outputs. To guarantee that the dimension agrees, we apply a `Conv2D` layer (32 filters, filter size $= 3 \times 3$, stride $= 3$, padding $= 1$) on the bypass route. Finally, batch normalization [IS15] and leaky-ReLU non-linearlity are used to produce the residual block output.

All models are trained by minimizing a weighted cross-entropy loss function where the ratio of between weights assigned to genuine and spoofed examples are 9:1, in order to mitigate the imbalance in the training data distribution. The cost function is minimized using Adam optimizer [KB14] with learning rate $= 5 \times 10^{-5}$ for 200 epochs with batch size $= 32$. After each epoch, we save the model parameters, and we finally use the parameters with the best performance on the validation dataset.

The final countermeasure score (CM) is computed from the softmax outputs using the log-likelihood ratio.

$$CM(s) = \log(p(\text{bona fide}|s;\theta)) - \log(p(\text{spoof}|s;\theta)) \tag{9.3}$$

where s is the given audio file and θ represents the model parameters.

CHAPTER 10

Model Evaluation

In this chapter, we quantitatively evaluate our model using selected metrics. Information about the dataset and baseline models will also be provided in the chapter. We implement our neural network model using PyTorch [PGC17] and train our models using a desktop machine with TitanX GPU. Feature extraction is done using the *librosa* [MRL15] python library.[1].

10.1 Dataset and Baseline Models

Dataset Overview. The data used in this work comes from ASVSpoof 2019 Challenge database [TWS19]. The database consists of two partitions separately for logical access (LA) and physical access (PA) scenarios. The non-overlapping short audio files come from 78 human speakers (33 males, 45 females), derived originally from the VCTK dataset. Each partition is then divided into three subsets with disjoint sets of speakers: *training* (8 male, 12 female), *development* (4 male, 6 female), and *evaluation* (21 male, 27 female). Data distributions over the subsets are shown in Table 10.1.

	Logical access		**Physical access**	
Subset	Bona fide	Spoof	Bona fide	Spoof
Training	2,580	22,800	5,400	48,600
Development	2,5480	22,296	5,400	24,300
Evaluation	7,355	63,882	18,090	116,640

Table 10.1: *Number of audio files in the ASVspoof2019 dataset.*

***Logical Access* Scenario.** The spoofed audio in the *logical access* scenario is generated

[1] For the CQCC for which we used the MATLAB code provided by competition organizers

using 17 different speech synthesis and voice conversion toolkits. Six of these attack types are considered *known* attacks and are used to generate the training and development datasets while the other 11 attacks are considered *unknown* and are used, along with two of the *known* attacks, to generate the evaluation dataset. For details of the 17 algorithms used to generate the spoofed speech, please refer to the Appendix A(1).

***Physical Access* Scenario.** In *physical access* scenario, the attacker records the voice of a genuine speaker and replay it to attack the system. The attack scenario is shown in Figure 10.1[2]. Human speakers standing at the blue point talk to the ASV system marked in yellow, and an attacker record the speech at the red point. Then during the attacking phase, the attacker replays the recording speech at the blue point to interact with the ASV system.

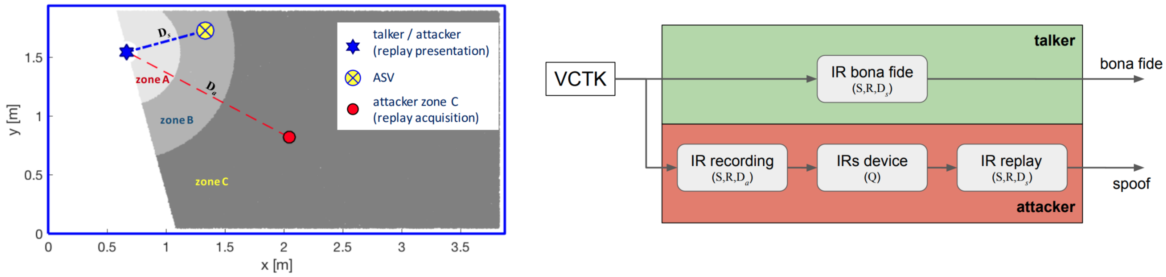


Figure 10.1: *Physical access attack scenario and data generation.*

The data in the physical access partition is simulated. The simulation of genuine speech is conducted using Roomsimove[3]. The room size S, reverberation R, distance D_s (distance between the speaker and the ASV system), and the speaker directivity is taken into consideration. The attacker audio, however, has two extra phases: The first one is the sound propagation channel response. This impulse response (IR) is similar to the IR of a bona fide speech except that the speaker is not facing the recorder, and (2) the distance D_a (distance between the speaker and the recorder) is different from D_s. Secondly, the IR of the recording device also needs to be considered. For example, the recorded speech using a Hi-Fi recorder will be much different from that of a tape recorder. This IR is simulated with the generalized polynomial Hammerstein model and

[2] The figure is directly imported from the challenge evaluation plan [con19].

[3] http://homepages.loria.fr/evincent/software/Roomsimove_1.4.zip

the Synchronized Swept Sine tool[4].

The Training and the Development subsets are recorded and replayed in the 27 different acoustic configurations (3 room sizes S, 3 levels of reverberation R, and 3 speaker-to-ASV distances D_s). Evaluation subset of the *physical access* partition are generated from different impulse responses and therefore represents *unknown* attacks. Also, for spoofed data inside those subsets, there are 9 different attack settings (3 record zone distances D_a and 3 recorder qualities Q).

Baseline Models. For each track of the competition, the organizers have provided implementations for two baseline models, which are using Gaussian mixture models (GMMs) [RR95, RQD00] using the Linear Frequency Cepstral Coefficients (LFCC) and CQCC features.

10.2 Evaluation Metrics

The evaluation scores are computed using the following metrics on both the development dataset (*known* attacks) and evaluation dataset (both *known* and *unknown* attacks):

EER: the Equal Error Rate is used as a secondary metric. EER is determined by the point at which the miss (false negative) rate and false alarm (false positive) rate are equal to each other.

t-DCF [KLD18]: the *tandem detection cost function* is the new primary metric in the ASVSpoof 2019 challenge. It was proposed as a reliable scoring metric to evaluate the combined performance of ASV and CMs. One of the disadvantages of EER is that this metric did not take the cooperation between the ASV system and the CM system. For example, under a zero-effort attack, where a genuine human claims himself to be someone else without any effort of voice imitation, the CM system will give a high CM score as the voice is indeed a human voice. However, the EER will drop as it is actually an attack. In reality, it is the job of the ASV system to reject such attacks.

As we mentioned before, our CM system work together will the ASV system in a serial manner. We can classify the system input into three categories: target speech, non-target

[4] https://ant-novak.com/pages/sss/

speech (human voice of another user), and spoof speech, with probability π_{tar}, π_{non} and π_{spoof}. An ideal CM system will assign a low score to the third case and a high score on the first two. Similarly, an ideal ASV system should only accept the target speech while rejecting the rest.

Both the ASV system and the CM system has two options: reject or accept. We define "miss" as the case that a legitimate trial is rejected, and "fa" (false acceptance) as a case where an attack or non-target speech is accepted. Specifically, for a CM system, a "miss" happens if it rejects a human speech, and a "fa" happens when it accepts a spoofed voice. For an ASV system, "miss" denotes the rejection of a target speech, and "fa" means it accepts anything else.

Also, we denote the threshold of CM system as s, where an utterance with score beyond s is treated as a bona fide speech. Similarly, the threshold of the ASV system is denoted as t. In our experiment, t is fixed[5], and s is flexible, so that we can measure the performance of our CM system. Then in total, there are two general cases that will cause a problem.

- The speech is a target speech, but one or more systems in ASV and CM reject this trial. This will cause a legitimate customer to lose his or her access, which will incur a cost $C_{miss} = 1$. The probability of this happening is

$$\begin{aligned} p_a(s) &= \pi_{tar}(p_{miss}^{asv} \times (1 - p_{miss}^{cm}(s)) + p_{miss}^{cm}(s)) \\ &= \pi_{tar}(p_{miss}^{asv} + p_{miss}^{cm}(s) - p_{miss}^{asv} p_{miss}^{cm}(s))). \end{aligned} \tag{10.1}$$

- The speech is a non-target or spoofed speech, and both the ASV system and the CM system let it through. This case will incur a high cost $C_{fa} = 10$. The probability that this case happens is

$$\begin{aligned} p_b(s) &= \pi_{non}(p_{fa}^{asv} \times (1 - p_{miss}^{cm}(s))) + \pi_{spoof} \times p_{fa}^{asv}(s) \times p_{fa}^{asv} \\ &= \pi_{non} p_{fa}^{asv} - \pi_{non} p_{miss}^{cm}(s) p_{fa}^{asv} + \pi_{spoof} p_{fa}^{asv}(s) p_{fa}^{asv}. \end{aligned} \tag{10.2}$$

The final empirical t-DCF is given by,

$$t - DCF = C_{miss} \times p_a(s) + C_{fa} \times p_b(s) = p_a(s) + 10 p_b(s). \tag{10.3}$$

[5] For notation simplicity we will ignore the t in the next few equations

All the π's, p_{fa}^{asv}, p_{miss}^{asv} is known once the ASV system and the dataset are fixed. The only unknown terms are $p_{miss}^{cm}(s)$ and $p_{fa}^{cm}(s)$. They can be calculated with

$$\begin{aligned} p_{miss}^{cm}(s) &= \frac{\text{\# of bona fide speech with score} \leq \text{s}}{\text{\# of bona fide speech}} \\ p_{fa}^{cm}(s) &= \frac{\text{\# of spoofed speech with score} \geq \text{s}}{\text{\# of spoofed speech}}. \end{aligned} \tag{10.4}$$

Finally, the empirical t-DCF is normalized by dividing a t-DCF score from a CM system that either accepts or rejects all the trials, whichever is better. For more details about t-DCF, please refer to [KLD18].

10.3 Results

Table 10.2 shows a comparison between the scores of our three model variants (`MFCC-Resnet`, `Spec-ResNet`, `CQCC-ResNet`) and the baseline algorithms (`LFCC-GMM`, and `CQCC-GMM`) on both the development and evaluation datasets. *Fusion* represents the result of doing a *weighted* average of the individual ResNet models' *CM* scores to provide a final *CM* score, where fusion weights are assigned based on the single model's performance on the validation dataset.

	Logical Access				**Physical Access**			
	Development		Evaluation		Development		Evaluation	
Model	t-DCF	EER	t-DCF	EER	t-DCF	EER	t-DCF	EER
Baseline LFCC-GMM	0.0663	2.71	0.2116	8.09	0.2554	11.96	0.3017	13.54
Baseline CQCC-GMM	0.0123	0.43	0.2366	9.57	0.1953	9.87	0.2454	11.04
MFCC-ResNet	0.1013	3.34	**0.2042**	9.33	0.3770	15.91	-	-
Spec-ResNet	0.0023	0.11	0.2741	9.68	0.0960	3.85	**0.0994**	**3.81**
CQCC-ResNet	0.0002	0.01	0.2166	**7.69**	0.1026	4.30	0.1070	4.43
Fusion	0.0000	0.00	**0.1569**	**6.02**	0.0581	2.65	**0.0693**	**2.78**

Table 10.2: *t-DCF and EER scores for the different models as measured on the development and evaluation sets for both logical and physical access scenarios.*

10.3.1 Logical Access Results

As shown in Table 10.2, Our `Spec-ResNet` and `CQCC-ResNet` have a significantly smaller t-DCF and EER scores than the baseline algorithms on the development set (*known* attacks) of the logical access scenario. The fusion of the models achieves a *perfect* score of zero EER and t-DCF on the development set. However, in the evaluation set results, our models outperform the baseline models only in the EER of `CQCC-ResNet` and the t-DCF score of `MFCC-ResNet`. This difference between results on the development set and the evaluation set (contains unseen attack types) highlights the difficulty of generalizing a spoofing detection system to *unknown* attack algorithms. Nevertheless, our model fusion shows t-DCF = 0.1569 and EER = 6.02, which are approximately a 25% improvement over the best scores of baseline algorithms.

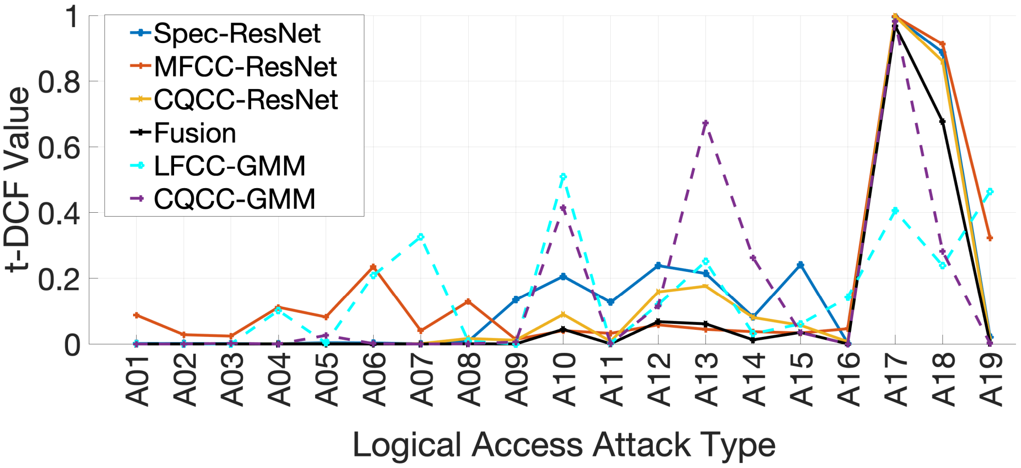


Figure 10.2: *t-DCF scores of different models against different attack types in the logical access scenario.*

To provide a better analysis of the performance of our model against both *known* and *unknown* attacks, the t-DCF scores of our models against each attack type are shown in Figure 10.2.

Attacks from A01 to A06 come from the development set, and are *known* attacks (i.e., the same attacks as that used in the training set). Meanwhile, attacks from A07 to A19 are the 11 *unknown* and two *known* attacks, where (A16 = A04, A06 = A16) are known algorithms. From Figure 10.2, we can see that our models achieve near-zero t-DCF on all known attacks. Meanwhile, it still works well against most attack types except for only two types of *unknown* attacks, namely A17 and A18.

Both A17 and A18 are voice conversion algorithms. A17 uses Variational Auto-Encoder (VAE) as the acoustic model and uses waveform filtering as the waveform generator. In comparison to the baseline models, the CQCC-GMM model also performs poorly on A17 (t-DCF=0.9820). It would be interesting in the future to understand when this VAE based method is capable of deceiving CQCC and neural networks. A18 uses i-Vector and PLDA as the acoustic model and uses ”MFCC-to-waveform” as a method of waveform generation. Both the CQCC-GMM and LFCC-GMM work fine on A18, so it is possible that ResNet is more vulnerable to this type of vocoder attack.

10.3.2 Physical Access Results

Attack Type	**CQCC-ResNet**		**Spec-ResNet**		**Fusion**	
	t-DCF	EER	t-DCF	EER	t-DCF	EER
AA	0.2857	10.59	**0.2473**	**9.17**	0.1845	6.78
AB	0.0690	2.57	**0.0638**	**2.22**	0.0468	1.77
AC	0.0464	1.75	**0.0436**	**1.56**	0.0219	0.80
BA	0.1404	5.46	**0.1300**	**4.82**	0.0855	3.29
BB	**0.0295**	**1.18**	0.0374	1.34	0.0230	0.79
BC	**0.0213**	**0.84**	0.0240	0.86	0.0086	0.36
CA	0.1173	4.55	**0.1105**	**4.01**	0.0705	2.71
CB	**0.0266**	**1.00**	0.0342	1.19	0.0171	0.59
CC	**0.0209**	**0.82**	0.0254	0.87	0.0074	0.28

Table 10.3: *Detailed comparison between the two best single models and the fusion model in Physical Access scenario under different replay attack settings.*

In the physical access scenario, both Spec-ResNet and CQCC-ResNet have significantly improved both the EER and t-DCF compared to the baseline. As shown in Table 10.2, our best single model (Spec-ResNet) is 50% and 60% better than the best baseline results according to the development set EER and t-DCF, respectively. According to the evaluation set scores, Spec-ResNet reduces the t-DCF and EER of baseline algorithms by 60% and 65%, respectively. Furthermore, the fusion of our models leads to 71% and 75% improvement.

Table 10.3 provides detailed results of model performance over different replay attack settings. Each setting is marked with two letters. The first letter stands for the distance between the recording device and the bona-fide speaker, i.e., D_a. 'A' means 10-50 cm, 'B' means 50-100 cm, and 'C' means >100cm. The second letter indicates the quality of replay devices, where A means perfect, B means high, and C means low. From the results, it is easy to see that, as the distance decreasing and the recording device getting better, the anti-spoof task becomes more and more difficult. The worst results are achieved at setting 'AA'. Another thing to notice is that, while Spec-ResNet is generally performing better than CQCC-ResNet while in some cases like 'BB', 'BC', 'CB', and 'CC', CQCC-ResNet outperforms Spec-ResNet.

Generally, the system performs better on physical access scenarios that on logical access. This is probably caused by the challenge of generalization, as in logical access, most attacks in the testing dataset are diverse and unknown, while in physical access, the features come from the replay channel properties and are easier to learn and generalize.

CHAPTER 11

Conclusion

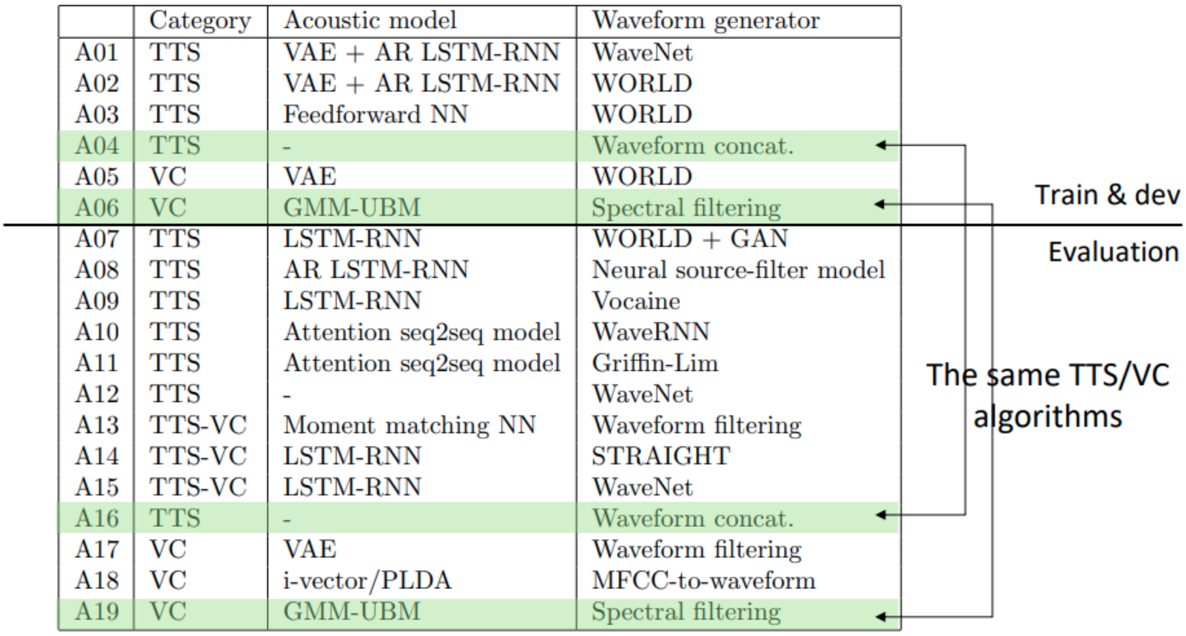
In the second part of this dissertation, we present a novel audio spoofing detection system for both logical access and physical access scenarios. We provide comparisons between the performance of our model combined with three feature different feature extraction algorithms. According to the evaluation dataset scores, against replay attacks, the fusion of our models CM scores improves the t-DCF and EER metrics of baseline algorithm by 71% and 75%, respectively. Also, against the TTS and VC attacks, our fusion of models improves the t-DCF and EER metrics by approximately 25% each. Our future work is to study how to improve the generalization of our model against *unknown* attacks. One possible solution is to employ advanced fusion to build a 'wide-and-deep' network, as proposed in [CKH16]. The key idea of this new proposal is to concatenate the features from each model's last fully connected layers and use a shared softmax layer as the output layer. This might be able to "teach" the networks to collaborate with each other and achieve a better fusion result.

As a summary, in this dissertation, we firstly work towards robust audio sensing using IR-UWB radar. This new sensing modality can give extra information about sound source distance, bringing new applications such as sound separation and through-wall eavesdropping. Then in the second part, we enable secure acoustic sensing by building a DNN-based countermeasure to protect the ASV system from audio spoofing attack. Our results show that this system is not only capable of defending attacks that the system is trained on, but can also generalize to some unseen attack types. Generally speaking, we are making a small step towards robust and secure acoustic sensing through the effort described in this work.

APPENDIX A

Additional Details of Part II

(1) Spoofing Algorithms in Logical Access[1]



	Category	Acoustic model	Waveform generator
A01	TTS	VAE + AR LSTM-RNN	WaveNet
A02	TTS	VAE + AR LSTM-RNN	WORLD
A03	TTS	Feedforward NN	WORLD
A04	TTS	-	Waveform concat.
A05	VC	VAE	WORLD
A06	VC	GMM-UBM	Spectral filtering
A07	TTS	LSTM-RNN	WORLD + GAN
A08	TTS	AR LSTM-RNN	Neural source-filter model
A09	TTS	LSTM-RNN	Vocaine
A10	TTS	Attention seq2seq model	WaveRNN
A11	TTS	Attention seq2seq model	Griffin-Lim
A12	TTS	-	WaveNet
A13	TTS-VC	Moment matching NN	Waveform filtering
A14	TTS-VC	LSTM-RNN	STRAIGHT
A15	TTS-VC	LSTM-RNN	WaveNet
A16	TTS	-	Waveform concat.
A17	VC	VAE	Waveform filtering
A18	VC	i-vector/PLDA	MFCC-to-waveform
A19	VC	GMM-UBM	Spectral filtering

Figure A.1: *Spoofed speech sample generation algorithms used in logical access*

(2) ResNet Model Variants Specifications

[1] **Information directly imported from https://www.asvspoof.org/interspeech2019_slides.pdf**

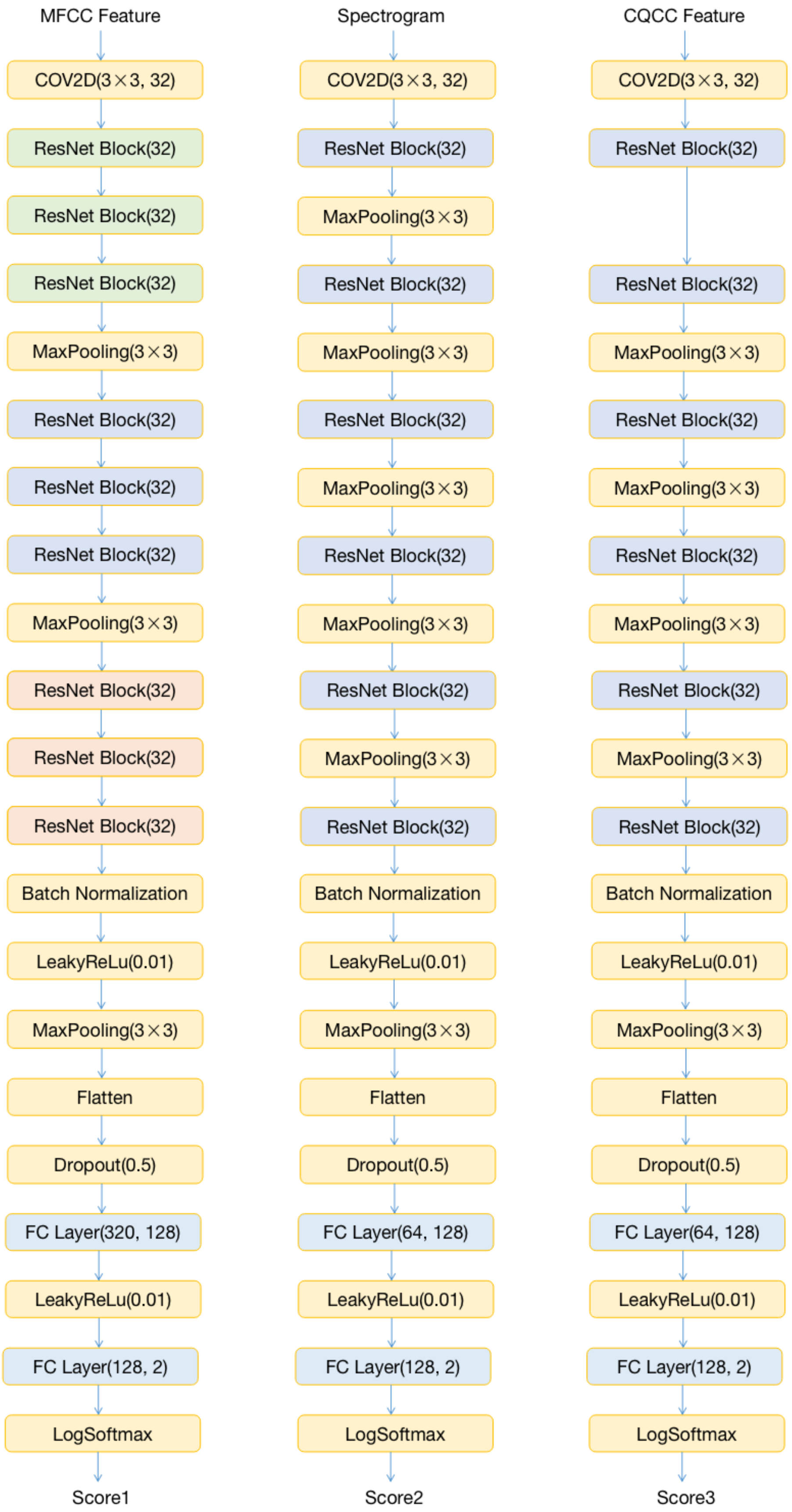


Figure A.2: *ResNet model variants specifications*

REFERENCES

[AAA16] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." In *International conference on machine learning*, pp. 173–182, 2016.

[AGM17] Nikolaj Andersen, Kristian Granhaug, Jørgen Andreas Michaelsen, Sumit Bagga, Håkon A Hjortland, Mats Risopatron Knutsen, Tor Sverre Lande, and Dag T Wisland. "A 118-mw pulse-based radar soc in 55-nm cmos for non-contact human vital signs detection." *IEEE Journal of Solid-State Circuits*, **52**(12):3421–3433, 2017.

[AMJ14] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. "Convolutional neural networks for speech recognition." *IEEE/ACM Transactions on audio, speech, and language processing*, **22**(10):1533–1545, 2014.

[APP] "XeThru X4 Radar User Guide." https://github.com/novelda/Legacy-Documentation/blob/master/Application-Notes/XTAN-13_XeThruX4RadarUserGuide_rev_a.pdf. Accessed: 2020-05-28.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." *IEEE transactions on pattern analysis and machine intelligence*, **35**(8):1798–1828, 2013.

[Bol79] S Boll. "A spectral subtraction algorithm for suppression of acoustic noise in speech." In *ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pp. 200–203. IEEE, 1979.

[BSM79] Michael Berouti, Richard Schwartz, and John Makhoul. "Enhancement of speech corrupted by acoustic noise." In *ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pp. 208–211. IEEE, 1979.

[CKH16] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. "Wide & deep learning for recommender systems." In *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10. ACM, 2016.

[con19] ASVspoof consortium. "ASVspoof 2019: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan." 2019.

[CQD15] Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, and Kai Yu. "Robust deep feature for spoofing detection—The SJTU system for ASVspoof 2015 challenge." In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[CXZ17] Zhuxin Chen, Zhifeng Xie, Weibin Zhang, and Xiangmin Xu. "ResNet and Model Fusion for Automatic Spoofing Detection." In *INTERSPEECH*, pp. 102–106, 2017.

[DGZ18] Ashutosh Dhekne, Mahanth Gowda, Yixuan Zhao, Haitham Hassanieh, and Romit Roy Choudhury. "Liquid: A wireless liquid identifier." In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 442–454. ACM, 2018.

[DRW14] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J Mysore, Frédo Durand, and William T Freeman. "The visual microphone: passive recovery of sound from video." 2014.

[EKN17] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. "Dermatologist-level classification of skin cancer with deep neural networks." *Nature*, **542**(7639):115, 2017.

[EKY13] Nicholas WD Evans, Tomi Kinnunen, and Junichi Yamagishi. "Spoofing and countermeasures for automatic speaker verification." In *Interspeech*, pp. 925–929, 2013.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[HLH18] Wen-Chin Huang, Chen-Chou Lo, Hsin-Te Hwang, Yu Tsao, and Hsin-Min Wang. "Wavenet vocoder and its applications in voice conversion." *ROCLING 2018*, p. 96, 2018.

[HZR15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[IS15] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167*, 2015.

[JBW18] Lauri Juvela, Bajibabu Bollepalli, Xin Wang, Hirokazu Kameoka, Manu Airaksinen, Junichi Yamagishi, and Paavo Alku. "Speech waveform synthesis from MFCC sequences with generative adversarial networks." In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5679–5683. IEEE, 2018.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014.

[KEY17] Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Kong Aik Lee, Md Sahidullah, Massimiliano Todisco, and Héctor Delgado. "ASVspoof 2017:

Automatic speaker verification spoofing and countermeasures challenge evaluation plan." *Training*, **10**(1508):1508, 2017.

[KL02] Sunil Kamath and Philipos Loizou. "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise." In *ICASSP*, volume 4, pp. 44164–44164. Citeseer, 2002.

[KLD18] Tomi Kinnunen, Kong Aik Lee, Héctor Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A Reynolds. "t-DCF: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification." *arXiv preprint arXiv:1804.09618*, 2018.

[KWE19] Ilya Kavalerov, Scott Wisdom, Hakan Erdogan, Brian Patton, Kevin Wilson, Jonathan Le Roux, and John R Hershey. "Universal sound separation." In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 175–179. IEEE, 2019.

[KXF] Andrew Kwong, Wenyuan Xu, and Kevin Fu. "Hard Drive of Hearing: Disks that Eavesdrop with a Synthesized Microphone." In *Hard Drive of Hearing: Disks that Eavesdrop with a Synthesized Microphone*, p. 0. IEEE.

[LNM17] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. "Audio Replay Attack Detection with Deep Learning Frameworks." In *Interspeech*, pp. 82–86, 2017.

[LYT18] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, and Zhenhua Kinnunen, Tomi a Ling. "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods." *arXiv preprint arXiv:1804.04262*, 2018.

[Mil16] Dan Miller. "Voice Biometrics Census: Steady Growth of Global Enrollments." 2016.

[MRL15] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and music signal analysis in python." In *Proceedings of the 14th python in science conference*, pp. 18–25, 2015.

[NGW15] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. "Contactless sleep apnea detection on smartphones." In *Proceedings of the 13th annual international conference on mobile systems, applications, and services*, pp. 45–57, 2015.

[nov] "Novelda presence sensor." https://novelda.com/novelda-presence-sensor.html.

[ODZ16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499*, 2016.

[PGC17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in pytorch." 2017.

[PNC] "XeThru X4 Phase Noise Correction." https://github.com/novelda/Legacy-Documentation/blob/master/Application-Notes/XTAN-14_XeThru_X4_Phase_Noise_Correction_rev_a.pdf. Accessed: 2020-05-28.

[RHH17] Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. "Cardiologist-level arrhythmia detection with convolutional neural networks." *arXiv preprint arXiv:1707.01836*, 2017.

[RQD00] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. "Speaker verification using adapted Gaussian mixture models." *Digital signal processing*, **10**(1-3):19–41, 2000.

[RR95] Douglas A Reynolds and Richard C Rose. "Robust text-independent speaker identification using Gaussian mixture speaker models." *IEEE transactions on speech and audio processing*, **3**(1):72–83, 1995.

[shaa] "Shazam." https://www.shazam.com. Accessed: 2019-12-08.

[Shab] Stephen Shankland. "Apple built UWB into the iPhone 11, so it's time to learn about this new wireless tech. Here's our FAQ.".

[SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research*, **15**(1):1929–1958, 2014.

[SLJ15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[TCS16] Tomoki Toda, Ling-Hui Chen, Daisuke Saito, Fernando Villavicencio, Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. "The Voice Conversion Challenge 2016." In *Interspeech*, pp. 1632–1636, 2016.

[TDE17] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification." *Computer Speech & Language*, **45**:516–535, 2017.

[TWS19] Massimiliano Todisco, Xin Wang, Md Sahidullah, H ector Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection." In *Proc. of Interspeech 2019*, 2019.

[VMO15] Jesus Villalba, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida. "Spoofing detection with DNN and one-class SVM for the ASVspoof 2015 challenge." In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[WHK18] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. "ESPnet: End-to-End Speech Processing Toolkit." In *Interspeech*, pp. 2207–2211, 2018.

[WKE15] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov. "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge." In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[WWK16] Zhizheng Wu, Oliver Watts, and Simon King. "Merlin: An Open Source Neural Network Speech Synthesis System." In *SSW*, pp. 202–207, 2016.

[WWZ15] Teng Wei, Shu Wang, Anfu Zhou, and Xinyu Zhang. "Acoustic eavesdropping through wireless vibrometry." In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 130–141. ACM, 2015.

[WYK15] Longbiao Wang, Yohei Yoshida, Yuta Kawakami, and Seiichi Nakagawa. "Relative phase information for detecting human speech and spoofed speech." In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[XED] "Xethru Raspberry Driver Example." https://github.com/novelda/Legacy-SW/tree/master/Examples/X4Driver_RaspberryPi. Accessed: 2020-05-28.

[XeT] XeThru. "Impulse radar chip XeThru X4, Available at https://novelda.com/x4-soc.html." Accessed: 2020-05-29.

[XLZ19] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. "WaveEar: Exploring a mmWave-based Noise-resistant Speech Sensing for Voice-User Interface." In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 14–26. ACM, 2019.

[YLL16] Lei Yang, Yao Li, Qiongzheng Lin, Xiang-Yang Li, and Yunhao Liu. "Making sense of mechanical vibration period with sub-millisecond accuracy using backscatter signals." In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pp. 16–28. ACM, 2016.

[Zava] Esfandiar Zavarehei. "Berouti Spectral Subtraction MATLAB implementation." https://www.mathworks.com/matlabcentral/fileexchange/7653-berouti-spectral-subtraction.

[Zavb] Esfandiar Zavarehei. "Boll Spectral Subtraction MATLAB implementation." https://jp.mathworks.com/matlabcentral/fileexchange/7675-boll-spectral-subtraction.

[Zavc] Esfandiar Zavarehei. "Multi-band Spectral Subtraction MATLAB implementation." https://www.mathworks.com/matlabcentral/fileexchange/7674-multi-band-spectral-subtraction.

[ZCC20] Tianyue Zheng, Zhe Chen, Chao Cai, Jun Luo, and Xu Zhang. "V2iFi: in-Vehicle Vital Sign Monitoring via Compact RF Sensing." *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol*, **4**(2), Jun 2020.

[ZHC19] Dongheng Zhang, Yang Hu, Yan Chen, and Bing Zeng. "BreathTrack: Tracking indoor human breath status via commodity WiFi." *IEEE Internet of Things Journal*, **6**(2):3899–3911, 2019.

[ZLH18] Yang Zhang, Gierad Laput, and Chris Harrison. "Vibrosight: Long-Range Vibrometry for Smart Environment Sensing." In *The 31st Annual ACM Symposium on User Interface Software and Technology*, pp. 225–236. ACM, 2018.

[ZZL15] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." In *Advances in neural information processing systems*, pp. 649–657, 2015.