Computational Wireless Sensing at Scale

Rajalakshmi Nandakumar

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Shyamnath Gollakota, Chair

Katharina Reinecke

Victor Bahl

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Computational Wireless Sensing at Scale

Rajalakshmi Nandakumar

Chair of the Supervisory Committee:
Shyamnath Gollakota
Computer Science and Engineering

Computational wireless sensing is an exciting field of research where wireless signals from everyday computing devices are used to enable sensing. The key challenge is to enable new sensing capabilities that can be deployed at scale and have an impact in the real world. In this dissertation, we show, for the first time, how to enable computational wireless sensing at scale by leveraging ubiquitous devices like smartphones. We present algorithms that can wirelessly sense motion and physiological signals such as breathing using just a smartphone, in a contactless manner. The key idea in this dissertation is to transform the smart devices into active sonar systems. We emit an inaudible sound signal from the device's speaker and these signals are reflected back by the different objects and people in the environment which we then capture using the device's microphones. By isolating the various reflections at the receiver, we can enable the sensing and tracking of different objects in the environment.

Building on this approach, we design, build and evaluate four key systems. We present the first contactless system that can diagnose sleep apnea using a smartphone by monitoring the minute millimeter level breathing motion. Next, we built a smartphone based system that can detect opioid overdoses by looking at respiratory depression and body movements. We then show that, beyond respiratory tracking, we can also track the 2D location of a finger around a smartphone or watch to enable new interaction modalities. Finally, we explore the privacy implications of enabling wireless sensing on commodity devices and propose defenses to prevent information leakage.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

This dissertation is the result of enormous support and encouragement from a large number of people who came into my life during my journey as a Ph.D. student. They were there as I grew as a person through the program; they answered my questions, supported me when I doubted, and made me smile. I am grateful for every one of these people and feel incredibly fortunate to receive so much from so many.

My advisor, Shyam Gollakota, has guided me through this program from Day 0. He is the most active person I have known, inducing the same hyper energy to the lab. From the first project where he had to stay with me during the experiments, to the last project I managed, he cultivated me to become an independent researcher. He was always there when I had questions and he never failed to push me further than what I thought I could achieve. In addition to teaching me how to do research, he was a huge support while I was adapting to the new country and also during my personal health struggles. I am honored to be one of his first students and to have been a part of the journey in forming his new lab at UW.

Special thanks to my committee members for their interest in my work and advice over the last few years. Thank you, Katharina, for teaching me to think through the HCI lens while designing systems. Thank you Victor, for always helping me to visualize the high level picture while I was lost in the minute details. I will always remember your advice on creating a successful research career. Thank you Jacob for working closely with me on the opioid overdose project. This project and the experience at the supervised injection facility taught me the challenges of conducting clinical studies. This experience has motivated me to further pursue a research career in the computational health domain. Thank you Anind for all your valuable feedback during the dissertation process.

Without the support of my labmates, this dissertation would not have been possible. Thank

you, Bryce and Vamsi for patiently answering all the hardware related questions that I had while working with software radios for the first time. A special thanks to Vikram Iyer, who has been a great collaborator for a number of my projects. He built the smart watch platform for the FingerIO project within a week during the crunch time of the submission! Starting with that project, he has been my go to person for any hardware related aspects of my research. All the outdoor experiments in the Seattle rain wouldn't have been as much fun without him. Thanks to Justin Chan for co-teaching my first course with me. It was a great experience designing the course with him. Thanks to my labmates Anran, Mehrdad and Ali for all the coffee break sessions and for being a great support during the course of my Ph.D. I would also like to thank my collaborators, Dr. Nate Watson, Alex Takakuwa and Tadayoshi Kohno for working with me and educating me in new domains. Thanks Hank, Ed and Tom for helping me immensely during my job search.

I wish to express my deepest gratitude to my parents and family members who have always supported and encouraged me to pursue my goal. They have been the role model teaching me to always aim high and succeed in life. A special thanks to my dad for introducing me to the field of health and medicine during my formative years in India. It not only prepared me to work in the hospitals with physicians but it also had a profound effect in me getting interested in the health domain with a goal to create an impact in society.

And finally, no amount of thanks can convey my regards to my partner, Raghav. His numerous phone calls during nights when nothing was working to his celebrations of all my minor successes has been the essential tonic that I required during the course of this Ph.D. By travelling every week from San Francisco to visit me in Seattle, he ensured that I managed a decent work life balance.

Thank you everyone, and this thesis is dedicated to all of you.

DEDICATION

to my parents, Jeyanthi and Nandakumar

Chapter 1

INTRODUCTION

In the superhero film Dark Knight, Lucius Fox used transmissions from cellphones to imitate active sonar and image the whole city. Over the past century, this concept of wireless sensing has appeared in various fantasy and gaming literature in different forms. Inspired by this fascination for wireless sensing, we explore the concept of computational wireless sensing in this thesis. Computational wireless sensing is the use of wireless signals from everyday computing devices like smart phones and watches to enable new sensing capabilities. At a high level, the wireless signals propagate through air and are reflected by the human body in the environment before they arrive at the receiver. By designing algorithms that can separate various reflections at the receiver, we enable sensing and tracking capabilities for objects and humans in the environment. More importantly, using these sensing capabilities, we also design and deploy various applications across domains ranging from mobile health to user interfaces.

The main challenge is to enable computational wireless sensing at the scale of millions of users while still achieving the precise millimeter resolution required by many applications like respiration monitoring or user interfaces. To do this, we take inspiration from the early days of the Internet, where existing telephone lines were repurposed to provide Internet connectivity to hundreds of millions of end users. Similarly, we would like to leverage the existing infrastructure of billions of smart phones to enable computational wireless sensing at large scale, without the need for additional custom hardware. The key algorithmic roadblock towards achieving this, however is the ability to support precise millimeter-level wireless sensing resolution using existing computing devices, which is essential to track minute changes in the human body due to breathing as well as other fine-grained movements in the environment.

1.1 Transforming devices into active sonar systems

In this dissertation, we take a novel approach that for the first time enables computational wireless sensing at scale. Specifically, we leveraged the speakers and microphones available on existing mobile devices to enable wireless sensing. At a high level, we transmitted inaudible high-frequency custom sound signals using the device's speaker. These signals were then reflected by the environment and recorded by the device's microphones. Any minute motion in the environment, such as finger motion or breathing' causes a change in the reflections as seen by the microphones. These changes were detected and localized using sophisticated signal-processing algorithms. Building on this idea, we design multiple sonar-based systems that enable wireless sensing and address long-standing problems in two key domains: mobile health and user interfaces.

There are two main advantages to our sonar-based approach. First, audio sensors, including microphones and speakers, are ubiquitous in today's devices. In fact, recent trends show an increase in the number of speakers and microphones in mobile devices: The iPhone 3 and X have two and four microphones respectively. Further, with the introduction of home assistants like Google Home and Amazon Echo dots, we now have 7-8 microphones in a single device that are widely deployed. The second advantage is that sound travels at much lower speed in air, so a basic audio sensor with a sampling rate of 48 kHz currently available in the marketplace can provide a resolution of approximately 7 mm. This is a very high resolution compared to even custom radio-based radar solutions. Radio signals travel at the speed of light which makes time of arrival computation harder to perform. Indeed to enable millimeter resolution using radio signals, one would need a high-rate sampling clock and Gigahertz of bandwidth [147, 212] or a large number of antenna [137, 138], none of them are feasible on existing devices. This makes active sonar a compelling design to achieve wireless sensing at the scale of billions of devices.

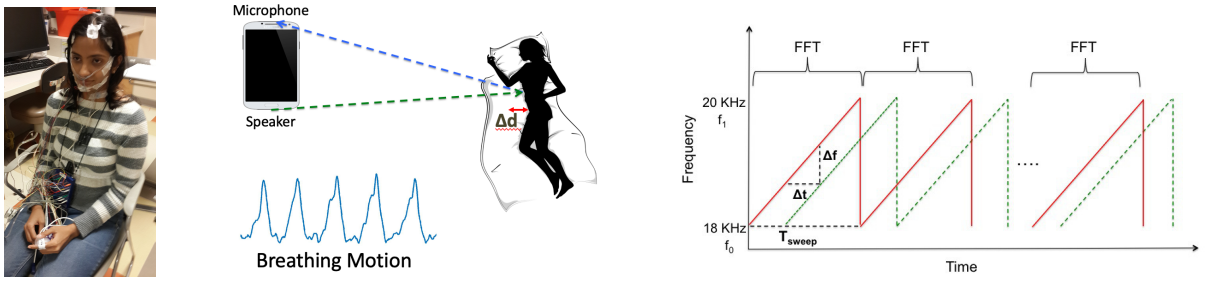
Here, we present algorithms that achieve different levels of sensing and tracking accuracies ranging from millimeter-level to the meter-level resolutions. We also explore the effect of the operational range of the sensing system and its effect on the accuracy of the system. By trading off accuracy to a meter level we show that we can enable sensing systems that can work across

rooms and through the walls. On the other hand, by limiting range to a few meters, we show that can we achieve millimeter-level accuracies for tracking minute human body motion such as breathing. Further, to demonstrate the utility of these sensing technologies, we build systems that enable applications across mobile health and user interfaces. By incorporating application-specific optimizations to these technologies, we build end-to-end systems that can be deployed and evaluated at scale in the real-world and have an impact.

Specifically, we introduce four active-sonar based systems in this dissertation.

- *Contactless diagnosis of sleep apnea [130].* We develop ApneaApp, which is the first contactless solution for detecting sleep apnea events by monitoring the minute chest and abdomen movements caused by breathing on smartphones. Our system works with the phone upto a meter away from the subject and can simultaneously identify and track the fine-grained breathing from multiple subjects. We do this by transforming the phone into an active sonar system that emits frequency-modulated inaudible sound signals and listens to their reflections. We have developed algorithms that identify various sleep apnea events including obstructive apnea, central apnea, and hypopnea from the sonar reflections. We deploy our system at the UW Medicine Sleep Center at Harborview and performa clinical study with 37 patients for a total of 296 hours demonstrating its ability to operate in the wild.

- *Opioid overdose detection using smartphones [129].* Early detection and rapid intervention can prevent death from opioid overdose. At high doses, opioids can cause rapid cessation of breathing, hypoxemic/hypercarbic respiratory failure, and death, the physiologic sequence by which people commonly succumb from unintentional opioid overdose. We present algorithms that run on smartphones and unobtrusively detect opioid overdoses and their precursors [49, 1]. Our short-range active sonar system uses acoustic frequency shifts to identify respiratory depression and gross motor movements associated with acute opioid toxicity. We develop algorithms and perform testing in an approved supervised injection facility (SIF), where people self-inject illicit opioids. Our testing 209 opioid users shows that our system can accurately identify opioid-related respiratory depression events.



(a) (b) (c)

Figure 1.1: **Contactless diagnoses of sleep apnea** (a) Polysomnography, today's clinical standard test for sleep apnea requires the patient to wear more than 20 sensors in their body. (b) Our system transforms the smartphone into an active sonar system to detect the minute breathing motion of a person. (c) The system transmits custom FMCW signals that translates minute breathing motion into frequency shifts which can be detected by taking an FFT across each chirp.

- *Fine-grained finger tracking [131].* Next, we explore the following question: Can we track a user's finger around a device, and can we do this even when the finger and device are occluded from each other? A positive answer would let users interact in more expressive ways, because they could fully utilize the screen without hand blockage, which would enable new interaction scenarios. We develop FingerIO, a finger-tracking system that doesn't require instrumenting the finger with sensors and that works even with occlusions between the finger and device. FingerIO tracks finger motion in the region around existing smartphones and achieves an average 2D tracking accuracy of 8 mm. It also tracks subtle finger motion around the device, even when the phone is in a pocket. Using FingerIO, we also prototype a smartwatch-form-factor device that can track the finger while extending the interaction space to a 0.5 0.25-square-meter region on either side of the device. Furthermore, the watch can continue tracking the finger even when the finger is fully occluded from the device.

- *Activity information leakage using music [133].* Finally, we explore the privacy implications of enabling active sonar on commodity devices. We develop Covertband, a low-cost, covert

physical sensing system that embeds inaudible signals into music and uses such signals to track multiple individuals' locations and activities both within a room and through barriers in 2D space. We achieve this by transforming a smartphone into an active sonar system that emits a combination of a sonar pulse and music and listens to the reflections off of humans in the environment. We evaluated CovertBand by running experiments in five homes in the Seattle area, showing that we can localize both single and multiple individuals through barriers like walls. These tests show CovertBand can track walking subjects with a mean tracking error of 18 cm and subjects moving at a fixed position with an accuracy of 8 cm at up to 6 m in line-of-sight and 3 m through barriers.

Before we dig into the details of these systems, we provide the high-level description of each of the systems and the impact they have had on the academic community as well as industry.

1.1.1 Contactless diagnosis of sleep apnea

ApneaApp, the subject of Chapter 2, presents a contactless solution for detecting sleep apnea events on smartphones. Sleep apnea is a common medical disorder that occurs when breathing is disrupted during sleep. It is estimated to affect more than 18 million American adults [21, 177] and is linked to attention deficit/hyperactivity disorder, high blood pressure, diabetes, heart attack, stroke, and increased motor vehicle accidents [18, 171]. Diagnosing sleep apnea in the clinic requires the polysomnography test which is an expensive, time-consuming and labor-intensive process. It requires a trained technician to attach and monitor various sensors on the patient for the sleep duration and is typically associated with long waiting lists [69]. While portable recording systems are being developed for use in home settings, they require instrumenting either the patient [101, 63, 63] or the bed [134] with various sensors and most still require a trained technician to setup the recording system [134]. To overcome this, we designed ApneaApp which is the first contactless system that detects sleep apnea using just an off-the-shelf smartphone.

To achieve this, we introduce a novel contactless system that monitors the minute chest and abdomen movements caused by breathing on smartphones. Our system works with the phone

away from the subject and can simultaneously identify and track the fine-grained breathing movements from multiple subjects. We do this by transforming the phone into an active sonar system that emits frequency-modulated continuous signals (FMCW) and listens to their reflections. An FMCW waveform is a chirp signal,where the transmitted frequency increases linearly with time between 18 kHz and 20 kHz, frequencies that are inaudible to adults. These signals reflect off the reflector (e.g., human body) and arrive at the microphone after a time delay. To determine this delay, at a high level, the receiver compares the frequencies of the transmitted and reflected signals. Since the transmitted frequency increases linearly in time, time delays in the reflected signals translate to frequency shifts in comparison to the transmitted signals. The changing frequency shifts corresponding to the breathing motion are then extracted by transforming the signal into frequency domain by taking a fast fourier transform (FFT) over the chirp duration. The challenge however is that breathing movements are minute and create a very small frequency shift. Specifically, a 2 cm breathing displacement creates a 11.7 Hz shift. This is however problematic because given our sampling rate and chirp duration, the width of each FFT bin is 93.75 Hz which is much greater than the frequency shifts created due to breathing. To address this constraint, we design an FMCW receiver algorithm that reduces the FFT bin width and allows us to track minute breathing motion. From the breathing motion signal, we then develop algorithms that allow us to identify various sleep apnea events including obstructive apnea, central apnea, and hypopnea, which are essential to diagnose sleep apnea. These algorithm is explained in detail in Chapter 2.

We deploy our system at the UW Medicine Sleep Center at Harborview and perform a clinical study with 37 patients for a total of 296 hours. Our study demonstrates that across patients, the number of central apnea, hypopnea and obstructive apnea events detected by our system is highly correlated with the ground truth. Specifically, the intra-class correlation coefficient between the ground truth and ApneaApp, is 0.9957, 0.9533 and 0.9860 for central apnea, hypopnea and obstructive apnea respectively. We also ran an audibility test with 87 sleep apnea patients (ages between 23 and 93 with a mean age of 50), none of whom reported any audible sounds.

Impact. ApneaApp was licensed by ResMed Inc., which is a leader in continuous positive airway pressure (CPAP) machines that are frequently used to manage sleep apnea. They released

(a)

(b)

Figure 1.2: **Opioid overdose detection using smartphones** (a) The figure shows the smartphone application that detects opioid overdose in a timely manner and connects to life saving interventions like 911. (b) Our system was evaluated at Insite, a supervised injection facility at Vancouver BC, where people can bring illicitly obtained drugs and can take them in the presence of medical supervisors. [Image copyright: NYTimes]

an app called SleepScore [24] that can now be downloaded in the app store across both the Android and iOS platforms. The app uses our technology to detect breathing motion and applies machine learning techniques to do sleep staging. Because our algorithms can run as a software app on a smartphone, this technology has been able to scale to around a million users in just 6-8 months.

1.1.2 Opioid overdose detection using smartphones

Fatal opioid overdose remains a public health epidemic in the United States [195, 102, 136, 67, 181, 74]. Each day, 115 Americans die from opioid overdose and data from the Centers for Disease Control and Prevention (CDC) indicate the epidemic is worsening [153, 30, 180]. Unlike many life-threatening medical emergencies, opioid toxicity is readily reversed with rapid identification and administration of the overdose antidote naloxone or supportive respiratory care [167, 156, 182, 78]. Thus, a fundamental challenge of fatal opioid overdose events is that victims die alone or among untrained or impaired bystanders, in each case with no or insufficiently timely diagnosis and treatment [59]. To help connect potential overdose victims with widely available life-saving

interventions, we developed algorithms for commodity smartphones that unobtrusively recognize opioid overdose by its physiologic precursors. Our system converts the phone into a short-range active sonar system, using frequency shifts to identify respiratory depression, apnea and gross motor movements associated with acute opioid toxicity. By creating overdose detection algorithms that can be deployed on devices most high risk individuals already own [125, 122], we hope to provide a harm reduction system that can automatically connect with naloxone-equipped friends and family or EMS to help prevent fatal overdose events [46, 75].

A mobile system that can detect opioid overdose precursors and events in real-time does not currently exist due to both design and validation challenges. Existing, human-based approaches to overdose diagnosis rely on medical grade equipment or trained recognition of diagnostic signs of opioid toxicity [49, 71, 77, 39, 81]. Achieving similar sensing capabilities on smartphones, without the need for medical grade equipment, is challenging since it requires tracking physiological parameters without being intrusive and violating privacy [60, 76]. In addition, validating the efficacy of any opioid toxicity system requires access to patients and data while high risk opioid use occurs, which is difficult because this can represent a medically life-threatening situation. We overcome these challenges with an active sonar-based monitoring solution, leveraging access to two unique environments where people routinely experience overdose respiratory physiology without harm: (1) a legally sanctioned supervised injection facility (SIF), where people self-inject previously obtained illicit opioids under medical supervision and (2) the operating room (OR), during routine induction of general anesthesia.

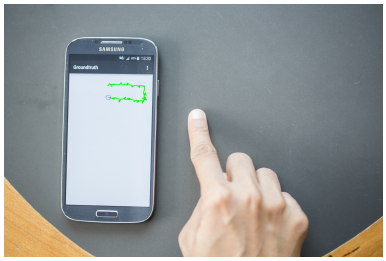
We designed a contactless smartphone-based system that matches the performance of an invasive respiratory impedance monitor in identifying 3 critical overdose precursors: opioid-induced respiratory depression, central apnea and simulated overdose events [87, 49, 1]. The system works by placing the phone within 1 meter of the subject as it monitors them during the post-injection period, the highest risk time for a fatal overdose event and the period when a victim would most benefit from rapid identification and resuscitation. Our system builds on our previous work using active sonar to detect sleep apnea, however the opioid use case differs from the sleep environment in several fundamental ways. First, breathing motion is diminished during opioid use, which can

complicate respiratory peak detection, and subjects may use opioids multiple times over the course of a day and thus may have a diminished breathing signal at the initiation of their use event. Additionally, the sleep laboratory in prior work is a controlled environment with a lone subject who is primarily stationary. In contrast, subjects using opioids have increased motion that can affect the time delay of the echo, may engage in high-risk opioid use behaviors in the presence of others, and are generally in a much less controlled environment, which introduces other sources of potential interference [61, 170]. Notably, the supervised injection facility is a highly dynamic and stimulating environment (recording devices are prohibited within the SIF, but the environment can be observed in this succinct public domain report by the New York Times [28]). For example, there are routinely several people around; there is talking amongst clients; staff and clients walk around; overhead music is playing and occasionally personal dogs are within the environment. In addition, there is climate control equipment, as well as a special indoor ventilation system (to remove the smoke from heroin preparation), all of which produce ambient noise. In short, there are several environmental elements in the high-risk opioid use domain that differ from the controlled setting of a sleep laboratory. Chapter 3 describes algorithms that address the above challenges in detail.

Impact: Given the reliable reversibility of acute opioid toxicity, smartphone-enabled overdose detection, capable of alerting naloxone-equipped friends and family or Emergency Medical Services (EMS), may hold potential as a low-barrier, harm reduction intervention. This work has seen significant interest from major public health organizations including the National Institute of Drug Abuse (NIDA) and has led to a startup Sound Life Sciences Inc that is commercializing this technology. Further, our *Science Translational Medicine* publication has led to public conversation about the use of technology to address the opioid crisis.

1.1.3 Fine-grained finger tracking

In the above chapters, we monitor the breathing motion of the entire chest of an human subject. Following this, we asked if it were possible to track and locate the exact 2D position of the user's finger around the device and can we do this even when they are occluded from each other? A positive answer would allow the user to interact in more expressive ways, utilize the screen fully



(a)

(b)

Figure 1.3: **Fine grained finger tracking** (a) The figure shows that the system can accurately track 2D location of the finger around a device thereby increasing the interaction space. (b) Our system benefits smaller devices such as smart watches by tracking the user's finger over the hand or the air enabling full screen utilization without hand blockage.

without hand blockage and also enable new interaction scenarios. For instance, the user can use her finger as a pen to provide input over a much larger surface area than that provided by the smartphone. Such a capability would also benefit smaller devices such as smart watches that could track the user's finger in air or over surfaces, even when fully occluded from it or when the watch is on a different plane from the interaction surface.

To enable this 2D tracking, we take advantage of the presence of the two microphones in a smartphone. The speaker periodically emits pulses of inaudible sound and as the finger moves, the amount of time it takes for sound waves to reflect off the finger and arrive back at the microphone will change. In this case, the finger is moving and produces changing echoes with respect to everything else in the environment and hence can be separated. The distance accuracy of the system depends on how accurately we can identify the arrival time of each echo. One approach is to use signals such as FMCW chirps that have high auto-correlation property and perform a cross-correlation operation to detect the echoes. However this operation is accurate only with 2-3 samples leading to an error of 3 cm which is twice the width of a finger. In order to achieve better tracking accuracies, we developed an innovative approach that uses a modulation technique

commonly used in wireless communication called Orthogonal Frequency Division Multiplexing (OFDM). OFDM transmissions are divided up into symbols, waveforms which are calculated by taking an inverse fourier transform of some frequency domain data to be sent over the air. In addition to the samples corresponding to data, OFDM also includes a cyclic suffix where the first few samples are simply repeated at the end of the signal. As we explain in chapter 4, this cyclic suffix is the key to using OFDM for time synchronization and it helps in correcting the 2-3 sample error caused in the correlation process. This way our design can achieve sub-centimeter level finger tracking. Finally to perform 2D tracking, we combine the distances of the finger with respect to two microphones as described in detail in chapter 4. Our design can track finger motion in the region around existing smartphones, and achieves an average 2-D tracking accuracy of 8 mm.

Impact: Since we published this paper there has been a lot of subsequent work from multiple groups across the world that span both HCI and mobile systems communities [121, 183, 188, 208, 172]. These systems have improved the resolution to around 2 mm using more sophisticated signal processing techniques that leverage the signal phase. Finally, since 2019, we are starting to see products from Google that incorporate active sonar in their smart speakers for activity recognition [12].

1.1.4 Activity information leakage using music

In the above chapters, we leveraged smart devices with speakers and microphones to enable millimeter level resolution sensing systems that work in a shorter range. We show that by trading off accuracy, we can simultaneously track multiple users through barriers like walls, doors and windows. In addition to tracking, we design algorithms to distinguish linear and rhythmic class of motions. This system explores the privacy leaks that are exposed when it is exploited by an attacker. To show the various security risks, our design expertly conceals the high frequency sonar pulses within the beats of popular songs making it indistinguishable. This means that the attacker can implement the attack even remotely by using music apps that play the modified versions of popular songs. These modified versions can then be used to spy on the subjects in a home.

The reason to hide the high frequency sonar pulses is because to achieve a longer range, we

need to increase the output power of the high frequency sonar pulses. However at such high power, the non-linearity in the hardware results in sub-harmonics that are audible to the human ear. To achieve tracking through walls and long range as well as concealing the attack, we conceal the sub-harmonics within the beats of popular songs. When the speaker plays our combined sonar signal, sound waves reflect off both static objects in the environment and moving persons before being reaching the microphone. We can analyze this signal to extract information about the activities performed by the subject remotely. We evaluated CovertBand by running experiments in five homes in the Seattle area and show that we can track subjects walking with a mean tracking error of 18 cm and subjects moving at a fixed position with an accuracy of 8 cm at up to 6 m in line-of-sight and 3 m through barriers. We test a variety of rhythmic motions such as pumping arms, jumping, and supine pelvic tilts in through-wall scenarios and show that they produce discernibly different spectrograms from walking in the acoustic reflections. In tests with 33 subjects, we also show that even in ideal scenarios, listeners were unlikely to distinguish a original song from the modified one and hence could not detect a Covertband attack.

Impact: With the proliferation of devices like smart phones and speakers, this work focuses on exploring the privacy leaks possible such these devices that go beyond the ability to simply record conversations in the home. In addition to creating an awareness, the work also puts forth some defenses to prevent such covert tracking attacks.

1.2 Organization

The rest of this dissertation is organized as follows. In Chapter 2, we describe ApneaApp in more detail and mechanisms to make it work across different smartphone platforms as well as enable concurrent tracking of breathing from multiple human subjects. Chapter 3 describes how the above algorithms are adapted to work in the opioid use case to deal with ambient noise as well as environmental motion. Chapter 4 presents FingerIO and describes in detail our algorithms to achieve sub-centimeter resolution for around-the-device interaction on smartphones and smart-watches. Chapter 5 describe how high frequency acoustic signals can be embedded in popular songs to extract activity information through walls and barriers as well as at a long range. Finally,

Chapter 6 concludes the dissertation with comments on the future direction for this line of work.

Chapter 2

CONTACTLESS SLEEP APNEA DETECTION ON SMARTPHONES

Sleep apnea is a common medical disorder that occurs when breathing is disrupted during sleep. It is estimated to affect more than 18 million American adults [21, 177] and is linked to attention deficit/hyperactivity disorder, high blood pressure, diabetes, heart attack, stroke, and increased motor vehicle accidents [18, 171]. Diagnosing sleep apnea in the clinic requires the polysomnography test which is an expensive, time-consuming and labor-intensive process. It requires a trained technician to attach and monitor various sensors on the patient for the sleep duration and is typically associated with long waiting lists [69]. While portable recording systems are being developed for use in home settings, they require instrumenting either the patient [101, 63, 63] or the bed [134] with various sensors and most still require a trained technician to setup the recording system [134].

In this chapter we ask the following question: Can we leverage smartphones to detect sleep apnea events without the need for sensor instrumentation? The key challenge is that detecting sleep apnea events requires tracking the fine-grained abdomen and chest movements due to breathing [52]. While the iPhone Respiratory app [15] can track the breathing movements, it requires placing the phone on the body between the ribcage and the stomach and hence is intrusive. Vision-based solutions [197] can track these movements without instrumenting users, but are limited to line-of-sight and good lighting conditions and hence are not applicable to the sleep environment, i.e., in the dark or under a blanket.

We introduce a novel contactless system that tracks the chest and abdomen movements on smartphones and works in the sleep environment. It operates with the phone away from the user and can concurrently track the breathing movements from multiple users. Using this design, we build *ApneaApp*, a smartphone-based solution for detecting sleep-related respiratory events reported in a clinical polysomnography test including hypopnea (when the subject's breathing becomes shal-

low), obstructive apnea (a complete or partial blockage of the subject's airway) and central apnea (when the subject holds his or her breath).

Our key insight is to transform the phone into an active sonar system. At a high level, we transmit 18-20 kHz sound waves from the phone speaker and listen to their reflections at the microphone. The chest and abdomen motion due to breathing creates changes to the reflected sound waves. These changes, however, are minute and extracting them reliably from other environmental reflections is challenging. To overcome this, we employ FMCW (frequency modulated continuous wave) transmissions that allow us to separate reflections arriving at different times by mapping time differences to shifts in the carrier frequency. Specifically, the reflections from the human body arrive at a specific time depending on the distance from the phone speaker. Thus, focusing on the corresponding frequency allows us to reliably extract the amplitude changes due to breathing, in the presence of all other environmental reflections. Further, since reflections from multiple subjects would arrive at different times, the corresponding frequencies provide us with the ability to simultaneously track multiple breathing signals. Finally, non-breathing body motion creates reflection patterns distinct from breathing, enabling us to distinguish between them.

We implement our design on off-the-shelf smartphones and run benchmark experiments with five healthy participants in a bedroom environment using the Vernier respiratory belt as a baseline. Our results show the following:

- Our system estimates the coarse-grained breathing frequency[1] to within 99.2% of the baseline at distances of up to a meter from the subject. This translated to an error of less than 0.11 breaths/min. These accuracies remain this high even when the subjects use blankets.

- The above accuracies remain unaffected by audible noise in the environment from vehicles on a nearby street as well as human conversations. This is because, we use a high-pass filter to filter out audible signals below 18 kHz.

[1]Detecting sleep apnea events requires tracking the fine-grained abdomen and chest motion variations in addition to the coarse-grained breathing frequency. We evaluate ApneaApp's ability to track these variations in our clinical study.

- It can separate and concurrently track the breathing movements of two subjects on the bed separated by 20 cm.

Building on the above system, we design algorithms to compute the number of central, obstructive and hypopnea events as well as the apnea-hypopnea index which is the average rate of apnea and hypopnea events during the sleep duration. We achieve this by processing both the fine- and coarse- grained changes due to the chest and abdomen movements as well as non-breathing body motion. We deploy ApneaApp at the UW Medicine Sleep Center at Harborview and perform a clinical study with 37 patients for a total of 296 hours. The patients in our study were ordered by their physicians to undergo the polysomnography (PSG) test. Our study was done concurrently with the PSG test and we consider the sensor data and diagnosis from the latter as the ground truth for evaluating our system. Our study shows the following:

- Across patients, the number of central apnea, hypopnea and obstructive apnea events detected by our system is highly correlated with the ground truth. Specifically, the intra-class correlation coefficient between PSG and ApneaApp, is 0.9957, 0.9533 and 0.9860 for central apnea, hypopnea and obstructive apnea respectively.

- The average error in computing the rate of apnea and hypopnea events is 1.9 events/hr; this is a clinically acceptable value [134].

- Our system accurately classifies 32 out of 37 patients between four sleep apnea levels (no apnea, mild, moderate, and severe apnea). The five misclassifications occur between no-apnea and mild-apnea; four of them happen right at the boundary between the two levels with an error less than 1 event/hr. These boundary cases are handled separately by physicians depending on the patient preferences, symptoms, and insurance; thus, effectively reducing the number of misclassifications to one.

- We ran an audibility test with 87 sleep apnea patients (ages between 23 and 93 with a mean age of 50) and 57 healthy undergraduate students at UW CSE. None of the 87 sleep apnea

patients reported any audible sounds from ApneaApp. Only two of the 57 undergraduates reported hearing audible sounds. This demonstrates that ApneaApp is inaudible for most of the adult population.

Contributions: We make four key contributions: (1) We introduce a novel contactless technique for tracking chest and abdomen movements due to breathing on smartphones. We achieve this by analyzing the reflections from FMCW sonar transmissions. (2) We design algorithms to detect central apnea, obstructive apnea, and hypopnea as well as estimate the apnea-hypopnea index from the sonar reflections. (3) We implement our design on off-the-shelf smartphones and demonstrate the ability to concurrently track breathing movements from multiple subjects. (4) We perform a clinical study with 37 patients demonstrating the feasibility of using our system to accurately compute the number of central, obstructive, and hypopnea events as well as the apnea-hypopnea index.

2.1 Related Work

Our work is related to prior art in three domains.

(a) Mobile Health and Wellness Systems: There has been recent interest in the mobile health and wellness community to better understand sleep quality. Wearable sleep sensors such as FitBit [11], WakeMate [27], and Jawbone [16] capture the movements from an accelerometer to sense basic sleep patterns such as hours slept and the number of wakings. Lullaby [104] leverages sensors deployed in the bedroom to understand the effect of environmental factors (e.g., ambient noise and lighting conditions) on sleep quality. More recent work including Toss-N-Turn and iSleep leverage smartphone sensors as sleep quality detectors [126, 57, 85]. Specifically, they use accelerometers to detect coarse non-breathing body movements [126, 85] and microphones for cough [85] and snoring [85, 23]. ApneaApp builds on this foundational work but is complementary to it. We introduce a novel contactless system for smartphones that can track fine-grained chest and abdomen movements due to breathing. We then demonstrate the ability to detect sleep apnea events on smartphones.

Our work is also related to prior work on monitoring physiological signals using wearable sensors [25] including mouthpieces and chest bands [26, 19]. Designs such as BodyBeat [146] use custom-built piezoelectric microphones on the body surface to monitor body sounds such as respiration, food intake and laughter. More recent work [151] leverages audible breathing sounds from an ear bud to infer the breathing frequency. The Android app, iBreath [84], combines the audible breathing sounds from a Bluetooth headset with the accelerometer on the phone to infer the breathing frequency while running. These systems are very promising for their application domains. For diagnosing sleep apnea, however, we are unaware of clinical-validated research that demonstrates the ability to compute AHI values and detect sleep apnea events using only audible breathing sounds/frequency. In contrast, we propose a novel contactless solution that can monitor the chest movements by actively transmitting FMCW audio signals. We then demonstrate the ability to detect various sleep apnea conditions including obstructive, central, and hypopnea as well as compute the AHI values.

(b) RF-based Breathing Rate Monitors: Prior RF-based approaches [207, 161, 73] leverage changes in radio signals to estimate the breathing frequency. This includes using a network of 2.4 GHz wireless sensors in the bedroom [137, 138] as well as expensive ultra-wideband RF radar systems with 500 MHz to 2 GHz of bandwidth [147, 212]. While promising, none of these approaches have been demonstrated to work on off-the-shelf smartphones. We also believe that the sonar approach introduced in this chapter is more attractive for two main reasons. First, a microphone that samples at 48 kHz gives a range resolution of 0.7 cm; RF based approaches need Gigahertz of bandwidth to get a similar resolution. Second, the one-meter operational range of our sonar system reduces the effect of environmental movements making it more robust in real world sleep scenarios.

(c) Sleep Apnea Research and Systems: Polysomnography [52] is the standard clinical test for diagnosing sleep apnea in the medical setting. This however is an expensive test that requires attaching various sensors to the patient for the sleep duration. While portable recording systems [101, 63] are being developed for use in home settings, they require the attachment of various sensors and

most still require a trained technician to setup the system [134].

Mobile phone app developers have also designed questionnaires (e.g., Home Sleep Apnea A-Z [14], Sleep Access [20]) to screen patients for the PSG test. Sleep Apnea Monitor app [22] help users diagnosed with sleep apnea to sleep on their side instead of their back by buzzing when the user moves to their back. It does so by attaching the phone to the person's leg. Prior research has also explored the feasibility of leveraging snoring and movement information [31, 40] to perform screening for PSG. While correlation existing between sleep apnea and snoring, such an approach does not monitor breathing movements and hence cannot identify sleep apnea events [179, 134]. To address this issue, commercial systems such as Sonomat [134] use embedded sensors in a mattress pad to track the breathing movements and sounds [9, 134]. Our work instead focuses on detecting sleep apnea events by tracking breathing movements on smartphones, without the need for any other sensors.

Finally, medical research has proposed the use of ultrasound imaging, instead of MRI, to measure the tongue volume and width of upper airway [38, 162]. This information has been demonstrated to be helpful in diagnosing obstructive apnea [116]. In contrast to this work, we transform a smartphone into an active sonar system and develop algorithms to extract hypopnea, obstructive apnea, and central apnea information from the sonar signal changes.

2.2 Polysomnography Overview

The clinical polysomnography test (PSG) is traditionally used to diagnose sleep apnea and other sleep disorders. It is conducted overnight in a sleep laboratory where a trained technician monitors the patient's sleeping patterns. To do this, the technician attaches the patient with a number of sensors including a chest and abdomen belt to measure breathing movements, a nasal pressure transducer, a snore microphone, a pulse oximeter to measure oxygen saturation, a movement sensor on each leg to detect movements and five EEG sensors to measure brain activity. The sensors are all connected using wires and the technician monitors the live data stream from the sensors, throughout the sleep duration.

Fig. 2.1 shows a snapshot of a PSG report. The key metric used for sleep apnea diagnosis is

Respiratory Summary:

Types of Respiratory Events / Respiratory Effort Related Arousal (RERA) Events

Respiratory Events	Number	Index	RERA	Parameter	Total	Index
Obstructive Apneas	123	15.7 /hr		Total:	129	16.4
Mixed Apneas	0	0.0 /hr		Non-REM:	105	16.5
Central Apneas	1	0.1 /hr		REM:	24	16.1
Total Apneas	**124**	**15.8 /hr**		Supine:	129	16.4
Total Hypopneas*	**371**	**47.2 /hr**		Lateral:	N/A	N/A
Apneas + Hypopneas*	**495**	**63.0 /hr**		Prone:	N/A	N/A

Figure 2.1: **Snapshot of a Clinical PSG Report.** It summarizes the number of obstructive, central and hypopnea events along with the apneas-hypopneas index (AHI). An AHI value between 0–5 is classified as no-apnea, values between 5–15 are classified as mild-apnea, AHI values between 15–30 are classified as moderate-apnea, and higher AHIs are severe apnea conditions.

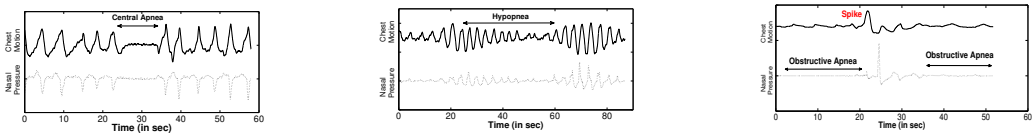
the AHI — the Apnea-Hypopnea Index — that represents the rate at which apnea and hypopnea events occur during the sleep period. Physicians classify the sleep apnea level using these AHI values. Specifically, AHI values between 0–5 are classified as no-apnea, those between 5–15 are classified as mild-apnea, AHI values between 15–30 are classified as moderate-apnea, and higher AHIs are severe apnea.

The apnea-hypopnea index is computed as follows:

$$AHI = \frac{\#central\ apnea + \#hypopnea + \#obstructive\ apnea}{total\ sleep\ time}$$

In the above equation, central apnea, hypopnea, and obstructive apnea denote the various apnea conditions that are tracked during the study. Mixed apneas are another class of apneas that are sometimes included in the above equation. However, none of our PSG reports showed non-zero mixed apneas and so we ignore them in our computation.

To compute the above parameters, the eight-hour sensor data is split into 30-second intervals called epochs. The scoring process of analyzing these epochs involves two main steps. The first step is staging, which identifies whether the patient is awake or asleep in each epoch. This is achieved by examining the brain activity obtained from the five EEG sensors. At the end of this step, each epoch is marked as being in either a wake or sleep stage. The second step involves



(a) Central Apnea Event (b) Hypopnea Event (c) Obstructive Apnea Event

Figure 2.2: **American Academy of Sleep Medicine (AASM) Signal Characterization of the Apnea Events.** The figures show the chest motion and nasal pressure signals for the three apneas. A central apnea event occurs when the subject holds her breath for a non-negligible duration. A hypopnea event occurs when the subject's chest motion drops by more than 30% with an accompanying 3–4% oxygen desaturation. Finally, an obstructive apnea event occurs when the subject makes an increased effort to pull air into the lungs but air does not reach the lungs due to blockage.

identifying the number of central apnea, hypopnea, and obstructive apnea events, using the AASM guidelines [52] outlined below.

Identifying central apnea events. A central apnea event occurs when the subject holds her breath for a non-negligible duration. Fig. 2.2(a) shows the nasal pressure and chest motion signals during a central apnea event. The figure shows that the chest movements are flat indicating the absence of breathing effort; consequentially the nasal pressure is also flat. If this persists for more than ten seconds, it is marked as a central apnea event.

Identifying hypopnea events. A hypopnea event occurs when the subject's breathing becomes shallow. Fig. 2.2(b) plots the nasal pressure and chest motion signals during a hypopnea event. The figure shows that during a hypopnea event, the chest movements reduce in amplitude. In particular, if this amplitude drops by more than 30% and has an accompanying 3–4% oxygen desaturation, then the corresponding epoch is marked as hypopnea. We note that recent clinical research [134] has shown that the 30% reduction alone can be used for detecting hypopneas without a significant loss of accuracies.

Identifying obstructive apnea events. Obstructive apnea occurs when there is a complete or

partial blockage of the upper airway during sleep. During an obstructive apnea event, the subject makes an effort to pull air into the lungs, however air does not reach the lungs because of blockage. Fig. 2.2(c) shows the signals where the breathing effort can be seen in the chest band signals and the air flow is flat in the nasal pressure sensor.

We note the following three points about PSG:

- The current procedure for sensor data collection and processing is both labor and time intensive. Specifically, it takes about an hour for the technician to fit each patient with these sensors. Throughout a sleep duration of eight hours, the technician monitors the sensors and ensures that they remain properly attached to the patient's body. The sensor data is then processed manually to tag every epoch with the sleep apnea events.

- While portable sleep apnea testing is performed in the home, it still require setting up the patient with chest and abdomen belts, nasal pressure sensors, transducer and thermistors, EKG and pulse oximetry. Home testing has a high failure rate of up to 33% due to the loss of signal resulting from detachment of wires and cables [145].

- A PSG test is also used to diagnosis other sleep-related conditions including upper airway resistance syndrome which involve respiratory effort related arousals (RERA) that are shown in Fig. 2.1. RERAs are sleep arousals that do not meet the above definitions of apneas and hypopneas. While these are respiratory-related and could be detected using our sonar-based system, exploring them in detail is not in the scope of this chapter.

2.3 ApneaApp

ApneaApp is a contactless system that enables detection of sleep apnea events using smartphones. To understand how ApneaApp operates, we first describe how we transform the phone into an active sonar system that tracks the chest and abdomen movements due to breathing. We then describe our algorithms to detect sleep apnea events from these movements.

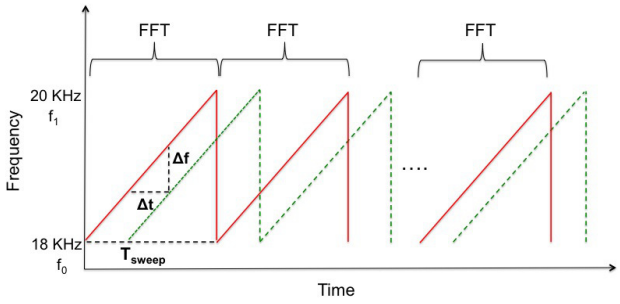


Figure 2.3: **Traditional FMCW Processing.** The transmitter continuously transmits signals where the frequency increases linearly with time between f_0 and f_1. The reflections that arrive with a time delay Δt create a frequency shift Δf. The receiver extracts this frequency shift by performing an FFT over the chirp duration.

2.3.1 Transforming the Phone into an Active Sonar

An FMCW waveform is a chirp signal, as shown in Fig. 2.3, where the transmitted frequency increases linearly with time between 18 kHz and 20 kHz. These signals reflect off the reflector (e.g., human body) and arrive at the microphone after a time delay. To determine this delay, at a high level, an FMCW receiver compares the frequencies of the transmitted and reflected signals. Since the transmitted frequency increases linearly in time, time delays in the reflected signals translate to frequency shifts in comparison to the transmitted signals.

For instance, the red line is the transmitted signal from the phone speaker and the green line is the reflected signal from a human body that arrives with a time delay Δt. This delay is given by $\frac{2d}{v_{sound}}$, where d is the distance from the human body and v_{sound} is the speed of sound. Now, the frequency shift Δf between the transmitted and reflected signal is:

$$\Delta f = \frac{f_1 - f_0}{T_{sweep}} \Delta t$$

When we have multiple reflectors that are at different distances from the receiver, their reflections translate to different frequency shifts in the signal. An FMCW receiver can extract all these frequency shifts by performing an Fourier transform over a chirp duration as shown in Fig. 2.3. The chirp duration, T_{sweep}, in practice is picked so that the reflections from all points within the desired

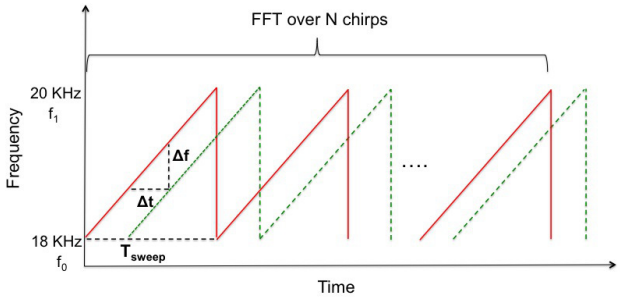


Figure 2.4: **FMCW Processing in ApneaApp.** To extract the minute frequency shifts created by breathing motion, ApneaApp performs an FFT over an integer number of chirp durations.

operational distance would start arriving before the chirp ends. Since our operational distance is a meter we pick a chirp duration of 10.75 ms in our implementation.

Challenge: The act of breathing creates minute chest and abdomen motion that can be captured by monitoring the corresponding bin in the Fourier transform as a function of time. The challenge however is that breathing movements are minute and create a very small frequency shift. Specifically, a 2 cm breathing displacement creates a 11.7 Hz shift.[2] This is problematic because given our sampling rate and chirp duration, the width of each FFT bin is 93.75 Hz which is much greater than the frequency shifts created due to breathing.

To address this problem, as shown in Fig. 2.4, the ApneaApp receiver performs a Fourier transform over N FMCW chirps. This is in contrast to a traditional FMCW receiver that computes a Fourier transform over the duration of a single FMCW chirp. Such an operation, decreases the width of each FFT bin by a factor of N. In our implementation we set N to ten which results in an FFT bin width of 9.37 Hz. This allows us to capture the minute frequency shifts resulting from the breathing movements. We note that performing an FFT over multiple chirp durations reduces our ability to track high-frequency movements that occur during these chirps. However, in our implementation, ten chirps correspond to a very short 107 ms, a duration within which significant

[2]Given the speed of sound, a 48 kHz sampling rate translates to a resolution of 0.71 cm per sample. Further, a 10.7 ms chirp duration corresponds to 512 samples. With 18–20 kHz FMCW chirps, each sample corresponds to a 3.9 Hz frequency shift. Thus, a displacement of 0.71 cm translates to a 3.9 Hz change in the frequency domain. Consequentially, a 2 cm breathing movement creates a 11.7 Hz frequency shift.

breathing movements are unlikely to occur.

The final question is: *how do we compute the distance of the subject from the phone?* At a high level, we start at a distance of zero and search for breathing movements at increasing distance values up to the maximum distance of one meter. Specifically, we search for breathing movements in the 58 Fourier bin corresponding to 18 kHz to 18.546 kHz.[3] In our implementation we reduce the computation by searching in every alternate FFT bin. To search for these breathing movements in each FFT bin, we perform another FFT over a 30s duration and search for peaks in the typical breathing frequencies of 0.2-0.3 Hz.

To summarize, the phone transmits FMCW signals in the 18-20 kHz range with a chirp duration of 10.7 ms from its speaker. The microphones receive the reflected signals and process them to track the breathing movements. Specifically, we first find the distance to the human by searching for a periodic breathing signal starting from the closed distance value to the maximum range of one meter. Once we find this distance, we track the breathing movements by performing in a shorter FFT over ten chirp durations and monitor the reflected signals corresponding to the estimated distance value. Note that the above procedure is repeated every time the user moves their position, which we identify using the algorithm in sec:sleepestimate. This prevents the need for manually calibrating the distance between the user and the phone.

We note the following points about our algorithm.

Computational Complexity. Our algorithm requires one 5120-point FFT and between one to twenty nine 24000-point FFTs to successfully extract the breathing motion. We stop our search at the first FFT bin that has the breathing movements.

Tracking breathing from multiple subjects. Reflections corresponding to subjects at different distances arrive with different time delays. Thus they create different frequency shifts with the FMCW signal. Therefore, to track breathing from two subjects, we modify the above algorithm to continue its search until it finds two FFT bins with the breathing motion.

---

[3]For 18-20 kHz FMCW transmissions, a distance of zero corresponds to the FFT bin for 18-kHz. The operational distance of 1 m corresponds to a frequency shift of 18.546 Hz.

Leveraging angle-of-arrival algorithms. One could track breathing movements from equidistant subjects who are at different angles, by using multiple microphones and implementing angle-of-arrival algorithms. Evaluating this, however, is not in the scope of this chapter.

FMCW versus pulse-modulated transmissions. Pulse-modulated transmissions use high-amplitude short pulses and are an alternative to continue-wave FMCW signals. In our experiment, however, they created low frequency components in the 0-18 kHz range that made them noticeably audible. FMCW transmissions, on the other hand, have lower-amplitudes and are limited to 18-20 kHz, making them inaudible for most of the adult population.

2.3.2 Sleep Apnea Detection Algorithm

As described in sec:psg, diagnosing sleep apnea requires estimating the Apnea-Hypopnea Index (AHI) which is the average rate of apnea events during the sleep duration. This requires computing the number of central, obstructive, and hypopneas as well the total sleep time. In this section, we first describe our algorithms to compute the number of apneas and then the total sleep time.

Estimating the Number of Apneas

ApneaApp detects a hypopnea event when the chest motion reduces below a threshold (30%).[4] A central apnea event is detected when the subject holds her breath and as a result the amplitude of the chest motion signal reduces to zero. Thus, ApneaApp identifies hypopnea when the amplitude of the breathing motion decreases below the threshold but the periodicity that is expected in a breathing signal still exists. But to identify central apnea, we couple an amplitude reduction in the chest motion signal with an absence of the breathing periodicity.

An obstructive apnea event, on the other hand, occurs when there is an obstruction in the airflow, i.e., the subject makes a breathing effort but the airflow is obstructed by a tissue overgrowth in the neck. In a clinical PSG study, obstructive apnea is detected using the nasal pressure sensor

[4] AASM and Medicare guidelines require the 30% reduction to be accompanied with a 3% and 4% oxygen saturation respectively. However, recent clinical research [134] has shown that the 30% reduction alone can be used for detecting hypopneas.

that directly measures the airflow; this, however, is not available in our system. We instead perform an analysis of the chest motion data that reveals that the subject usually tends to increase her breathing effort in these scenarios. This results in a clear spike in the amplitude of the chest motion, as shown in Fig. 2.2(c).

Thus, measuring the amplitude and periodicity in the chest motion signal is critical to detecting obstructive, central and hypopneas. Note that as shown in Fig. 2.2, the chest motion signal can be approximated as a periodic sinusoidal wave. Hence the magnitude of the peaks of these sinusoidal waves represents the amplitude and their peak locations determine their periodicity. Thus, we design a peak detection algorithm to compute the amplitude and periodicity of the chest motion signal.

ApneaApp's peak detection algorithm. Standard peak detection algorithms identify the transition point at which the signal changes from an increasing to a decreasing trend. In other words, for every set of three points, if the middle point is the maximum, then it is labeled as a peak. Such an algorithm, however, would result in a number of erroneous peaks with our chest motion signal. As an example, Fig. 2.5 plots a typical chest motion signal. Running the standard peak detection algorithm on this signal results in a number of unintended peaks as shown in the figure. To reduce the number of such peaks, we introduce two key heuristics.

The first heuristic is to set a threshold on the minimum distance between two consecutive peaks. In particular, the breathing frequency in an adult human typically varies between 12-18 breaths/min. Thus each breath takes 3.3 s at the maximum frequency of 18 breaths/min. We set a conservative threshold of three seconds in our implementation.

The second heuristic is to set a threshold on the minimum amplitude at which a peak is detected. To do this, we first run the above peak detection algorithm with the minimum distance heuristic on the chest motion data for the first hour to obtain an initial set of peaks. We then compute the minimum amplitude threshold as $\mu_{peaks} - 2\sigma_{peaks}$, where μ_{peaks} and σ_{peaks} are the mean and standard deviation of the peak amplitudes. Finally, we go back and apply both the amplitude as well as the minimum distance thresholds on the entire chest motion data for the eight-hour sleep duration to obtain the actual set of peaks. Fig. 2.5 shows that our peak detection algorithm identifies

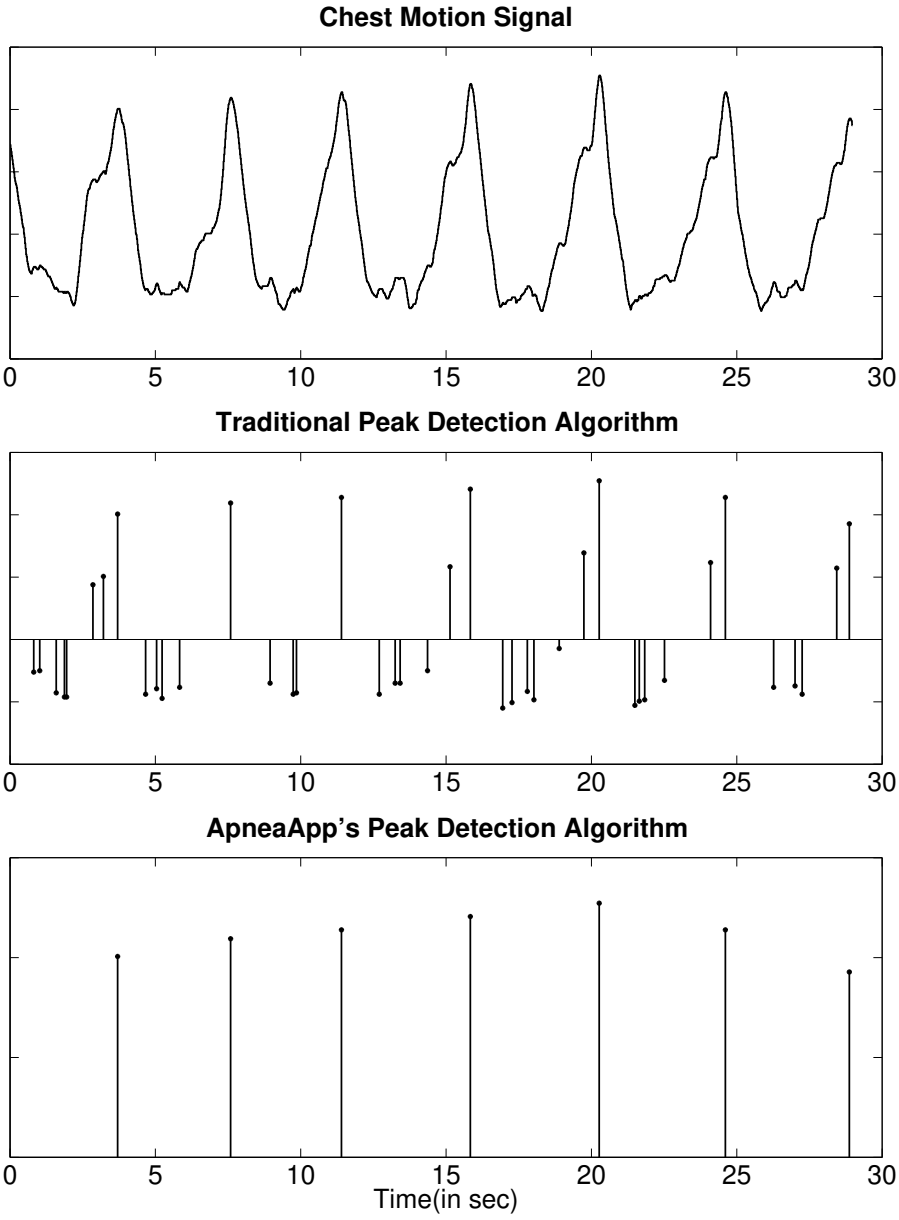


Figure 2.5: **Understanding ApneaApp's peak detection algorithm.** Standard peak detection algorithms identify the transition point at which the signal changes from an increasing trend to a decreasing trend. Thus, they identify a number of unintended peaks in our chest motion signal. ApneaApp's peak detection algorithm, in contrast, uses minimum peak distance and amplitude heuristics to reduce these unintended peaks.

the correct peaks.

Central apnea estimation algorithm. We run the peak detection algorithm to identify the locations of the peaks in the chest motion signal. We then compute the distance between these consecutive peaks. If this distance is greater than ten seconds, it means that the subject holds her breath for a non-negligible period of time and hence we declare it as a central apnea event.

Hypopnea estimation algorithm. We again use the peak detection algorithm to detect the peaks. When the peak values reduce beyond a threshold and still maintain their periodicity, we declare it as a hypopnea event. To compute this threshold for our sonar data, we perform a linear regression on the data from a single patient to maximize the hypopnea detection accuracy and identify a threshold of 38%. We use this threshold for all the patients in our clinical study.

Obstructive apnea estimation algorithm. As described earlier we detect obstructive apnea using sudden spikes in the chest motion signal. We identify the spikes from the peaks in the signal when the amplitude in these peaks increases by 50%. Further, if multiple spikes occur within a second we consider them to be part of a single obstructive apnea event. We note that some of the patients with periodic limb movement disorder periodically move their limbs. To avoid confusing them with obstructive apnea, we ignore periodic spikes in the chest motion signal.

Estimating the total sleep time

In a polysomnography test, the EEG sensors are used to measure the brain activity to determine whether the subject is asleep or awake; this is then used to measure the total sleep time of a subject. Recent work [134] has shown that one can use body movements to compute the total sleep time and the resulting accuracies are acceptable for the purposes of estimating the apnea-hypopnea index. Thus, ApneaApp measures the total sleep time by identifying the non-breathing body movements and subtracting their duration from the total experiment time.

To identify these movements, we leverage that when the subject's body moves, the sonar reflections experience large variations that do not have the periodicity of the breathing motion. By identifying these aperiodic signals that are shown in Fig. 2.6, ApneaApp detects body movements.

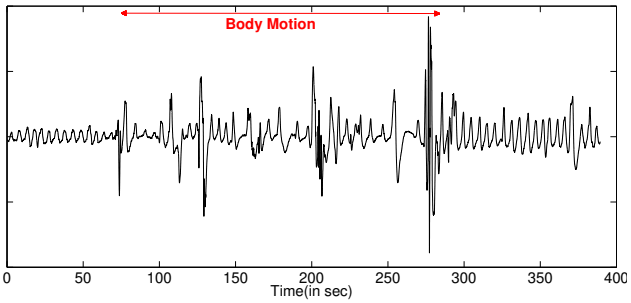


Figure 2.6: **Differentiating body movements from breathing.** ApneaApp leverages that when the subject moves her body, the sonar reflections experience large variations that do not have the periodicity of the breathing motion.

Specifically, we run our peak detection algorithm after disabling the minimum distance heuristic. We then check if the resulting peaks are aperiodic. To do this, we perform a second order derivative (difference of the difference) over the peak locations. For periodic breathing motion, this derivative is close to zero but is higher for aperiodic signals. We use a threshold of five to detect such aperiodic signals and compute the corresponding body motion durations.

Now, we compute the total sleep time by subtracting the body motion durations from the total experiment time. We note that if a patient moves twice within a short period, it is likely that she was awake for the whole duration between the movements. Thus, for the purposes of computing the sleep time, we combine any body movements that occur within a ten-minute period into a single awake duration.

2.4 Implementation

We implement ApneaApp as a third party Android app that does not require rooting the smartphone. ApneaApp plays frequency-modulated samples on the phone speaker and continuously record the raw samples from the microphone.

To understand the smartphone capabilities necessary to accurately assess sleep apnea, we experiment with four different Android phones: Samsung Galaxy S4, Samsung Galaxy S5, HTC One, and Galaxy Nexus. To operate ApneaApp the phone requires the ability to transmit and receive

	Samsung S4	S5	Galaxy Nexus	HTC One
18-20 kHz Mic	Yes	Yes	Yes	Yes
18-20 kHz Speaker	Yes	Yes	Yes	Yes
Separated Speaker/Mic	Yes	Yes	No	Yes
Apnea Detection	Yes	Yes	No	Yes
Breathing Frequency	Yes	Yes	Yes	Yes

Table 2.1: **Smartphone requirements for ApneaApp.** The table shows that accurate apnea detection requires a phone that has a speaker-microphone pair that is not co-located. This is because a co-located speaker-microphone pair results in unpredictable variations in the amplitude of the microphone signal. However, monitoring breathing frequency does not require stable amplitude and hence can work even with co-located speakers and microphones.

audio signals in the 18-20 kHz range from the speaker and microphone respectively. Table. 2.1 shows that all the tested smartphones satisfy this requirement.

Another critical requirement is a stable amplitude at the microphone. This is important since our algorithm leverages amplitude changes to detect sleep apnea events. Our experiments reveal that, across all the tested smartphones, when the co-located microphone-speaker pair concurrently transmit and receive signals, the amplitude at the microphone experiences unpredictable variations. This happens because of a feedback loop between the co-located microphone and speaker and can result in sleep apnea misclassifications.

To address this issue, we leverage that many smartphones today come with an additional microphone that is not co-located with the speaker. Our results show that using this microphone eliminates any unintended interaction with the phone speaker. This means that smartphones that do not have this additional microphone, such as the Galaxy Nexus, cannot be used to accurately detect sleep apnea events with ApneaApp. Note, however, that the measurement of breathing frequency is not affected by this feedback loop and hence ApneaApp can still be used to track the breathing frequency on Galaxy Nexus phones.

2.5 Clinical Study

We conducted a clinical sleep study with 37 patients (17 female and 20 male) between ages of 23-93 (mean: 50) for a total of 296 hours and compare the results from ApneaApp with the PSG study reports. The study was conducted at the sleep laboratory in the UW Medicine Sleep Center at Harborview.

Participants. The patients who participated in our study were examined by a sleep physician for common sleep disorders like excessive snoring or daytime fatigue and prescribed to undergo the in-laboratory polysomnography (PSG) study. The sleep lab is equipped to conduct PSG studies for a maximum of eleven patients per night; we randomly choose up to five subjects per night for our study. We do not screen patients based on their gender, race or national origin, but only consider adults. The patients who participated in our study were not provided any monetary benefits. We note two key points: First, our participant group included both patients undergoing the regular PSG study to diagnose sleep apnea as well as those undergoing the CPAP treatment [115] after being diagnosed positive for sleep apnea. Second, out of the 38 patients we approached, 37 consented to participate in our sleep study. Further, there was significant interest amongst them in using a smartphone instead of the existing PSG procedure to diagnose their sleep illness.

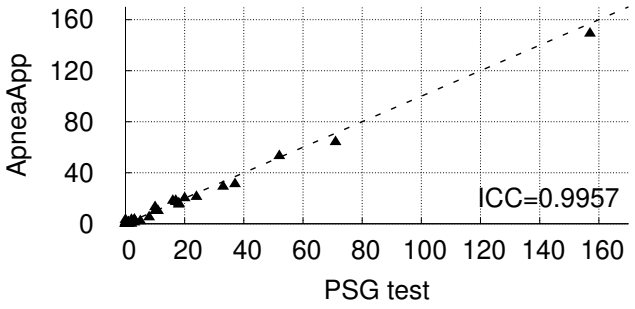
Protocols. In clinical PSG studies, the sleep lab technician assigns each patient to a separate room furnished with an adjustable king size bed. Each room is also fitted with a camera to get a visual log of the patient. Polysomnography was performed as follows according to American Academy of Sleep Medicine criteria: Lectro-encephalographic electrodes were positioned at two frontal (F7, F8), two central (C3, C4), and two occipital (O1, O2) locations (International 10-20 system of measurement) and were referenced to the contralateral mastoids. Chin electromyogram and right and left electroculogram electrodes were also applied. Airflow was measured using a nasal pressure cannula placed in the nose and a thermistor placed in the nose and over the mouth, allowing differentiation between nasal and oral breathing. Chest and abdominal respiratory effort were assessed by piezo respiratory-effort bands placed around the chest and abdomen, and snoring with a small microphone sensor placed on the throat just lateral to the trachea (Pro-Tech Services, Inc., Muk-

ilteo, WA). Oxygen saturation was measured from the index finger via pulse oximetry (Nellcor, Pleasanton, CA). Bilateral electromyogram electrodes were placed on the anterior tibialis muscle to monitor leg movements. After fitting the sensors, the technician performs bio-calibrations by asking the patient to do specific activities like leg motion, snoring and breathing exercises and ensure that the sensors are properly attached.

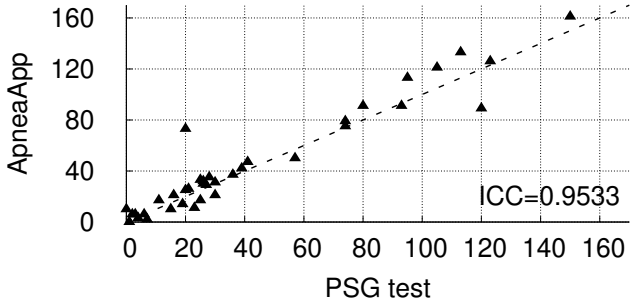
The procedures for our ApneaApp study occur concurrently with existing PSG protocols. Specifically, as the sleep technician attaches sensors to the patient, the first author places a Samsung Galaxy S4 smartphone running ApneaApp at the other end of the bed. The position and the orientation of the phone vary across patients depending on their sleep habits. For example, some patients prefer to surround themselves with pillows in which case the phone is placed behind the pillows at a farther distance. The distance of the phone from the patient varies between 0.3 and 0.7 meters. We leave the room and return in the morning to collect the smartphones from the sleep lab. The phone is plugged in throughout the study.[5] The above procedures were reviewed and approved by the UW Human Subjects Division.

Data Processing. The data from the clinical PSG study is sent to a third-party entity that scores the sensor data and provides the number of central and obstructive apneas, and hypopneas. Currently, a trained sleep technician manually performs the above scoring process resulting in a minimum delay of three days to obtain the scored data. Specifically, obstructive apneas, hypopneas, and central apneas were scored by the technician as follows: Obstructive apneas were defined by at least a 90% reduction in the pressure-flow signal with corresponding respiratory effort; central apneas were defined by at least a 90% reduction in pressure-flow signal without respiratory effort; and hypopneas were defined as a greater than 30% reduction in amplitude in pressure-flow signal lasting at least 10 seconds with an associated 4% oxygen desaturation.

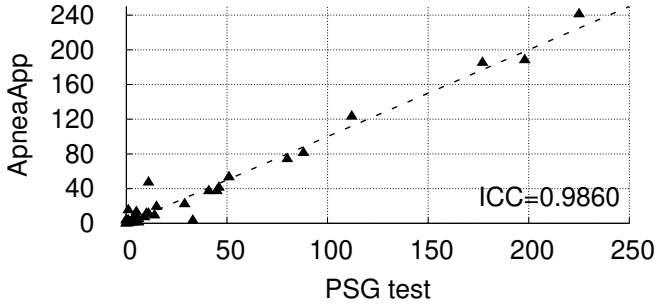
[5] A full-charged Samsung Galaxy S4 running ApneaApp lasts around four hours. When plugged in, however, the phone can simultaneously run ApneaApp and go from a no-charge state to being full-charged in four and a half hours. Since phones are often left charging during the night, we believe that the latter would be a typical ApneaApp use case.



(a) Central Apnea Events



(b) Hypopnea Events



(c) Obstructive Apnea Events

Figure 2.7: **Scatter-plots of the Number of Apnea Events.** The x-axis and the y-axis show the number of apnea events estimating by PSG and ApneaApp respectively. The plots show that there is a high correlation between the number of central, obstructive and hypopnea events identified by PSG and ApneaApp.

2.5.1 Sleep Apnea Detection Accuracy

The key metric used for sleep apnea diagnosis is the Apnea-Hypopnea Index (AHI), which represents the average rate at which apnea and hypopnea events occur during the sleep duration. To compute this, we need to estimate the number of central, obstructive and hypopnea events as well as the total sleep time. In this section, we evaluate ApneaApp's effectiveness in estimating these parameters.

Accuracy of apnea events. First we examine ApneaApp's effectiveness in computing the number of central apnea, hypopnea, and obstructive apnea events. We compare the number of apnea events computed by our algorithm with the baseline numbers provided by the PSG study. Figs. 2.7 (a)-(c) show the scatter plots for central, obstructive and hypopnea events. The 45-degree line denotes the ideal scenario when our estimates match the baseline.

The plots show that the number of apnea events detected by ApneaApp is highly correlated with the PSG test. Specifically, the interclass correlation coefficient (ICC)[6] between ApneaApp and PSG is 0.9957, 0.9533 and 0.9860 for central apnea, hypopnea and obstructive apnea respectively. We further note the following.

Amongst the apneas, the number of central apnea events computed by ApneaApp has the highest correlation (ICC=0.9957). This is because a central apnea event is detected as the absence of chest motion; this can be done accurately using the algorithm in sec:apneaalgo.

The number of hypopneas has a lower correlation coefficient (ICC=0.9533). The key reason for this is that hypopneas are detected when the amplitude of the breathing motion reduces by more than a threshold. Thus, small errors near this threshold can result in misclassifications. We note that in the PSG study, the technicians mark the hypopnea events manually by ball parking the 30% threshold; this introduces a number of misclassified hypopnea events in the ground truth data

[6]ICC is a standard statistic used in the sleep literature for quantifying the effectiveness of two procedures (in our case, PSG and ApneaApp) to produce correlated observations across a range of values [134]. ICC is also a more appropriate metric for us compared to recall accuracy: missing one of four apnea events results in a low recall accuracy of 75%. This however does not capture the system-level performance since for a eight-hour duration, the resulting AHI error is only 0.125.

False Positives	False Negatives
0.0023% ($\frac{59}{26070}$)	0.0336% ($\frac{43}{1281}$)

Table 2.2: **Detecting Body Motion.** ApneaApp detects body motion using abrupt changes to the periodicity of breathing movements. We use the body motion sensors from the PSG study as our ground truth. The table shows that we can accurately detect body movements without any sensor instrumentation of the human body.

itself. We also note that the error due to the lack of oxygen desaturation information in the case of ApneaApp was negligible.

ApneaApp can accurately compute the number of obstructive apneas by monitoring the chest movements alone (ICC =0.9860), without a nasal pressure transducer that can directly monitor air flow. Analyzing the misclassifications further reveals that ApneaApp misses obstructive apnea events where the airflow is obstructed for a very small duration (10-15s). This is expected because such events might result in less noticeable changes in the breathing effort/movements but would be noticeable at the nasal pressure transducer signal used in the PSG test.

Body Movements and sleep time estimation. To compute AHI, we need to measure the total sleep time. In the clinical PSG study, the staging step in the scoring process calculates the total sleep time using the brain activity (EEG) information. Since ApneaApp does not have the brain activity data, it measures the sleep time indirectly using body movements: the absence of body movements is used as an indicator that the patient is asleep. We evaluate ApneaApp's accuracy first in detecting non-breathing body movements and then in estimating the sleep time.

(i) Body Motion Detection Accuracy. ApneaApp detects non-breathing body movements using abrupt changes in the periodicity of the breathing movements (see sec:sleepestimate). To evaluate this, we use the body motion sensor data from the PSG study as the ground truth. Every 30 second epoch in the PSG data is annotated with the movement information; we consider each of these epochs as an event. We count a false positive when a body movement is detected by ApneaApp in

the absence of one and a false negative when ApneaApp does not detect a body movement that is present in the PSG data.

Table. 2.2 shows the number of false positive and false negatives across the 37 patients. The results show that ApneaApp can accurately detect 1238 of the 1281 body movement epochs. Further, ApneaApp misclassifies only 59 epochs as body movements across all the 37 patients. This is expected because body movements create significant changes to the reflected signal and hence can be easily distinguished from breathing movements. Thus we conclude that ApneaApp can accurately detect body movements without the need for placing sensors on the human body.

(ii) Sleep Time Estimation Accuracy. ApneaApp computes sleep time by subtracting the total duration of body movements from the total experiment time. Next, we evaluate how well this correlates with the sleep time computed in the PSG study using the brain activity (EEG sensors). Fig. 2.8 plots the total sleep time computed from both ApneaApp and the EEG sensor data across the 37 subjects. The plot shows that the mean and median sleep time error is 36 and 27 mins respectively. Further, ApneaApp overestimates the sleep time by an hour for six patients. This happens because these patients woke up in the middle of the night and lie on the bed without frequent movements, while trying to fall asleep. The brain activity during this period is high but there is little to no frequent motion. This is a fundamental limitation of estimating the total sleep time without access to the brain activity data. However, as we will see next, ApneaApp's sleep time accuracies are acceptable for diagnosing sleep apnea.

Accuracy of AHI estimation. Finally, we evaluate how well ApneaApp can compute the apnea-hypopnea index and diagnose sleep apnea. The results are as follows:

- Fig. 2.10 plots a scatter plot of the AHI values computed by ApneaApp versus PSG for the 37 patients. The figure shows that the AHI values are highly correlated between ApneaApp and PSG with an interclass correlation coefficient of 0.9816. Further, the average error in computing AHI is 1.9 events/hr. Larger errors tend to happen at higher AHI values where the diagnosis is fortunately more tolerant to large errors, i.e., the diagnosis is the same whether the AHI is 40 or 50.

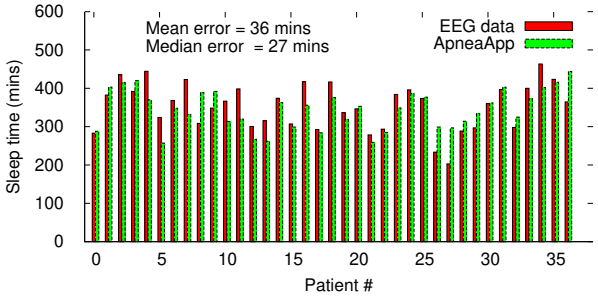


Figure 2.8: **Sleep time Estimation Error.** We compare the sleep time estimated using body movements (ApneaApp) and brain activity (EEG data). The mean and median error is 36 and 27 mins. ApneaApp tends to overestimate the sleep time for patients who wake up and lie on the bed without any movements while trying to fall asleep. ApneaApp's sleep time accuracies are however sufficient for the purposes of diagnosing sleep apnea. We also note that in practice patients could provide direct feedback about these unusually long awake periods to improve our estimation further.

- Fig. 2.9 shows the confusion matrix of the sleep apnea levels diagnosed by ApneaApp with thresholds of 5, 15, and 30. The table shows that ApneaApp diagnoses the sleep apnea level accurately for 32 out of 37 patients. The five misclassifications are between the no-apnea and mild-apnea diagnosis. Specifically, four patients who have mild-apnea are diagnosed by ApneaApp as having no apnea while one patient with no-apnea is diagnosed as having mild sleep apnea.

- For four of the five misclassified patients, the AHI values are in the 4.5-5.5 range centered at the threshold between no-apnea and mild-apnea. These boundary cases are handled separately by physicians depending on the patient preferences, symptoms, and insurance, effectively reducing the number of misclassifications to one.

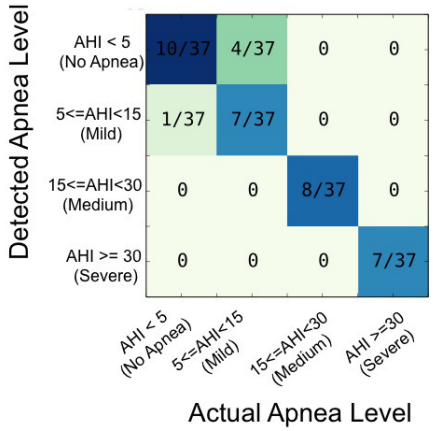


Figure 2.9: **Confusion Matrix of the Sleep Apnea Levels.** We accurately classifies 32 out of 37 patients between four sleep apnea levels. The five misclassifications occur between no-apnea and mild-apnea; four of them happen right at the boundary between the two levels with an error less than 1 event/hr. These boundary cases are handled separately by physicians depending on the patient preferences, symptoms, and insurance, effectively reducing the number of misclassifications to one.

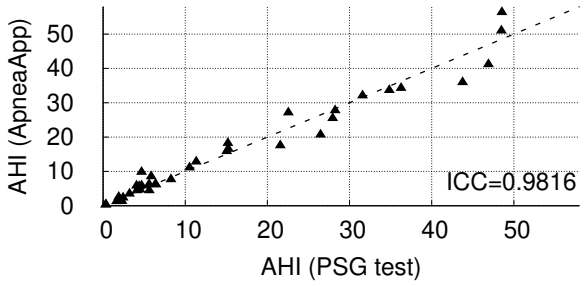


Figure 2.10: **Scatter-plot of the AHI Values.** The mean error in the AHI values computed between PSG and ApneaApp is 1.9. Larger errors typically happen at higher AHIs where the diagnosis is fortunately more tolerant to large error, i.e., the diagnosis is the same whether the AHI is 40 or 50.

2.6 Microbenchmarks

We run experiments with five participants (ages 24–25) in a home bedroom environment to evaluate the effects of various parameters such as orientation, position and distance. The bedroom is next to a major street with significant foot and vehicular traffic. Since we cannot imitate the sleep apnea events where the chest motion experiences amplitude variations, in these set of experiments we use the coarse-grained breathing frequency as a proxy to understand the effects of the various parameters. To obtain the ground truth data for the breathing frequency, we use a Vernier respiratory belt worn at the abdomen level.

2.6.1 Effect of Phone Distance, Orientation and Position

We first evaluate the effect of the phone distance, orientation and position on the breathing frequency accuracies.

Effect of the phone's distance. We place the phone on the left side of the subject sleeping in the supine position. In each trial, we monitor the breathing frequency over a three-minute duration after which the subject is asked to get out of the bed. For each distance value, we perform five trials for a total of ten minutes per location. We measure the accuracy of the breathing frequency estimated by ApneaApp versus the Vernier respiratory belt and plot the results in Fig. 2.11. The figure shows that:

- The breathing frequency accuracy is as high as 99.90% at distances up to one meter. We note that these accuracies are much higher than those observed for sleep apnea. This is because, as described in sec:imp, the breathing frequency on the phone is relatively stable compared to the amplitude variations that are necessary for detecting sleep apnea events.

- As the distance increases beyond a meter the accuracies decrease. This is because the strength of the reflections due to breathing reduces with distance, making the breathing signal noisy. The one-meter range is, however large enough to enable contactless breath monitoring

that is non-intrusive, as demonstrated in our clinical study. It also limits the negative effects of environmental changes farther away than a meter on ApneaApp's accuracies.

- The accuracies are unaffected by audible noise in the environment from the vehicular and foot traffic on the street. Introducing human conversations in the vicinity of the experiments also does not affect these accuracies. This is because, we use a high-pass filter to filter out audible signals below 18 kHz.

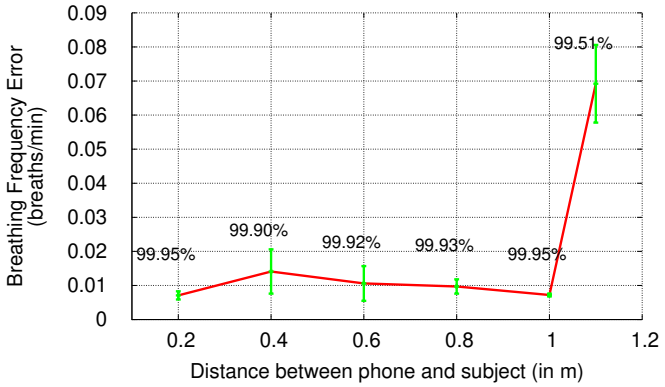
Effect of the phone's orientation. We place the phone 20 cm away and to the left of the subject. We then rotate the phone and compute the accuracies for each phone orientation. As before, for each trial, we perform five experiments for a total of 10 minutes per phone orientation. Fig. 2.11 plots the results as a function of the phone's orientation. We observe that the accuracies remain high, demonstrating that during ApneaApp's operation we do not need to fix the phone orientation.

Effect of the phone's position. Next, we experiment with the phone at different positions around the subject. Specifically we place the phone is four different positions — near the head, near the legs, and two positions to the left — along a semicircle of radius 40 cm centered at the subject as shown in Fig. 2.12. Fig. 2.11 shows that the accuracies are high when the phone is in the left positions and slightly lower when placed near the head and the feet. This is because in the latter positions, the head and the leg effectively block the chest/abdomen motion. We however note that the maximum error is less than 0.13 breaths per minute across all the phone positions.

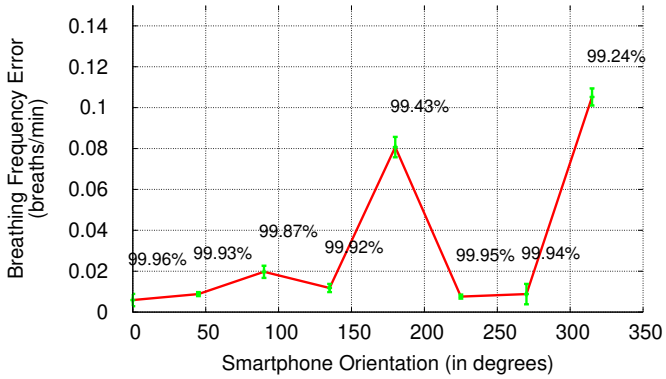
2.6.2 Effect of Sleeping Position and Blankets

Next, we evaluate the accuracies for different sleeping positions and in the presence of blankets.

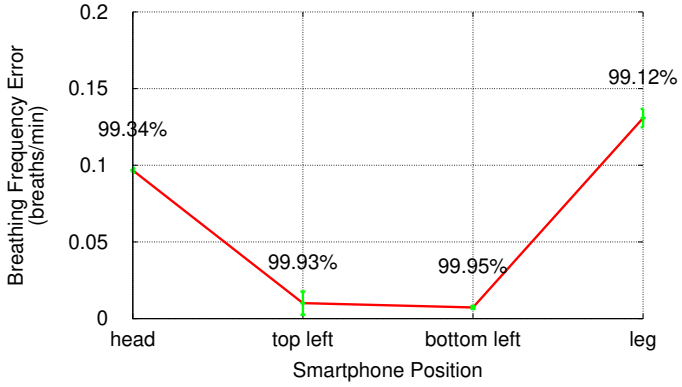
Effect of the subject's sleeping position. We consider four different sleeping positions: supine (on the back), prone (on the abdomen), lying on the left, and the right. We place the smartphone at a distance of 20 cm to the left of the subjects and measure the breathing rate accuracies. As before, for each sleep position we monitor the breathing rate over chunks of two minutes for a total of ten minutes. Fig. 2.13 shows that the average residual error is below 0.16 breaths per minute across



(a)



(b)



(c)

Figure 2.11: **Effects of phone distance, orientation and position.** The plots show the results for a subject sleeping in the supine position.

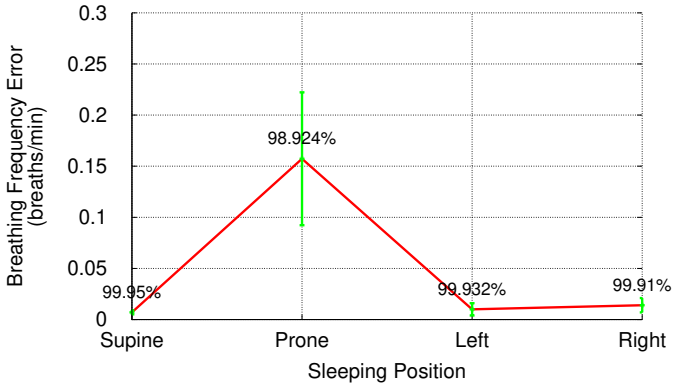
Figure 2.12: **The various phone positions used for the results in Fig. 2.11.** We experiment with four different positions along a semicircle centered at the subject.

all the sleeping positions. We note that the accuracy is lower when the patient is lying with her face down (prone). In this position, we noticed that both the signals from the Vernier belt and ApneaApp experience a larger variation. We however note that our clinical study tracks the chest movements throughout the sleep duration where the patient's sleeping position was not controlled.

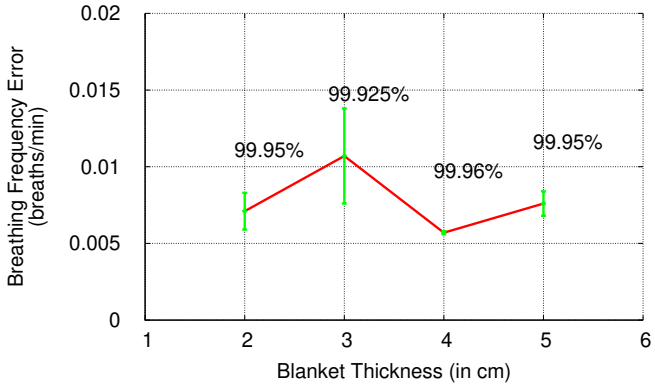
Effect of Blankets. We measure the breathing frequency accuracy for various blanket thicknesses. The subjects are asked to sleep in the supine position and the phone is placed left of the subject at a distance of 40 cm. We use four blankets with thicknesses varying from 2-5 cm. Fig. 2.13 shows that the accuracies are not noticeably degraded by the use of blankets. This demonstrates that ApneaApp is well suited for the sleep environment, which is further validated by our clinical study where all the patients used blankets.

2.6.3 Breathing Signals from Multiple Subjects

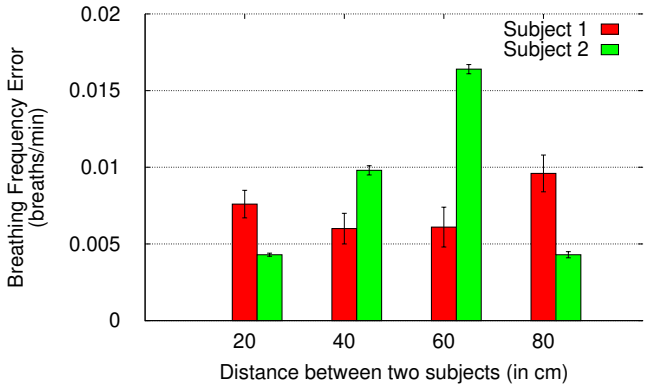
As discussed in subsec:sonar, the sonar reflections from multiple subject arrive at different times at the microphone. Thus, ApneaApp can simultaneously track breathing movements from more than one subject. To evaluate this, we monitor the breathing rate from two subjects sleeping in the supine position on the bed. Both the subjects where fitted with the Vernier respiratory strap to



(a)



(b)



(c)

Figure 2.13: **Effects of sleeping position, blanket thickness and multiple people.** The plots show the results for a subject sleeping in the supine position.

record the ground truth. The phone was placed at one side of the bed at a distance of 20 cm from the first subject. We increase the distance between the two subjects at intervals of 20 cm starting at the minimum distance of 20 cm where the two subjects could sleep without overlapping. As before, we repeat the experiments over five trials for a total of 15 minutes at each distance value.

Fig. 2.13 plots the accuracies for the two subjects as a function of the distance between them. The plots show that ApneaApp can accurately distinguish between breathing signals of subjects even when they are as close as 20 cm from each other. This is expected, because as described in subsec:sonar, our FMCW transmissions can be used to resolve reflections with a greater resolution than 20 cm. The key point, however, is that we can identify and track the breathing signals from more than one subject — a capability that is important for use in the home environment.

2.6.4 Evaluating ApneaApp's Audibility

ApneaApp transmits FMCW audio signals in the 18-20 kHz range. To evaluate the audibility of these signals, we ran ApneaApp with 144 subjects: 37 of whom were sleep apnea patients who took part in our clinical study, 50 additional sleep apnea patients not part of our clinical study, and 57 healthy undergraduate subjects who are seniors in the UW CSE department.

None of the 87 sleep patients reported any audible sounds from the phone. Further, the 37 patients who took part in our clinical study had no audibility issues for the eight-hour sleep duration. One of the undergraduates could hear faint sounds when the phone was placed next to his ear and a second undergraduate reported hearing a sound similar to TV static at distances of up to one meter. However 55 of the 57 undergraduate students reported no audible sounds from the phone even when placed right next to the ear. This shows that ApneaApp is inaudible for most adults, which is the target population for ApneaApp.

2.7 Limitations and Future Directions

We discuss the limitations of our system as well as various opportunities to improve it.

Increasing the AHI estimation accuracies. We design ApneaApp to operate in a home setting

without the need for instrumenting the users with sensors. We note however that a clinical PSG test uses a number of sensors that provide physiological information that is not available in our current design. For example, EEG sensors provide the brain activity data that helps accurately find different sleep stages (REM, non-REM, and awake). One could imagine analyzing fine-grained changes in the sonar reflections to find correlations with the sleep stages. Specifically, we plan to explore correlations between physiological parameters such as breathing rate and motion with the various sleep stages.

Similarly, since the PSG test uses cameras to produce a visual log of the patient, the report also summarizes the patient's sleeping position and the corresponding number of sleep apnea events. We plan to explore the feasibility of using accelerometers to predict these sleeping positions. Another direction is to augment the chest movements from our system with audible audio signals including snoring and evaluate the resulting accuracy improvements. Supplementing our system with these capabilities can improve our accuracies and also provide additional diagnostic information.

Detecting other respiratory-related events. In addition to diagnosing sleep apnea, a PSG test is also used to detect other events such as the respiratory-effort related-arousals (RERAs). These are arousals in the sleep that do not meet the definition of apnea and hypopnea events. A direction worth exploring is to leverage our sonar design with other smartphone sensors to detect these non-sleep apnea conditions.

Extracting other physiological information. In this chapter, we focus on extracting the minute chest and abdomen movements due to breathing. We note however that in principle our sonar design can also be used to extract the heart rate without requiring contact with the human body. Prior RF-based radar systems have demonstrated the feasibility of extracting heart rate from radio signals with Gigahertz of bandwidth. While heart rate is not necessary for sleep apnea diagnosis, we plan to use narrowband sonar transmissions on a smartphone and enable contactless heart rate detection, with the goal of better understanding the sleep architecture [175].

2.8 Discussion

This chapter presents a contactless solution that detects sleep apnea events on smartphones. To achieve this, we introduced a novel technique that can track chest and abdomen movements due to breathing on smartphones using FMCW sonar transmissions. We designed algorithms to detect central apnea, obstructive apnea, and hypopnea as well as estimate the total sleep time by analyzing the FMCW reflections from the human body. Compared to existing wellness systems that track sleep, ApneaApp detects the specific apnea events and computes the AHI index similar to the clinical standard polysomnography process. Hence it can generate the report that can be examined by a sleep physician. Further, the polysomnography process has a high variance as the subject is asked to sleep in an unfamiliar controlled environment and the results are based on a single night. ApneaApp has the potential to reduce this variance as it can be easily run at home for multiple days in the subject's smartphone.

Further, the breathing motion signal is a key physiological signal that is closely monitored to diagnose other conditions in addition to sleep apnea. For example, the breathing signal detection module in apneaapp can be used in baby monitors to avoid Sudden Infant Death Syndrome (SIDS). It can also be used to diagnose Chronic Obstructive Pulmonary Disease (COPD).

Chapter 3

OPIOID OVERDOSE DETECTION USING SMARTPHONES

Fatal opioid overdose remains a public health epidemic in the United States [195, 102, 136, 67, 181, 74]. Each day, 115 Americans die from opioid overdose and data from the Centers for Disease Control and Prevention (CDC) indicate the epidemic is worsening [153, 30, 180]. Unlike many life-threatening medical emergencies, opioid toxicity is readily reversed with rapid identification and administration of the overdose antidote naloxone or supportive respiratory care [167, 156, 182, 78]. Thus, a fundamental challenge of fatal opioid overdose events is that victims die alone or among untrained or impaired bystanders, in each case with no or insufficiently timely diagnosis and treatment [59]. To help connect potential overdose victims with widely available life-saving interventions, we developed algorithms for commodity smartphones that unobtrusively recognize opioid overdose by its physiologic precursors. Our system converts the phone into a short-range active sonar system, using frequency shifts to identify respiratory depression, apnea and gross motor movements associated with acute opioid toxicity. By creating overdose detection algorithms that can be deployed on devices most high risk individuals already own [125, 122], we hope to provide a harm reduction system that can automatically connect with naloxone-equipped friends and family or EMS to help prevent fatal overdose events [46, 75].

A mobile system that can detect opioid overdose precursors and events in real-time does not currently exist due to both design and validation challenges. Existing, human-based approaches to overdose diagnosis rely on medical grade equipment or trained recognition of diagnostic signs of opioid toxicity [49, 71, 77, 39, 81]. Achieving similar sensing capabilities on smartphones, without the need for medical grade equipment, is challenging since it requires tracking physiological parameters without being intrusive and violating privacy [60, 76]. In addition, validating the efficacy of any opioid toxicity system requires access to patients and data while high risk opioid use occurs,

which is difficult because this can represent a medically life-threatening situation. We overcome these challenges with an active sonar-based monitoring solution, leveraging access to two unique environments where people routinely experience overdose respiratory physiology without harm: (1) a legally sanctioned supervised injection facility (SIF), where people self-inject previously obtained illicit opioids under medical supervision and (2) the operating room (OR), during routine induction of general anesthesia.

3.1 System Design

This session describes a contactless smartphone-based system that matches the performance of an invasive respiratory impedance monitor in identifying 3 critical overdose precursors: opioid-induced respiratory depression, central apnea and simulated overdose events [87, 49, 1]. The system works by placing the phone within 1 meter of the subject as it monitors them during the post-injection period, the highest risk time for a fatal overdose event and the period when a victim would most benefit from rapid identification and resuscitation.

All participants provided informed consent and the studies were approved by the University of Washington Institutional Review Board, the University of British Columbia Office of Research Ethics and the Vancouver Coastal Health (VCH) Ethics Services (VCH operates the supervised injection facility, InSite).

3.1.1 Active sonar detection of breathing signals

Our design transmits frequency modulated continuous wave (FMCW) signals and analyses the frequency shifts resulting from human motion. The frequency shift, from which the respiratory rate is derived, was determined by performing a fast Fourier transform (FFT). We chose a FMCW chirp period of 10 ms, which gives a frequency resolution of 100 Hz. The unique challenge with opioids acting on the central nervous system is that breathing motion can be severely depressed. For our FMCW signal with a 4 kHz bandwidth this minute motion can translate to a frequency shift less than 8.33 Hz. To extract this, we perform an FFT over 15 consecutive chirps which linearly

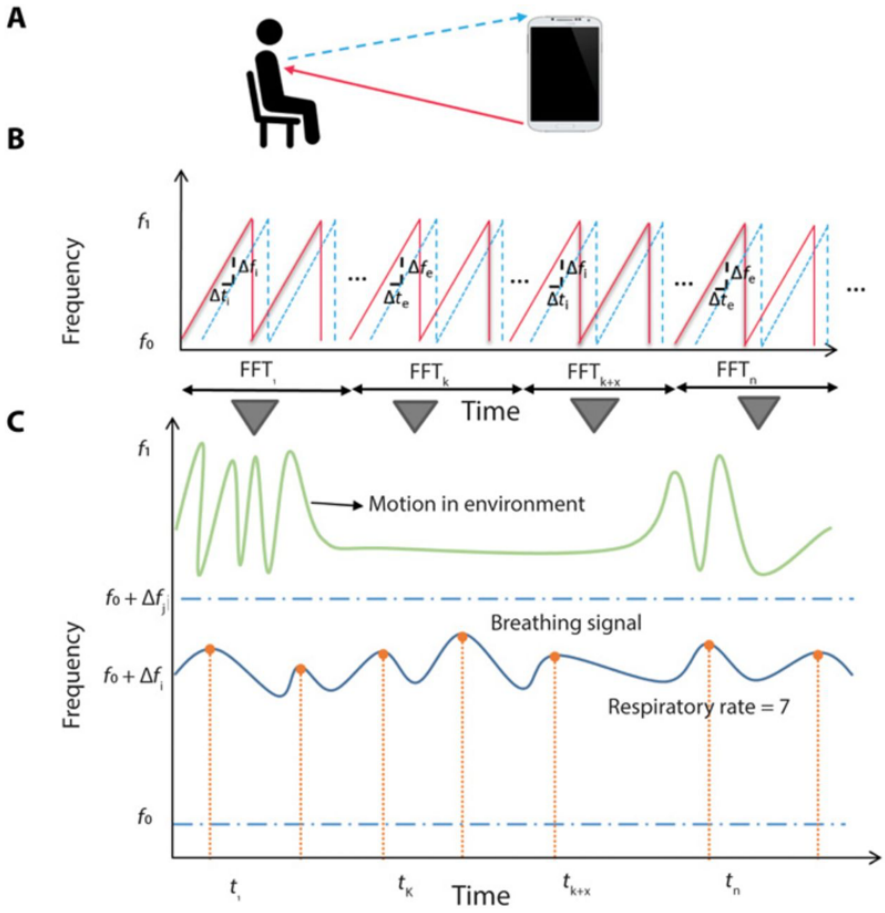


Fig. 1. Converting a smartphone into an active sonar monitoring system. **(A)** The smartphone's speaker transmits an inaudible, custom frequency modulated continuous waveform (FMCW) signal, which is reflected by the subject and recorded using the smartphone's microphone. **(B)** The reflections arrive at time delays Δt_i and Δt_e during inhalation and exhalation; the changes translate to unique frequency shifts Δf_i and Δf_e. **(C)** The frequency shifts can be estimated by taking a fast Fourier transform (FFT) over 15 chirps; the breathing signal is found in a frequency bin corresponding to the subject's distance from the smartphone. Motion in the environment from a different distance would appear at a different frequency bin and hence can be separated.

increases the frequency resolution to 6.66 Hz and thus captures even severely depressed breathing motion, down to a chest movement of 0.7 cm. By taking an FFT over 15 chirps, any high frequency motion within this duration is averaged and hence lost. However, since the average breathing rate of human subjects is less than 20 breaths per minute, which is a relatively low frequency motion, no significant breathing motion is lost within the 150 ms FFT duration.

To extract the breathing signal, the algorithm first estimates the distance of the person's chest from the smartphone over time. As described previously, the breathing signal is present in a unique frequency bin corresponding to this distance. To identify this bin, we examine the FFT bins corresponding to the frequency of the custom acoustic chirp (i.e., 18-22 kHz). The algorithm starts by looking from the 18 kHz bin (corresponding to distance zero) and proceeds to 18.320 kHz (corresponding to a distance of one meter). For each bin, it examines changes in the power value over a duration of 30 seconds by performing a second FFT over it. If a peak between 0.5 to 0.7 Hz (the typical breathing frequency of a human) is observed, then that bin corresponds to the breathing signal. Therefore, the second FFT occurs until the bin that corresponds to the breathing signal is found. In the worst-case scenario, the system may iterate through 48 bins before isolating the breathing signal. Once found, the signal recurs within the same bin as long as the subject remains in place. However, when a subject moves significantly, the bin corresponding to the distance has a motion signal instead of a breathing signal. In this scenario, the system re-initiates the search for the new bin containing the breathing signal. In particular, if the distance of the subject from the smartphone changes, we estimate the new distance by computing the bin corresponding to the breathing signal after the motion.

3.1.2 Detecting opioid-induced depressed breathing

Breathing motion is diminished when people use opioids, which makes peak detection difficult. Moreover, in the setting of opioid-induced diminished breathing, a small amplitude motion can add noise to the breathing signal and thus cause errors in peak count estimation. Further, subjects may use opioids multiple times over the course of a day and thus may have a depressed breathing signal at the initiation of another use event. To overcome this, we make two changes to the peak

detection algorithm. First, to remove any small motion noise, we run the data through a bandpass decimating Cascaded Integrated Comb filter. The filter removes any motion noise higher than a frequency of 1 Hz and also decimates the signal by a factor of two. Second, we collect a baseline breathing signal for a duration of one minute prior to the self-injection event. From the baseline collection period, the algorithm calculates the subject's average peak amplitude, peak prominence and average peak distance. These parameters are used to identify the peaks during post-injection monitoring. For the baseline signal, the algorithm leverages the periodicity of the breathing signal and the frequency limits of the breathing signal (less than 20 breaths/min) to estimate breathing peak parameters. Specifically, only peaks that are separated by a minimum of 20 samples (corresponding to a maximum breathing rate of 20 breaths/minute) are considered. During post injection monitoring we combine this condition along with the average peak parameters that we estimated in the first step. Peaks separated by a minimum of 20 samples that have an amplitude of at least 50% of the baseline and 30% of the peak prominence are classified as breathing peaks. If the number of peaks is less than or equal to 7, the epoch is marked as a respiratory depression event. If the distance between the peaks is greater than 10 seconds, we mark a central apnea event. If the number of breaths in the epoch is at least greater than 3, we update peak amplitude, peak prominence and distance values with the combined average of new peak values. Further, during our training periods, we noticed that people occasionally take deep breaths that have much higher amplitude than the rest of the peaks in an epoch. If a specific peak value is twice as great as the average peak values, the system does not use that peak value in average peak parameter computations.

3.1.3 Differentiating breathing from motion

Subjects using opioids may move their heads or hands, which are motions that can affect the time delay of the echo. Since subjects' faces and hands are closer to their chests and are approximately at the same distance from the smartphone, the change caused by these motions can be added to the breathing signal in its frequency bin during the primary algorithm's FFT operation. Moreover, this motion has higher amplitude compared to the more subdued breathing motion and can overpower the breathing signal, making it difficult to extract the breathing motion. While normally such

motion noise would be problematic, the presence of motion provides additional information about the subject. Specifically, sustained motion indicates that the subject is active and not overdosed. Similarly, motion that is followed by breathing indicates that the subject is active and thus not overdosed. On the other hand, motion within the operational range that is followed by an absence of breathing likely indicates an overdose scenario. Hence, we modify the algorithm to differentiate between a signal corresponding to periodic, low-frequency breathing motion and one that belongs to high frequency body motion, which is aperiodic and high amplitude. We identify this by looking at the peaks in the second FFT operation of the 30-second signal corresponding to each bin. If the peaks have higher frequencies and an amplitude at least twice that of the breathing frequency peaks, then the instance is classified as a motion epoch. If the motion is absent or present only for a few seconds, the algorithm considers it to be a breathing signal and processes it to identify the respiration rate.

3.1.4 Distance recalibration

When we encounter a motion epoch, the distance of the subject with respect to the smartphone can change. Hence after every motion epoch we need to run the recalibration step to detect the new frequency bin that corresponds to the new distance of the subject from the smartphone. When we encounter the first motion epoch, we set the motion bit to 1 and examine the next epochs. For subsequent epochs, we search all the nearby FFT bins until we detect the bin that has the breathing signal. We then use this new bin for the next set of epochs until we see the next motion epoch. For the first breathing epoch after the motion epoch, we update the peak parameter values to the average values of the new epoch corresponding to the new distance of the subject.

To review, we first filter the recorded signal using a high pass filter to remove audible environmental noise. We then split the data into 30-second epochs and run the distance estimation step described above on the first epoch to identify the bin that contains the breathing signal. We estimate average breathing peak amplitude, peak distance and average peak prominence for this epoch. For subsequent epochs, we check the same frequency bin in the distance estimation algorithm. If it contains the breathing signal, we use the previously estimated amplitude and prominence values to

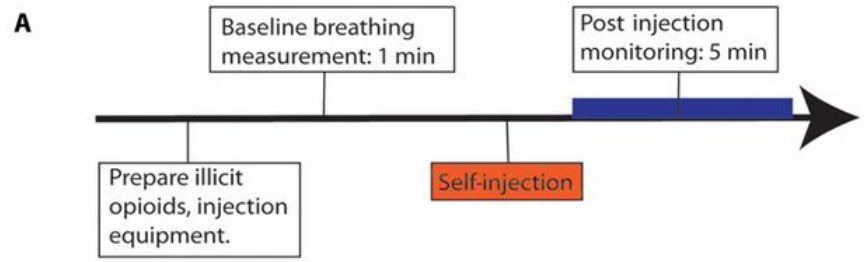
determine the breathing peaks in this epoch and subsequently update them with the new peaks of the current epoch. This continues until the bin contains a motion signal instead of breathing signal. If the subsequent epoch does not contain the breathing signal and instead contains the motion signal (high amplitudes, more peaks), we mark it as a motion epoch and run the recalibration step for the subsequent epochs until we find the new breathing signal.

3.1.5 Suppressing environmental motion

High-risk opioid use is commonly done with others, which introduces another source of potential interference [61, 170]. In fact, in the SIF environment, there are routinely several people around; there is talking, overhead music playing and occasionally personal dogs present. Thus, another potential source of interference is the breathing and motion of those other than the subject. In this case, the interfering subject's breathing or motion may change the received echoes at the smartphone. However, since the interfering subject(s) will mostly be located at different distances with respect to the smartphone, their breathing motions (as determined by the primary FFT operation) would occur at different frequency bins than that of the subject of interest. Assuming that the smartphone is closest to the intended subject, viz., within one meter, the first frequency bin containing the breathing signal likely corresponds to the breathing motion of the intended subject. The algorithm therefore filters out any breathing detected at farther distances. This is an advantage of using FMCW since it separate echoes as a function of distance from the smartphone.

3.1.6 Computational complexity

In the worst case, our algorithm performs 54 FFT computations per second and one linear peak estimation algorithm. Such operations can each be computed within a few milliseconds on an off-the-shelf smartphone [2, 3]. This delay is within the expected human response time (for visual overdose identification) of a few seconds. Finally, based on the duration of high-risk opioid self-injection events, we expect the application to run typically for less than 45 minutes per day and not more than 15 minutes per event. The algorithm's computations along with the sensor data collec-



B

		Reference standard	
		RD(+)	RD(–)
Algorithm	RD(+)	41	5
	RD(–)	6	42
Sensitivity	87.2% (95% CI, 74.2–95.1%)		
Specificity	89.3% (95% CI, 76.9–96.4%)		

C

		Reference standard	
		CAE(+)	CAE(–)
Algorithm	CAE(+)	47	2
	CAE(–)	1	44
Sensitivity	95.9% (95% CI, 86.0–99.5%)		
Specificity	97.7% (95% CI, 88.2–99.9%)		

Fig. 4. Measurement of real-world, high-risk opioid use events. (A) Protocol for the studies conducted in the supervised injection facility.(B-C) Sensitivity and specificity for respiratory depression (RD) and central apnea events (CAE).

tion consumes 6-18% of a phone's battery power for this duration. In addition, most fatal overdose events occur within a private residence, hotel or motel [94], which should have an available power source.

3.2 System Evaluation

3.3 Vancouver Supervised Injection Facility (SIF)

We first report data of our system deployment in Vancouver, British Columbia, within an approved supervised injection facility (see Figure 3). While our primary target is people who use opioids when alone (the demographic at highest risk for fatal overdose), we choose the SIF environment because it facilitates safe, real-world testing and algorithm development based on actual opioid self-injection events. Acute, life-threatening overdose events requiring medical intervention still remain relatively uncommon in this environment, occurring in less than 1% of opioid use events (approximately 500 supervised injections occur per day in the facility) [108]. Therefore our pri-

mary outcomes of interest were post-injection central apnea (cessation of breathing for 10 seconds or more) [1] and opioid-induced respiratory depression (respiratory rate $\leq$7 breath/minute) [4, 64], both of which are necessary precursors to lethal opioid intoxication events.

3.3.1 Participants

All injection opioid users who utilize the SIF, were over 18 years old, and had capacity to provide informed consent (as determined by Insite staff), were eligible for study inclusion. Opioid users under age 18 and impaired users were not eligible (per SIF protocol, severely impaired users are assisted and unable to use the facility). Potential participants were identified at the time they checked into the SIF for the purposes of supervised opioid self-injection, and approached by a research assistant for informed consent. Participants were approached consecutively following check-in into the facility. Participants were given a $5 coffee card for participation.

3.3.2 Measures

We compute the breathing rate in order to identify central apnea occurrences respiratory depression, both of which can indicate or precede a fatal opioid overdose. We define a breathing rate of $\leq$ 7 breaths/min to be a respiratory depression event; the absence of breathing for 10 seconds or more to be an opioid-induced central apnea event. We chose a respiratory rate of $\leq$ 7 breaths per minute because the Agency for Healthcare Research and Quality (AHRQ) finds this rate sufficiently dangerous to recommend as a trigger for a hospital's Rapid Response System [4, 64]. The Food and Drug Administration (FDA) defines an apnea event as cessation of breathing for 10 seconds or more and requires FDA-approved apnea devices to detect this threshold [1].

3.3.3 Protocol

Clients who consented obtained sterile injecting equipment per routine and were assigned to a monitored injection stall (see Fig. S3A) and were asked to prepare their drugs as they normally would. Monitored stalls were equipped with a study smartphone on the tabletop. All subjects, regardless

of participation, received standard clinical monitoring by the SIF clinical staff according to institutional protocols. Post-injection overdose detection by staff was defined by standard institutional triggers listed in Extended Data Table 1. Of note, staff monitoring relies on visual monitoring for acute clinical distress and does not involve active respiratory monitoring equipment.

Once participants had prepared their equipment and drugs, the participant was fitted with a respiratory impedance monitor for reference standard monitoring (see Fig. S3B). Then the participants were asked to remain seated and breathe normally for one minute to establish a baseline respiratory rate. The smartphone, placed within one meter of the participant on the injection stall table, began respiratory monitoring at the initiation of the one minute baseline measurement (see Fig. S3C). Participants then self-injected opioids and monitoring continued for five minutes. We chose five minutes because this represents the critical period when an acute overdose would occur; from a pharmacology perspective, fentanyl reaches a peak plasma concentration within 3-5 minutes and more than 80% of the injected dose leaves the plasma by five minutes [139]. If an overdose event occurred, or a patient was in a physiologic state sufficiently concerning that a trained medical staff member walked over to check on a patient, it was recorded by the research assistant and counted as an intervention event.

We also examined the baseline and post-injection nadir respiratory rate of participants (see Fig. S4). Not unexpectedly, most participants experienced a decrement in breathing rate following opioid self-injection. We did not observe meaningful differences in post-injection breathing rate decrements based on what users reported they were injecting. Heroin users had a mean post-injection nadir decrease of 4.5 $\pm$ 3.2 breaths/minute; fentanyl users, a mean decrement of 3.7 $\pm$ 3.1 breaths/minute; hydromorphone/morphine users, a mean decrement 2.5 $\pm$ 2.4 breaths/minute. In this environment we were unable to account for unmeasured variability in substance purity, dosage or tolerance-related physiologic response. Towards that end, a recent report identified approximately 93% of seized heroin contained fentanyl or fentanyl analogues in Vancouver, British Columbia [5].

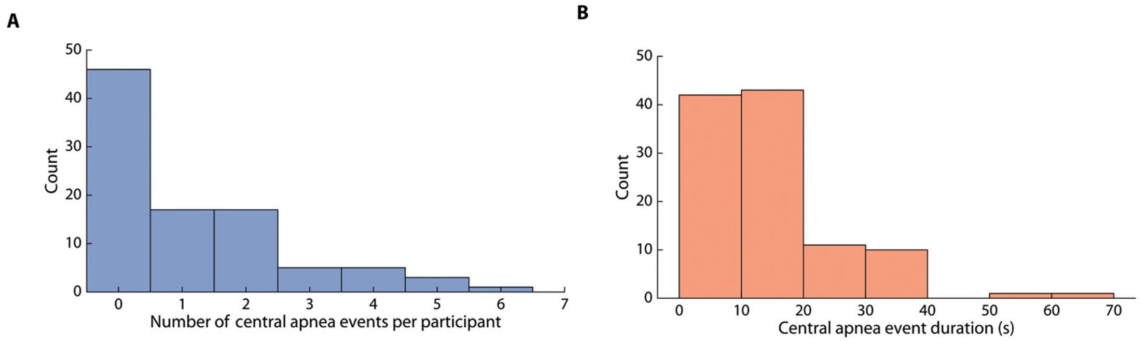


Figure 3.1: **Distribution of observed central apnea events in the supervised injection facility. (A)** Histogram of central apnea events per participant. **(B)** Histogram of the duration of the central apnea events identified by our system.

3.3.4 Results

We recruited participants over 209 self-injection instances (194 unique participants): 115 injection events were used as a development set and 94 were used as an evaluation set to measure algorithm performance. Results from the evaluation set are presented here. The average age was 43 $\pm$ 11.0 years; the average height and weight were 178 $\pm$ 8.3 cm and 77 $\pm$ 12.4 kg, respectively. Sixty-four participants (68%) reported using heroin; 19% reported using fentanyl; 13% reported using morphine or hydromorphone. Following injection, 47 participants (50%) experienced clinically significant respiratory depression; 49 participants (52%) experienced at least one post-injection central apnea event; 8 participants (8.5%) had a manual intervention by clinical staff, of which 2 participants (2.3%) experienced an overdose event requiring clinical resuscitation (i.e., oxygen, bag-mask ventilation and/or naloxone therapy). Both overdosed participants were successfully resuscitated by the clinical staff without issue. Across the 209 self-injection events, no participants heard the FMCW acoustic signals emitted from the phone. Further, during the act of illicit opioid self-injection, we note that people generally did not wear restrictive clothing (e.g., bulky jackets) because it complicates obtaining venous access.

The system had strong performance detecting post-injection central apnea events and respira-

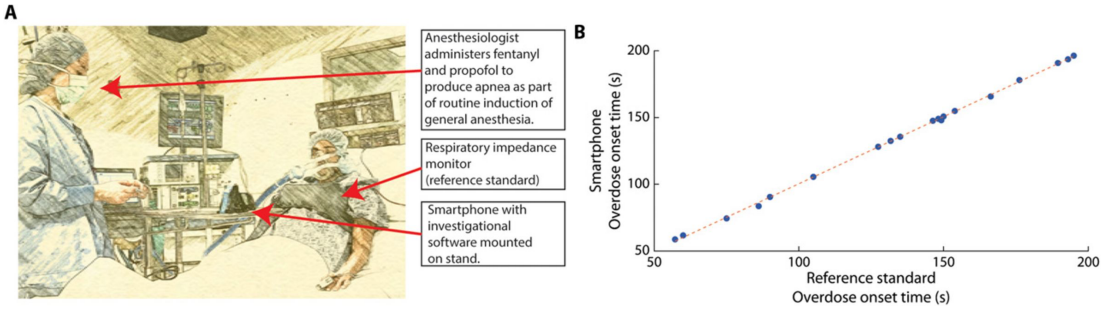


Figure 3.2: **Measurement of simulated overdose events in the operating room. (A)** The phone is placed within 1 meter of the patient on a surgical (Mayo) stand. A respiratory impedance monitor is fit around the patient's chest to measure the true respiratory rate and apnea status. Healthy patients wearing all standard operating room monitors have general anesthesia induced. **(B)** Comparison of time to detection of simulated overdose, based on algorithm-identified respiratory failure onset, smartphone vs. real-time detection by the reference standard.

tory depression, compared to the reference standard (Figure 4 C,D). It had 95.9% (95% CI, 86.0% - 99.5%) sensitivity for identifying post-injection central apnea events (cessation of breathing for 10 seconds or longer) and specificity of 97.7% (95% CI, 88.2% - 99.9%). The system had 87.2% (95% CI, 74.2% - 95.1%) sensitivity for identifying post-injection respiratory depression (respiratory rate $\leq$7 breaths/minute) and 89.3% (95% CI, 76.9% - 96.4%) specificity (see Figure 4 A,B).

Figure 3.1 shows the distribution of the number of central apnea events per participant, as identified by our system. Forty-eight percent of the participants had no central apnea events, all of whom required no intervention. Among those who experienced a post-injection central apnea event, 71% had 1–2 central apnea events during the 5 minute post-injection monitoring period. Figure 5B is a histogram of the durations of these central apnea events. The plot shows that 66% of central apnea events were $\leq$20 seconds in duration. We also note that both overdosed participants had a central apnea event of at least 30 seconds prior to clinical intervention.

3.4 Simulated Overdose Detection, Operating Room

A key limitation of the SIF environment is the limited occurrences of overdose events requiring clinical resuscitation. To address this, we next report data of our system deployment in the operating room. We choose this environment because it is controlled and allows us to safely simulate the worst-case scenario of acute opioid toxicity: immediate loss of consciousness coupled with respiratory depression that would be fatal or critically morbid without intervention. Such conditions are safely reproduced during routine induction of general anesthesia (see Figure 6) when patients receive fentanyl and other anesthetic drugs. To optimize surgical conditions, patients are routinely given fentanyl and other potent drugs immediately prior to surgery to purposefully induce apnea. In other words, the physiologic equivalent of drug overdose occurs each time a patient undergoes general anesthesia. However, patients are unharmed by this induced "overdose" because of their supra-physiologic oxygen stores and the timely ventilatory support the anesthesiologist provides. Therefore, anesthetizing a patient in the OR offers a unique way to safely simulate an otherwise lethal overdose event induced by fentanyl and other opioids.

Healthy patients free of cardiopulmonary disease, aged 18-55 and scheduled for elective surgery, were eligible for the operating room study and were approached on the day of surgery. Participants were given a $50 Amazon gift card for participation.

3.4.1 Measures

The primary outcome in the operating room study, simulated overdose, was detection of the presence of sustained apnea (defined as cessation of breathing for 30 seconds). We chose 30 seconds of apnea as a suitably safe period for a pre-oxygenated individual to be completely apneic before intervention [192].

3.4.2 Protocol

Once inside the operating room, patients were fitted with standard anesthesiology cardiopulmonary monitors: pulse oximeter, blood pressure cuff, 5-lead EKG. In addition to the standard anesthesiol-

ogy monitors, participants were fitted with a Vernier respiratory belt to provide reference standard respiratory monitoring. The smartphone was placed on a surgical stand $\leq$1 meter away at approximately chest level (see Fig. S2). Next, the clinical team conducted their standard operating room pre-induction safety procedures (i.e., safety surgical and pre-anesthesia checklists). The ventilation mask was affixed to the patient with a strap. Operating room personnel were asked to stand away from the patient; the attending anesthesiologist was beside the patient and immediately available. Next, per standard anesthetic procedure, the participant breathed 100% oxygen for 3-5 minutes until their expired oxygen levels were greater than 85%–a level deemed safe to administer induction doses of anesthetic agents to induce apnea [163]. This standard pre-oxygenation procedure allows the body to be safely apneic without having oxygen saturation levels fall to unsafe levels during apnea periods that may persist for as long as 7 minutes under normal conditions [192]. The attending anesthesiologist then administered induction doses of fentanyl and propofol in doses at his/her discretion for standard induction of general anesthesia (the average fentanyl dose was 1.4 mcg/kg; the average propofol dose was 2.9 mg/kg). The attending anesthesiologist announced the moment when the patient had become apneic, at which time a timer was started. The timekeeper announced the elapsing time in 10 second intervals. At 30 seconds, the protocol was officially over, at which point the anesthesiology team assumed control of the airway (administered a neuromuscular blocking agent, if indicated) and provided manual ventilation and inserted an endotracheal tube or laryngeal mask airway. Once the participant's airway was secured and it was deemed safe by the clinical team, the study team removed the research equipment and exited the operating room. Per the IRB protocol, the attending anesthesiologist could intervene at any moment and for any reason during the protocol should the patient require intervention. Breaking protocol was in no cases required during the study as all patients safely tolerated the procedure.

3.4.3 Results

We recruited for 35 instances of simulated overdose (34 unique participants): 15 patients were used as a development set to generate the algorithm, and 20 were used as an evaluation set to validate algorithm performance. Results from the evaluation cohort are presented here. The average

participant age was 33 $\pm$ 10.8 years; the average weight was 75 $\pm$ 14.6 kg; 60% of participants were female. In the evaluation set, as expected all 20 patients experienced true overdose physiology, characterized by post-injection loss of consciousness and diminished or absent breathing. Our algorithm identified 19 of the 20 simulated overdoses as having disordered breathing. Of the 19 correctly identified patients, 18 patients experienced sustained apnea (terminated per protocol after 30 seconds); and 1 patient had significantly diminished breathing that the algorithm identified as an overdose. The 1 patient who was incorrectly classified had a breathing signal just above the algorithm's threshold. In each case where the algorithm correctly identified the overdose event, it detected the onset of respiratory failure similarly to the real-time reference standard (Figure 3.2). We note that specificity is not meaningful in this environment since all participants experience the simulated overdose event (i.e., all patients undergoing induction of general anesthesia lose consciousness and experience depressed breathing) [124].

3.5 9-1-1/Emergency Medical Services (EMS) response

Because the city of Seattle and outlying King county have separate data collection procedures, separate files were supplied with the relevant out-of-hospital cardiac arrest response times. The out-of-hospital cardiac arrest response times for the city of Seattle were from the years 2009-2014 (n=1,730); the response times for cardiac arrests outside of the city were from the years 2013-2017 (n=4,219). EMS response time indicates the time from 9-1-1 call to when EMS is at the patient's side, which is calculated by when the defibrillator is turned on or when CPR begins, whichever occurs first. The fatal overdose location data were from the years 2010-2017 (data include intentional and unintentional opioid overdose events). If an out-of-hospital overdose victim died en route to the hospital or at the hospital, the geocoded location of where they were initially found by EMS was used.

Fig. S1 was generated with QGIS [6]. Using the response time data from Seattle and King County we gridded the area into 1 km across hexagons and binned the median response time per hexbin. We chose median response time to mitigate the effect of extreme outliers. Fatal opioid overdose locations were then overlaid. We added a Mapbox base layer which stylizes Open-

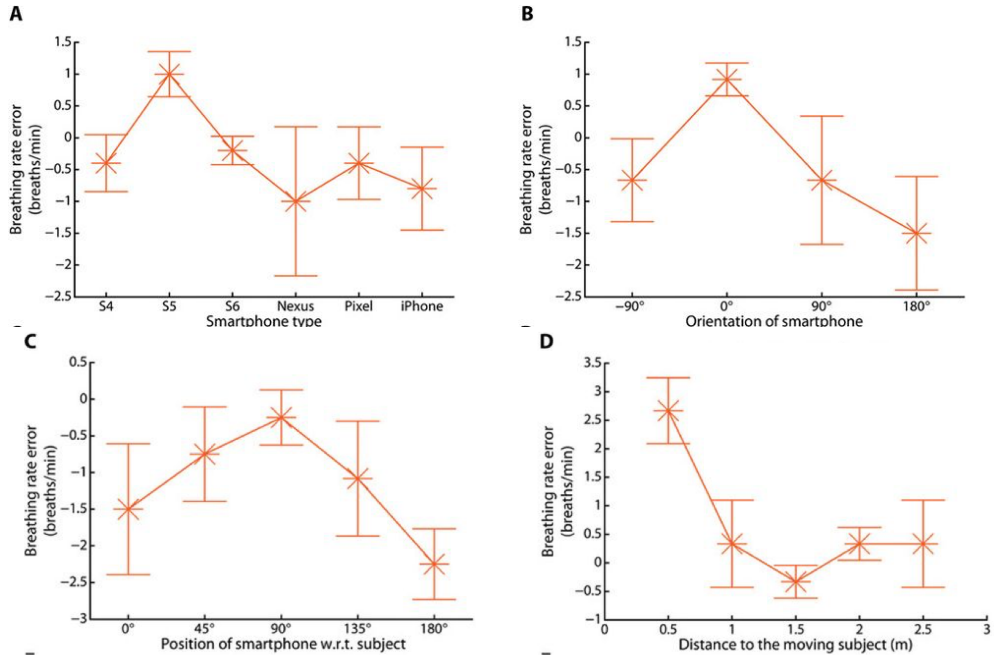


Figure 3.3: **Breathing rate accuracy across different scenarios.** The system is evaluated across **(A)** different smartphone models, **(B)** orientations of the smartphone, **(C)** various positions of the smartphone w.r.t the subject, **(D)** in the presence of interference from another nearby moving subject

StreetMap data. The map uses geographic coordinates and WGS84 datum. Overdose coordinates were converted from a state plane projection ("NAD 1983 HARN State Plane Washington North FIPS 4601 Feet").

3.6 Microbenchmarks

We run experiments to evaluate our algorithm in various benchmark scenarios. Figure 3.3A shows the performance of our system across various smartphone models — Samsung S4, S5, S6, Nexus, Google Pixel and iPhone 5S. We ran the same Android application across the first five phone models and a custom app programmed using Swift on the iPhone. For all devices, the five subjects who participated in this experiment were around 50 cm from the smartphone, and their breathing was monitored for 5 minutes each. The result shows that the breathing rate accuracy is consistent

across the tested phones. Slight variations are due to changes in the frequency response of the hardware at the higher frequencies. Finally, the speaker and microphone are located in very similar locations for all smartphones: one speaker is closer to the ear for voice calls and the external speaker is at the other end for music; one microphone is at the top close to the camera for video recording and the second is close to the bottom of the phone for voice calls. This explains why we did not have to customize our algorithms across the various phone models. Note that positive errors denote over-counting the number of breaths per minute and negative errors denote under-counting the number of breaths per minute.

Next we run experiments to evaluate our algorithm in various smartphone orientations (Figure 3.3B). We first place a smartphone 50 cm in front of the subject. The phone is initially placed in a portrait orientation, with an orientation angle of zero degrees. The subject wears a Vernier respiratory impedance monitor that is used as a reference standard. We monitor the breathing rate of the subject for a period of 2 minutes. We then repeat the above experiment by changing the orientation of the smartphone to three other angles: 180 degrees (portrait with smartphone upside down); 90 degrees (landscape orientation); and -90 degrees (the opposite landscape orientation). We perform this experiment for all six smartphone models and the figure plots the average breathing rate error.

In the SIF experiments across the 209 self-injection events, participants wore a range of different clothing. In addition, we also ran a benchmark experiment with four different materials of clothing: cotton shirt, silk shirt, hoodie jacket and jean jacket. The subject was at a distance of 50 cm from the smartphone and the breathing rate was monitored for 5 minutes in each case. The breathing rate error was 0.01 ± 0.70 for cotton shirt, -0.6 ± 0.54 for silk shirt, 0.4 ± 0.54 for a pull-over and -1.6 ± 1.6 for jean jacket. We note that very bulky clothing can absorb acoustic signals, however such clothing is not conducive to obtaining venous access and are not commonly worn while performing real-world opioid injections, as demonstrated in our Vancouver results.

Figure 3.3C shows the results for smartphones placed at five different angles with respect to the subject: zero degrees (phone placed on right side of the subject); 45 degrees; 90 degrees (phone placed in front of the subject); 135 degrees; and 180 degrees (phone located on the left side of the

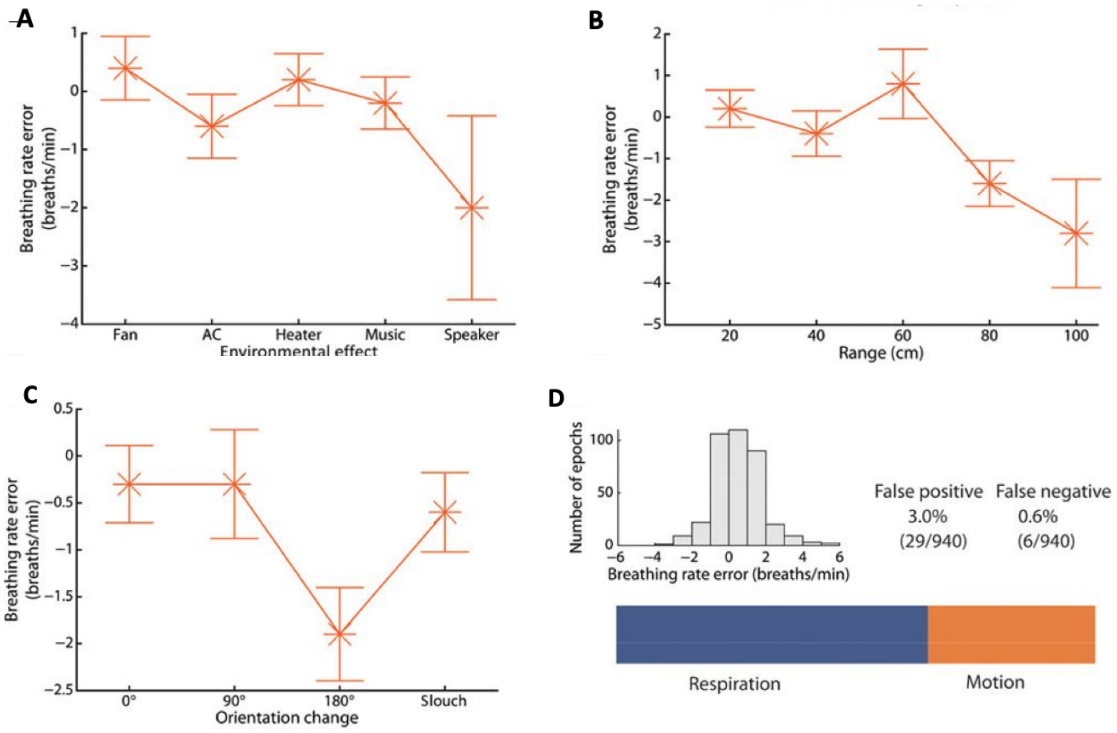


Figure 3.4: **Breathing rate accuracy across different scenarios.** The system is evaluated across **(A)** with environmental noise from devices placed 75 cm from the subject, **(B)** as a function of distance from the smartphone and **(C)** recalibration accuracy after the subject changes the orientation of the phone as well as slouches. In addition, **(D)** shows the fraction of time the subject's respiration and other motion was captured by the algorithm in the SIF deployment. W.r.t = with respect to.

subject). We monitor the breathing rate of the subject at each angle for two minutes and repeat this experiment for all the six smartphone models to compute the average breathing rate error. We can see that the accuracy is best at 90 degrees, i.e., when the smartphone is directly in front of the subject. As the smartphone moves away from that angle, the relative motion recorded by the microphone decreases, causing it to miss breaths. Further, the phone speakers and microphones typically have a directionality of 180 degrees; hence, the system works best when the speaker and microphone are facing the subject.

As described above, the FMCW algorithm can detect the subject's breathing motion in the presence of other motion in the environment occurring at different distances from the phone. To verify this, we conducted a benchmark experiment with two subjects, the results of which are shown in Figure 3.3D. One subject, sitting in front of the phone, had their breathing rate monitored, and the second subject was moving at different distances from the first subject. We computed the accuracy of the breathing rate of the first subject when the second subject was moving at varying distances of 0.5 m to 2.5 m. The plot shows that after 1 m, the distance of the moving subject did not affect the breathing rate of the first subject. For 50 cm, the interference was slightly higher, but the breathing rate error was still as low as 2.6 breaths/min.

As we note, our system first filters audible sound signals in the recording and therefore is not significantly affected by environmental noise. To stress test this performance, we conducted breathing experiments in the presence of five common environment noise producers: a fan, an AC unit, a heater, background music and a nearby loudspeaker. In all cases, the subject was in front of the phone at a distance of 50 cm, and the sound-producing device was at a very close distance of 75 cm from the subject. We chose that distance to stress test our system and would expect that such noise-producing devices would be much farther away in practical scenarios. Figure 3.4A shows that the fan, AC unit, heater and music devices do not affect the system performance. The loud speaker decreases performance because the loud sound from it saturates the analog-to-digital converter (ADC) of the microphone causing a higher breathing error rate, given its proximity. We expect that in practice subjects could be instructed not to have such high volume loudspeakers less than a meter away from them.

Next, the performance of our system depends upon the distance of the subject from the phone. To understand this, we conducted experiments with subjects at varying distances from the phone ranging from 20 cm to 1 m (Figure 3.4B). The subjects were monitored for two minutes at each distance, and we repeated this experiment for three different subjects. We can see that breathing rate accuracy slightly decreases with distance. At 1 m, the SNR becomes low causing the system to miss multiple breaths. This demonstrates that the operational range of our design is less than one meter, which can be considered an advantage since it is not affected by environmental motion outside that range.

Our FMCW algorithm re-computes the distance of the subject whenever the subject or the phone position changes. To verify this, we conducted a benchmark experiment with the subject at a distance of 50 cm from a smartphone initially placed at zero degree orientation. After the first minute, the subject changed the phone orientation to 90 degrees and after the second minute to 180 degrees. Finally, the subject slouched, and the breathing motion was monitored at that position for one minute. This process was repeated by four other subjects and the breathing rate was monitored for each smartphone and subject position. The figure plots average breathing rate accuracy. Figure 3.4C shows that the smartphone finds the new FFT bin corresponding to the new distance every time the orientation changes, including slouching instances, and can extract the breathing rate from the correct bin.

Finally, we also tested our algorithm's ability to discriminate between gross motor movements and breathing motion in the SIF, which is important because gross motor movement provides a surrogate indication of the absence of unconsciousness due to opioid toxicity. We observed that approximately 15% of the post-injection epochs contained gross motor movement that precluded measurement of breathing because the motion overpowered the signal; the algorithm had excellent performance with respect to identifying this type of motion (Figure 3.4D).

3.7 Discussion

This chapter focuses on identification of opioid overdose precursors, which are crucial indicators given the readily reversible nature of overdose events with early detection. In the setting of real-

world high risk opioid use, our results highlight the need for a multi-tier interactive alarm system on the phone that escalates or deescalates based on user feedback. Put differently, we do not envision the system alerting a third party or disturbing the user based on an isolated central apnea or respiratory depression event; rather, an alert should be sent only after a subject is unable to respond to a stimulus from the phone following a sustained central apnea or respiratory depression event, representing a potentially life threatening overdose. Evaluating such a multi-tiered system would be the next step in enabling an end-to-end overdose detection system using commodity smartphones.

In summary, we chapter development of a proof-of-concept system that can be implemented on commodity smartphones and can identify simulated and real-world opioid overdose events and their precursors. As a harm reduction intervention, such a system could connect people experiencing potentially fatal overdose events with known life-saving interventions (e.g., naloxone-equipped friends, family, shelter personnel, or EMS) in real time. In order for this harm reduction intervention to be efficacious, further work is needed to ensure the system meets the needs, values and preferences of people who use opioids, in addition to establishing the system's safety vis-à-vis its potential to encourage moral hazard. Importantly, other harm reduction interventions such as take-home naloxone programs have been found not to increase risky behavior or lead to adverse health consequences [65, 157, 44]. Further, prior data on harm reduction interventions show that people are willing to engage in behaviors to help keep themselves safe, e.g., by utilizing needle exchanges, take-home naloxone, face shields for mouth-to-mouth respiratory support and supervised injection facilities [72, 107, 168].

Another potential concern is whether the system can reliably alert pre-hospital EMS providers in a time-frame that enables successful resuscitation with naloxone or supportive respiratory care. Based on historical data from Seattle, King County involving fatal overdose events and average EMS response times, we believe meaningful EMS integration is possible. Such a program would need to incorporate the detection algorithm's performance characteristics to leverage the operations and resources of a given EMS system [105]. Any integrated program must also acknowledge that even with rapid connection to EMS, victims still could experience morbidity and mortality

following opioid overdose [166].

As the number of deaths attributable to opioid overdose continues to rise, new strategies are needed to help mitigate the risk of death and disability from this public health epidemic. One key tool to reverse these events–naloxone administration–is increasingly available to first responders (including police), to people through take-home naloxone programs, and is now endorsed by the US government [33]. In addition to its use by EMS providers, the administration of naloxone by trained friends and family has also been shown to be a safe and effective means of reversing overdose [123, 78]. However, neither EMS, friends or family can intervene with naloxone or supportive respiratory care in an emergency if they are not immediately aware that an overdose is taking place. Non-invasive self-monitoring via smartphone, as we have described, could address this critical shortcoming and may represent an easily accessible strategy to help keep people safe until they are able to access long-term treatment.

Chapter 4

FINGERIO: USING ACTIVE SONAR FOR FINE-GRAINED FINGER TRACKING

In this chapter, we explore the following question: Can we track the user's finger around the device and can we do this even when they are occluded from each other? A positive answer would allow the user to interact in more expressive ways, utilize the screen fully without hand blockage and also enable new interaction scenarios. For instance, the user can use her finger as a pen to provide input over a much larger surface area than the smartphone. She can also perform subtle finger motion around the device that can be tracked even when the phone is in a pocket. Such a capability would also benefit smaller devices such as smart watches that could track the user's finger, even when fully occluded from it or when the watch is on a different plane from the interaction surface. Existing solutions for finger tracking, however, either instrument the finger with sensors [111, 204, 53, 56] or use cameras/infrared sensors at the device [51, 112, 114]. The former approach is burdensome while the latter does not work with occlusions.

We present *FingerIO*, a fine-grained finger tracking system that does not require instrumenting the finger with sensors and works even with occlusions between the finger and the device. FingerIO tracks finger motion in the region around existing smartphones, and achieves an average 2-D tracking accuracy of 8 mm. It also tracks subtle finger motion around the device, even when the phone is in the pocket. Using FingerIO, we also prototype a smart watch-form factor device that can track the finger, while extending the interaction space to a $0.5\times0.25\ m^2$ region on either side of the device. Further, the watch can continue tracking the finger even when they are fully occluded from each other.

Our key insight is to transform mobile devices (e.g., smartphones) into active sonar systems. At a high level, we transmit inaudible 18-20 kHz sound waves from the device's speaker. These

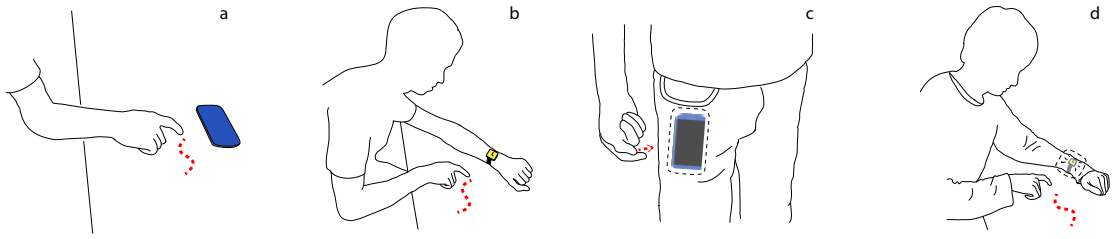


Figure 4.1: **Applications of FingerIO.** a) Transform any surface into a writing interface; b) provide a new interface for smartwatch form factor devices; c) enable gesture interaction with a phone in a pocket; d) work even when the watch is occluded.

signals get reflected from the finger and can be recorded as an echo at the device's microphones. Finger motion results in changes to the arrival time of the echo at multiple microphones. By tracking the time at which these echoes begin, FingerIO continuously tracks the finger location. The echo from the finger however is noisy and hence estimating the exact time when the echo is received, in the presence of all other reflections, is challenging. To appreciate the challenge, microphones on today's mobile devices have a sampling rates of 48 kHz [130]. Given the speed of sound in air, an error of just 3-4 samples in estimating the start of the echo results in a 2.1-2.8 cm error in the estimated distance from each microphone. Since the finger location is determined by computing the distance from multiple microphones, the finger tracking error would be much higher. In addition, speakers and microphones on mobile devices typically run independently and do not sample at the exact same time. This results in an additional sampling offset, further increasing the tracking error.

FingerIO addresses the above technical challenges by borrowing insights from wireless communication. In wireless systems, the transmitter and the receiver are not synchronized and do not sample at the same time. Wireless receivers however are designed to estimate the sampling offset for every transmission so as to decode the transmitted information. Modern wireless system uses a modulation technique called Orthogonal Frequency Divison Multiplexing (OFDM) to achieve this goal. Inspired by this, FingerIO achieves accurate finger tracking using OFDM. At a high level,

we create 18-20 kHz OFDM symbols by computing the FFT of N random bits and generating N samples. We then compute a cyclic suffix of S samples, as shown in the Fig. 4.2, which we transmit from the speaker. The key property of OFDM with cyclic suffixes is that, a sample error in identifying the beginning of the symbol, translates linearly into phase changes in the frequency domain; these changes can be extracted at the microphones using an FFT. For instance, an error of E samples translates into the phase linearly increasing from 0 to 2Eπ at the output of the FFT. Further, fractional sampling errors that occur because of sampling drifts between the microphone and the speaker also result in a similar phase change at the output of the FFT. Using this, FingerIO corrects for the sampling errors and achieves fine-grained finger tracking.

While active sonar has been proposed before to perform coarse-level gesture recognition [82, 55], we are not aware of prior attempts to use it to achieve fine-grained finger tracking on existing devices. Hence, our contributions are:

1. We introduce a novel approach to fine-grained finger tracking for around device interaction that does not require instrumenting the finger with sensors and works even in the presence of occlusions between the finger and the device.

2. We propose and develop an active sonar solution to finger tracking. To achieve this goal with high accuracies, we introduce algorithms that use the properties of OFDM to track the changes in echoes caused due to finger motion.

3. We implement our design on a Samsung Galaxy S4 using its in-built speaker and microphones and demonstrate finger tracking around the phone, with no additional hardware. We also built a prototype of our design in a smart watch form factor device using off-the-shelf hardware.

4. We conduct experimental evaluations that show that FingerIO can achieve average 2-D finger tracking accuracies of 8 mm and 1.2 cm at 169 frames/s for the smartphone and smart watch prototypes. Further, it accurately tracks subtle finger motion even when the phone in a pocket as well as with the smart watch fully occluded from the finger.

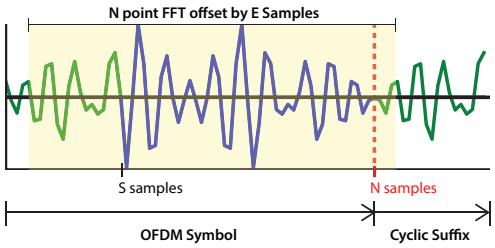


Figure 4.2: **OFDM signal structure.** The first S samples of the OFDM samples are appended to the end to create the cyclic suffix. Taking an N point FFT of the signal beginning at offset E will include part of the cyclic suffix resulting in a phase difference.

4.1 Related Work

Prior work falls in three key domains.

Near Device Interaction. Prior work in this domain can be broadly categorized as either requiring instrumenting the human body or vision based systems. iRing [135] uses an infrared sensor to recognize rotation, bending and external force on the finger. LightRing [111] designs a ring-form factor device that consists of a infrared proximity sensor and a 1-axis gyroscope to measure the finger flexion and rotation respectively. Magic finger [204] designs a finger-worn device that uses an optical mouse sensor and a micro RGB camera to sense touch as well as the texture of the surface being touched. Fingerpad [53] is a nail-mounted device that uses the tip of the index finger as a touchpad using magnetic tracking. uTrack [56] proposes to instrument the back of the fingers with magnetometers and the back of the thumb with a magnet to enable a 3D input system using magnetic sensing.

Systems such as Digits [112] and SideSight [51] do not require instrumenting the finger with sensors but use vision/infrared sensors and hence do not work with occlusions. Specifically, Digits [112] uses a wrist-worn 3D infrared camera to recover the full pose of the user hand. SideSight [51] instruments the sides of a mobile device (e.g., smartphone) with an array of infrared proximity sensors which detect the presence and position of the fingers in line-of-sight of the sensors. Hoverflow [114] uses an array of IR sensors placed along the edges of the phone to

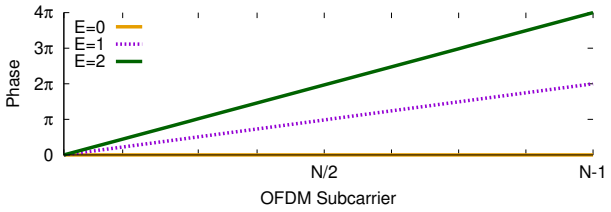


Figure 4.3: **OFDM phase change due to sample error.** The cyclic suffix introduces a phase change of $2E\pi$ across the OFDM subcarriers when the beginning of the OFDM symbol is estimated incorrectly by E samples. This phase can be used to correct the sample error and improve the accuracy of the system.

detect hand gestures. [165] uses the built-in camera of a phone to detect in-air gestures and tracking of hand parts in the camera's view. [205] attaches an omni-directional mirror to the phone's camera to increase its field of view. In contrast to these systems, FingerIO leverage the microphones and speakers that are common on most mobile devices to design an active sonar system that can track finger motion in the vicinity of the device. Further, FingerIO uses acoustic signals and hence can operate even when the finger is occluded.

Finally, PocketTouch [155] detects finger-strokes through fabric using a custom capacitive sensing hardware. In contrast, FingerIO can track subtle finger motion through pockets with existing smartphones, without the need to touch the device.

Active Acoustic Localization. Device localization systems such as Cricket [140], Doplink [42], Spartacus [174], and Shake and Walk [90] localize and determine the direction of a device movement using acoustic transmissions. AAmouse [206] uses Doppler shifts to track the phone position using anchor devices in the room. Whiteboard technologies such as Mimio [17] use an active stylus with ultrasound and infrared and localize using an anchor device placed at the corner of the board. In contrast, FingerIO is a device-free localization solution that tracks an uninstrumented finger using existing devices; this is achieved using the properties of OFDM. While OFDM has been used in wireless communication and device localization systems [109, 110] due to its resilience to

multipath, we are unaware of prior work that uses it for finger tracking.

SoundWave [82] leverage Doppler shifts from acoustic transmissions to recognize gestures such as moving the hand towards or away from the device. Airlink [55] and Surfacelink [79] use Doppler shifts and surface-mounted piezoelectric sensors respectively to detect hand waving gestures from one device towards the other. These design focus on pre-defined set of hand and arm gestures and are not designed for finger tracking. Finally, ApneaApp [130] tracks the periodic breathing movements in a sleep environment using FMCW reflections of the inaudible transmissions from the phone.

Chirp microsystems [10] designs on-chip ultrasonic rangefinders operating at 217 kHz with a bandwidth of 12 kHz using an array of seven transducers to perform angle-of-arrival techniques and get an angular resolution of 15 degrees [142]. The key motivation was to reduce the power consumption of cameras and instead use an ultrasonic design. Our approach differs from this in three key ways. First, we use existing devices with one to two microphones and do not need any custom chips. Second, we leverage OFDM and show that using just 2 kHz of bandwidth we can achieve centimeter level localization. Third, while the performance of these on-chip designs has not been evaluated for finger tracking, we apply our design in various finger tracking applications and show that it can enable a number of interesting interaction scenarios.

Passive Acoustic Localization. [185, 118] use the audible sounds made when clicking on a keyboard to snoop on the keys typed by the users. [150] localizes taps on solid aluminum and glass surfaces by using a piezoelectric shock sensor to sense the sound propagation through the material. Toffee [199] uses vibro-acoustic piezo sensors to find the direction of the audible sound and vibration waves that propagate as the user taps on a table. It achieves a mean angular resolution of 4.3 and 18.3 degrees using four sensors at the corners of a laptop and smartphone respectively. More recently, [50] uses contact microphones attached to the surface to distinguish between various impact events such as touch, knock and swipe. In contrast to this work, FingerIO uses an active sonar approach that transmits inaudible signals and achieves centimeter level finger tracking both on surfaces as well as in the air.

RF-based Gesture Systems. WiSee [143], AllSee [106] and SideSwipe [209] uses Wi-Fi, TV and cellular transmissions respectively to recognize coarse hand, arm and leg gestures. WiTrack [34] uses custom radar transmissions to detect pointing gestures. WiDraw [173] tracks the arm motion in the vicinity of a Wi-Fi device using transmissions from 20-30 other Wi-Fi devices in the network. Google has reported that project Soli is exploring the use of 60 GHz radar to recognize subtle finger gestures [80]. None of these approaches have been demonstrated on smartphones and require custom sensor hardware. We also believe that the active sonar approach introduced in this chapter is more attractive for two reasons: RF signals propagate at the speed of light and so to get a centimeter resolution requires processing GigaHertz of bandwidth. In contrast, the speed of sound is significant lower and hence with 48 kHz, FingerIO could achieve centimeter level accuracies. Further, our approach uses microphones and speakers that are already available on existing mobile devices and hence the bar for adoption is much lower.

4.2 FingerIO

FingerIO achieves centimeter level finger tracking by transforming the mobile device into an active sonar system. At a high level, we transmit an inaudible sound signal in the frequency range of 18-20 KHz from the device's speaker. These signal are reflected by all the objects in the environment and can be recorded by the microphone as echoes. When the user moves her finger, the time of arrival for the corresponding echo changes. By comparing the echo profile from one instance to the other, we can extract the echoes that correspond to the moving finger. As described earlier, since these echoes are noisy, the challenge is in accurately identifying the beginning of the echo so that we can achieve finger tracking with high accuracies. FingerIO leverages a modulation technique called OFDM to achieve this goal.

In the rest of this section, we first explain the properties of OFDM. Next, we describe how we generate OFDM transmissions using speakers in the 18-20 kHz range. We then show how FingerIO uses OFDM to measure the distance of a moving finger from a single microphone. Finally, we discuss how to use two microphones to achieve 2D tracking.

4.2.1 Understanding OFDM

Orthogonal frequency division multiplexing (OFDM) is a common modulation technique used in modern wireless communication systems including Wi-Fi and LTE. In this section, we focus on the properties of OFDM that are relevant to our design. See [47, 86, 141] for more extensive discussion about OFDM. OFDM splits up the bandwidth into orthogonal subcarriers and transmits data independently on each of the subcarriers. For example, in Wi-Fi, the 20 MHz bandwidth is divided into 64 subcarriers each with a width of 312.5 kHz. The data is then transmitted on each of these 64-subcarriers. To achieve this, OFDM uses Fourier transforms. Say we divide the bandwidth into N subcarriers and transmit the data bit $\mathbf{X}_n$ on the nth subcarrier. OFDM generates the time-domain samples that are sent at the transmitter by performing an inverse Fast Fourier transform (IFFT) over these data bits, i.e.,

$$\mathbf{x}_k = \sum_{n=0}^{N-1} \mathbf{X}_n e^{i2\pi kn/N} \qquad k = 0 \; to \; N-1$$

This creates N time-domain samples, $\mathbf{x}_k$, that are then sent by the transmitter. An ideal receiver would receive these time-domain samples and performs a Fast Fourier transform (FFT) to recover the data bits, i.e.,

$$\mathbf{X}_n = \sum_{k=0}^{N-1} \mathbf{x}_k e^{-i2\pi kn/N} \qquad n = 0 \; to \; N-1$$

In practice since the receiver is not perfectly synchronized, it does not know the exact beginning of this symbol. To help address this problem, the transmitter sends a cyclic suffix which is a repetition of the first S time-domain samples as shown in Fig. 4.2. To see why this helps, say the receiver has an error of E samples in estimating the beginning of the OFDM symbol. Given this error, it would perform an FFT over the N time-domain samples that are offset by E, as shown in the figure. Since we use a cyclic suffix, these new time-domain samples can be written as $\mathbf{x}_{(k+E)modN}$. Now when the receiver performs an FFT over these samples, we get,

$$\begin{aligned} \mathbf{X^E}_n &= \sum_{k=0}^{N-1} \mathbf{x}_{(k+E)modN} e^{-i2\pi kn/N} \\ &= \mathbf{X}_n e^{i2\pi En/N} \end{aligned}$$

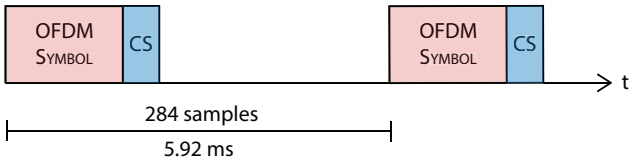


Figure 4.4: **FingerIO transmissions at the speaker.** The 84 samples for the OFDM symbol and the cyclic suffix are followed by 200 samples of silence. This silence duration is sufficient to receive echoes from all objects within 1 m from the device. Given a 48 kHz sample rate the above transmissions, the above transmissions achieves a frame rate of 169 Hz.

We see that the new frequency-domain data is the same as the original data but with an additional phase that depends on the error in estimating the beginning of the symbol (E). It also linearly increases with the subcarrier number n as shown in the Fig. 4.3. For example, an error of one sample results in the phase linearly increasing from 0 to 2π across the subcarriers. More generally, an error of E samples, results in the phase increasing from 0 to $2E\pi$ across the N OFDM subcarriers.

To summarize, if the receiver knows the data bits, $\mathbf{X}_n$, that are been transmitted, it can compute the error E in estimating the beginning of the OFDM symbol. Further, the above analysis holds even when there is a fractional time offset between the transmitter and the receiver, allowing us to estimate it. We leverage this OFDM property in our design to achieve centimeter-level finger tracking accuracies.

4.2.2 FingerIO transmissions at the speaker

FingerIO generates OFDM signals in the inaudible frequency range of 18-20 KHz which is then played by the device's speaker. There are however two key subtleties with creating an acoustic OFDM system: i) acoustic devices do not use oscillators to generate and transmit a carrier frequency. This is because the audio sampling rate of 48 kHz is sufficient to cover the entire frequency range of typical speaker and microphones. ii) The input to the speaker is a 16-bit real number and cannot transmit the complex numbers generated by the IFFT. So FingerIO generates a carrier-less real value OFDM symbol. To do this, given a sampling rate of 48 kHz, we first split

the operational frequency of 0–24 kHz into 64 subcarriers each spanning a width of 375 Hz. Since we want to operate only in the inaudible frequencies of 18–20 kHz, we set the subcarriers outside this range to zero. For the rest, we set each subcarrier to either +1 or -1. Then, we compute an IFFT that gives us a complex time-domain signal with 64 samples, $\mathbf{x}_k$. We convert these complex numbers into real values by computing the real value of these complex numbers. Specifically, we use the following transformation,

$$\mathbf{real}_k = |\mathbf{x}_k| cos(\angle \mathbf{x}_k)$$

Here $|.|$ and $\angle$ denote the amplitude and phase of the complex number. These 64 real values form the real-valued OFDM symbol that is transmitted by the speaker. We append the first 20 of these values to create a cyclic suffix that together is played repeatedly from the speaker, as shown in Fig. 4.4.

At a sampling rate of 48 kHz, these 84 samples (including the cyclic suffix) form a pulse that occupies 1.75 ms. We separate these pulses by 200 samples which in turn translates to a separation of 4.17 ms. We pick this duration to ensure that all the echoes from a distance of 1 m can arrive before the beginning of the next pulse. Given these parameters, we transmit an OFDM pulse once every 5.92 ms and achieve a frame rate of 169 Hz.

4.2.3 Measuring the distance from the microphone

The OFDM signal played by the speaker gets reflected off different objects including the hand and is then recorded by the microphone. To find the distance from the finger in the presence of all these reflections, we perform three key steps: (1) generate the echo profile of all the reflections at the microphone, (2) identify the echo corresponding to the moving finger, (3) process the OFDM symbol that is echoed by the finger to fine-tune its distance from the microphone.

Step 1. Generating the echo profile. While processing the recording at the microphone, we first identify the individual echoes that occur due to all objects within a distance. To do that, we perform correlation of the received signal with the original OFDM symbol. The output of this correlation process is an echo profile as shown in Fig. 5.3. Each of the peaks in this profile corresponds to the

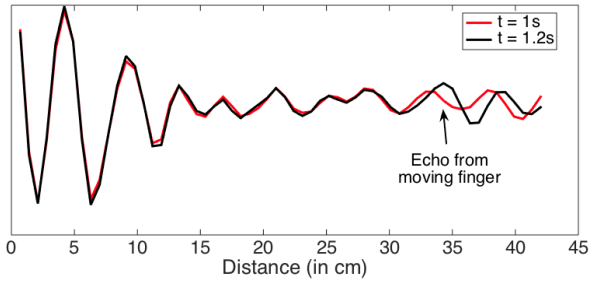


Figure 4.5: **Echo profile at two time instances.** Each peak indicates the arrival of an echo and the X-axis shows the corresponding distance computed based on the speed of sound. When the finger moves from 34 cm to 35 cm with respect to the device's microphone we can see a shift in the peak due to the change in the arrival time of the echo.

beginning of an echo from an object around the mobile device. This can be translated to a distance at which the reflecting object is present from the microphone and speaker. While OFDM has good autocorrelation properties, given noise, this step gives us the beginning of each echo with an error of 2–3 samples. This translates to an error in distance estimate of 1.4–2.1 cm.

Step 2. Identifying the echo corresponding to the finger. When the finger moves, the time of arrival for the echo corresponding to the finger changes. Fig. 5.3 shows the echo from the finger as it moves from a distance of 34 cm to 35 cm from the microphone. We can see from the figure that the position of the echo changes as the finger moves around the device. We identify this change by performing a subtraction between the echo profiles of consecutive OFDM pulses. To recognize if a change has occurred at a specific distance value, we use a threshold based approach. Specifically, if the changes across all the distance values in the echo profile are smaller than a threshold, we conclude that there has been no finger motion in the vicinity of the device. Changes that are greater than the threshold value are designated as those corresponding to motion. We set this threshold value to 15% of the amplitude of the speaker signal as heard directly by the speaker. We pick this relative threshold value to ensure that amplitude fluctuations that occur because of non-linearities in microphones and speakers do not lead to false positives.

The width of a human adult finger is around a centimeter. This is close to the distance resolution (0.7 cm) provided by an active sonar system where the sampling rate if 48 kHz. Hence, when the finger moves, one would expect a change to occur at only one-two samples corresponding to the old and new locations of the finger. However, close examination of Fig. 5.3 reveals that the changes occur not only at the finger's location but also near its vicinity. This is because a finger motion also causes a hand movement that traces a similar path as the finger. For our design, however, this means that we need to correctly identifying the changes caused by the finger motion in the presence of changes corresponding to other small hand movements. We achieve this by picking the closest sample, where the change crosses the threshold value. The reason why this works is that the finger is always closer to the microphone compared to the rest of the hand. Further, as the user draws using their finger, the maximum displacement (and hence changes) occurs at the tip of the finger.

Step 3. Fine-tuning the distance from the microphone. Finally, once we identified the echo corresponding to the finger, we use the properties of OFDM to accurately estimate the beginning of this echo. Specifically as described earlier, when we perform an FFT over the echo of an OFDM pulse, any error in estimating the beginning of the echo translates to a linear phase shift at the output of the FFT. Thus, at a high level, we first compute a 64-point FFT starting at the approximate location of the echo as estimated by the correlation in step 1. We then use the linear phase shift at the output of the FFT to accurately estimate the beginning of the echo, which in turn gives us the distance from the microphone. We note that since the microphones receive a real OFDM signal, we need to first transform it into a complex signal before performing the FFT. To do this, we use a technique called negative sideband suppression [128] where we obtain a complex representation by setting the negative frequency components of the signal to zero. This overall process allows us to fine-tune the distance estimate of the finger from the microphone.

4.2.4 2D finger tracking using two microphones

To perform 2D tracking, we compute the distances of the finger with respect to two microphones and combine them to measure the 2D location. Note that the distance we are measuring is actually the sum of the distance traveled by the signal from the speaker to the finger and the distance traveled

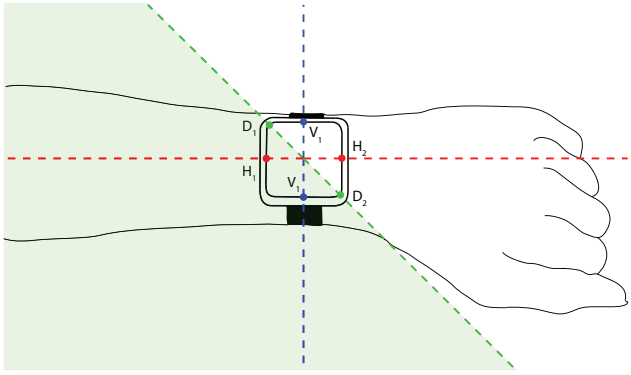


Figure 4.6: **Design choices for the smart watch.** Placing microphones on opposite sides of a smart watch either limit the user from drawing on the arm or on part of the surface below. Placing microphones along the diagonal could be a reasonable compromise as would allow users to interact both on the arm and a surface.

by the echo from the finger to the microphone. This means that given the distance measurements computing on a single microphone, the finger can lie on any point in a 2D space where the sum of its distance from the microphone and the speaker is equal to the measured distance. Said differently, the finger can lie on an ellipse where the speaker and the microphone locations are the two focii of the ellipse and the distance measured is twice the length of the major axis of the ellipse. Given the measurements from two microphones, FingerIO can find the intersection of the corresponding two ellipses to narrow down the possibilities for the finger location.

While in general, two ellipses can intersect in up to four points, in our case there can only be two intersection points. This is because, two ellipses that share a focal point (the speaker) can intersect in a maximum of two points [120]. These two points lie on either side of the line joining the two microphones. This means that the system is symmetrical along the line joining the two microphones and the 2-D location can lie on either side of the phone. The implication for our design is that we cannot distinguish between when the user is moving her finger on either side of the line connecting the two microphones. For the smartphone use case, this means that the user could draw on only one side of the phone at any instance. It is acceptable since the user is likely

to interact in the region between the phone and themself. This however raises interesting design choices for the smartwatch. If we place the two microphone in positions H_1 and H_2 as shown in Fig. 4.6 then the user can easily interact on either side of the arm but cannot draw along her arm. Placing them in the vertical positions V_1 and V_1 would allow us to interact on the arm but on a truncated region between the hand and the user. A reasonable compromise could be to place them along the diagonal positions D_1 and D_2. Another solution is to use three microphones and eliminate this symmetry. The implementation in this chapter uses the horizontal locations, H_1 and H_2 for the microphone positions.

Using two microphones, our design can track the motion on any 2D plane. Motion along a different plane will be projected to the plane along which the microphones lie. In the case of a smart watch, the user places the wrist on the interaction surface (e.g., table). Note that the width of a user's arm is around a couple of centimeters and the plane along which the smart watch lies can be approximated to be parallel to that the interaction surface. Thus, the projections of the finger-motion along the interaction surface are similar to that of the actual motion. Note that using more than two microphones, e.g., two microphones near the display and an additional one along the strap, will avoid the need for the above approximations.

4.3 Implementation

We implement FingerIO prototypes on two platforms.

Off-the-shelf smartphones. We developed a third party Android app and tested it on a Samsung Galaxy S4 smartphone. The phone has one speaker at the bottom and two microphones one at the top right and the other at the bottom right spaced apart by 13.5 cm. The app generates the OFDM symbols and uses the AudioTrack class that is an existing Android API to play it on the phone's speaker. The app is set to record simultaneously from both microphones using the stereo mode using Android's built-in AudioRecord class.

Smart watch form-factor device. While Apple smart watch has effective speakers and microphones that allow them to make calls and use Siri [8], they are currently not as programmable as smart phones and further have only a single microphone. While 1D tracking can be achieved with

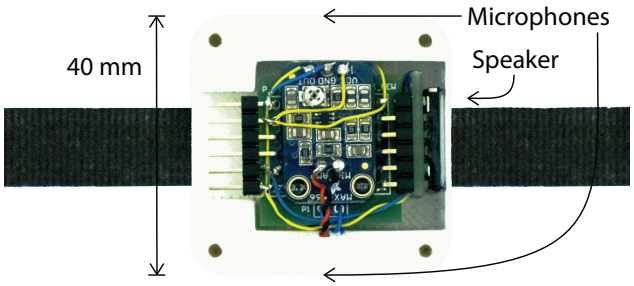


Figure 4.7: **Our Smart watch form-factor prototype.** The prototype consists of two microphones (embedded in the case) and a speaker mounted in a 3D printed case (shown transparent in figure) with a Velcro strap.

a single microphone, since we also want to demonstrate 2D tracking, we built a simple prototype consisting of two microphones and a speaker mounted in a 3D printed case with a Velcro strap, as shown in Fig. 4.7. The electret microphones (CMA-4544PF-W, CUI Inc) are mounted on opposite sides of the watch case spaced apart by 40 mm. The output from the microphones are connected to a Adafruit development board [7] containing a preamplifier (MAX4466, Maxim) which is then connected to the NI MyDAQ for data acquisition. The same OFDM signal used for the smartphone experiments was supplied to a class D amplifier (MAX98306, Maxim) that takes input from the phone's 3.5 mm headphone jack and drives the speaker. The system was powered using the Keysight E3631A DC power supply.

4.4 Evaluation

We recruited ten participants (5 female and 5 male) between the ages of 20–25; none of them were provided any monetary benefits. The participants were asked to draw any pattern they wanted using their finger and we evaluate FingerIO's accuracy in various scenarios for both the smartphone and smart watch implementations. Since the participants could draw any pattern, we use a second touch-based mobile device that collects the ground truth data. Specifically, we place an Android smartphone at different locations around our FingerIO-enabled smartphone/smart watch and ask the participants to draw freely using their finger on this device. We use the Android OnTouch Event

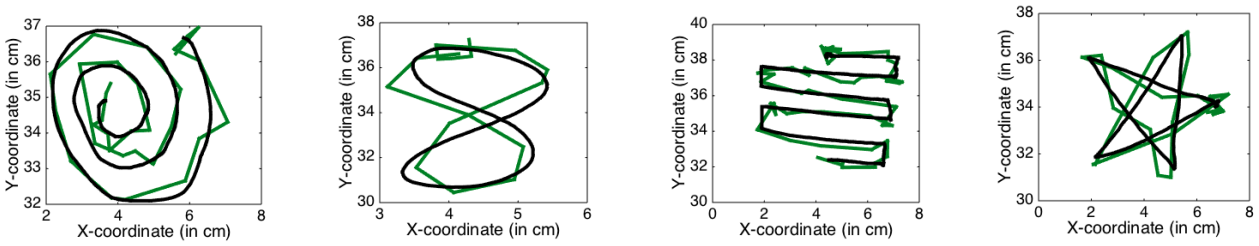


Figure 4.8: **Traces computed using FingerIO for smartphone setup.** The figures show both the ground truth trace (black lines) as well as FingerIO's estimated trace (green lines) for four of our participants.

Listener to obtain the pixel locations that the participants touch. We also extended the draw API to simultaneously display the path traversed by the user's finger. The OnTouch API only provides the locations as pixels in the screen space. We convert this into a screen location (in cm) by scaling the pixel value with the number of pixels per centimeter. We then offset this screen location with the distance between the FingerIO-enabled smart watch/smartphone and the smartphone used for ground truth data collection.

In this rest of this section, we first present 2D finger tracking accuracy for both the smartphone and the smart watch prototype implementations in line-of-sight and occluded scenarios. We then evaluate FingerIO's interaction surface (i.e., 2D-range) for both the prototypes. Finally, we address unintentional motion tracking with FingerIO.

4.4.1 FingerIO's Finger Tracking Accuracy

We run experiments in one of the offices in our organization. We place a smartphone running FingerIO on a table, lengthwise in front of the participants. A separate smartphone that is used to collect the ground truth was placed 25 cm from the FingerIO-enabled smartphone. The participants were given a demonstration of the system and were allowed to practice with it. Once the participants became familiar with the setup, they were instructed to draw a pattern of their choice that consisted a single continuous finger movement. The participants were asked to repeat their

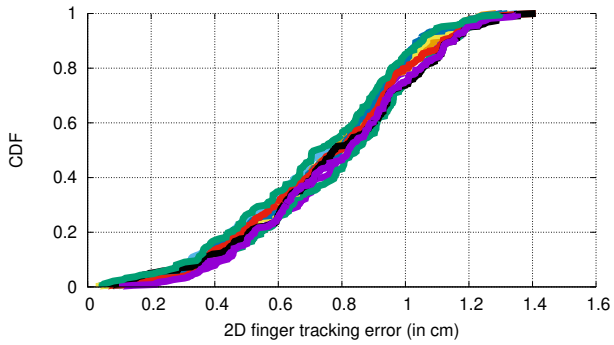


Figure 4.9: **Finger tracking accuracies with smartphone.** Cumulative distribution functions (CDFs) for the 2D tracking errors for each of the ten participants. The median tracking error across all the participants is 8 mm.

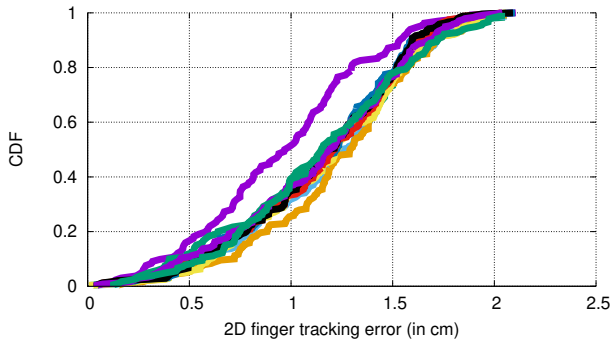


Figure 4.10: **Finger tracking accuracies with smart watch.** CDFs for the 2D tracking errors for each of the ten participants. The mean tracking error was 1.2 cm.

patterns thrice. All the participants placed their palm on the table while using their finger to draw. We process all this data and compute the average tracking error from the ground truth data. To compute the finger tracking error, we measure the average least perpendicular distance of each point along the trace computed by FingerIO with the ground truth.

Fig. 4.9 shows the CDF of the tracking error for all ten participants. The figure shows that the tracking error is similar across all participants. Further, the median error across all participants was 0.8 cm. This demonstrates that FingerIO achieves its goal of centimeter-level finger tracking

in practice. Fig. 4.8 shows four of the most complex patterns picked by the participants. The black line in these traces is the ground truth trace of the participants and the green traces are the ones computed by FingerIO, subsampled by a factor of five. The figures show that the two traces are close to each other and that FingerIO's algorithm can also deal with intricate motion such as those shown in the traces.

We repeat the above set of experiments with our smart watch prototype. In particular, we ask the participants to wear the smart watch on their arm and place it on a table. The participants drew patterns on the table at a distance of around 25 cm from the smart watch. As before, we compute the finger tracking error by measured the least perpendicular distance between the traces for FingerIO and the ground truth.

Fig. 4.10 shows the CDF of the tracking error for the smart watch experiments. As with the smartphone, the tracking error is similar for most participants. The median error across all participants was around 1.2 cm. We observe that the accuracy with the smart watch setup is lower than with the smartphone scenario. This is because we see higher noise levels at the microphones we use in our smart watch prototype, even in the absence of any finger motion. This is likely because of insufficient isolation between the speaker and the microphone in our hardware setup. Better isolation, could in principle improve the accuracies further.

4.4.2 FingerIO's Interaction Surface

We evaluate how FingerIO extends the interaction surface around our prototype smart watch and smartphone.

FingerIO's interaction surface with the smart watch prototype. The participants were asked to wear the watch on their hand and place the hand on the table. We divide the area in front of the hand into 5×5 cm grids as shown in Fig. 4.11. Within each grid, the participants draw a straight line with a length of 4 cm twice. In each trial, we compute the trajectory of the finger with FingerIO. We compare this with the ground truth collected from our experiments. To compute the finger tracking error, we measure the average least perpendicular distance of each point along the trace computed by FingerIO with the ground truth. We compute the average error in each grid by

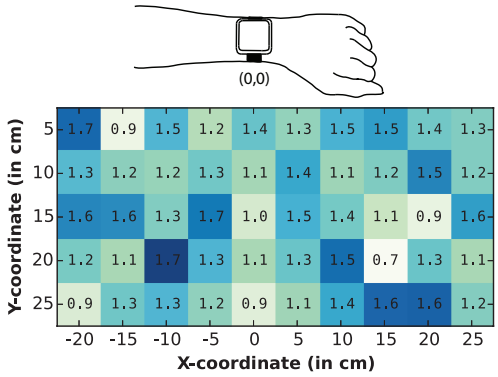


Figure 4.11: **Interaction surface for smart watch.** The surface around the hand was divided into 5×5 cm grids and the average finger tracking accuracy was computed for each grid. FingerIO enables an $0.5 \times 0.25\ m^2$ interaction surface with an average finger tracking accuracy of 1.2 cm.

averaging across trials in that grid. The figure shows that the interaction surface with the smart watch is a $0.5 \times 0.25\ m^2$ region on one side of the arm. Since the performance is symmetric across the line joining the two microphones, the actual interaction surface is double is region. The average error is about 1.2 cm and is uniform across the region. We note that beyond the grids shown in the figure, the accuracies quickly drop off with distance. As observed before, because of insufficient isolation between the speaker and the microphone in our hardware prototype, we see increased noise even in the absence of any motion. This contributes to slightly higher tracking errors than a smartphone scenario.

FingerIO's interaction space with the smartphone. In contrast to our smartwatch prototype, the output power of the smartphone speaker is set to the maximum value of 15 allowed by the Android API. This is around 10 dB greater than that in our smart watch implementation. Further, commercial devices such as smartphones have better isolation between microphones and speakers. So next we evaluate the interaction space for FingerIO using a off-the-shelf smartphone. To measure this, we place the smartphone on a table and divide the area around the phone into 10×10 cm grids. Since we would like to see what the maximum interaction surface area can be with good isolation and higher transmission power, we place the phone screen down. This ensures that the transmis-

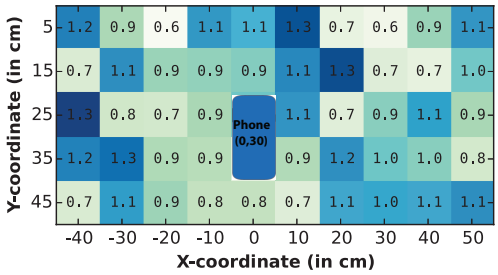


Figure 4.12: **Interaction surface with smartphone.** The surface around the device was divided into 10×10 cm grids and the average tracking accuracy was computed for each grid.

sions on the speaker that is on the back of the phone can reach much further distances and imitates a better version of the smart watch prototype. We perform the same experiments as in the previous scenario and measure the tracking error in each grid locations. Fig. 4.12 shows the smartphone and the finger tracking accuracy for the grids around it. The figure shows that, FingerIO expands the interactive surface to about a $0.5\ m^2$ region around the smartphone. The average error within this range is less than 1 cm and is fairly uniform; beyond this distance however this error increases to 3 cm. Further, these accuracies are similar on all four sides of the smartphone, demonstrating that FingerIO can perform well for different phone orientations from the user. The increased interaction area demonstrates that with better isolation and higher power, we can achieve a larger interaction space and better tracking accuracies. We believe however, a tracking range of less than a meter is sufficient for some interaction applications when the user is interacting with their smartphones or watches.

4.4.3 FingerIO in Occluded Scenarios

We evaluate two specific occlusion scenarios.

Smart watch occluded behind a jacket. We ask the participants to wear a polyester jacket that fully covers the smart watch. We use a similar setup as before to compute the tracking error. The participants were first asked to wear the smart watch and place their hand on the table as shown

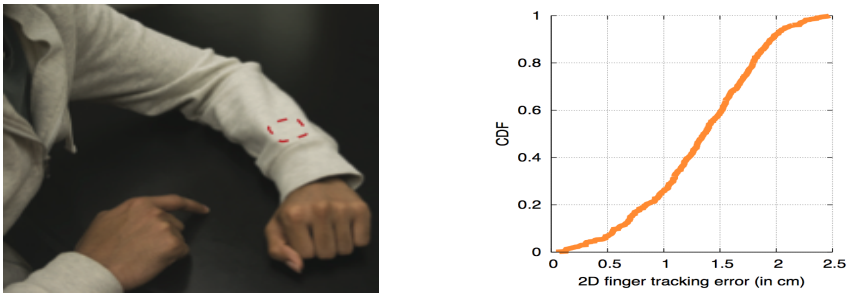


Figure 4.13: **Smart watch occluded behind a jacket.** The figure plots the CDF for the 2D tracking errors for five participants when the smart watch was occluded behind a jacket. The median tracking error across all the participants is 1.35 cm compared to 1.2 cm when the smart watch was not occluded.

in fig. 4.13(a). We then asked the participants to draw any pattern with their finger. We run these experiments with five of our ten participants where each of them was asked to repeat the pattern they drew 3 times. Fig. 4.13(b) shows the CDF of the tracking errors between FingerIO and the ground truth across all five participants. The median error across all the participants was 1.35 cm. This is slightly greater than the error in the absence of the occlusion. This demonstrates that FingerIO operates even in the presence of occlusions.

Smartphone in the pocket. Next, we run experiments with a smartphone placed inside the pocket of a pair of jeans, with the back of the smartphone facing outward as shown in fig. 4.14(a). Five of the participants were instructed to perform a finger swipe in the air in front of the pocket. The swipe motion consists of the thumb moving over the index finger. We configure the FingerIO algorithm to compute the distance moved by the finger by processing the data from a single microphone. The participants were allowed to perform the above swipe finger motion from any angle to the smartphone. We ask the participants to only move their fingers at a resolution of the prominent lines on their finger. This provides us with the ground truth data. On average, the participants moved their index finger by around 5 cm. We compute the error as the difference in the distance estimated by FingerIO and the ground truth motion. Fig. 4.14(b) shows the CDF of the

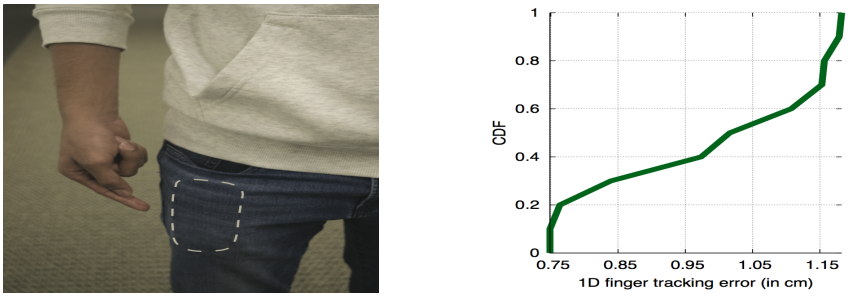


Figure 4.14: **Smartphone in a pocket.** The figure shows the CDF for the 1D tracking errors for five participants when the smart phone was inside the pocket. The median tracking error across all the participants is 1 cm compared to 8 mm when the smartphone was not occluded.

errors in this computed distance. The plots show an average error of 1 cm across all participants. In all cases, our algorithm correctly identifies the direction of the finger motion, i.e., either towards or away from the phone. This demonstrates the feasibility of through-the-pocket finger motion tracking.

4.4.4 Addressing Unintended Motion with FingerIO

In a system like FingerIO, we need a mechanism to inform the device the beginning and end times of when it should track the finger. This would prevent random motion in the vicinity of the device from being confused for finger motion for the purpose of interaction. To do this, we introduce a double swipe as a start and stop motion. A swipe is defined as a finger motion in a straight line for a length of at least 4 cm. A double swipe requires the user to perform a swipe motion in two opposite directions; we detect this by looking for distance values linearly increasing for 4 cm and then decreases for at least 4 cm. We consider a double swipe that is performed within a range of 5 cm from our device to be our start/stop motion. We pick the 5 cm range to ensure that similar finger motion that occurs at a farther distance is not confused for the start/stop motion.

Experiments: To evaluate how well this start/stop motion works, we ask our ten participants to perform the double swipe motion with both our smartphone and smart watch setups within 5 cm

from the devices. Each user performs the finger motion twice for both the devices. For each of these motions, if our algorithm fails to identify it as a start/stop motion, we consider it to be a false negative. To compute false positives, we ask our participants to draw random patterns other than the double swipe within 10 cm from the device for a period of 30 seconds. Each user performed it twice and then we look for the double swipe gesture over a total duration of 10 minutes across all the participants. The start/stop motion detected by our algorithm during this duration, are considered to be false positives.

False negatives: Our algorithm detected the double swipe start/stop motion 19 out of the 20 times the participants performed it with the smartphone. Similarly, we detected this motion 18 out of 20 times when it was performed with the smart watch. The undetected motions were because during a double swipe gesture, the participants move their whole hand along with the finger. While our algorithm tracks only the motion at the closest distance, i.e., the finger, for the three start/stop motions that were missed, the participants forcefully moved their entire arm in a different direction than the finger. Since our current implementation can only deal with a single motion direction, it was confused for this. This is however less likely to be the case if we use the active sonar approach to track multiple concurrent motions. We also note that the current false negative rate is still acceptable and could likely become more reliable as the users get accustomed to our system.

False positives: With the smart watch prototype, we did not detect any start/stop motion during the 10 min duration, i.e., the number of false positives during this duration was zero. This is because our algorithm requires a strict double swipe pattern within a small range from the device. With the smartphone, however we detected two start/stop motions during the 10 min duration. Further analysis of the data showed that the two false positives came from a single participant who drew a wiggly pattern tracing 4 cm distance in both directions. This triggered our algorithm to classify this as a start/stop motion. Given that the rest of the participants did not have any false positives, we believe that the double-swipe motion is sufficient in most scenarios.

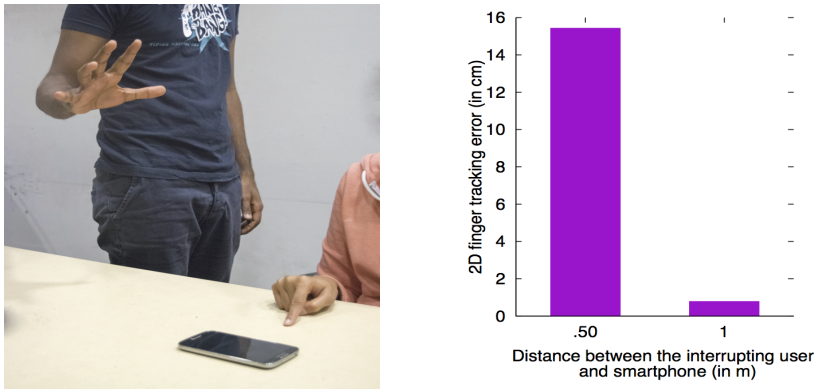


Figure 4.15: **Addressing Random Motion in the Surrounding.** The figure shoes the 2D tracking errors when there was a second interrupting user in the environment. The accuracy decreases when there is a stronger motion within 50 cm of the device. However, the accuracies remain consistently high when the interrupting user is beyond one meter.

4.4.5 Addressing Random Motion in the Surroundings

The maximum operational range of FingerIO is less than a meter. This is a key advantage, as it remains unaffected by motion in the surroundings. In this section, we evaluate this property by measuring the finger tracking accuracy of a participant while another participant creates random motion in the surroundings. We conduct experiments with one subject drawing a straight line of distance 4 cm. A phone running FingerIO is placed around 25 cm from the finger location. While the subject performs finger motion, another interfering subject continuously waves their hand toward the first subject. We repeat the experiments with two different distances for the second subject. We compute the finger tracking accuracy for each of these distance values.

Fig. 4.15 plots these accuracies as a function of distance of the interfering subject from the phone. The plot shows that when the interfering subject is within 50 cm from the phone, the finger tracking accuracies significantly suffer. This is not a fundamental limit of our sonar-based design. Rather this is because our current algorithm is designed only to track a single motion. In principle, one may design algorithms to track independent motions concurrently from multiple

distance values since the echoes arrive at different times for each of these distances. The key observation however is that for distances greater than a meter for the interfering subject, the finger tracking accuracies are again high. This is because, at these distances, the reflections caused due to the motion from the interfering subject are significantly attenuated and hence are weaker compared to the echoes from the finger motion that is performed at a closer distance.

4.5 Limitations and Future Directions

We discuss the limitations of our current design as well as opportunities to improve it.

Tracking 3-D motion and non-cursive writing. Our current implementation uses two microphones and cannot achieve 3-D tracking of finger motion. This is however not a fundamental limitation of our approach and can be addressed by using a third microphone. Specifically, three microphones can be used to triangulate the position of the finger in the 3-D space enabling 3-D finger tracking. A similar problem occurs with non-cursive writing, where the user could slightly lift her finger from the 2D surface to move it across different points on the surface. Using only two microphones, this would be tracked as a continuous motion and our algorithm will project this motion on the 2-D drawing plane as part of the input. We note that this is a similar issue faced by camera-based systems where the user draws with her finger in front of a camera. One direction worth exploring is to incorporate a third microphone on a different plane (which can be done on the smart watch setup) and use it to identify this 3-D motion.

Tracking multiple concurrent motions. While this chapter focuses on tracking a single finger, in principle, the algorithms presented could track concurrent changes from multiple fingers as long as they occur at different distances from the microphones. This can be used to detect pitch, zoom out and zoom in gestures that require multiple fingers moving at the same time. It can also be used to detect and separate the finger/body motion from other people near the device. We expect the algorithms for doing so to be similar to radar based approaches such as Google Soli. Exploring how well this works in practice is not in the scope of this chapter.

FingerIO's power consumption. A full-charged Samsung Galaxy S4 running FingerIO lasts around four hours. As with other always-on mobile gesture sensing techniques, there are a number

of power-accuracy tradeoffs that could be made. For instance, we transmit OFDM pulses once every 5.92 ms which translated to a frame rate of 169 frames/s. Since human motion is unlikely to change at this rate, we can operate FingerIO with a much lower frame rate. Further, as shown in Fig. 4.8, subsampling by a factor of five, still gives us values that look similar to the ground truth. Optimizing this further and reducing the power consumption, would be a worthwhile future direction.

Finger tracking with a moving device. The finger tracking algorithms developed in this chapter work under the assumption that the phone or the smart watch is static. To address mobility of the devices, we envision using the accelerometer/gyro, already present in the devices we imagine operating on, to compensate for the motion in our algorithms. Intuitively this would be similar to imitating a synthetic aperture radar system (SAR). We leave the development of such algorithms for future work.

4.6 Discussion

In this chapter, we introduce a novel active sonar design for fine-grained finger tracking that does not require instrumenting the finger with sensors and works even in the presence of occlusions between the finger and the device. While the existing body of work in this domain requires instrumenting the body or are vision based systems that does not work in occlusions, this system can be implemented in any off-the-shelf smart device that has a speaker and a microphone.

Today's smart IoT devices such as Fitbit, smartwatches and home assistants have small or no screen for interaction due to their size constraints. Hence they are equipped with speakers and microphones to enable voice based interaction with the device. This interaction mode requires the user to speak the commands that can be audible to others and the speech recognition software systems are susceptible to environment noise and varying accents. Instead our system leverages the existing speakers and microphones to provide an alternative mode of interaction for these devices.

Chapter 5

COVERTBAND: ACTIVITY INFORMATION LEAKAGE USING MUSIC

Smart devices and appliances are becoming increasingly prevalent, but as a consequence of adding these connected devices such as smart TVs, phones, and hubs like the Amazon Echo [68] to our homes, there are an increased number of connected speakers and microphones with access to our private environment. This provides a lot of value for consumers, but there are also privacy threats involved with increased connected sensing capabilities. In this chapter, we show that in the case of microphones and speakers there are privacy leaks possible with today's devices that go beyond the ability to simply record conversations in the home. For example, what if an attacker could remotely co-opt your television to track you as you move around, without you knowing? Further, what if that attacker could figure out *what* you were doing in addition to where you were? Could she even figure out if you were doing something *with* another person? A positive answer could leak information about user activities that are inaudible to a microphone and so far have been considered to be private.

While there has been significant research interest in the use of RF for localization and activity recognition (see §5.4), no existing RF mechanisms using Wi-Fi hardware on commodity devices routers, laptops, and smartphones permit device-free localization of unsuspecting victims in either through-barrier or remote-attack scenarios. We create CovertBand , which, for the first time, transforms commodity devices with microphones and speakers into active sonar systems to track users and differentiate between different classes of motion. At a high level, we transmit acoustic pulses in the 18-20 kHz range from the speaker and track reflections from the human body on the microphones. To accomplish our goals, we had to overcome two key challenges:

1. *How to perform passive localization using acoustic signals.* Due to the nature of indoor environments, there are significant multipath effects from static reflectors. To address this,

we borrow Orthogonal Frequency-Division Multiplexing (OFDM), a modulation technique commonly used in wireless communication systems such as Wi-Fi and LTE [119]. OFDM's strong autocorrelative properties allow CovertBand to function in the presence of multi-path, where a signal bounces off multiple objects in the environment before arriving at the receiver. This lets the receiver perform channel correlation to estimate the multi-path effects in the transmitted signal.

2. *How to perform acoustic localization through barriers.* A naïve solution to this challenge would simply increase the volume of pulses in the 18-20 kHz range until enough sufficient energy pentrates the barrier, reflects off a subject, passes through the wall, reflects off of a subject, and returns to the receiver. However, CovertBand uses speakers on existing devices which are not specifically built to transmit in the 18-20 kHz range at high volume. As a result, they create harmonics in audible frequency ranges. To mitigate this effect, we show how to mix the harmonics with cover music. We also show how to choose OFDM symbols that music can best conceal.

We implemented CovertBand on a Samsung Galaxy S4 with common audio devices, including 4 portable speakers [43, 45, 100, 48] and a home theater system [89]. To demonstrate the potential for privacy attacks on varied devices, we implemented CovertBand on a 42 inch SHARP TV [160]. We ran experiments in five homes in the Seattle area to demonstrate CovertBand's ability to help an attacker both localize victims and leak information about activities even in scenarios where those activities are not audible.

We summarize our experimental results below:

- CovertBand can track multiple subjects independently through barriers in a 2D plane. We ran experiments in five homes to track both a single subject and multiple subjects and found that we could localize with tracking error comparable to the state-of-the-art in RF localization [113, 62]. Specifically, CovertBand localized walking subjects with a mean tracking error of 18 cm and subjects moving in a fixed position with a mean tracking error of 8 cm.

For comparison, WiTrack2 [62], which uses custom FMCW radar hardware, has an accuracy of 10.9–19.2 cm when tracking moving subjects through walls.

- We evaluated CovertBand's range through a variety of materials using a portable speaker [100], showing that it can track at up to 6 m without barriers and 3 m in through-wall scenarios.

- CovertBand can differentiate between rhythmic and linear motions. We tested a variety of rhythmic motions pumping arms, jumping, and supine pelvic tilts in through-wall scenarios and show that they produced discernibly different spectrograms from walking.

- We compared performance in the 18-20 kHz range across multiple speakers to prove CovertBand could work on a wide variety of hardware. We also demonstrated CovertBand on an LG G4 connected to a Sharp TV [160] without fine tuning our algorithms to demonstrate the possibility of sensing with diverse sets of hardware without device-specific training.

- We evaluated our ability to conceal CovertBand with music by playing unmodified songs and songs with an additional sonar signal back-to-back in random order for 33 subjects in an isolated environment. We found that subjects could correctly differentiate only 58% of the pairs, which is close to random guessing, showing that even in ideal scenarios victims are unlikely to to detect the attack.

Contributions To the best of our knowledge, we are the first to demonstrate active sonar for through-barrier sensing on a wide range of commodity devices available to standard consumers. Specifically:

- We demonstrate the first device-free localization capability on commodity devices in both through-barrier and remote-attack scenarios. We show how to perform localization, tracking, and motion classification for multiple subjects in a 2D plane using changes in the audio channel.

- We ran experiments in five real homes to show that attacks are possible with our prototype. In particular, we show through multiple scenarios that an attacker can use active sonar to glean information about victims through walls, even when the attacker cannot see the victim nor hear any movements, and that such an attack is feasible using many common, off-the-shelf devices.

- We show how to conceal the attack from a victim by mixing active sonar pulses with music. We ran user studies to evaluate our methods, showing that such an attack could be done covertly to avoid detection even in ideal scenarios.

- We reflect on the broader implications of this work, including privacy implications and future research in this space. Specifically, we demonstrate the feasibility of potential threats through three case-studies, including spying on: 1) multiple people in a dormitory room when they are engaging in private activities, 2) a person's private activities in a bathroom (even when these activities are inaudible from outside), 3) remote victims by adding CovertBand-based malware to gadgets, like phones and TVs, that are commonly present in homes.

We note that our work intends to show the possibility of information leakage with commodity speakers and microphones. Maximizing the range and resolution for different materials and configurations, or building applications to utilize this capability is beyond the scope of this chapter.

5.1 Motivation and Goals

We begin by considering several motivating scenarios to derive our key goals.

5.1.1 Scenarios

These scenarios survey the utility of understanding the feasibility of covert, through-barrier sensing. Such attacks provide a new avenue for leaking information about obscured activities even in the presence of background or cover noise.

National intelligence Imagine a spy (Alice) entering a foreign country. She rents a hotel room adjacent to an individual (Bob), whom she intends to discretely and covertly monitor. To be a good spy, Alice cannot enter the country with dedicated surveillance hardware, and she cannot acquire any suspicious new hardware while in-country. But she still wants to monitor her neighbor to know when he is in the bedroom or bathroom, an ideal opportunity to enter the apartment and gather additional information). Bob does not know that he is being monitored, or even that he is the potential target of monitoring. Alice would benefit from using a covert monitoring mechanism, something she could run on her phone and that would avoid arousing Bob's suspicion.

Vigilante Justice In some cases, revealing certain private activities can be dangerous to victims. For example, many countries or non-government entities persecute pre-mairtal or other sexual partnerships [169, 97, 93]. We note that in many of these cases, vigilantes do not seek conclusive evidence before condemning victims; as such, the possibility of even circumstantial evidence could pose security threats for these individuals.

Remote Hacking of Phones and Smart TVs We also consider attacks that leverage devices already inside a victim's home. Because our attack requires access only to a speaker and microphone, an attacker can leverage many devices that already exist in the home environment. Smart TV apps and voice assistants, like Amazon Echo [68] and Google Home [88], already have access to speakers and microphones and let users install applications. A remote adversary who compromises one of these devices, perhaps via a Trojan application in an app store or via a remote exploit, could use our methods to remotely glean information about an individual's home activities. An attacker could also find more surreptitious ways to execute such an attack. For example, a streaming music app with voice control has all the permissions (speaker and microphone) needed to execute our attack. As a simple example, an attacker could utilize the advertising library embedded inside a music application to determine whether the user is near the phone when an ad is played.

5.1.2 Goals and Non-goals

Inspired by the preceding scenarios, we enumerate five goals for our system:

1. *Major motion detection and tracking.* The system should be able to detect motion and perform 2D tracking for each of several individuals in an environment.

2. *Distinguish between movements.* Our technique should be able to convey information about the type of movement occurring.

3. *Re-purpose commonly available devices.* Our technique should be implementable on devices that people might already own for several reasons. First is cost, ensuring that our approach is affordable enough to be commonly available. Second, using devices commonly found in household environments increases the potential attack surface. Further, using such devices provides "plausible deniability": a camera installed in an environment leaves physical evidence, and the purchase of dedicated radar equipment leaves little ambiguity as to someone's intended monitoring activities.

4. *Through-barrier sensing.* Each of the previous capabilities should be possible despite the presence of common barriers, e.g., walls, doors, windows, etc.

5. *Not detectable to unknowing target.* An unknowing target, unaware of our this type of attack, should have a low probability of detecting our attack.

We do not attempt to avoid detection by knowing targets. If targets know that they might be the subjects of monitoring, then they might be able to detect it by setting up sensors. Because this is true of any existing, active through-wall imaging techniques, including radar, we argue that this is a reasonable non-goal.

To our knowledge, we are the first to study, demonstrate, and evaluate an attack using the preceding goals. More specifically, we are unaware of any solution that performs through-barrier

detection and tracking on common devices. Note that RF-based systems that perform through-barrier device-free detection and tracking [34, 143, 58, 144, 193, 148, 194, 36, 203] require custom hardware, such as USRPs, or are limited to Wi-Fi chipsets with access to channel state information (CSI) in the vicinity of the device [173, 196, 190]. Commodity smartphones do not typically provide software-level access to CSI information. Thus, we know of no existing solution that tracks through barriers with commodity smartphone devices.

5.2 CovertBand Design

Section Outline This section is organized as follows: §5.2.1 describes the attack surfaces and attacker requirements. §5.2.2 describes our choice of signal, and §5.2.3 describes how we play this signal and our method for obscuring it. Sections §5.2.4 and §5.2.5 explain how to use the signal to calculate the distance to moving objects and how to use that distance to perform 2D localization.

5.2.1 Adversary Model

CovertBand enables two unique attack surfaces.

1. The first is a remote attack where an attacker compromises speakers and microphones already in a victim's home. This may be as innocuous as a music or video application with access to the microphone and speakers on a smartphone, Amazon Echo, or a Smart TV. Because CovertBand uses common devices, it can use an over-permissioned or malicious application, a common and well-known problem with mobile applications [152], to monitor an individual's location and activity. In principle, the hardware need only have multiple microphones and a single speaker, a configuration common in smarthphones and home assistants (e.g., Amazon Echo has a 7-microphone array). The attacker can likely reference hardware specifications to get information about speaker locations and microphones but must make some assumptions about device location. An Amazon Echo and television, for example, are unlikely to move. Thus, if attackers learn their location (for example, in a bedroom), they could use that information to breach privacy. State-of-the-art RF approaches generally do

not permit such remote attacks because through-wall approaches require specialized hardware (USRPs and FMCW radar arrays) not present in a victim's home [62, 148]. Other approaches use common Wi-Fi access points but require multiple access points and multiple devices in the environment at known locations and under attacker control. Further, they require a training phase that may demand victim cooperation [117, 198, 159, 154].

2. The second attack is a through-wall scenario where the attacker places a speaker and microphones near a barrier to sense obscured activities. Though this attack works best when victims are in the forward direction relative to the speaker, it: (1) does not require any particular speaker and microphone placement, (2) can be executed with a diverse combination of speakers and microphones, and (3) can be placed anywhere along (or up) the barrier as long as the victim is within range. Existing sonar approaches for through-barrier sensing require more specialized hardware and setups. For example, the DoD funded a through-wall sonar detector [58] with specialized hardware, which was meant for presence detection, not localization. Finally, while the principles in our chapter can generalize to two (stereo) or more speakers and a single microphone or a synthetic aperture by moving the speakers in a line, demonstrating these generalizations are beyond the scope of this chapter.

5.2.2 Strong Autocorrelative Signal

CovertBand leverages autocorrelation to identify the beginning of an echo from a human. For this reason, we selected OFDM signals for our sonar pulses. OFDM, a modulation technique commonly used in wireless communication systems including Wi-Fi [95] and LTE [119], has strong autocorrelative properties. These properties let it work in the presence of multi-path reflections, where a signal bounces multiple objects in the environment before arriving at the receiver. The receiver can thereby perform channel correlation to estimate the multi-path effects in the transmitted signal.

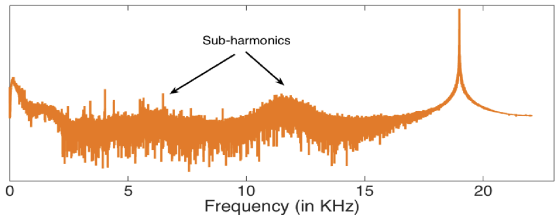


Figure 5.1: The figure plots the frequency spectrum of the signal recorded in a smartphone when the speaker plays a 19 kHz tone. While playing it creates sub-harmonics in lower frequencies. Recorded in a quiet lab environment by placing a speaker directly in front of a smartphone microphone.

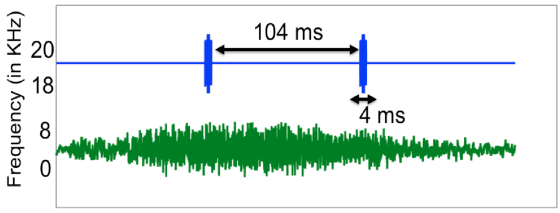


Figure 5.2: The figure shows the sonar OFDM signals (18-20 kHz) mixed with the cover music (sub-10 kHz). Generated in software to show the separation in frequency space between the OFDM symbols and the cover music.

5.2.3 Signal Generation at the Speaker

CovertBand generates OFDM symbols in the upper audible range (18-20 kHz), which we play through a connected portable speaker. Since our phones' microphones accept samples at 48 kHz, by the Nyquist condition, the effective bandwidth is 24 kHz. We divide this bandwidth into 64 subcarriers, each with a width of 375 Hz. We assign random data (either 1 or -1) to the seven subcarriers between 48 and 54 that correspond to the frequencies of 18 to 20 kHz and perform an IFFT on these 64 values to generate a 200-sample OFDM symbol in the time domain. When sampling at 48 kHz, a 200 sample OFDM symbol forms a pulse that spans 4.2 ms. Though radios have oscillators that let them transmit quadrature and in-phase components, and hence transmit the complex numbers output by the IFFT, speakers only accept 16-bit real numbers, so we transmit the amplitude of the IFFT samples and discard the phase.

Although not all adults can hear frequencies at 18-20 kHz, when played through an off-the-shelf portable speaker at high volume there are audible sub-harmonics at lower frequencies. Fig. 5.1 shows the spectrogram of a 19 kHz tone played from a JBL portable Bluetooth speaker and recorded using a smartphone. The plot shows sub-harmonics in the 11 kHz range. To hide these sub-harmonics, we combine the OFDM symbols with an audible song (see Fig. 5.2). Specifically, we play music continuously in the 0.1-8 kHz range and transmit the OFDM symbol every 105 ms,

i.e., about 9.6 OFDM symbols every second. Since the song and OFDM symbols use different frequency ranges, we can isolate the OFDM symbol at the receiver with a high pass filter, removing both our song and any environmental or cover noise below the 18-20 kHz band.

We note two key points about our design. First, we generated a number of random OFDM symbols and found that each symbol has different audibility level when played on a speaker. When we compared the structure of the OFDM symbol with its audibility level, we found that OFDM symbols that do not ramp up to full volume near the beginning or end of the signal are much less audible than symbols with a sudden change in amplitude (i.e., a crescendo and decrescendo, rather than immediate spikes at the beginning or end of the symbol). We use one such symbol in our design, though we did not excessively optimize OFDM symbols for this purpose.

Second, the songs selected effected the detectability of our sonar signal. We found that songs with more percussive events easily obscured the sonar signal but songs and speeches with many silent pauses were unable to mask certain elements of the signal. In our design, attackers can modify the ratio of song volume to sonar signal volume to better hide the OFDM signal. We found that song volumes higher than a quarter of the sonar signal volume were sufficient to our ears. That is, songs played at much lower volumes than the signal proved sufficient to mask it in our tests. See§5.3.5 for the evaluation of covertness.

5.2.4 Computing Distance from Microphone

When the speaker plays our sonar signal, sound waves reflect off both static objects in the environment and moving persons before being reaching the microphone. To find the distance of the people from the microphone, CovertBand performs to steps. It: (1) estimates the channel correlation for each transmitted OFDM symbol to find all reflectors, and (2) compares consecutive correlation profiles in time to seek moving reflectors, which we assume to be humans (but could be other moving objects).

Step 1: Generating the channel correlation profile. To generate the correlation profile, we pass the recorded signal through a high-pass filter to remove the song and isolate sonar signal reflections.

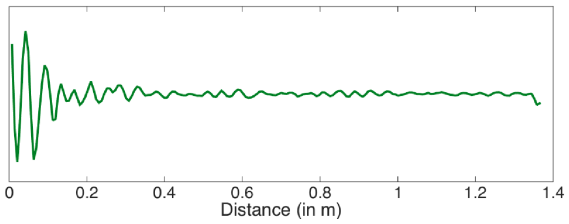


Figure 5.3: The figure plots an example correlation profile for a time instance recorded through a door in one of our five home experiments. The peaks represent all the major reflectors including the static objects and the human subject present at the corresponding distances.

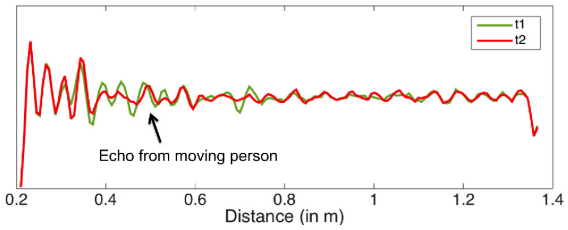


Figure 5.4: The figure plots an example correlation profile at two time instances t1 and t2 also recorded in a real home environment. The peaks shift farther to a distance of 0.45 m when the person moves away from the smartphone.

Fig. 5.3 shows a sample correlation profile at a specific time instance. Each peak corresponds to an object's echo. We can find the distance to the object using the speed of sound and the observed time delay. Since OFDM has strong autocorrelation properties, the correlation profile is accurate within a range of 2 to 3 samples (1-2 cm sampling at 48 kHz).

Step 2: Identifying the echo from the person. When a human moves, the resulting echo occurs at different distances over time. Fig. 5.4 shows the correlation profile at two time instances separated by 0.02 s when the subject moves from 0.45 m to 0.5 m. We clearly see the change in the echo's position. We extract this change by performing a consecutive subtraction of the channel correlation profiles every 110 ms. We should note that the consecutive subtraction operation removes the constant echo from all the static objects in the environment. During human motion, we see a significant change in echoes mainly at the distance corresponding to the individual's location. This change occurs in a range of distances corresponding to the human profile (height and weight). However, we also see some minor changes at slightly large distances due to the dynamic multi-path i.e. a multi-path that changes when the human moves. For example an echo from the human might reach a nearby strong static reflector first before reaching the receiver. In most cases, the dynamic multi-path causes small changes at larger distance than the direct path and hence we identify the smallest distance at which the difference is above a relative threshold to identify the new location

of the human. CovertBand uses 60% of the maximum change as the minimum threshold. The dynamic multipath causes larger changes in the echo only when there is a strong reflector very close to the human subject. Since the reflector is basically in contact with the human subject, this leads only to a small error of less than 10 cm.

To perform the correlation, enough energy must pass through the barrier, regardless of path, and reflect back to the microphone. We find this to be the case in our real-world experiments. Our intuition is that, in normal environments, sufficient gaps and holes (light switches, power outlets, door frames, windows, etc.) let sound propagate and return to the microphone for sensing.

Multipath In cases where the wall is not permeable to our sonar signal, the large correlation will correspond to an indirect path that over-estimates distance. This would cause error in our experiments. However, our experiments in realistic environments with various configurations show that some energy does pass in a sufficiently direct path to moving reflectors. Although the direct path may not be the highest energy reflection, we can see that it moves in the same way as do the higher energy reflections, indicating that it is a shorter path to the same moving object. We use the lowest-distance reflection to get an upper bound on the individual's distance.

In principle, we can infer the actual position of an individual even without *any* energy traveling along the direct path. One potential technique would build on [201], which uses angle of arrival to locate the source of reflections even if the direct path is fully blocked. We did not need to do this for our proof of concept information leakage demonstration because sufficient energy passed directly perform sensing.

Multiple people When multiple people move in an environment, consecutive subtraction of correlation profiles will show changes at the different distances corresponding to each human. We record on two microphones so multiple people at the same distance from one microphone are likely to show up as distinct reflectors on the other microphone, unless they are, in fact, in the same position. To identify each individual, we first scan through the entire correlation profile and attribute each distance where the difference in the correlation profile exceeds the minimum thresh-

old to a single person. In our implementation, we attribute motions that occur within a distance of 20 cm to the same person. We use the distance that corresponds to the maximum difference in the correlation profile in each group to compute the individual's location. Note that multi-path reflections corresponding to a single subject move at a fixed rate and distance. Using this, we can disambiguate between reflections from different subjects. In some theoretical situations where two subjects are in precise locations and move in concert at the same time, we will see only one change in the correlation profile and thus recognize one person even when there are two separate individuals. This highly unlikely situation is beyondthe scope of a commodity system such as CovertBand .

5.2.5 Tracking with Multiple Microphones

We can track an individual's location using the distance from multiple microphones. Note that the distance measured in the previous step sums of the distance from the speaker to the individual and the distance from the individual back to the microphone. In a two dimensional space, given this distance, the human can be at any point in the 2D plane along an ellipse with the locations of the speaker and microphone as the foci and the measured distance being twice the length of the major axis. Thus, each microphone creates an ellipse; the intersection of the ellipses from multiple microphones provides the individual's location in the plane. In the case where one cannot assume the subject is moving in a 2D plane, we would need a third microphone (for example, one plugged into the audio jack) or a different device to do trilateration in 3D. For example, the Amazon Echo has a 7 speaker array [96] that would help immensely for 3D localization.

Since the phone has two microphones, the individual's position can be at any location that occurs at the intersection of the two corresponding ellipses. While two ellipses can intersect at four different points, in our case both ellipses share a common focus (i.e., location of the speaker). Hence, they intersect at only two points that are symmetrical along the line joining the two microphones on the smartphone. These two points lie on either side of the barrier and can therefore be used to disambiguate the motion on either side.

5.3 Experimental Evaluation

Here we evaluate CovertBand 's ability to achieve our goals from §5.1. Because sonar has not been used with these goals in mind, namely for covert, through-barrier sensing on commodity devices we aim to show the constraints of our design choices from §5.2 and demonstrate feasibility in scenarios that represent realistic threats.

We had five main goals from §5.1:

1. Identify different classes of motion and activities.

2. Track multiple people in a 2-D environment.

3. Evaluate range through different materials.

4. Ensure that our methods work with existing, cheap, off the shelf hardware.

5. Conceal the attack from the victims.

In this section, we outline our experiments, implementation details and present results. We performed experiments for each of our goals to show feasibility in real world environments.

Implementation Details We implement our design of CovertBand as a third party Android app that does not require rooting the phone. The app uses the *AudioTrack* API to play the acoustic signals and the *AudioRecord* API to record simultaneously on both microphones in stereo. Our design requires the following from the phone: (1) transmit acoustic signals at 18-20 kHz, (2) sample the received signals on the microphones at 48 KHz, and (3) have two microphones to achieve 2D tracking (recall that a smartphone with a single microphone can be used to estimate the distance but not 2D position). Many Android phones including Samsung Galaxy S4, Samsung Galaxy S5 and HTC One satisfy the above requirements.

For each of our experiments, we used a Samsung Galaxy S4 connected to a portable speaker [100] through the audio jack. Like most common smartphones, the S4 has two microphones, one at the

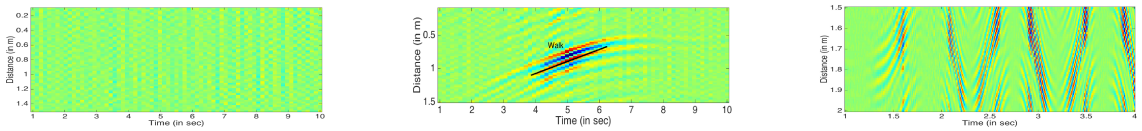
top and one at the bottom, separated by approximately 15 cm. Our Android app transmits a song along with our sonar signal through the speaker and records the backscattered signals using the two microphones as described in §5.2. We record and process the raw sample data and send the results to a laptop over Bluetooth for offline processing. We note that this attack can potentially be done on the phone, rather than offline, if needed.

Microphone Orientation. For each of our experiments, we placed the phone on its side, so that the two microphones (one at the top of the phone and one at the bottom) were in the same horizontal plane and perpendicular to the direction of the subject. For distinguishing between activities, any orientation should work. However, for some of the experiments discussed later, such as 2D tracking, the only relevant property of this orientation is that the two microphones are in the same plane as the target, as we can only make inferences in this plane. With more microphones we could potentially sense in 3D.

5.3.1 Distinguishing Between Activities

First we demonstrate CovertBand's ability to help an attacker infer information about *what* a person is doing using two basic methods: (1) inference based on characteristics of motion and (2) inference based on timing.

Inference based on characteristics of motion We show how CovertBand can potentially enable an attacker to differentiate between different classes of movements even when subjects are in different body positions and orientations. Specifically, we focus on two classes of motion: (1) linear motion (the subject walks in a straight line) and (2) periodic motion (pelvic tilt where the subject remains in approximately the same position (lying on his or her back on the floor) but performs a periodic exercise). These motions are sufficiently different that we should be able to differentiate them by looking at the spectrograms, but are also realistic enough to potentially enable privacy leakage. For example, (1) models information that might be of interest to intelligence community members, e.g., to track the location of a target within a room and (2) could be used to infer sexual activity, for which the importance of protecting might vary depending on the target's culture and



(a) No action (b) Walking (c) Repeated pelvic tilts

Figure 5.5: Spectrograms of the change in the correlation profile at all distances (increasing from the top in the y-axis of the figure) over time (x-axis) for three different cases a) When the person is stationary, there is no change. b) For a linear motion like walking, the maximum change occurs at the new location of the user. c) For a rhythmic activity like pelvic-tilt exercise, the changes repeat over time.

cultural norms or might vary depending on the target's public visibility, e.g., celebrity status or political status.

To run these experiments, we placed our phone and speaker 20 cm from a standard interior wall [99]. A subject 1 m from the inside of the wall was asked to perform each of the above activities. We then transmit our covert sonar signal and track the changes in the echoes as a function of time as described in §5.2. Fig. 5.5 shows the spectrogram plots for the different activities. The x-axis denotes the time and the y-axis is the distance of the subject from a single microphone at the smartphone. The spectrogram plots the difference in the echoes received where the differences are computed over successive 10 ms durations. The plots show that:

- First, when there is no activity on the other side of the wall, as expected, we do not see any significant changes in the echoes as received by the microphone.

- As the subject walks towards the phone on the other side of the wall, we see a strong change in consecutive echoes occurring at decreasing distances from the phone. The black line in Fig. 5.5(b) shows the actual distance of the subject as a function of time. By looking at the areas where the changes in the echoes are the highest, we can see that CovertBand accurately tracks the distance.

- When the subject performs a repetitive motion from a stationary position, we see a repetitive signal at a fixed distance (1.5 m).

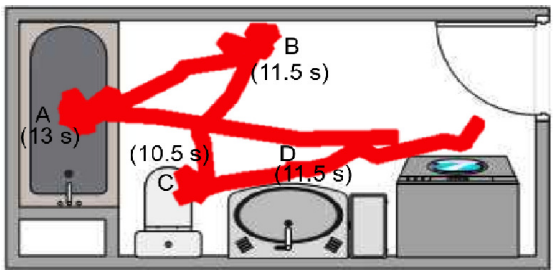


Figure 5.6: We were able to localize the person in different areas of the bathroom. According to our sonar readings, Bob spent 13s at station A, 11.5 s at station B, 10.5 s at station C and 11.5 s at station D. (Ground truth: 19.5 s at A, 13.4 s at B, 12 s at C and 14 s at D). Trajectory line thickened for visibility.

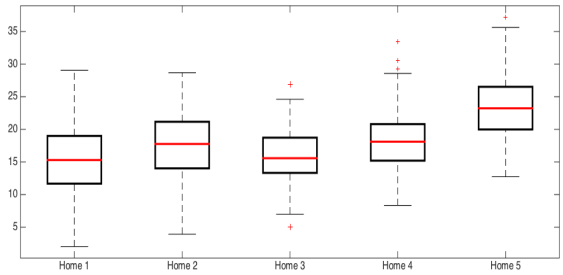


Figure 5.7: Experimental Setup 1: Box and whiskers plot of tracking error for one moving subject in the bathroom. We calculate error by comparing observations with trajectories based on starting and ending points marked on the floor.

We also tested other rhythmic motions, such as jumping and pumping arms with the subject in a standing position. The plots look similar to the pelvic tilt in that they are clearly repetitive, but have different energies and distances associated with them. Though this is clearly an example of a rudimentary classification, it requires no training phase to generate the data above that enables an additional privacy breach. More sophisticated attackers could potentially train models to do more accurate classification or detect additional types of movements. And though we do not aim to do gesture recognition in this work, even recognizing a motion as repetitive may be sufficient as a privacy threat in situations where circumstantial evidence can be damning. We also note that CovertBand should still be able to differentiate between linear and rhythmic motion in cases where subjects come into contact with stationary objects in the environment. We verified in §5.3.2 that CovertBand could detect the motion and localize a person sitting in an office chair, performing rhythmic motions.

Inference based on timing As we will see in §5.3.2, we can use CovertBand to do 2D tracking of subjects even through walls, which can further leak information about potentially private activities. To demonstrate this, we show a scenario where one subject (Bob) pretended to go through a routine

in the bathroom while the other (Alice) used CovertBand to track his movements. We placed the speaker setup 15 cm outside the bathroom door and performed four trials during which Bob spent less than 20 seconds doing each of the following: showering, drying off on the scale, sitting on the toilet, and brushing his teeth. *During the experiment, the bathroom fan was ON and we could not hear Bob performing any of the activities inside the bathroom.*

Using CovertBand , we were able to localize Bob at different areas of the bathroom as he performed different tasks. Fig. 5.6 shows a 2D mapping of Bob's movements with observed timing as he stopped in each location. We were also able to download the publicly available floorplan for this particular apartment, which allowed us to map each of the stops to different stations within the bathroom. For example, we can guess that Bob was probably using the sink during D, or showering during A. Floorplans for many apartments and hotels are available online.

We note that the notably higher error for timing while Bob was at station A is due to Alice's inability to differentiate between Bob standing in the shower, and Bob opening/closing the curtain or simulating drying off directly outside the shower. If we had simulated Bob showering for a more realistic amount of time, the relative error would be much more reasonable.

During this experiment, the participant did not spend a realistic amount of time at each station, rather simply paused and used a timer to record how long he stayed at each stop. We can see that being able to localize Bob within 20cm can tell us a lot about which part of the room he is in and with what he could be interacting.

5.3.2 2D Tracking

As we mentioned above, CovertBand can track 2D movements using echoes from multiple microphones on a phone (recall §5.2.4). Here we strive to demonstrate and quantify our 2D location tracking ability by performing experiments in real-world setups both from within the home and from outside of it.

Home Environment: In our first setup, we ran two sets of experiments in each of five real-world homes in a metropolitan area. For each, we asked volunteers of different height and weight to

perform various actions. We confirmed in separate tests that larger subjects (largest was 6'3", 180 lbs) reflected more energy, than smaller ones (smallest was 5'3", 130 lbs). However, though they reflected more energy, this did not affect our ability to localize them or change the observed error. In some cases, larger people were easier to detect, but caused more observed error as they reflected energy from a wider space.

Experimental scenario 1 - Single subject. For the first experiment, we ran three trials at each location where we placed the phone and speaker on a chair **outside a closed wooden bathroom door** and asked a volunteer to walk along a straight 1 m line marked on the bathroom floor. The thicknesses of the wooden doors were standard [98], but some were hollow and some were solid wood. Also, in a couple of homes the fan inside the bathroom was on, and in all the homes we could not visually see or hear the subject performing the activity. For each trial, we compared the 2D trajectory computed by our system with the marked trajectory on the inside of the bathroom. Fig. 5.7 shows the mean 2-D tracking error across the bathrooms in the five homes. While we see a variation in the errors across the homes, owing to differences in the bathroom door material and the natural variation in movements across different subjects and trials, across the five home environments the mean tracking error for the bathroom experiments was just 18 cm.

In a similar scenario, we were also able to localize a subject in close contact with a strong multi-path reflector. To demonstrate this, we placed a large metal sheet next to a subject, who performed rhythmic motion. As mentioned in §5.2.4, the close proximity of a strong multi-path reflector causes dynamic multi-path. However, we were still able to correctly localize the moving subject within an average error of 13 cm.

Experimental scenario 2 - Multiple subjects, multiple motions. We placed the speaker and phone **outside a bedroom** to detect and localize subjects inside. In three of five homes, we placed the setup outside the bedroom wall; in the the other two, we placed it outside a closed bedroom door. All walls and doors were standard and all doors were closed for the experiments. To show that we can simultaneously track two people, we asked one person to stand 2 m from the wall closest to the speaker and continuously move his torso in a rhythmic fashion. At the same time, a second person

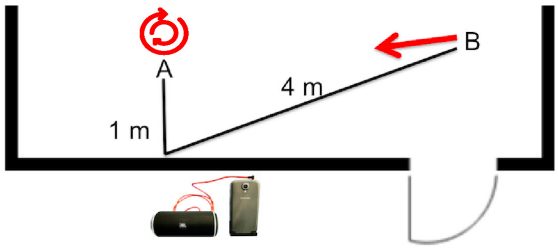


Figure 5.8: Experimental scenario 2: Multiple people, multiple motions. An example of one of the bedroom layouts. Subject A twisted at the hips, while subject B walked toward him. For all experiments, doors/windows were closed.

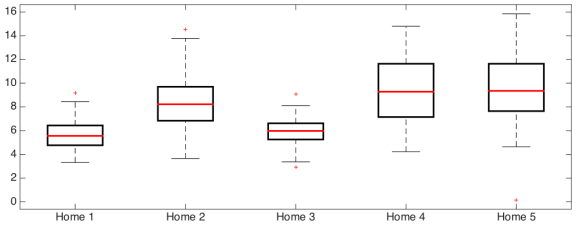


Figure 5.9: Experimental Scenario 2: Box and whiskers plot of tracking error for the twisting subject from Fig. 5.8 (Subject A). The error is smaller because he is stationary (only moves his torso).

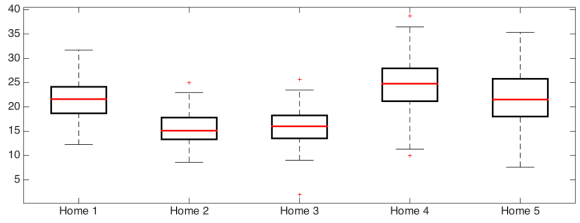


Figure 5.10: Experimental scenario 2: Box and whiskers plot of tracking error for the walking subject from in Fig. 5.8 (Subject B). We compute error based on the trajectory marked on the floor.

walked 2 m towards the first person. For these experiments, distances and orientations were a bit different in each setup depending on the layout of the room. The experimental setup for one of these layouts is depicted in Fig. 5.8. For subject A, we calculated the tracking error as the distance between the computed 2D location and the true location. For subject B, we computed the tracking error as the difference between the computed trajectory and the direct line between the starting and ending points.

In the bedroom experiments, the mean tracking error was 8 cm for the subject twisting in a fixed location, but 20 cm for the walking subject. The discrepancy in tracking error between the subjects is in part because we measured the accuracy against a reference line which was directly between the starting point and the ending point. In reality, it is likely that the walking subject deviated

slightly from this direct line while moving 2 m toward the twisting subject. For example, even for a subject walking directly on the trajectory line, normal walking motion includes arms, separated by more than 20 cm, swinging with pendulum-like motion. As such, we do not try to optimize our results for errors less than 50 cm. Despite this, all calculated errors were still less than 25cm (less than 1 foot) and in each trial we were able to both localize each movement and identify it as either the linear movement or the rhythmic one. Though the experiments for Fig. 5.7 were done in the same respective homes as those for Fig. 5.9 and 5.10, they were done in different rooms with different layouts, so we expect some deviation in the results. However, the mean tracking error for the bathroom experiments are similar to errors for the walking subject in the bedroom setup, owing to the similarity in movement, the similarity of materials within homes and similar variability in subject trajectory.

Experimental Scenario 3: Multiple people, convergence. To demonstrate this behavior, we performed two more experiments to confirm that CovertBand could deal with two subjects converging to a single location. In these experiments, both subjects performed a whole body motion by walking towards and away from each other inside a room with the speaker setup placed outside a standard interior wall. Specifically, for the first experiment, the two subjects were present at 1 m and 1.7 m from the wall and walked towards each other; for the second experiment, both the subjects started from the same point 1.5 m from the wall and walked in diagonally opposite directions. Fig. 5.12 shows the CDF of the 2-D localization errors of the two subjects. From the results, we found that we were able to localize both the subjects with an average error of less than 20 cm. When the subjects get sufficiently close, i.e., less than 20 cm, our thresholds begin to treat them as a single person. As they walk away, the threshold separates them into two distinct people again.

A Note About Multiple People. As described in *Experimental Scenario 3*, when multiple people are far apart, CovertBand can distinguish between them, but when they come within close contact, it treats them as a single person. One person moving rhythmically and one static will be treated as a single person moving rhythmically. Furthermore, if both are moving linearly together, they will simply be treated as a single walking person, but they will likely reflect more energy. Because CovertBand is only accurate to 18 cm, we leave it out of the scope of this work to distinguish

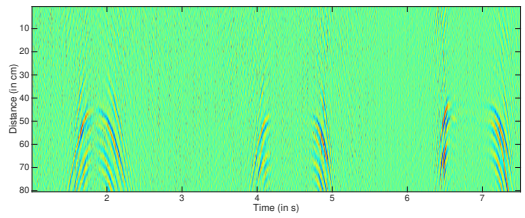


Figure 5.11: Spectrogram shows the seated person performing the rythmic motion when the other subject stands in contact with the chair and the subject.

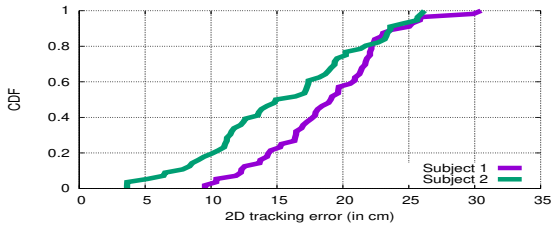


Figure 5.12: CDF of localization errors for two subjects when they are walking towards each other as in experimental scenario 3.

situations where there is only one person in a location vs. when there are two. An attacker using CovertBand would have to deduce this using prior knowledge. For example, she could use CovertBand to observe that there were previously two people moving toward each other. We note that an attacker can make inferences about activities in a bedroom for example, where two people converge to a location, spend time performing rhythmic activity in the same location, and then separate some time later without obtaining concrete evidence. As mentioned in §5.1.1, this could be a significant potential security threat for certain individuals. However, as we discuss in §5.6, CovertBand will not allow an attacker to identify either subject once they have separated.

Two people moving rhythmically in the same position will simply look like rhythmic motion (albeit with potentially different frequencies) on a spectrogram, similar to the pumping arms scenario, where the subject's arms were not moving in unison. We also note that a static person between the attacker and a moving subject will act like a stationary object in between a victim and attacker, serving to attenuate the sonar signal to some degree. To show that CovertBand can perform this type of tracking when subjects are in contact with stationary objects in the environment, we ran a similar experiment where the stationary person sat in an office chair and performed rhythmic motion while the walking subject walked 1 m to that position and stood next to the seated subject (who was performing rhythmic motion) both in contact with the chair and the person. CovertBand correctly recognized that rhythmic motion and was able to localize both the seated and walking subjects. When they came within 20 cm of each other, it treated them as a single per-

son continuing rhythmic motion. Our system was able to localize this subject with an average error of 9.57 cm. Fig. 5.11 shows the spectrogram for the seated person performing rhythmic motion. We can see that despite contact with a static object and another human, we are still able to see the rhythmic motion of the subject.

Overall, we find that resolution and accuracy were not drastically affected in any of our real-world experiments despite very different circumstances, building structures and layouts. This implies that in common scenarios, CovertBand can be resilient to changes in location and, unlike some alternative approaches, does not require a training phase before performing through wall detection and tracking.

Spying through external walls and doors: In addition to detecting and localizing subjects across rooms inside a home environment, we also tested CovertBand through external doors and walls. These barriers are thicker and are often designed so that any outsider cannot visually see or hear anything that happens within the home. We conducted experiments in three different locations one apartment and two standalone homes to test our ability to do 2D localization through external materials. Specifically, we tested a 4.5 cm thick solid wood door, a 30 cm thick exterior foundational basement wall composed of a combination of different wood paneling and sheet-rock, and an external double-paned window with curtains drawn. In each experiment, we placed the speaker close to the barrier and ensured that all windows and doors were shut. In the apartment, this meant placing the speaker in a neighboring apartment along a shared wall.

In all experiments, the speaker was placed 5 cm from the barrier on the house's exterior and a test subject walked a distance of 1 m inside the home. We then localize this subject over time using their walking motion and compute the average of shortest perpendicular distance to the original trajectory as the localization error.

We found that we were able to localize the subject within an error of about 30 cm in the case of the external door an window. As an example, Fig. 5.13 shows the speaker setup and the measured trajectory for the experiment that we conducted in the apartment. For that experiment, we placed the speaker setup along the shared wall in the neighboring apartment an asked a subject to walk toward the wall on the other side. We were able to localize this subject with an average error of 30

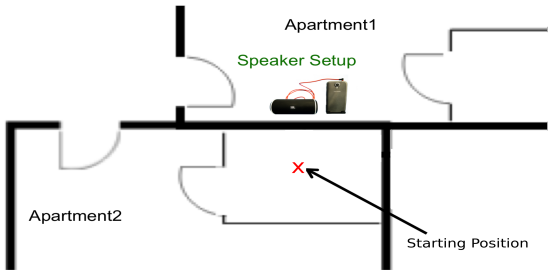


Figure 5.13: The figure shows the speaker layout placed in apartment1 close to the wall shared with apartment2. The subject walked in a line toward the speaker on the other side of the drywall.

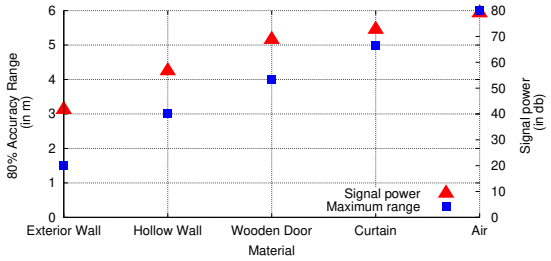


Figure 5.14: **Effect of barrier material.** The figure plots the measured signal power and the maximum range at which we can accurately locate an individual through five different barrier materials.

cm. However, we were not able to track the subject through the foundational basement wall due to much thicker external materials. However, an attacker could potentially increase the range and penetrate thicker materials by using a better speaker with higher output power.

5.3.3 Evaluating Range through Materials

To estimate upper bounds on CovertBand 's ability to sense in different environments, we ran experiments through various barriers in a two-bedroom apartment and measured the maximum ranges at which CovertBand could detect movement. In particular, we placed the speaker with no barrier as a baseline and 30 cm from the following visual obstructions: a hollow wooden door, a hollow interior wall, and a hollow exterior wall. In each case, we performed the experiments in the context of the home so as to study feasibility in realistic situations. The interior walls were all standard hollow walls [99] and the doors were made of hollow wood were standard interior doors [98]. The external walls were also drywall and had an estimated thickness of 10 cm.

For each situation, we placed the Samsung Galaxy S4 connected to a JBL portable Bluetooth speaker around 30 cm from the barrier. We also placed an Amprobe sound pressure meter [41] 20 cm away on the other side of the barrier to measure the attenuation. We asked a volunteer to take two steps (around 60 cm) on the other side of the barrier at various distances so that we can

calculate the maximum distance at which CovertBand can successfully detect the motion. The volunteer repeats the motion five times at each distance until we cannot successfully detect it with an error of less than 30 cm, 80% of the time, i.e., four of the five times. This is sufficient in our opinion to constitute a privacy breach.

Fig. 5.14 shows the pressure values reported by the sound pressure meter and the maximum distance at which we can detect the movement correctly in 80% of our trials. In order to count as correctly detected, we require that CovertBand track and localize the movement with an error of less than 30 cm from the ground truth trajectory, marked on the floor with tape. Logically, as the attenuation caused by the barrier increases, the distance at which we can detect the motion decreases. The maximum distances are around 6, 5, 4, 3 and 1.5 meters for air, curtain, wooden door, hollow wall and exterior wall, respectively. In addition to using speakers capable of higher volumes, we mention a few ways to increase this range across all materials in §5.6. Though these results are only for a single apartment, our experiments in five other homes in later sections showed similar results. We note that our system can detect movement with a lower probability (e.g., one out of five) at a further distance.

5.3.4 Using Common Existing Hardware

For our detection to work, the speaker needs to transmit signals at the higher end of the audible frequency range. As such, rather than re-running all of our experiments in homes with many different speakers, we find it equally effective to compare the relative volumes at which a variety of speakers can play frequencies in this range.

We compared 4 cheap portable speakers, including the JBL [100] we used in the above experiments, and a home theater system. All were able to play tones at comparable volumes at frequencies from 200 Hz-20 kHz with the exception of a Bose Bluetooth speaker, which had a noticeable dip at 17.5 kHz. We also noticed more speaker "clipping", across all portable speakers at full volume when we played some of the higher frequencies. Despite this, as we demonstrate in other sections, the power is still sufficient for the purposes of our attack and the "clipping" noises don't harm our ability to perform tracking. The clipping is a result of the speaker not being able

to drive the speaker cone with the expected amplitude sufficiently fast, so even at 90% volume the clipping disappears. Better speakers capable of higher volumes would not necessarily have this problem. The home theater system we tested, for example, did not exhibit clipping at any of the volumes we measured.

We believe that all five of the speakers we tested are fully capable of executing our attack with ranges from 2-6 m, with the caveat that the signal may need to be altered to use the Bose speaker. The only other components we need are a pair (or more) of microphones that can sample at 48 kHz or higher. Many common smartphones, such as the Galaxy S4/5, LG G4, and HTC One, have microphones that satisfy this requirement.

Demonstrating information leakage with smart TVs As we mentioned, the ability to use common hardware for this type of attack opens up a variety of potential devices for an attacker to use to get information about a target; an attacker with access to a speaker and microphone that already exist in the environment can potentially leverage them to glean information about remote targets. We show that such an attack is possible if an attacker can get an over-permissioned application on a smart TV. To demonstrate, we use CovertBand to play some of our altered songs through the television speakers of a 42 inch SHARP TV [160]. Though all of our previous tests were done on a Samsung Galaxy S4 with software tuned for the orientation of the speakers on that device, an attacker may not know the exact orientation and power of speakers on a victim's device, or the relative position with respect to the speaker they now control. To account for this, we used the dual microphones from a larger LG G4 phone (with microphones in different positions) and placed it on a table next to the TV. As we mean to demonstrate feasibility, we conservatively did not change any of the software to tune it for the new hardware. We had a volunteer stand two meters in front of the television and perform repetitive motions for a short interval in a stationary position. We found that using the standard TV front-facing speakers we could correctly estimate distance of a subject within a maximum error of up to 30 cm.

5.3.5 Evaluating Covertness of our Design

Our final goal was to conceal the signal, and its subharmonics, against unknowing attackers. We designed an experiment to get an estimate of an upper bound on detectability by testing whether subjects who were both familiar with the experiment and knowledgeable of the signal could differentiate between unaltered songs and songs containing our sonar signal. Our logic is that subjects who have been exposed to the signal, who are not obscured by barriers, and who have been told that the signal will be present should detect our attack at higher rates than unknowing victims.

Experimental Setup Participants were asked to identify which of two clips (played back-to-back in a random order) was the original unmodified song and which was the clip with our added sonar signal. The volume of the added sonar signal was half the volume of the song (6 dB lower). Our subjects (17 female, 16 male) spanned ages between 18 and 45 and were composed of students and staff from different entities in our organizations as well as other local individuals. None of the subjects were monetarily compensated. All the experiments reported in this were given an exemption by our organization's IRB. Fig. 5.15 shows a histogram of ages and genders.

We used sixteen different songs chosen from a list of popular songs and selected ones with sufficient amounts of percussion. We did not use any sophisticated methods for choosing songs or try to maximize our ability to hide the signal within. The songs are listed in the appendix. Each participant was seated in front of a Beats Bluetooth speaker [45] with no barrier blocking the speaker. They sat within a 30 degree angle of the speaker's face and were allowed to move around as close or far as needed. To give our victims as much power as possible, we spent the first few minutes of each experiment training their ears by playing our signal un-obscured. We also allowed them to ask for a replay of any clip at any time. Every subject we tested was able to identify the sonar signal when played without music cover. They reported hearing something similar to static.

Results We expect our subjects to be able to guess the correct clip with slightly better than 50% accuracy. Intuitively, this is because with no information, we would expect them to succeed at 50%. However, we have given our subjects a number of advantages and have not tailored our signal to

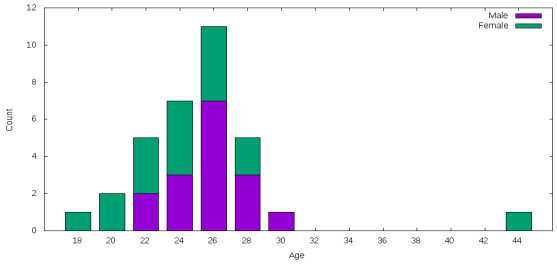


Figure 5.15: Histogram of ages and genders for test subjects from experiments in §5.3.5.

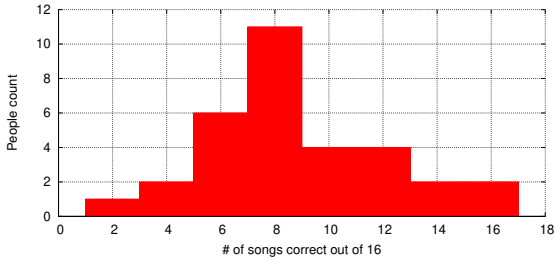


Figure 5.16: Distribution of detection scores. Random guessing would on average result in 8 correct songs.

hide specifically within songs. Further, some subjects are likely to be able to hear frequencies at which we transmit [37], driving up the expected percentage of correct answers.

Our results fall mostly in line with our hypothesis. Overall, our 33 subjects guessed the correct clip 58% of the time, for a mean of 9.3 out of 16, a median of 9, and a standard deviation of 3.077.

We also asked each of our subjects to record any pairs for which they were very confident they had correctly identified the signal. Of the 528 total trials, this resulted in 71 claims of high confidence, of which 60 (85%) were chosen correctly. These high confidence claims tended to be of two varieties: 1) Three subjects claimed to be able to hear the sonar signal clearly in almost every song. Of those, two chose the correct clip in all sixteen trials. One (Female, 21) was able to hear the high frequency tone in one of two ears. The other (Female, 22) heard both the clipping and the high frequency sound with ease. The third (Male, 27) guessed incorrectly twice before figuring out that he could feel the clipping sound if he leaned in directly in front of the speaker. 2) There was one song for which almost everyone was able to identify the sonar signal due to many short silent periods in the song. A third of subjects marked this song with high confidence and all of those subjects correctly identified the modified version. Many explicitly mentioned this song after the conclusion of the experiment. If we remove this song from the statistics, even with the three subjects above included, the detection rate drops to 56%. If we additionally remove the three subjects who were able to consistently detect readings, we show a detection rate of under 53%, less than 3% above random guessing ($p = 0.12$).

Most of our subjects admitted that they guessed randomly on every trial in the experiment despite the lack of barriers, close proximity, and prior training.

Key Observations When played at full volume (as we did in our experiments), whether or not we obscured with music, there was an audible clipping sound due to the sonar signal. However, reducing the volume by just ten percent removes the clipping, so we would expect even lower detection rates at more modest volumes.

Through-Barrier Follow-Up To get an upper bound on detectability in through-barrier scenarios, we re-tested two of the three subjects who claimed to hear the sonar signal in every song in a through-the-wall scenario. We placed them inside one of the rooms we used for experiments in §5.3.2 and placed the speaker [100] along the same wall we used before. Using the exact same songs which the subjects were able to detect from the unobstructed trials we asked them if they could identify the sonar signal. We only collected subjective feedback on their experience due to a very small sample size and limited subject availability. Though both were able to hear the unobscured signal through the wall, when mixed with music they admitted to having low confidence on over two thirds of pairs. Subjects had the greatest success when lying on the floor to listen through the space at the bottom of the door, indicating they were able to identify the signal much better when unobstructed.

Improvements We believe that a sophisticated attacker could hide the signal during percussive events in the music on a per-song basis, making it incredibly difficult for most people to detect. We also randomly generated our OFDM symbols. An attacker who really wanted to make it undetectable could generate a tone specifically meant to avoid clipping noises (some symbols were better than others) and avoid playing the signal at 100% volume. Future work could inclue specifically analyzing the ratios of music to OFDM symbol and tailor the volumes to match. However, even simple improvements, such as removing the symbols during silent portions of songs would reduce detectability based on our responses. We also note that very young children and animals

often hear higher frequencies more clearly, and thus may be able to detect our signal. We are currently placing these detection mechanisms out of scope.

In summary, for most people, detection was very difficult despite all the advantages we gave our subjects. We hypothesize that for adult targets who are not aware of the presence of the signal, detection would be exceedingly unlikely even without a wall or other barrier. Even subjects who can hear the added signal may just think there is something strange with the speaker or music, like static from the radio.

5.4 Related Work

Acoustic Systems Acoustic transmissions have been used extensively to localize devices in systems such as Cricket [140]. [42, 174, 91, 206] demonstrated the feasibility of localizing and determining the direction of device movement using acoustic transmissions. Unlike prior work, we demonstrate the feasibility of tracking users without instrumenting them with any devices.

Prior work leverages Doppler shifts from inaudible acoustic transmissions to perform gesture recognition [82, 55]. These designs enable recognition of a pre-defined set of gestures in close vicinity to the mobile devices using the resulting Doppler shifts. [200, 186] achieve finger-level localization close to a phone; these techniques do not use active sonar but instead localize the sounds of fingers tapping on a surface. FingerIO [131] uses active sonar to track finger motion around a wearable device, but it is not designed to go through barriers. ApneaApp [130] detects breathing movements using acoustic transmissions on the phone with a range of one meter but is not designed to operate through barriers. Medical imaging uses high-frequency ($>$ 1 MHz) ultrasound signals to perform imaging inside the body [127]. These systems require the acoustic transducer to be placed directly on the skin's surface. Sonar imaging has also been used extensively in underwater settings [164, 92, 29].

Many sonar-based approaches address through-wall detection to aid Law Enforcement. However, these approaches currently require non-standard equipment placed directly against a wall [32, 58, 193]. As such, they would be expensive, forego plausible deniability, and could not be leveraged by remote attackers.

A rich body of work focuses on mapping environments using acoustics, for example [66]. However, this work does not pertain to ours in execution or motivation. In general, it uses first order echoes to reconstruct environments and is not designed to work through barriers or track movement.

RF-based Designs Prior work has proposed radio-based solutions for human motion detection [148, 36, 34], localization [148, 34, 62], and gesture recognition [143, 106, 132]. While promising, none of the RF-based designs have been demonstrated to work on off-the-shelf smartphones. Specifically, these designs use expensive, ultra-wideband transceiver and/or specialized hardware that are not available on mobile devices. Further, they typically require multiple antennas separated by half a wavelength, which is difficult on smartphones due to their size constraints. In particular, [148, 34, 62] use radar hardware that transmits, receives and processes 500 MHz to 3 GHz wideband signals and requires multiple antennas. Researchers have recently proposed ultra-wideband radar designs that operate in the millimeter wave [211] and terahertz bands of the electromagnetic spectrum [83, 176] where the wavelengths are significantly smaller, permitting multiple antennas to be packed together. Further, it remains to be seen whether the power and processing required for such wideband signals could be achieved on smartphone-grade consumer devices.

Recent work also leverages Wi-Fi for human motion detection and gesture recognition. [36] does human motion detection (walking forward and backward) in through-wall scenarios using 20 MHz wide transmissions but requires specialized interference cancellation hardware that is not available on commodity devices. [143] extracts Doppler shifts on RF transmissions to perform gesture recognition in through-wall scenarios. [54] uses specialized, full-duplex hardware to track finger strokes using RF signals. [191] extracts the minute changes that occur on a loudspeaker when playing a song using Doppler effects from wireless signals. However, these require custom hardware processing (e.g., software radios) and do not work with commodity devices.

Wi-Fi gestures [132] can enable recognition of a pre-defined set of gestures in the vicinity of an Intel Wi-Fi chipset. WiDraw [173] tracks arm motion using transmissions from 20-30 other

Wi-Fi devices. These systems work only when the user is close to the Wi-Fi chipset and have not been demonstrated in through-barrier scenarios. Work on tomography imaging [210, 194] tracks motion by deploying 10-30 sensors spread throughout the environment that measure the attenuation between every pair of sensors. WiDir [196] estimates the direction of motion using Wi-Fi CSI values; however, CSI information is not available on commodity smartphones at the software level. In contrast, to the best of our knowledge, ours is the first work to demonstrate user motion detection or 2D tracking through walls and barriers using just a smartphone and a Bluetooth speaker, opening up a new attack vector.

Finally, thermal imaging cameras have been designed to interface with smartphones and detect the heat radiated by humans using an infrared sensor array [70, 158]. These cameras can detect changes in heat radiation patterns and can hence see in the dark or detect pipes within walls. However, they cannot be used to see through walls or even glass surfaces [178].

Summary of limitations and comparisons to prior work While there has been extensive work on device tracking [140, 113, 42, 174, 91, 206], we focus our comparisons on device-free tracking, which does not require the victim to carry a device. Here, we compare our work to previous work in the device-free localization and tracking space in both sonar and RF domains. Table 5.1 lists those works grouped by approach and their capabilities and quantitative results. To be representative of the general approach, we have used the strongest listed results from each group of citations that are comparable to our work.

Of the compared systems, only CovertBand and the sonar gesture papers [131, 189, 130] can be implemented on commodity hardware. The gesture-based papers, however, do not focus on full-body detection. The Wi-Fi solutions exploit existing Wi-Fi infrastructure in a space to do 2D localization. However, they require control over multiple access points and network devices connected to the localization software in order to function. For example, to achieve 70 cm accuracy, [184] uses four access points and seven laptops to do localization in the home, in effect creating an 11 antenna array. Further, because they work by looking at disturbances in multiple wireless streams, they need those devices to get good coverage of the space, and need to know the

	CovertBand	FMCW Hardware [148, 34, 62, 35]	Software Radio [103, 36]	Wi-Fi [117, 154] [159, 198, 184]	Gesture Sonar [131, 189, 130]	Coupled Acoustic [58]
Can be done on COS hardware	•			•	•	
Detects moving subjects	•	•	•	•	•	•
Detects stationary subjects	•	•	•	•	•	•
Works in non-LOS	•	•	•	•		•
Differentiates different types of motion	•	•	•		•	
Enables remote attacks	•			•	•	
2D localization	•	•	•	•	•	
Range in ideal circumstances	6 m	20 m	4.9 m	20 m	0.3 m	13.7 m
2D Accuracy	18 cm	10.9 cm	80 cm	70 cm	0.8 cm	-

Table 5.1: Comparing CovertBand constraints and capabilities against similar approaches in RF and Sonar. Cells with • are considered to meet the criteria. Cells with • satisfy the criteria with constraints. Blank cells do not satisfy the criteria.

locations of each device. They also require training or a background collection phase, which would have to be redone should any device move.

All compared approaches permit detection of moving subjects (though we have marked the gesture sonar paperss as "satisfy with constraints" because they were not designed for human level tracking). However, CovertBand , software radio-based approaches [36, 103], and the DoD solution [58] can detect stationary people only when there is sufficient movement, i.e., arm movement or twisting motions. The Wi-Fi ecosystem approach [154, 159, 198, 117] and the FMCW approaches [62, 148] can localize static people using either breathing motion or by monitoring changes in an already mapped environment.

Of the attacks possible on commodity hardware, we mark the Wi-Fi approaches as "satisfy with constraints" because they require a detailed understanding of the WiFi AP and device placements in the environment and would require compromising many devices. The very small range of the sonar-based gesture papers have earned it the same score, though microphones could be utilized solely for eavesdropping purposes.

Most of these localization approaches are fairly particular about the placement of sensing

equipment. As mentioned, the Wi-Fi papers require a detailed understanding of device placements and retraining if devices move. The DoD approach [58] requires placing the acoustic coupler directly against the wall. CovertBand , the FMCW papers, and the software radio papers benefit from directionality, though they do not necessarily need subjects to be in any particular direction: for RF papers, range will be best when the subject is in the main lobe of the antenna. Similarly, CovertBand gets best results when subjects are in the forward direction of the speaker owing to the directional nature of most commodity speakers.

The main constraint of our work is range. Because RF propagates well through materials, it permits a longer range in through-barrier scenarios. We discuss some methods to increase range in §5.6. The custom FMCW builds [148] can increase transmit power to penetrate thick materials and measure up to 20 m. Similarly, because Wi-Fi approaches use existing Wi-Fi infrastructure, they can in principle work as long as multiple Wi-Fi streams intersect a location. This depends greatly on the wireless network's layout. Alternatively, because the coupled acoustics [58] are placed directly against the wall, they eliminate one of the biggest reflectors (known as "the flash problem"), turning the barrier into a speaker of sorts. This enables for much farther propagation than is achievable with audio on commodity hardware.

As noted, RF approaches without significant antenna arrays are far less accurate due to the vastly larger RF wavelengths. CovertBand benefits from shorter acoustic wavelengths and speed of sound, permitting accuracies similar to those of large antenna arrays in the FMCW approaches [148, 34, 62]. The DoD solution does only presence detection, not localization, so its accuracy is not listed. In effect, the Wi-Fi approach also has a large array. As mentioned above, To achieve 70 cm accuracy [184] effectively uses 11 antennas. The figures listed for the gesture sonar come from FingerIO [131], which was designed for tracking finger movement very close to the phone. As such, range and accuracy are very small. The best comparison to CovertBand is probably WiTrack2 [62], which operates at slightly better ranges around 10 m and accuracies around 10.9 cm. We calculated the accuracy using median x and y errors for the first detected subject (likely an underestimate). Again, however, this approach will not enable remote attacks as it cannot be done on commodity equipment.

5.5 Defenses

In this section we discuss some defenses against CovertBand attacks.

(1) Victims could prevent some versions of this attack by soundproofing their homes. This may be infeasible for many people, especially those who want windows that can open to the outside. However, one could soundproof more private areas of the home and remove speakers and microphones to prevent CovertBand attacks.

(2) Another defense that does not require structural changes involves jamming signals in the victim's inaudible range. However, jamming in the 18-20 kHz range may be audible to pets or children and requires playing inaudible sounds whenever a victim notices music near private areas. Furthermore, an attacker could allay suspicion by simulating natural sounds, like birds chirping or leaves ruffling, to do sparse sensing without explicit cover traffic (music). It does not always seem feasible to play sounds like this, though playing random noise across all potential frequencies in the inaudible range in private areas would thwart the attack. We note that although background noise such as loud music can thwart eavesdroppers, it will not thwart CovertBand , which filters out signal in the audible range.

(3) One could set up a dedicated sensor, like a Raspberry pi with an attached microphone, that listens for transmissions at frequencies that exceed a victim's hearing threshold. A potential defense could combine this with jamming, sensing when there is a potential external sonar signal and responding by jamming with a known pseudorandom sequence in a comparable frequency band.

(4) Finally, a simple smartphone application could be built to fool or jam a CovertBand attack. Upon detecting high frequencies, the app could match the frequency range and signal power and jam with random noise in that band. We verified the effectiveness of such a defense against CovertBand attacks by conducting an experiment in the bedroom of one of our test homes. With the speaker setup 20 cm outside the wooden door of the bedroom, a subject walked 1 m towards the door from the inside. At the same time, we set up another smartphone as a jammer inside the bedroom 2 m from the door and played a random OFDM signal in the same frequency range of

18-20 khz from the smartphone speaker. We repeated this experiment with the jammer set to play at five different volume levels 6, 8, 10, 12 and 15 (in android phones, the volume level ranges from 0 to 15). For this particular layout, with the jamming volume set to 8, i.e. 50 % of output power, the tracking error increased to 64 cm. At volumes higher than that, our system could not detect the motion of the subject. We note a possible extension to this defense: given that CovertBand repeats OFDM symbols, the defending phone app could even spoof locations and activities by transmitting altered signals at comparable amplitudes during expected intervals. In this way, a phone left in a particular location or carried by a potential victim could be set up to do detection and jam/spoof locations.

We note that although these defenses are effective to varying degrees and have different drawbacks, they rely on victims to understand that they may be under attack and take actions to mitigate the harm.

5.6 Discussion

With a proof-of-concept prototype that uses active sonar pulses in the 18-10 kHz range played on commonly available devices that already exist in many homes, we show that an attacker can glean information about *what* a person is doing even when that attacker can neither hear the person nor see his movements. This section outlines our system's limitations as well as future research opportunities.

Achieving a Higher Accuracy and Resolution. One could incorporate phase-based algorithms [131, 206] to achieve a higher resolution than that demonstrated in this chapter. One could also use multiple phones or move the phone along a straight line (potentially in conjunction with the accelerometer) to improve localization and gesture detection. Moving the phone creates a virtual microphone array by taking measurements at different points in space. One could then use tradition angle of arrival algorithms to gain both resolution and accuracy even in the absence of a direct path, enabling the sensing of more subtle motions such as the movement of hands, arms, or even fingers.

Achieving a Higher Range. We evaluate CovertBand's range in a variety of materials, showing that it can track at up to 6 m without barriers and 3 m in through-wall scenarios. While this in itself

is a privacy leakage, further research is required to achieve better range. For example, we currently place the microphone next to the speaker to make it easier to administer experiments. This limits our range because the close proximity of the speaker means we can play only a limited volume before we exceed the microphone's capabilities. However, an attacker could use any method to supply audio. Therefore, it is possible to simply position the microphone far from the speaker, for example, by connecting to it over Bluetooth, allowing the use of significantly higher volumes to increase range. Additionally, an attacker could use longer OFDM symbols and perform correlation-based techniques to decode the minute changes due to human motion at larger distances. As discussed in §5.3.5, a sophisticated attacker could improve CovertBand by using various clever methods to hide the sonar signal. Some of the more subtle methods including tailoring each signal to match the song in cadence and distortive sound. More sophisticated methods may even emulate natural sounds or natural frequencies simulating a car driving, truck moving, or a jackhammer allowing for the use of lower frequency sonar pulses which would permit much higher volumes and better sound propagation on existing commodity speakers.

Tracking More than Two Subjects. Our current implementation is tested to track up to two concurrent subjects. In principle, as long as the subjects are at different distances from the microphones, CovertBand can distinguish more than two subjects. One can further generalize the techniques described in this chapter to leverage more than two microphones, potentially on multiple devices, to achieve higher angular resolution. If the microphones are placed on different sides of the subject, this could also help solve the near-far problem, where one of the subjects is much closer to the microphones and can have much stronger reflections.

Expanding the Set of Activities Classified. Our chapter explores a limited set of activities like pumping arms, jumping, and supine pelvic tilts, which in certain contexts can expose private activities. However, one could explore the use of active sonar to achieve imaging of the environment and a much richer set of activity recognition. At a high level, if we can combine the acoustic reflections from multiple devices spread across a home, one can start creating images of the environment with a higher resolution.

Identifying Individual Subjects. Our implementation can distinguish between users in different

locations, but it cannot identify them. Therefore, it cannot continue to track a particular person if two people move to the same location and then separate. Similarly, CovertBand can not currently differentiate between movement caused by different objects, such as a dog, a fan, or a human. However, there may be differences in the types of motion caused by each. For example, prior work has demonstrated that gait information can be used to identify human subjects. In principle, one could extract gait information from the acoustic reflections and achieve subject identification. Similar work has been done in the RF space [187, 202]. One could imagine that these techniques could generalize to classify certain movements as non-human or as generated by a particular object, like a fan.

Generalizing to More Devices CovertBand could also adapt to instances where an attacker has access to only a single microphone by using multiple speakers. A system with stereo speakers and a single microphone would still create two ellipses which share a focus at the microphone. For example, a smart television with a single microphone will almost assuredly have stereo sound; this could be used to track motion in a 2-D plane.

5.7 Songs Used

See §5.3.5 for details: American Woman - Lenny Kravitz, Bad - Michael Jackson, Barbara Streisand - Duck Sauce, Baby Don't Cry - 2pac, Five Hours - Deorro, What Goes Around Comes Around - Justin Timberlake, Guerilla Radio - Rage Against The Machine, Ice Ice Baby - Vanilla Ice, O.P.P. - Naughty By Nature, Revolutionary Warfare - Nas, Rockefeller Skank - Fatboy Slim, Save Me San Francisco - Train, Hey, Soul Sister - Train, Spoils - Protest the Hero, Uptown Funk - Bruno Mars, What Would You Do? - City High

Speakers Used (See §5.3.4): JBL [100], Beats [45], Auvio [43], Bose [48], Bose Acoustimass Home Theater [89]

Chapter 6

CONTRIBUTIONS TO WIRELESS SENSING AND LIMITATIONS

The concept of sonar is not new and have been traditionally used in underwater communication. Sonar imaging has also been used extensively in underwater settings [164, 92, 29]. However the transmission properties of sound in water medium is very different and requires bulky custom hardware. Above water, acoustic transmissions have been used extensively to localize devices in systems such as Cricket [140]. [42, 174, 91, 206] systems have demonstrated the feasibility of localizing and determining the direction of device movement using acoustic transmissions. Unlike these prior work that requires infrastructure deployment, we demonstrate the feasibility of tracking users without instrumenting them with any devices, e.g., device-free sensing. Our work demonstrates the ability, for the first time, to achieve millimeter level tracking of respiratory motion as well as fine finger motion using the reflection of acoustic waves.

This dissertation proposes active sonar based wireless sensing systems that can enable contactless motion tracking with millimeter level precision on commercial off-the-shelf devices. This ability to track minute motion can enable various applications across different domains including mobile health, user interaction and IoT devices. First, we showed that we can transform off-the-shelf smartphone devices into active sonar systems that can detect minute breathing motion. We employed custom FMCW signals to distinguish motion that occurs at different distances from the device. Following this, we developed FingerIO, a fine grained 2D finger tracking system that can work even in the presence of occlusions. In this work, we prove that active sonar based systems can not only detect minute motion but can also localize and track the motion in a 2D space. We achieved this by adopting custom OFDM modulation that is commonly used in WiFi transmissions. In our work on opioid overdose detection, we address the effect of motion in the environment to the accuracy of the wireless sensing system. As opposed to a sleeping subject in the ApneaApp

scenario, here the subjects were present in a dynamic environment and were also awake and moving. To address this challenge, we designed re-calibration algorithms to handle changes in the distance of the subject from the smartphone. So the sonar based wireless sensing system can be used in a crowded environment and can adjust to the motion of the subject. Finally in Chapter 5, we increased the range of the sensing system by using off-the-shelf smart speakers that have a higher output power than the smartphone speaker. We showed that we can increase the range of the system to 6 m in normal settings and 3 m when the transmission is through-the-wall.

Comparison to RF-based wireless sensing systems. Prior work has proposed radio-based solutions for human motion detection [148, 36, 34], localization [148, 34, 62], and gesture recognition [143, 106, 132]. While promising, none of the RF-based designs have been demonstrated to work on off-the-shelf smartphones. Specifically, these designs use expensive, ultra-wideband transceiver and/or specialized hardware that are not available on mobile devices. Further, they typically require multiple antennas separated by half a wavelength, which is difficult on smartphones due to their size constraints. In particular, [148, 34, 62] use radar hardware that transmits, receives and processes 500 MHz to 3 GHz wideband signals and requires multiple antennas. Researchers have recently proposed ultra-wideband radar designs that operate in the millimeter wave [211] and terahertz bands of the electromagnetic spectrum [83, 176] where the wavelengths are significantly smaller, permitting multiple antennas to be packed together. Further, it remains to be seen whether the power and processing required for such wideband signals at the sub-10 GHz frequencies could be achieved on smartphone-grade consumer devices.

Recent work also leverages Wi-Fi for human motion detection and gesture recognition. [143] extracts Doppler shifts on RF transmissions to perform gesture recognition in through-wall scenarios. [36] does human motion detection (walking forward and backward) in through-wall scenarios using 20 MHz wide transmissions but requires specialized interference cancellation hardware that is not available on commodity devices. [54] uses specialized, full-duplex hardware to track finger strokes using RF signals. However, these require custom hardware processing (e.g., software radios) and do not work with commodity devices. Wi-Fi gestures [132] can enable recognition of

a pre-defined set of gestures in the vicinity of an Intel Wi-Fi chipset. WiDraw [173] tracks arm motion using transmissions from 20-30 other Wi-Fi devices. These systems work only when the user is close to the Wi-Fi chipset and have not been demonstrated in through-barrier scenarios.

6.1 Limitations and Discussion

While our active sonar based wireless sensing systems achieve mm-level motion tracking resolution on existing off-the-shelf devices, they also have some limitations due to the property of the sound signals and the limitations of today's devices. This section outlines these limitations as well as future research opportunities.

Achieving a Higher Range. Active sonar based wireless sensing systems are typically designed to achieve high accuracy while operating at small ranges. Currently the operating range for the systems running on a smartphone device is 1 m. For Covertband, which uses a smart speaker, the working range of the system is 6 m when there are no barriers and 3 m in through the wall scenarios. This operating range is low compared to RF based systems [149, 147] where the RF signals propagate upto 30 m and can easily travel through walls.

Further the accuracy of the sonar system reduces as the range increases. For distances upto 1 m, the system can detect minute mm-level breathing and finger motion. However at larger distances, the resolution decreases to tens of centimeters. This is because i) the output power of a smartphone speaker for the high inaudible frequencies used in these systems is low and it cannot travel large distances. The acoustic sensors in these devices are built for audible speech and music whose frequencies are $< 10\ Khz$, ii) the speaker and microphone in off-the-shelf devices only have a 16 bit resolution compared to RF sensors that have 32 bits. Hence it is difficult to distinguish the lower SNR signals that arrive from larger distances to the microphone.

While the output power of the speaker can be increased by using external speakers to increase the range of the system, the solution is not straightforward. For example, if the microphone is placed next to the speaker, higher volume signal will overwhelm the 16 bit microphone. Hence the placement of the sensors will also be a key factor in enabling larger ranges. Another solution is to employ sophisticated coding schemes to the signal to enable decoding of lesser SNR signals.

Distinguishing multiple co-located motions. Our system can distinguish multiple motions in the environment as long as they are separated by a distance of at least 20 cm. We use custom modulated signals such as FMCW and OFDM to distinguish motion that happens at different distances. However our system cannot differentiate two motions that happen at a closer distance. For example, when a person moves their hand, this large motion overwhelms the nearby chest motion and hence it becomes difficult to detect the breathing motion. Similarly in FingerIO, it is difficult to distinguish the motion between two adjacent fingers. This is because, the algorithms assume that the motion occurs due to a point source. However both the chest and the finger are not point sources and the motion happens throughout the length of the finger and the chest. The algorithm addresses this by considering the closest point to the device. However any other motion that happens within the span of the finger or chest length will be affected and hence cannot be distinguished. In the future, this problem can potentially be addressed by adding more microphones and enabling beam forming techniques to focus on specific points. With the trend of increasing microphones in smart devices, it is becoming more feasible to implement these solutions in off-the-shelf devices such as home assistants that have 7-8 microphones.

Effect of smartphone motion. All the four systems discussed in this paper assume the smartphone with the acoustic sensors to be in a static position, i.e., the location of the device along with its speaker and the microphone remains static. This is a key assumption because all the distances measured in these systems are relative to the location of the speaker and the microphone and any change in the position changes the location axes of the system. For example, we do not address a scenario where the device is present in a user's hand which is constantly shaking. However this can potentially be solved by leveraging the phone's inertial sensors. The challenge is that the output of inertial sensors such as accelerometer and gyroscope are typically very noisy and the distance computation involves a double integration leading to fast accumulation of error.

Security and privacy. Finally the major concern of active sonar based wireless sensing systems is security and privacy. These systems enable contactless tracking of minute mm-level motion such as the breathing motion without requiring any contact or initiation from the subject. Further the signals used in these systems are mostly inaudible to adults and hence it is difficult to detect any

attack by an adversary. In our work Corvertband, we also discuss a scenario where a malicious third party smartphone app can execute a remote attack on a subject without their knowledge. The apps does not require any explicit permission from the user to turn on the speaker and hence can play the inaudible sound signals. While the user typically needs to give permission to access the microphone on a smartphone app, it has been shown that these devices can be hacked [13]. Also an app getting microphone access from the user for a video/audio chat can now potentially monitor the subject's breathing without their knowledge. While the limited range of the sonar system inherently provides some security from attackers at large ranges, this issue is largely under explored.

Chapter 7

CONCLUSION

In conclusion, this dissertation is about enabling computational wireless sensing at scale. Traditionally wireless sensors are used for communication. In this dissertation, we take a novel approach and show that we can leverage the existing infrastructure of billions of mobile devices that have multiple acoustic sensors to enable new wireless sensing capabilities. Specifically we develop algorithms that can transform these devices into active sonar systems and detect minute millimeter-level motion in the environment. Using these algorithms, we were able to design and build practical applications across two different domains: mobile health and user interfaces. In addition to building these systems, we also deploy and evaluate them in the real world. Specifically, we built four systems to demonstrate the sensing technologies.

- *Contactless sleep apnea detection.* In contrast to the current clinical diagnostic test that is cumbersome, we presented ApneaApp which is the first contactless system that detects sleep apnea using just an off-the-shelf smartphone. It enables this by tracking the fine-grained breathing movements from multiple subjects without requiring any contact with the phone.

- *Opioid Overdose detection using smartphones.* We designed a novel adaptive tracking algorithm for commodity smartphones that unobtrusively recognize opioid overdose by detecting the respiratory distress that occurs as its pre-cursor. This smartphone-enabled overdose detection system, capable of alerting naloxone-equipped friends and family or Emergency Medical Services (EMS), may hold potential as a low-barrier, harm reduction intervention.

- *Fine grained finger tracking.* We designed the first active sonar based fine-grained 2D finger tracking system that does not require instrumenting the finger with sensors and works even

with occlusions between the finger and the device. This capability is also beneficial for smaller devices such as smart watches that could track the user's finger, even when fully occluded from it or when the watch is on a different plane from the interaction surface.

- *Activity information leakage using music.* Finally, we explore the privacy implication of enabling contactless sensing on commodity devices. We presented CovertBand that enables high frequency acoustic signals into the beats of popular songs to simultaneously track multiple users through barriers like walls, doors and windows.

7.1 Looking Forward

The past few years has seen a rapid proliferation of smart devices such as smartphones, smart watches and home assistants. These connected smart devices equipped with multiple wireless sensors enable a rich infrastructure that can be leveraged to enable end-to-end systems across different domains that can create a huge impact in the society. This dissertation on using wireless technologies to enable applications in the health and HCI space is only a beginning and the next few years are going to be exciting because of the diverse systems that can be enabled ranging from sonar vision, the science fiction concept showed in the Batman movie to the detection of life threatening medical conditions in the health domain. Specifically, the detection of the physiological signals can help in early diagnoses of many diseases including cardio-vascular conditions. Further, these physiological signals can be used to make psychological inferences about a subject. By scaling these systems to crowd sourcing platforms like Amazon Mechanical Turk, we can potentially transform the way psychological studies are performed by enabling remote studies with billions of users.

While these technologies are intended to improve the quality of life of people, they could also enable significant privacy risks either inadvertently or when used with malicious intents. As engineers come up with creative new uses for these existing sensors, ensuring security and privacy for the users is still an open problem and needs to be addressed by enabling novel defense mechanisms and revisiting the privacy policies. Enabling these technologies in a secure and privacy preserving fashion is a necessity to fully benefit from the potential of these inventions.

BIBLIOGRAPHY

[1] Anesthesiology and Respiratory Devices Branch, Office of Device Evaluation. Class II Special Controls Guidance Document: Apnea Monitors; Guidance for Industry and FDA. 2011.

[2] IOS accelerate framework. https://developer.apple.com/documentation/accelerate.

[3] Fast FFT library for android. http://www.fftw.org/#documentation.

[4] Rapid Response Systems; https://psnet.ahrq.gov/primers/primer/4/rapid-response-systems; accessed February 14th 2018.

[5] Health Canada. Drug Situation Summary Report – British Columbia 4th Quarter (accessed april 2nd).

[6] QGIS Development Team (2018). QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org.

[7] Adafruit. https://www.adafruit.com/products/1063.

[8] Apple watch - guided tour: Phone calls. https://www.youtube.com/watch?v=_Zj5KisMVv8.

[9] Bam labs, sleep iq only from sleep number. http://bamlabs.com.

[10] Chirp microsystems. http://www.chirpmicro.com/technology.html.

[11] Fitbit. www.fitbit.com.

[12] Google updates the nest hub to sense where you are without a camera. https://www.engadget.com/2019/11/06/google-nest-hub-max-ultrasound-sensing-now-available/.

[13] Hacking an apple iphone within minutes. https://www.forbes.com/sites/kateoflahertyuk/2020/01/10/google-reveals-how-to-hack-an-apple-iphone-within-minutes/#6e9d7d9816c1.

[14] Home sleep apnea a-z. https://play.google.com/store/apps/details?id=com.app_aviisha123.layout&hl=en.

[15] Iphone app: Respiratory rate. https://itunes.apple.com/us/app/respiratory-rate/id538726900?mt=8.

[16] Jawbone. https://jawbone.com/.

[17] A mimioteach interaction whiteboard. http://www.mimio.com/en-NA/Products/MimioTeach-Interactive-Whiteboard.aspx.

[18] Morbidity and mortality weekly report,centers for disease control and prevention. In Weekly, Vol.60, No. 8, 2011.

[19] Nose breathe mouthpiece. http://www.nosebreathe.com/.

[20] Sleep access. http://sleepaccess.com/.

[21] Sleep apnea: Epidemiology. http://www.sleepdex.org/epidemiologyapnea.htm.

[22] Sleep apnea monitor. https://itunes.apple.com/gb/app/sleep-apnea-monitor/id464587229?

[23] Sleep as android: Snoring detection. https://sites.google.com/site/sleepasandroid/.

[24] Sleepscore. https://www.sleepscore.com/sleepscore-app/.

[25] Snuza - portable baby monitors. http://www.snuza.com/.

[26] Vernier respiration monitor belt. http://www.vernier.com/products/sensors/rmb/.

[27] Wakemate. https://play.google.com/store/apps/details?id=com.wakemate.android.

[28] Health: A safe place. a new york times, published february 8, 2011, accessed, 2011. https://youtu.be/v6NJPjpelKM.

[29] Videoray rov sonar, videoray imaging sonar, 2016. http://www.videoray.com/homepage/options/sonar/blueview-imaging-sonars.html.

[30] Provisional counts of drug overdose deaths, as of 8/6/2017, 2017. National Center for Health Statistics and others.

[31] U Abeyratne, S Silva, C Hukins, and B Duce. Obstructive sleep apnea screening by integrating snore feature classes. In Physiological Measurement, 2013.

[32] Acoustic localization through wall/ceiling/floor, 2017. https://redecomposition.wordpress.com/acousticvideo/.

[33] Jerome M Adams. Increasing naloxone awareness and use: The role of health care practitioners. JAMA, 2018.

[34] Fadel Adib, Za adelec, Dina Katabi, and Rob Miller. 3d localization via human body reflections. In NSDI, 2014.

[35] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. Capturing the human figure through a wall. ACM Trans. Graph., 34:219:1–219:13, 2015.

[36] Fadel Adib and Dina Katabi. Seeing Through Walls Using WiFi! In SIGCOMM, 2013.

[37] HO Ahmed, JH Dennis, O Badran, M Ismail, SG Ballal, A Ashoor, and D Jerwood. High-frequency (10–18 khz) hearing thresholds: reliability, and effects of age and occupational noise exposure. Occupational Medicine, 51(4):245–258, 2001.

[38] M.A. Al-Abed, P. Antich, D.E. Watenpaugh, and K. Behbehani. Detection of airway occlusion in simulated obstructive sleep apnea/hypopnea using ultrasound: An in vitro study. In Engineering in Medicine and Biology Society (EMBC), 2013.

[39] Farah Q AL-Khalidi, Reza Saatchi, Derek Burke, H Elphick, and Stephen Tan. Respiration rate monitoring methods: A review. Pediatric Pulmonology, 46(6):523–529, 2011.

[40] S. Alqassim, M. Ganesh, S. Khoja, M. Zaidi, F. Aloul, and A. Sagahyroon. Sleep apnea monitoring using mobile phones. In e-Health Networking, Applications and Services (Healthcom), 2012.

[41] Amprobe sound power meter, 2016. http://www.amprobe.com/amprobe/usen/environmental-test/sound/amp-sm-10.htm?PID=73334.

[42] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. Doplink: Using the doppler effect for multi-device interaction. *UbiComp 2013*, 583-586.

[43] Auvio Portable Bluetooth Speaker, 2016. http://www.amazon.com/Portable-Bluetooth-Speaker-PBT1000-Gadgets/dp/B00D2D46S4.

[44] Caleb J Banta-Green, Phillip O Coffin, Joseph O Merrill, Jeanne M Sears, Chris Dunn, Anthony S Floyd, Lauren K Whiteside, Norbert D Yanez, and Dennis M Donovan. Impacts of an opioid overdose prevention intervention delivered subsequent to acute care. Injury Prevention, pages injuryprev–2017, 2018.

[45] Beats Pill, 2016. http://www.beatsbydre.com/beatspill.html.

[46] Daniel Belz, Jacob Lieb, Tom Rea, and Mickey S Eisenberg. Naloxone use in a tiered-response emergency medical services system. Prehospital Emergency Care, 10(4):468–471, 2006.

[47] H Boleskei. Principles of mimo-ofdm wireless systems. *2004*.

[48] Bose Sound Link 2, 2016. https://www.bose.com/products/speakers/wireless_speakers/soundlink_mini_ii.html.

[49] Edward W Boyer. Management of opioid analgesic overdose. New England Journal of Medicine, 367(2):146–155, 2012.

[50] Andreas Braun, Stefan Krepp, and Arjan Kuijper. Acoustic tracking of hand activities on surfaces. *WOAR 2015*, 1-5.

[51] Alex Butler, Shahram Izadi, and Steve Hodges. Sidesight: Multi-"touch" interaction around small devices. *UIST 2008*, 201-204.

[52] Iber C, Ancoli-Israel S, Chesson AL Jr., and Quan SF. The aasm manual for the scoring of sleep and associated events. In 1st ed. Westchester IL: American Academy of Sleep Medicine, 2007.

[53] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. Fingerpad: Private and subtle interaction using fingertips. *UIST 2013*, 255-260.

[54] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. Tracking keystrokes using wireless signals. In Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, 2015.

[55] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. Airlink: Sharing files between multiple devices using in-air gestures. *UbiComp 2014*, 565-569.

[56] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. utrack: 3d input using two magnetic sensors. *UIST 2013*, 237-244.

[57] Zhenyu Chen, Mu Lin, Fanglin Chen, Nicholas D. Lane, Giuseppe Cardone, Rui Wang, Tianxing Li, Yiqiang Chen, Tanzeem Choudhury, and Andrew T. Campbell. Unobtrusive sleep monitoring using smartphones. In PervasiveHealth 2013.

[58] Ivan Cowie. Through-wall surveillance for locating individuals within buildings. Technical report, Time Domain Corporation and United States of America, 2008. Final Scientific and Technical Report.

[59] Shane Darke, Joanne Ross, D Zador, and S Sunjic. Heroin-related deaths in new south wales. Australia, 1996:141–150, 1992.

[60] Peter J Davidson, Andrea M Lopez, and Alex H Kral. Using drugs in un/safe spaces: Impact of perceived illegality on an underground supervised injecting facility in the united states. International Journal of Drug Policy, 53:37–44, 2018.

[61] Peter J Davidson, Kristen C Ochoa, Judith A Hahn, Jennifer L Evans, and Andrew R Moss. Witnessing heroin-related overdoses: the experiences of young injectors in san francisco. Addiction, 97(12):1511–1516, 2002.

[62] Fadel Ddib, Zachary Kabelac, and Dina Katabi. Multi-Person Localization via RF Body Reflections. In NSDI, 2015.

[63] Wallace JM Deutsch PA, Simmons MS. Cost-effectiveness of split-night polysomnography and home studies in the evaluation of obstructive sleep apnea syndrome. In J Clin Sleep Med 2006;2:145-53.

[64] Michael A DeVita, Rinaldo Bellomo, Kenneth Hillman, John Kellum, Armando Rotondi, Dan Teres, Andrew Auerbach, Wen-Jon Chen, Kathy Duncan, Gary Kenward, et al. Findings of the first consensus conference on medical emergency teams. Critical Care Medicine, 34(9):2463–2478, 2006.

[65] Maya Doe-Simkins, Alexander Y Walley, Andy Epstein, and Peter Moyer. Saved by the nose: bystander-administered intranasal naloxone hydrochloride for opioid overdose. American Journal of Public Health, 99(5):788–791, 2009.

[66] Ivan Dokmanića, Reza Parhizkara, Andreas Walthera, Yue M. Lub, and Martin Vetterlia. Acoustic echoes reveal room shape. In Proceedings of the National Academy of Sciences, volume 110, pages 12186–12191, 2013.

[67] Deborah Dowell, Rita K Noonan, and Debra Houry. Underlying factors in drug overdose deaths. JAMA, 318(23):2295–2296, 2017.

[68] Amazon Echo, 2016. http://www.amazon.com/Amazon-SK705DI-Echo/dp/B00X4WHP5E.

[69] W. Ward Flemons, Neil J. Douglas, Samuel T. Kuna, Daniel O. Rodenstein, and John Wheatley. Access to diagnosis and treatment of patients with suspected sleep apnea. In American Journal of Respiratory and Critical Care Medicine, Vol. 169, No. 6 (2004).

[70] Flir one thermal imaging, 2016. http://www.flir.com/flirone/explore.cfm.

[71] Mia Folke, L Cernerud, M Ekström, and Bertil Hök. Critical review of non-invasive respiratory monitoring in medical care. Medical and Biological Engineering and Computing, 41(4):377–383, 2003.

[72] European Monitoring Centre for Drugs and Drug Addiction (EMCDDA). Preventing fatal overdoses: a systematic review of the effectiveness of take-home naloxone. 2015.

[73] N.A. Fox, C. Heneghan, M. Gonzalez, R.B. Shouldice, and P. de Chazal. An evaluation of a non-contact biomotion sensor with actimetry. In Engineering in Medicine and Biology Society, 2007.

[74] Richard G Frank and Harold A Pollack. Addressing the fentanyl threat to public health. New England Journal of Medicine, 376(7):605–607, 2017.

[75] Robert L Frank, Mark A Rausch, James J Menegazzi, and Myron Rickens. The locations of nonresidential out-of-hospital cardiac arrests in the city of pittsburgh over a three-year period: implications for automated external defibrillator placement. Prehospital Emergency Care, 5(3):247–251, 2001.

[76] Taeko M Frost. An investigation of public injection drug use in new york city: A mixed-methods study. 2017.

[77] A Gaucher, D Frasca, O Mimoz, and B Debaene. Accuracy of respiratory rate monitoring by capnometry using the capnomask® in extubated patients receiving supplemental oxygen after surgery. British Journal of Anaesthesia, 108(2):316–320, 2012.

[78] Rebecca E Giglio, Guohua Li, and Charles J DiMaggio. Effectiveness of bystander naloxone administration and overdose education programs: a meta-analysis. Injury Epidemiology, 2(1):10, 2015.

[79] Mayank Goel, Brendan Lee, Md. Tanvir Islam Aumi, Shwetak Patel, Gaetano Borriello, Stacie Hibino, and Bo Begole. Surfacelink: Using inertial and acoustic sensing to enable multi-device interaction on a surface. *CHI 2014*, 1387-1396.

[80] Google. Project soli. https://www.youtube.com/watch?v=_Zj5KisMVv8.

[81] Traci C Green, Robert Heimer, and Lauretta E Grau. Distinguishing signs of opioid overdose and indication for naloxone: an evaluation of six overdose training and naloxone distribution programs in the united states. Addiction, 103(6):979–989, 2008.

[82] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. Soundwave: Using the doppler effect to sense gestures. *CHI 2012*, 1911-1914.

[83] Ruonan Han, Yaming Zhang, Youngwan Kim, Dae Yeon Kim, H. Shichijo, E. Afshari, and O. Kenneth. 280GHz and 860GHz image sensors using Schottky-barrier diodes in 0.13 um digital CMOS. In ISSCC, 2012.

[84] Tian Hao, Guoliang Xing, and Gang Zhou. ibreath: A convenient mobile app that tracks breathing during running.

[85] Tian Hao, Guoliang Xing, and Gang Zhou. isleep: Unobtrusive sleep quality monitoring using smartphones. In SenSys '13.

[86] J Heiskala and J Terry. Ofdm wireless lans: A theoretical and practical guide. *Sams publishing, 2001*.

[87] Jerome R Hoffman, David L Schriger, and John S Luo. The empiric use of naloxone in patients with altered mental status: a reappraisal. Annals of emergency medicine, 20(3):246–252, 1991.

[88] Google Home, 2017. https://madeby.google.com/home/.

[89] Costco: Bose Acoustimass 10 Series V Home Theater Onkyo Bundle, 2016. http://www.costco.com/Bose%C2%AE-Acoustimass-10-Series-V-Home-Theater-Onkyo-Bundle.product.100147047.html.

[90] Wenchao Huang, Yan Xiong, Xiang-Yang Li, Hao Lin, Xufei Mao, Panlong Yang, and Yunhao Liu. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. *INFOCOM 2014*, 370-278.

[91] Wenchao Huang, Yan Xiong, Xiang-Yang Li, Hao Lin, XuFei Mao, Panlong Yang, and Yunhao Liu. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In INFOCOM, 2014.

[92] Humming bird side imaging, 2016. http://www.humminbird.com/Category/Technology/Side-Imaging/.

[93] Hunted: The War Against Gays in Russia, 2016. http://www.hbo.com/documentaries/hunted-the-war-against-gays-in-russia.

[94] Emily Hurstak, Christopher Rowe, Caitlin Turner, Emily Behar, Rachel Cabugao, Nikolas P Lemos, and Phillip Coffin. Using medical examiner case narratives to improve opioid overdose surveillance. International Journal of Drug Policy, 54:35–42, 2018.

[95] IEEE. Ieee 802.11g-2003: Further higher data rate extension in the 2.4 ghz band. In Standard, 2003.

[96] Amazon Echo Teardown, 2017. https://www.ifixit.com/Teardown/Amazon+Echo+Teardown/33953.

[97] Man killed on suspicion of incest. http://timesofindia.indiatimes.com/city/ahmedabad/Man-killed-on-suspicion-of-incest/articleshow/35271666.cms, 2016.

[98] Standard Inside Door Sizes, 2017. http://homeguides.sfgate.com/standard-inside-door-sizes-84805.html.

[99] Building Requirements for Partition Walls, 2017.

[100] JBL Flip 2, 2016. http://www.jbl.com/bluetooth-speakers/JBL+FLIP+II.html.

[101] Masa JF, Corral J, and et al. Pereira R. Effectiveness of home respiratory polygraphy for the diagnosis of sleep apnoea and hypopnoea syndrome. In Thorax 2011;66:567-73.

[102] Christopher M Jones, Emily B Einstein, and Wilson M Compton. Changes in synthetic opioid involvement in drug overdose deaths in the united states, 2010-2016. Jama, 319(17):1819–1821, 2018.

[103] Kiran Raj Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. Wideo: Fine-grained device-free motion tracing using rf backscatter. In NSDI, 2015.

[104] Matthew Kay, Eun Kyoung Choe, Jesse Shepherd, Benjamin Greenstein, Nathaniel Watson, Sunny Consolvo, and Julie A. Kientz. Lullaby: A capture & access system for understanding the sleep environment. In UbiComp '12.

[105] Andjelka Kelic, Munaf Syed Aamir, Ahmad M Jrad, and Roger Mitchell. A generalized framework for modeling next generation 911 implementations. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2018.

[106] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. Bringing gesture recognition to all devices. In NSDI, 2014.

[107] Thomas Kerr, Jo-Anne Stoltz, Mark Tyndall, Kathy Li, Ruth Zhang, Julio Montaner, and Evan Wood. Impact of a medically supervised safer injection facility on community drug use patterns: a before and after study. BMJ: British Medical Journal, 332(7535):220, 2006.

[108] Thomas Kerr, Mark W Tyndall, Calvin Lai, Julio SG Montaner, and Evan Wood. Drug-related overdoses within a medically supervised safer injection facility. International Journal of Drug Policy, 17(5):436–441, 2006.

[109] M.O. Khyam, M.J. Alam, A.J. Lambert, C.R. Benson, and M.R. Pickering. High precision multiple ultrasonic transducer positioning using a robust optimization approach. *IS-SPIT 2013*, 192-197.

[110] M.O. Khyam, M.J. Alam, and M. Pickering. Ofdm based low-complexity time of arrival estimation in active sonar. *OCEANS 2014*, 1-5.

[111] Wolf Kienzle and Ken Hinckley. Lightring: Always-available 2d input on any surface. *UIST 2014*, 157-160.

[112] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. *UIST 2012*, 167-176.

[113] Manikanta Kotaru, Kiran Raj Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. Computer Communication Review, 45:269–282, 2015.

[114] Sven Kratz and Michael Rohs. Hoverflow: Expanding the design space of around-device interaction. *MobileHCI 2009*, 1-8.

[115] Clete A. Kushida, Alejandro Chediak, Richard B. Berry, Lee K. Brown, David Gozal, Conrad Iber, Sairam Parthasarathy, Stuart F. Quan, and James A. Rowley. Clinical guidelines for the manual titration of positive airway pressure in patients with obstructive sleep apnea. Journal of Clinical Sleep Medicine.

[116] Y Lahav, E Rosenzweig, Z Heyman, J Doljansky, A Green, and Y Dagan. Tongue base ultrasound: a diagnostic tool for predicting obstructive sleep apnea. In Ann Otol Rhinol Laryngol, 2009.

[117] Qiang Lin and Yuan Yue. Device-free passive human detection using wi-fi technology: Current state and future trend. 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), pages 1717–1723, 2015.

[118] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. Snooping keystrokes with mm-level audio ranging on a single phone. *MobiCom 2015*, 142-154.

[119] An Introduction to LTE, 3GPP LTE Encyclopedia, 2016. `https://sites.google.com/site/lteencyclopedia/home`.

[120] MacNeish. The intersections of two conic sections with a common focus. The American Mathematical Monthly, 28(6/7):260–262.

[121] Wenguang Mao, Jian He, and Lili Qiu. Cat: high-precision acoustic motion tracking. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, pages 69–81, 2016.

[122] Erin A McClure, Shauna P Acquavita, Emily Harding, and Maxine L Stitzer. Utilization of communication technology by patients enrolled in substance abuse treatment. Drug & Alcohol Dependence, 129(1):145–150, 2013.

[123] Rebecca McDonald and John Strang. Are take-home naloxone programmes effective? systematic review utilizing application of the bradford hill criteria. Addiction, 111(7):1177–1187, 2016.

[124] Ronald D Miller, Lars I Eriksson, Lee A Fleisher, Jeanine P Wiener-Kronish, Neal H Cohen, and William L Young. Miller's Anesthesia E-Book. Elsevier Health Sciences, 2014.

[125] Joanna Milward, Edward Day, Elle Wadsworth, John Strang, and Michael Lynskey. Mobile phone ownership, usage and readiness to use by patients in drug treatment. Drug & Alcohol Dependence, 146:111–115, 2015.

[126] Jun-Ki Min, Afsaneh Doryab, Jason Wiese, Shahriyar Amini, John Zimmerman, and Jason I. Hong. Toss 'n' turn: Smartphone as sleep and sleep quality detector. In CHI '14.

[127] Smartphone ultrasound: The mobius sp1 system, 2016. http://www.mobisante.com/products/product-overview/.

[128] Rajalakshmi Nandakumar, Krishna Chinatalapudi, Venkat Padmanaban, and Ramarathnam Venkatesan. Dhwani : Secure peer-to-peer acoustic nfc. *Sigcomm 2013* 2013.

[129] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Jacob E Sunshine. Opioid overdose detection using smartphones. Science translational medicine, 11(474):eaau8914, 2019.

[130] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. Contactless sleep apnea detection on smartphones. *Mobisys 2015*, 45-57.

[131] Rajalakshmi Nandakumar, Vikram Iyer, Shyamnath Gollakota, and Desney Tan. Fingerio: Fine-grained finger tracking using active sonar. In CHI, 2016.

[132] Rajalakshmi Nandakumar, Bryce Kellogg, and Shyamnath Gollakota. Wi-fi gesture recognition on existing devices. CoRR, abs/1411.5394, 2014.

[133] Rajalakshmi Nandakumar, Alex Takakuwa, Tadayoshi Kohno, and Shyamnath Gollakota. Covertband: Activity information leakage using music. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(3):1–24, 2017.

[134] MB Norman, S Middleton, O Erskin, PG Middleton, JR Wheatley, and CE Sullivan. Validation of the sonomat: a contactless monitoring system used for the diagnosis of sleep disordered breathing. In Sleep, Sep 2014.

[135] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. iring: Intelligent ring using infrared reflection. *UIST 2012*, 131-136.

[136] Julie K O'Donnell, R Matthew Gladden, and Puja Seth. Trends in deaths involving heroin and synthetic opioids excluding methadone, and law enforcement drug product reports, by census region—united states, 2006–2015. MMWR. Morbidity and mortality weekly report, 66(34):897, 2017.

[137] N. Patwari, L. Brewer, Q. Tate, O. Kaltiokallio, and M. Bocca. Breathfinding: A wireless network that monitors and locates breathing in a home. Selected Topics in Signal Processing, IEEE Journal of, 2014.

[138] N. Patwari, J. Wilson, S. Ananthanarayanan, S.K. Kasera, and D.R. Westenskow. Monitoring breathing via signal strength in wireless networks. Mobile Computing, IEEE Transactions on, 2014.

[139] Philip WH Peng and Alan N Sandler. A review of the use of fentanyl analgesia in the management of acute pain in adults. Anesthesiology: The Journal of the American Society of Anesthesiologists, 90(2):576–599, 1999.

[140] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. *Mobicom 2000*, 32-43.

[141] J Proakis and M Salehi. Digital communications. *McGraw-hill, 2007.*

[142] R.J. Przybyla, Hao-Yen Tang, A. Guedes, S.E. Shelton, D.A. Horsley, and B.E. Boser. *IEEE Journal of Solid-State Circuits 2015*, 320-334.

[143] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-Home Gesture Recognition Using Wireless Signals. In MOBICOM, 2013.

[144] Through-the-wall sensors for law enforcement, October 2012. `https://www.justnet.org/pdf/00-WallSensorReport-508.pdf`.

[145] Rafael Golpe and Antonio Jiménez and Rosario Carpizo. Home sleep studies in the assessment of sleep apnea/ hypopnea syndrome. In CHEST, 2002.

[146] Tauhidur Rahman, Alexander T. Adams, Mi Zhang, Erin Cherry, Bobby Zhou, Huaishu Peng, and Tanzeem Choudhury. Bodybeat: A mobile system for sensing non-speech body sounds. In MobiSys '14.

[147] T.S. Ralston, G.L. Charvat, and J.E. Peabody. Real-time through-wall imaging using an ultrawideband multiple-input multiple-output (mimo) phased array radar system. In Phased Array Systems and Technology (ARRAY), 2010.

[148] T.S. Ralston, G.L. Charvat, and J.E. Peabody. Real-time through-wall imaging using an ultrawideband multiple-input multiple-output (MIMO) phased array radar system. In ARRAY, 2010.

[149] S. Ram and H Ling. Through-wall tracking of human movers using joint doppler and array processing. In Geoscience and Remote Sensing, 2008.

[150] V.G. Reju, A.W.H. Khong, and A.B. Sulaiman. Localization of taps on solid surfaces for human-computer touch interfaces. *IEEE Trans. on Multimedia 2013*, 1365-1376.

[151] Ren, Yanzh and Wang, Chen and Chen, Yingying and Yang, Jie. Poster: Hearing your breathing: fine-grained sleep monitoring using smartphones. In MOBICOM Posters, 2014.

[152] Franziska Roesner, Tadayoshi Kohno, Alexander Moshchuk, Bryan Parno, Helen J. Wang, and Crispin Cowan. User-driven access control: Rethinking permission granting in modern operating systems. 2012 IEEE Symposium on Security and Privacy, pages 224–238, 2012.

[153] Rose A Rudd. Increases in drug and opioid-involved overdose deaths - united states, 2010–2015. MMWR. Morbidity and mortality weekly report, 65, 2016.

[154] Ahmed Saeed, Ahmed E. Kosba, and Moustafa Youssef. Ichnaea: A low-overhead robust wlan device-free passive localization system. IEEE Journal of Selected Topics in Signal Processing, 8:5–15, 2014.

[155] T. Scott Saponas, Chris Harrison, and Hrvoje Benko. Pockettouch: Through-fabric capacitive touch input. *UIST 2011*, 303-308.

[156] Robert P Schwartz, Jan Gryczynski, Kevin E O'Grady, Joshua M Sharfstein, Gregory Warren, Yngvild Olsen, Shannon G Mitchell, and Jerome H Jaffe. Opioid agonist treatments and heroin overdose deaths in baltimore, maryland, 1995–2009. American Journal of Public Health, 103(5):917–922, 2013.

[157] Karen H Seal, Moher Downing, Alex H Kral, Shannon Singleton-Banks, Jon-Paul Hammond, Jennifer Lorvick, Dan Ciccarone, and Brian R Edlin. Attitudes about prescribing take-home naloxone to injection drug users for the management of heroin overdose: a survey of street-recruited injectors in the san francisco bay area. Journal of Urban Health, 80(2):291–301, 2003.

[158] Seek thermal camera review: Smartphone thermal vision in a tiny package, 2016. `http://www.cnet.com/products/seek-thermal-camera/`.

[159] Moustafa Seifeldin, Ahmed Saeed, Ahmed E. Kosba, Amr El-Keyi, and Moustafa Youssef. Nuzzer: A large-scale device-free passive localization system for wireless environments. IEEE Transactions on Mobile Computing, 12:1321–1334, 2013.

[160] Sharp LC-42SB45UT 42" 1080p LCD TV , 2017. `https://www.amazon.com/Sharp-LC-42SB45UT-42-1080p-LCD/dp/B001F0QS9G`.

[161] R.B. Shouldice, C. Heneghan, G. Petres, A. Zaffaroni, P. Boyle, W. McNicholas, and P. de Chazal. Real time breathing rate estimation from a non contact biosensor. In Engineering in Medicine and Biology Society (EMBC), 2010.

[162] Chin-Chung Shu, Peilin Lee, Jou-Wei Lin, Chun-Ta Huang, Yeun-Chung Chang, Chong-Jen Yu, and Hao-Chien Wang. The use of sub-mental ultrasonography for identifying patients with severe obstructive sleep apnea. PLoS ONE, 2013.

[163] R Sirian and Jonathan Wills. Physiology of apnoea and the benefits of preoxygenation. Continuing Education in Anaesthesia, Critical Care & Pain, 9(4):105–108, 2009.

[164] Sonardyne: Sound in depth, 2016. http://www.sonardyne.com/.

[165] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. In-air gestures around unmodified mobile devices. *UIST 2014*, 319-329.

[166] Karl A Sporer. Acute heroin overdose. Annals of internal medicine, 130(7):584–590, 1999.

[167] Ian G Stiell, Daniel W Spaite, Brian Field, Lisa P Nesbitt, Doug Munkley, Justin Maloney, Jon Dreyer, Lorraine Luinstra Toohey, Tony Campeau, Eugene Dagnone, et al. Advanced life support for out-of-hospital respiratory distress. New England Journal of Medicine, 356(21):2156–2164, 2007.

[168] Jo-Anne Stoltz, Evan Wood, Will Small, Kathy Li, Mark Tyndall, Julio Montaner, and Thomas Kerr. Changes in injecting practices associated with the use of a medically supervised safer injection facility. Journal of Public Health, 29(1):35–39, 2007.

[169] Afghan woman stoned to death for 'adultery', 2015. http://www.cnn.com/2015/11/04/asia/afghanistan-taliban-woman-stoning/.

[170] John Strang, Beverly Powis, David Best, Louisa Vingoe, Paul Griffiths, Colin Taylor, Sarah Welch, and Michael Gossop. Preventing opiate overdose fatalities with take-home naloxone: pre-launch study of possible impact and acceptability. Addiction, 94(2):199–204, 1999.

[171] K Strohl, S Merritt, J Blatt, A Pack, F Council, S Rogus, K Georges, T Roth, J Kiley, j Stutts, R Kurrus, P Waller, A McCartt, and Willis. Ncsr/nhtsa epert panel on driver fatigue and sleepiness. In Drowsy Driving and Automobile Crashes.

[172] Ke Sun, Wei Wang, Alex X Liu, and Haipeng Dai. Depth aware finger tapping on virtual displays. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, pages 283–295, 2018.

[173] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. *Mobicom 2015*, 77-89.

[174] Zheng Sun, Aveek Purohit, Raja Bose, and Pei Zhang. Spartacus: Spatially-aware interaction for mobile devices through energy-efficient audio sensing. *MobiSys 2013*, 263-276.

[175] Keiko Tanida, Masashi Shibata, and Margaret Heitkemper. Sleep stage assessment using power spectral indices of heart rate variability with a simple algorithm: limitations clarified from preliminary study. Biol Res Nurs, 2013.

[176] H. Tanoto, J. H. Teng, Q. Y. Wu, Z. N. Chen, S. A. Maier, B. Wang, C. C. Chum, G. Y. Si, and A. J. Danner. Retraction: Greatly enhanced continuous-wave terahertz emission by nano-electrodes in a photoconductive photomixer. In Nat Photon, 2013.

[177] Terry Young and Mari Palta and Jerome Dempsey and James Skatrud and Steven Weber and Safwan Badr. The occurrence of sleep-disordered breathing among middle-aged adults. In The New England Journal of Medicine, 1993.

[178] Thermal imaging: Facts versus fiction, 2016. `https://pr-infrared.com/about-thermal-imaging/thermal-imaging-facts-vs-fiction/`.

[179] A Visvanathan, AP Gibb, and RR Brady. Increasing clinical presence of mobile communication technology: avoiding the pitfalls. In Telemed J E Health, 2011.

[180] Alana M Vivolo-Kantor, Puja Seth, R Matthew Gladden, Christine L Mattson, Grant T Baldwin, Aaron Kite-Powell, and Michael A Coletta. Vital signs: trends in emergency department visits for suspected opioid overdoses - united states, july 2016–september 2017. Morbidity and Mortality Weekly Report, 67(9):279, 2018.

[181] Nora D Volkow and Francis S Collins. The role of science in addressing the opioid crisis. New England Journal of Medicine, 377(4):391–394, 2017.

[182] Alexander Y Walley, Ziming Xuan, H Holly Hackman, Emily Quinn, Maya Doe-Simkins, Amy Sorensen-Alawad, Sarah Ruiz, and Al Ozonoff. Opioid overdose rates and implementation of overdose education and nasal naloxone distribution in massachusetts: interrupted time series analysis. BMJ, 346:f174, 2013.

[183] Anran Wang and Shyamnath Gollakota. Millisonic: Pushing the limits of acoustic motion tracking. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pages 1–11, 2019.

[184] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. Lifs: low human-effort, device-free localization with fine-grained subcarrier information. In MobiCom, 2016.

[185] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization. *MobiSys 2014*, 14-27.

[186] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. Ubiquitous Keyboard for Small Mobile Devices: Harnessing Multipath Fading for Fine-grained Keystroke Localization. In MobiSys, 2014.

[187] Wei Wang, Alex X. Liu, and Muhammad Shahzad. Gait recognition using wifi signals. In UbiComp, 2016.

[188] Wei Wang, Alex X Liu, and Ke Sun. Device-free gesture tracking using acoustic signals. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, pages 82–94, 2016.

[189] Wei Wang, Alex X. Liu, and Ke Sun. Device-free gesture tracking using acoustic signals. In MobiCom, 2016.

[190] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures. In MOBICOM, 2014.

[191] 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, 2015.

[192] Scott D Weingart and Richard M Levitan. Preoxygenation and prevention of desaturation during emergency airway management. Annals of Emergency M edicine, 59(3):165–175, 2012.

[193] Norbert Wild. Ultrasonic through-the-wall surveillance system. Technical report, International Society for Optics and Photonics, 2001.

[194] J. Wilson and N. Patwari. See-through walls: Motion tracking using variance-based radio tomography networks. IEEE Transactions on Mobile Computing, 2011.

[195] Evan Wood. Strategies for reducing opioid-overdose deaths—lessons from canada. New England Journal of Medicine, 378(17):1565–1567, 2018.

[196] Dan Wu, Zhang Daqing, Chenren Xu, Yasha Wang, and Hao Wang. Widir: walking direction estimation using wireless signals. In UBICOMP, 2016.

[197] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. ACM Trans. Graph., 2012.

[198] Jiang Xiao, Kaishun Wu, Youwen Yi, Lu Wang, and Lionel M. Ni. Pilot: Passive device-free indoor localization using channel state information. 2013 IEEE 33rd International Conference on Distributed Computing Systems, pages 236–245, 2013.

[199] Robert Xiao, Greg Lew, James Marsanico, Divya Hariharan, Scott Hudson, and Chris Harrison. Toffee: Enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. *MobileHCI 2014*, 67-76.

[200] Robert Xiao, Greg Lew, James Marsanico, Divya Hariharan, Scott Hudson, and Chris Harrison. Toffee: Enabling Ad Hoc, Around-device Interaction with Acoustic Time-of-arrival Correlation. In MobileHCI, 2014.

[201] Jie Xiong and Kyle Jamieson. Towards fine-grained radio-based indoor location. In HotMobile, 2012.

[202] Qinyi Xu, Yan Chen, BeiBei Wang, and K. J. Ray Liu. Radio biometrics: Human recognition through a wall. IEEE Transactions on Information Forensics and Security, 12:1141–1155, 2017.

[203] Lei Yang, Qiongzheng Lin, Xiangyang Li, Tianci Liu, and Yunhao Liu. See through walls with cots rfid system! In MOBICOM, 2015.

[204] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. Magic finger: Always-available input through finger instrumentation. *UIST 2012*, 147-156.

[205] Xing-Dong Yang, Khalad Hasan, Neil Bruce, and Pourang Irani. Surround-see: Enabling peripheral vision on smartphones during active use. *UIST 2013*, 291-300.

[206] Sangki Yun, Yi-Chao Chen, and Lili Qiu. Turning a Mobile Device into a Mouse in the Air. In MobiSys, 2015.

[207] A Zaffaroni, P de Chazal, C Heneghan, P Boyle, PR Mppm, and WT McNicholas. Sleepminder: an innovative contact-free device for the estimation of the apnoea-hypopnoea index. In Proc. IEEE Engineering Medical biology Society, 2009.

[208] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A Cunefare, Omer T Inan, and Gregory D Abowd. Soundtrak: Continuous 3d tracking of a finger using active acoustics. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(2):1–25, 2017.

[209] Chen Zhao, Ke-Yu Chen, Md Tanvir Islam Aumi, Shwetak Patel, and Matthew S. Reynolds. Sideswipe: Detecting in-air gestures around mobile devices using actual gsm signal. *UIST 2014*, 527-534.

[210] Y. Zhao and N. Patwari. Robust estimators for variance-based device-free localization and tracking. IEEE Transactions on Mobile Computing, 2014.

[211] Yanzi Zhu, Yibo Zhu, Ben Y. Zhao, and Haitao Zheng. Reusing 60ghz radios for mobile radar imaging. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom '15, 2015.

[212] D. Zito, D. Pepe, M. Mincica, and F. Zito. A 90nm cmos soc uwb pulse radar for respiratory rate monitoring. In Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011.