Learning to Listen: An Active Acoustic Approach to Sensing Spaces

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Oliver Shih

B.S., Computer Science and Information Engineering, National Taiwan University
M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

August, 2018

To my family

Abstract

Recently, technologies utilizing ubiquitous sensory data have started to revolutionize our perception and interaction with the physical world, as the Internet of Things (IoT) continues to bring billions of sensors online. Most systems are currently architected with sensor data collection at the edge and processing in the cloud. However, as embedded processing becomes cheaper, faster, and more efficient, we are seeing the opportunity to apply learning on raw data samples closer to the sensor devices. The combination of edge computing and *in situ* learning not only improves a system's sensing and analysis ability, but also maintains low transportation cost, low latency, and good scalability.

In this dissertation, we explore this new class of agile sensing applied to active wide-band acoustic sensors. Unlike conventional approaches that rely on signal processing and well-engineered acoustic features, we propose generic and adaptive learning algorithms that operate closer to raw waveforms. We demonstrate this applied to the field of modern architectural acoustics, where modeling and manipulating space acoustics remains a big challenge, and show its potential in applications such as occupancy estimation, room geometry sensing, acoustic model reconstruction, and microphone localization. We address multiple challenges in designing a lightweight and adaptive learning algorithm, and evaluate trade-offs between estimation accuracy, memory consumption, and energy efficiency on an embedded platform in various real-world environments.

Acknowledgments

I would like to express my deepest gratitude to my advisor and thesis committee chair, Anthony Rowe. He has been a great mentor, more than I could ever hope for, not only for promoting my academic success, but for caring about my personal development. His energy and enthusiasm for life never cease to amaze me, and it has kept me passionate about my research in the face of adversity. Thank you, Anthony, for being an incredible mentor and friend.

I would also like to thank my doctoral committee members: Mario Berges, who provided invaluable discussion and insights on my research; Raj Rajkumar, who mentored and encouraged me during multiple stages of my Ph.D. journey; and Xiaolin Lu, who gave me the opportunity to intern at Texas Instruments and guided me in solving interesting and challenging research problems. It has been a great pleasure working with them and a privilege to have them on my doctoral committee.

I am extremely fortunate to work with my extraordinary labmates from WiSE Lab. I would like to offer my special thanks to Patrick Lazik, Niranjini Rajagopal, and Adwait Dongare, as parts of this dissertation are the result of our fruitful collaboration. My Ph.D. life would not have been as wonderful without the company of Max Buevich, Luis Pinto, Chris Palmer, Craig Hesling, Artur Balanuta, Khushboo Bhatia, Anh Luong, and John Miller. I enjoyed every moment spent with this entertaining crowd. Many thanks to Toni Fox and Chelsea Mendenhall for making the reimbursements hassle-free and keeping the lab running "behind the scenes."

A special mention to Hsu-Chieh Hu, for all the insightful discussions and hard work he contributed to this dissertation; Anand Bhat, who is a great badminton pal and an even better friend; and Hyoseung Kim, who has been generous in sharing his survival guide to Ph.D. life. In addition, this dissertation would not be nearly as reader-friendly without the scrutiny of Nick Wilkerson and Vicky Chou.

Finally, I would never be where I stand today without the support of my family. To my parents, Ming-Tang Shih and Tsung-Mei Wang, and my big brother, Paul Shih: thank you for exposing me to all kinds of opportunities and challenges as I grew up, giving me unwavering support and unconditional love, and encouraging me to pursue my passion and do the things I love. A very special thank you to my fiancé,

Jacky Chou, who endured years of long-distance before joining me at CMU, loved me through the ups and downs, and filled my life with unending inspiration and happiness.

The work in this dissertation was supported in part by the Bosch Research and Technology Center in Pittsburgh as well as the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and the National Science Foundation under award number 1534114.

Contents

List of Figures

List of Tables

Acronyms

ADC	Analog to Digital Converter
ADV	Acoustic Doppler Velocimetry
AI	Artificial Intelligence
ANN	Artificial Neural Network
AOA	Angle of Arrival
AP	Access Point
AR	Augmented Reality
ASR	Automatic Speech Recognition
AURES	Adaptive Ultrasonic Response Estimation Sensor
BLE	Bluetooth Low Energy
CNN	Convolutional Neural Network
CSS	Chirp Spread Spectrum
DAC	Digital to Analog Converter
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNN	Deep Neural Network
DOA	Direction of Arrival
EDM	Euclidean Distance Matrix
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
GDOP	Geometric Dilution of Precision
GMM	Gaussian Mixture Model
GPS	Global Positioning System
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
HVAC	Heating, Ventilation, and Air Conditioning
ICA	Independent Component Analysis

IMU Inertial Measurement Unit
IoT Internet of Things
IS Image Source

k-NN k-Nearest Neighbors
KF Kalman Filter

LDA Linear Discriminant Analysis
LIDAR Light Detection and Ranging
LOS Line of Sight
LPC Linear Predictive Coding
LSTM Long Short-Term Memory

MCTS Monte Carlo Tree Search
MDS Multidimensional Scaling
MEMS Microelectromechanical Systems
MFCC Mel-Frequency Cepstrum Coefficient
ML Machine Learning
MSE Mean Squared Error

NEDM Nearest Euclidean Distance Matrix
NLOS Non-Line of Sight
NN Neural Network
NP Non-deterministic Polynomial acceptable

PCA Principal Component Analysis
PIR Passive Infrared

RADAR Radio Detection and Ranging
RANSAC Random Sample Consensus
RF Random Forest
RMS Root Mean Squared
RNN Recurrent Neural Network
RSSI Received Signal Strength Indicator
RT Reverberation Time
RTOF Round Trip Time of Flight

SDP Semi-definite Programming
SLAM Simultaneous Localization and Mapping
SNR Signal-to-noise Ratio
SONAR Sound Navigation and Ranging
SPL Sound Pressure Level
SVM Support Vector Machine

TDMA Time Division Multiple Access
TDOA Time Difference of Arrival
TOA Time of Arrival
TOF Time of Flight

UWB Ultra Wide Band

VIO Visual Inertial Odometry
VO Visual Odometry
VR Virtual Reality

Nomenclature

$\bar{A}$	estimation of matrix A
$\bigcap$	union of sets
$\boldsymbol{e}$	vector of ones
$\circ$	Hadamard product
$\emptyset$	the null set
$\forall$	for all
$\iff$	if and only if
$\in$	belong to
$\|\|A\|\|_F$	Frobenius norm of matrix A
$\mathbb{R}$	set of real numbers
$\mathcal{E}^n$	space of $n \times n$ Euclidean distance matrices
$\mathcal{S}^n$	space of $n \times n$ symmetric matrices
$\mathcal{S}^n_+$	subspace of positive semi-definite matrices in $\mathcal{S}^n$
$\mathcal{S}^n_C$	centered subspace in $\mathcal{S}^n$
$\mathcal{S}^n_H$	hollow subspace in $\mathcal{S}^n$
$\propto$	proportional to
$\rightarrow$	mapping of spaces
$\succeq$	Löewner partial order
$\mathbf{0}$	the origin of a space
A^T	transpose of matrix A
a^T	transpose of vector a
H	Hermitian transpose
diag(·)	diagonal elements of a matrix
e	exponential
offDiag(·)	orthogonal projection onto the hollow matrix

Chapter 1

Introduction

1.1 Background

Sound waves can tell us a great deal about the world around us. When sound waves are generated, information about their sources, such as pitch, duration, and loudness, is embedded into pressure waves that move through air. These pressure waves impart energy into our eardrums, allowing us to hear words and music. This same principle applies to echoes, where each can be treated as an individual sound wave carrying information about its reflector. This mechanism is the foundation of active acoustic sensing, and it can be found both in nature and in man-made systems. Bats make high-pitched calls to navigate and forage; toothed whales use echolocation to hunt; shrews emit ultrasound to locate insects. As early as the mid-18th century [68], reports show that blind individuals are able to locate silent objects using sound and hearing [1]. These intriguing findings have since inspired many remarkable inventions such as SOund Navigation And Ranging (SONAR).

[1]Interestingly, the mechanism driving this ability was originally believed to be pressure changes on the skin [68]

However, most applications and advanced techniques have been limited to underwater sensing and outdoor environments. Acoustic signals are notoriously difficult to model and manipulate in indoor environments where spaces are confined. Not only does sound dissipate faster in air than in water, when transmitted into a confined space it creates numerous echoes. These multipath reflections can blend in with each other or cancel each other out, while decaying at different rates based on the absorption of the reflected surfaces. This phenomenon makes it extremely difficult to capture all the dynamics and nuances in the environment. Scientists and engineers in the field of architectural acoustics have been studying the impact of room geometry and absorption on indoor acoustics for more than a hundred years, yet the state-of-the-art still relies on computer simulations that struggle to be accurate.

Over centuries, the potentials of active acoustic sensing have continued to thrive as measurement tools and processing techniques have advanced. Perhaps what it can achieve is only limited by our imagination. In the 2008 Christopher Nolan movie, *The Dark Knight*, the Batman and his confidant hack into people's cellphones, activating their speakers and microphones to "see" the surrounding environment and nearby people, as a bat would. While this might seem futuristic, the line separating science fiction and reality begins to blur given the rapid pace of technology. From nature's gifts to science fiction, all of these abilities are built upon a central understanding of sound and echoes; they are driven by a comprehension of *how* the echo is reflected and *where* the echo is reflected. This dissertation takes initial steps toward expanding our understanding of echoes and their potentials in active acoustic sensing driven by machine learning approaches.

1.2 Motivation

Recently, technologies utilizing ubiquitous sensory data have started to revolutionize our perception and interaction with the physical world, as the Internet of Things (IoT) continues to bring billions of sensors online. Real-time networked sensors have demonstrated huge potential in making numerous applications smarter through rich inferences derived from surrounding environment and nearby people. Driven by advances in embedded devices with faster processing power, lower energy consumption, and decreases in cost, we are seeing the opportunity to apply *in situ* learning on the sensor devices to improve their sensing and analysis capabilities while maintaining low transportation cost, low latency, and good scalability.

In this dissertation, we explore the use of embedded machine learning techniques on active acoustic sensing systems like smart speakers, where sound signals are transmitted into the environment and the reflected signals are recorded at one or multiple locations. Applications in active acoustic sensing have taken many forms in the past, such as SONAR, ultrasonic imaging, and motion and proximity sensing. Most of these applications have been limited in terms of both scope and performance, because modeling and making inferences about acoustic signals is difficult due to the highly dynamic nature of multipath reflections in confined spaces. Classic signal processing approaches use building blocks that attempt to perform generalized processing. However, this processing approach often fails due to variability from space to space. Given how significantly the environment impacts acoustic signals, we demonstrate that learning can be used to capture the nuances of specific environments to help model acoustic

properties. For example, work in the domain of architectural acoustics has studied acoustic modeling in space in order to improve sound quality within buildings. Acoustic engineers use relatively basic first-order models for a variety of applications ranging from enhancing speech clarity in auditoriums to reducing background noise in restaurants or improving music quality in concert halls and home theatres. However, manipulating space acoustics, which requires a strong understanding of sound absorption and reflection, still remains a process of trial and error. By sensing and learning about specific environmental features, we believe we can significantly improve upon this process and, more importantly, expand on the types of sensing that are possible with active acoustic systems.

To expand the capabilities of active acoustic sensing, we propose an approach that leverages learning to accurately capture both acoustic absorption and reflection at the same time. We demonstrate this capability in real-world environments and show its potential in applications such as occupancy estimation, room geometry sensing, acoustic model reconstruction, and microphone localization. We also discuss practical challenges of enabling learning on resource-constrained devices and evaluate their impacts on various aspects of the system performance.

1.3 Problem Statement

Active acoustic sensing relies on understanding the physical nature of how sound interacts with the environment, as shown in Figure 1.1. On one axis, we see that sonic energy is altered either through absorption or reflection. On the other axis, we see the impact of the room geometry and humans in the space. Various different active

acoustic sensing applications lie at the intersection of each of these dimensions. For example, room geometry reconstruction and microphone localization require modeling the environment and locating the sources of reflections. Improving sound quality depends on understanding room surface absorption. Counting humans in a space requires an understanding of both absorption and reflection.

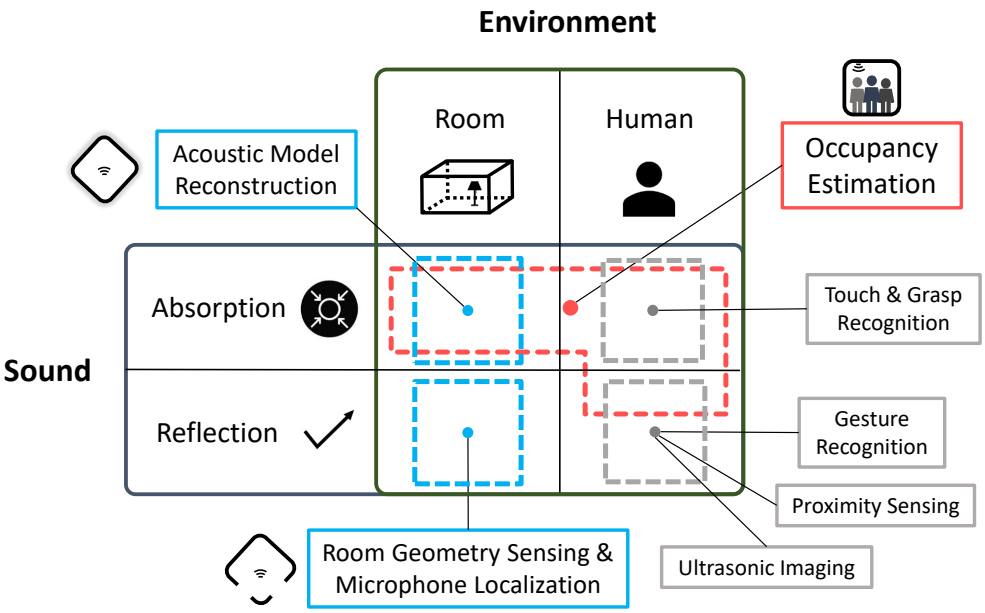


Figure 1.1: Active acoustic sensing applications and their interactions with the environment.

Through learning different aspects of these interactions, we aim to answer the following questions:

1. Can active acoustic sensing be used to estimate occupancy?
    - How well can the technique estimate the number of occupants in a variety of spaces?
    - What parameters impact the estimation accuracy?
    - How does the environment affect the system performance?
    - What Machine Learning techniques are best suited for embedded

systems?

- How can we adapt a trained model to environmental changes?

2. Can active acoustic sensing be used to derive room geometries, microphone locations, and acoustic models of spaces?
    - How well can we estimate the room geometry in a variety of spaces?
    - How well can we locate a microphone?
    - What parameters impact the reconstruction's accuracy?
    - How well can we estimate architectural acoustic properties such as absorption coefficient of surfaces?
    - How does the system perform compared to the state-of-the-art?

1.4 Thesis Statement

"*Machine learning applied to active acoustic systems can improve their ability to sense and derive inferences from the environment; we present new techniques for occupancy estimation, room geometry sensing, acoustic model reconstruction, and microphone localization.*"

1.5 Contribution

The contribution of this dissertation is presented in the following applications within the class of active acoustic sensing:

1. **Occupancy Estimation**
    - The design and evaluation of an occupancy estimation algorithm based

on acoustic properties

- The design and evaluation of a presence detection algorithm and recalibration algorithm that adapt the occupancy estimation model to account for changes in the background environment over time
- The design and implementation of a self-contained energy-harvesting platform with wireless communication that executes the occupancy estimation algorithm in real-time and leverages a smartphone for training
- A comparative analysis of the proposed approach and state-of-the-art solutions in real-world environments

2. **Room Geometry Sensing, Acoustic Model Reconstruction, and Microphone Localization**
    - The design and evaluation of a room model reconstruction and microphone localization algorithm
    - The design and evaluation of a gradient-based searching algorithm
    - The design of a prototype system evaluated in real-world environments
    - A comparative analysis of the proposed approach and state-of-the-art solutions
    - An AR application for visualizing the reconstructed acoustic model

Chapter 2

Related Work

In this chapter, we discuss the background related to active acoustic sensing (see Section 2.1), followed by an overview of Machine Learning (ML) frameworks in the domain of acoustic sensing (see Section 2.2). Next, we address challenges specific to the field of architectural acoustics (see Section 2.3) and discuss their close relationships with occupancy estimation, room geometry sensing, and acoustic model reconstruction. Finally, we detail technologies related to these specific topics in Section 2.3.1 and Section 2.3.2, respectively.

2.1 Active Acoustic Technologies

Active acoustic approaches have shown great potential in multiple forms of sensing. As early as 1912, the first underwater echo-ranging device was invented in response to the sinking of the *Titanic*, in an attempt to echolocate the ship in the same way bats use sound for navigation and localization. During World War I, the military developed active SONAR (SOund Navigation And Ranging) system for the detection of submarines. SONAR works by sending out a ping signal and then

listening for the reflected echo of the pulse. To measure the distance to an object, the time between the transmission and the reception of the pulse can be converted into a range based on propagation given the speed of sound. To measure the bearing, multiple hydrophones are used to measure the relative amplitude and arrival time between each beam. To measure the radial speed of the target, the Doppler effect is used to convert the difference in frequency between the transmitted and received signal into a velocity. Numerous studies have continued to improve these measuring methods over the years, and techniques such as pulse compression, beamforming, and Acoustic Doppler Velocimetry (ADV) have since enabled improved ranging accuracy, signal strength, and velocity estimation.

Recently, acoustic sensing and related techniques have been widely applied to non-military applications. For example, acoustic ranging techniques have been utilized in several indoor localization systems [53, 78, 101, 104]. These systems typically share the same operating principle as the Global Positioning System (GPS); the Line-of-Sight (LOS) ranges from multiple synchronized transmitters/receivers with known locations are used to determine the receiver's/transmitter's location. Depending on whether the transmitter(s) and receiver(s) are synchronized, the localization process can either follow a Time of Arrival (TOA) or Time Difference of Arrival (TDOA) based approach. The ranging measurements from multipath reflections can also be exploited in a similar way and used in many applications. Many researchers used a speaker and/or microphone array to determine the shape of a room or surrounding reflectors based on echo ranges [6, 18, 19, 27, 35, 60, 88, 110, 112, 113, 127]. To improve ranging resolution and SNR, most systems utilize pulse compression on the transmitted

signal along with a matched filter at the receiver. Aside from pulse compression, wireless communication modulation schemes can also be applied to acoustic signals to improve ranging resolution. For example, in [91], the authors used OFDM modulation to achieve sub-centimeter ranging accuracy and demonstrated accurate gesture tracking and recognition using a mobile phone.

Active acoustic sensing has been shown effective for various applications in classification. In [82, 99, 134], the authors demonstrated how an attached speaker/microphone pair on human bodies or common objects allows recognition of various touch and grasp gestures. As the sound wave propagates through the body, different postures cause varying fluctuations in the signal's power spectrum which can be reliably classified using Support Vector Machine (SVM). The micro-Doppler effect, which is the shift in frequency caused by an object vibrating or spinning commonly observed in RADAR systems, has also been studied in acoustic signals. In [11], the authors built a k-NN and Bayesian classifier with micro-Doppler signature to differentiate walking gaits of different people, or different actions undertaken by the same person. The same gait signature has also been used to distinguish humans from four-leg animals [139], or to classify underwater vehicles [64]. A few studies also exploited ranging readings in addition to reflected Doppler signals, to classify speech, walking motion, and gestures tracking [107]. It is worth noting that most of these classification-based applications apply machine learning algorithms on top of signal processing tools to derive more complex inferences from received signals.

In this dissertation, we utilize many of the aforementioned acoustic inferences including frequency shift, ranging estimation, and changes in the spectrum and apply relevant techniques such as Doppler shift, pulse compression, and machine

learning for estimating occupancy and reconstructing an acoustic model of a room.

2.2 Machine Learning Frameworks

In the domain of acoustic sensing, automatic speech recognition (ASR) is one of the first fields to widely adopt machine learning techniques. In fact, ASR was one of the main drivers for machine learning in general and has a substantial influence on the development of acoustic signal processing. Techniques such as Support Vector Machine (SVM), Gaussian Mixture Model/Hidden Markov Model (GMM-HMM), and Artificial Neural Networks (ANN) have been commonly used with well-engineered features, such as zero-crossing rate, linear predictive coding (LPC), Mel-frequency cepstrum coefficient (MFCC), for the recognition and translation of spoken languages [33]. In particular, MFCC [31] is by far the most widely used feature in state-of-the-art speech recognition systems, given its robustness to additive noise and uniqueness in approximating human auditory response. Other common generalized feature extraction techniques include Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA), which have also been widely employed in speech recognition and sometimes used in combination with MFCC [81, 122, 128]. Following their success in ASR, many of these techniques and features have been adopted in other applications as well, such as human activity recognition [24], gesture recognition [12], and sound source localization [124].

Over the past decade, advances in hardware and the availability of large datasets have reignited interest in Deep Neural Networks (DNN) based approaches

due to their powerful ability to generalize feature extraction and transformation [67] while perfectly matching modern GPU architectures. Deep-learning methods are typically composed of multiple layers of representation with transformation/activation functions in between. While the representation in each layer can be relatively simple, the combined non-linear transformation allows the model to approximate extremely complex functions as the number of layers grows. Several studies have shown that DNN-based approaches can obtain comparable performance even when trained with raw waveforms or amplitude spectra instead of conventional features like MFCC [129]. Methods combining Long short-term memory (LSTM), deep Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN) have become part of state-of-the-art systems in ASR [40, 49, 59, 115].

However, DNN-based approaches face several obstacles in practice. Despite having superior performance in general, these approaches rely on advanced hardware with large memory size, GPU acceleration, and efficient parallelization for data processing. These drawbacks make them more difficult to implement on embedded devices that have weak processors and limited memory. To solve these embedded challenges, some have proposed compressing the model size of existing algorithms. Several methods have been proposed to compress DNN [52], Random Forest (RF) [90], and k-Nearest Neighbor (k-NN) [73], but they still struggle at the scale of embedded devices. Offloading the computation to the cloud provides an attractive alternative, yet most cloud computing frameworks introduce several drawbacks including latency, bandwidth, reliability, and privacy issues [9]. In order to strike a balance between efficiency, accuracy, and robustness, the majority of applications maintain a traditional pipeline of signal processing followed by feature

extraction and classification. With this conventional framework, however, applications continue to rely on carefully crafted features and a considerable amount of labeled training data.

Recently, more attention has been drawn to developing embedded machine learning algorithms in order to bridge the gap between the two frameworks. Enabling *in situ* learning on edge devices can potentially create a new paradigm that combines the merits from both frameworks; devices can utilize learning without inducing the disadvantages of transporting data over the network. In [72], the authors proposed an embedded tree-based algorithm with extreme resource efficiency in energy consumption, execution time, and model size ($\leq 2kB$) while retaining good prediction accuracy in classification tasks. Their algorithm achieved the gain from a sparse tree learned in a low-dimensional space using dimensional reduction, joint learning, and gradient descent with iterative hard thresholding. Several papers [50, 131, 141] focused on embedded k-NN algorithms. Particularly in [50], the authors presented a compressed k-NN model that greatly reduced the model size from $6MB$ to $16kB$ using low-dimensional projection, prototype representation, and joint optimization. An efficient algorithm for outlier robust Principal Component Analysis (OR-PCA) is also studied in [25]. The authors proposed a thresholding based algorithm to effectively reduce the time complexity from quadratic to linear, which is equivalent to vanilla PCA, under specific noise models. Unfortunately, the method does not apply to the additive Gaussian noise model that is often observed in practice.

The majority of practical machine learning applications rely on supervised learning, which requires tremendous amounts of well-labeled data for training. However, such databases can often be expensive and difficult to obtain especially

when human annotators, special devices, or lengthy experiments are involved. In the domain of ASR, for example, accurately transcribing speech at the phonetic level can be extremely time-consuming and often requires experts [97]. In computer vision, labeling footage frame by frame is often laborious and tedious. In protein structure prediction, it may take weeks or months of laboratory work to identify a specific DNA sequence of a protein. On the other hand, unlabeled data is often available in large quantity and can be easily acquired: speech recordings can be obtained from radio broadcasts or audio books; video footage is readily available from online streaming services; and DNA sequences can be extracted directly from gene databases. This practical factor leads to a learning paradigm so-called semi-supervised learning, which considers the problems with a paucity of data labeled. The goal of semi-supervised learning is to utilize both labeled and unlabeled data to achieve better performance than using either alone. From a different perspective, semi-supervised learning could achieve the same performance as supervised learning, but with less labeled data required. Depending on problem formulation, semi-supervised learning can be utilized in multiple settings such as classification, clustering, or regression [143].

In this dissertation, we apply various data-driven algorithms to learn various aspects of acoustic properties while minimizing training effort, computational complexity, and memory consumption. To estimate room occupancy (see Chapter 3), we propose a semi-supervised learning framework based on raw frequency features that bypass the process of feature engineering. Our algorithm uses a PCA-based approach for dimensional reduction to reduce the model size, but assigns different weights based on relevancy to prevent outliers. We also apply a clustering algorithm in combination with a regression model to further minimize

the model size, impact of noise, and training effort. To reconstruct the acoustic model of a room (see Chapter 4), we propose a pipeline of optimization tools, including semi-definite programming and combinatorial optimization, to derive consensus on room geometry based on echo ranges. In addition, we use a clustering algorithm and design a searching algorithm based on gradient descent to reduce estimation inaccuracy and run-time complexity.

2.3 Architectural Acoustic

Over the last 120 years, work in the area of architectural acoustics has emerged to scientifically improve sound quality within buildings. Acoustic engineers have looked at problems ranging from enhancing speech clarity in auditoriums to reducing background noise in restaurants or improving music quality in concert halls and home theatres. However, manipulating space acoustics, which requires precise modeling of sound absorption and reflection, remains a big challenge. Current acoustic treatments rely on cumbersome trial and error processes performed by domain experts that involve constant adjustments of sound absorbers/blockers and repeated measurements using high-end equipment at specific locations, or in some more difficult cases, with a well-trained ear. Other benchmarks such as reverberation time RT_{60} (the time it takes an audio signal to decay $60dB$) can be used in combination with room size to estimate room response, but are limited in depicting room geometry and deriving absorption coefficients from all of the surfaces that play important roles in overall acoustics.

In the early 1900s, Wallace Sabine began to model the impact of people, frequency, and the geometry of spaces on acoustics [114]. Significant follow-on

work in acoustics has shown that people in a space significantly impact reverberation and that reverberation is frequency- as well as room geometry-dependent [15]. To model overall acoustic properties, recent work in this area has used computer simulations [30, 56, 57, 118] with precise 3D model and acoustic details of the space. It is clear from this large body of research that creating simple, generalizable models of reverberation is quite challenging. For this reason, we propose using machine learning techniques to identify room geometry while learning and classifying the reverberation response on a per-installation basis.

2.3.1 Occupancy Estimation

Occupancy sensing spans a variety of technologies with a number of design trade-offs. Conventional solutions use largely binary sensors with PIR, microwave, or ultrasound to detect the presence of people. Most recent work has used cameras or fusion of multiple sensor types to measure occupancy level. All of these approaches generally fall into two categories based on their capabilities. One group focuses only on detecting the vacancy/presence of people [17, 28, 51, 96, 126, 133], which often comes with an analysis of detailed user behaviors and actions. Other categories focus more on people-counting systems [16, 22, 43, 58, 66, 76, 85, 89, 92, 94, 132, 133, 140, 144], usually involving more sophisticated algorithms and learning-based approaches.

Presence Detection

In the category of presence detection, two common sensors have been widely deployed in modern buildings: passive infrared sensor and ultrasonic sensor.

Passive infrared (PIR) sensors detect the differences in infrared radiation, such as heat, emitted from human movement and those from the background environment. These sensors typically have a limited field-of-view and require clear LOS to the targets, which make them more suitable for a small, enclosed space or a narrow entrance. Ultrasonic sensors, on the other hand, use active transmission of ultrasonic signals to detect the presence of occupants. These sensors rely on the detection of Doppler shifts that occurs when there is movement toward or away from the transmitted signal. Ultrasonic sensors do not require strict LOS of the targets and can detect people around corners, due to diffraction and multipath reflection. In general, they are more effective at detecting sudden movements and tend to have larger coverage areas compared to PIR sensors, which makes them work well in open spaces and spaces with obstacles.

Recently, presence detection has been expanded to other sensor types. For example, in [133], the authors combined motion sensors and smart meter feeds to detect the presence of occupants and even infer the occupancy. In [28], the authors focused primarily on WiFi signals and used "sniffers" that monitor WiFi APs to detect occupants' mobile devices and their whereabouts. In both cases, the approaches do not perform as well in large spaces like auditoriums, unless each occupant is carrying a mobile device that cooperates with the system. Two of the recent works use similar approaches based on ultrasonic signals [17, 126]. In [126], the author proposed a sonar system using four microphones at a constant frequency of $20kHz$ in order to detect the user's attention state and several pre-defined activities. They built a classifier by characterizing the variance of the reflection intensity from a user's body. Their experimental results show supportive evidence that a user's presence impacts the intensity of the echoes, which is a

fundamental characteristic we leverage in our approach. Nevertheless, this technique requires a copious amount of training data to predict the pre-defined activities, and assumes the environment to be free from interference. Similar work in [17] proposed an ultrasonic array sensor and tracking algorithm to detect the presence and capture the movement of targets. This is achieved by taking the difference between the received echo signals to estimate the direction of arrival (DOA) with the array of sensors, and utilizing the received signal-to-noise ratio (SNR) as an indicator of occupancy. A simple tracking algorithm is also proposed to increase the performance of presence detection. While this method shows better performance than PIR sensors, the detection zone is limited to a certain area and confined by DOA angle.

Some other approaches take advantages of using multiple co-located sensors [51, 94, 96]. In [96], *TelosB* nodes are deployed with pressure sensors, PIR sensors, and audio sensors. The system is able to predict pre-defined activities by correlating the binary readings from multiple sensors. The overall classification accuracy is more than 90%, but it requires a careful deployment of multiple sensors at different locations in the room. With a similar choice of sensors, the author in [51] adopts additional light and CO_2 sensors. Classification is done using a decision tree in order to determine which sensors are most important. The results indicate that the motion sensor is dominant, and accounts for 97% of accuracy even when used alone. Even more diverse sensor types are utilized in [94] including temperature, humidity, light, and CO_2. The authors evaluated their performance with multiple classification algorithms. The model achieved 99% of accuracy with readings from light sensors as the most effective parameter. However, in case of changes in location, the model has to be retrained to adapt to

the new environment, which is a prerequisite for most classification models.

To summarize, although most of the presence detection techniques have the advantage of low-cost and low-complexity, their applications are limited due to their binary sensing. They also suffer from scalability and deployment difficulties due to the confined detection area of the sensors.

People Counting

The most common commercial solution for people-counting uses RGB video cameras [16, 22, 85, 132, 140]. While camera-based approaches tend to have high accuracy, they often suffer in practice due to lighting conditions, clutter in the background, privacy issues, and extensive training and setup efforts. An early work for fine-grained indoor people-counting is presented in [132], where the locations of the objects are first measured by their silhouettes by image sensors deployed around the room. The system shows accurate results up to 12 people moving in a room, but requires careful placement of multiple image sensors. Also, the computational complexity grows proportionally to the number of sensors. In [140], the authors used face detection with Kalman filtering and a k-NN classifier to track the trajectories of occupants. The results show high tracking accuracy, but the method does not scale well as the number of occupants increases. For counting larger groups of people, a crowd-counting algorithm proposed in [22] shows accurate results for tens of pedestrians with an error of less than two people. The algorithm claims to be privacy preserving by segmenting the crowd into groups using low-level features, then using a regression model to count people within each segment. A pedestrian database is required for providing a large number of training images, which is often costly and thus makes it less

feasible in more constrained use-cases, like on an embedded sensor. To reduce the effort in acquiring labeled data, several pieces of research proposed using a semi-supervised learning method for crowd counting [16, 125]. These algorithms first perform a spectral clustering on the unlabeled data to select the most representative data for labeling, then use feature mapping to facilitate learning of a new target model. This concept enables the use of knowledge from a previous scene and thus reduces required training data for bootstrapping learning in the new scene, but the assumption is that the two scenes must share similar manifold representations.

Recently, there have been several attempts at extending data fusion approaches from presence detection to occupancy estimation [43, 76, 144]. In [76], the authors evaluated three different learning methods including SVM, NN, and Hidden Markov Model (HMM) over a dozen of different sensor inputs, and were able to estimate $0-3$ occupants in an open office area with 75% accuracy. In [43], a classification model built on WiFi access, user activity, calendar, and time-of-day information achieved as high as 90% accuracy among five occupants. In [144], the authors developed an indoor air quality measurement system with CO_2 sensor, total volatile organic compounds (TVOC), air temperature, and air relative humidity sensors. They adopted supervised learning algorithms and compared decision tree, Logistic Regression, k-NN, and RF in their capability to detect and estimate occupancy of three people. Although the learning model could only achieve an accuracy of 75%, it was shown to be insensitive to the training participants.

Rather than estimating the occupancy in an instantaneous manner, another popular approach is to track the inbound and outbound traffic of the room by monitoring its entrance [58, 66, 89, 92]. An early work in [66] used a camera hung

from the ceiling to track people passing through the door. The algorithm used background subtraction followed by object extraction and tracking to monitor occupancy. This approach, however, assumes all moving objects are human and they do not overlap with each other. This assumption is more relaxed in [58], where Hnat et al. introduced the *Doorjamb* tracking system using ultrasonic range finders mounted on door frames to monitor room access. By using probabilistic inference and associating people's identities with their heights, the system performs well on people-tracking in specific environments, such as labs or residential homes with a high room-tracking accuracy. A similar system in [92] used weight sensors and Microsoft Kinect with Naïve Bayes and SVM classifiers to identify people based on their weight and height. However, both of these systems are unable to detect multiple people crossing at the same time and are unsuitable for environments with wide entrances. More recently in [89], the authors presented the *FORK* system utilizing Kinect depth sensors above doorways to track the heads and shoulders of passing occupants. The system is highly accurate and tolerant to multiple people-crossing scenarios. They also compared identification performance among several classification algorithms using tens of biometric features. However, the system is expensive to install and the performance degrades as the number of people crossing increases. In general, there are a few fundamental challenges associated with door monitoring approaches. First, they typically require careful deployment and hand-tuning of detection thresholds. Second, system performance degrades when a crowd of people passes through simultaneously, and the estimated error accumulates over time without a reliable calibration mechanism. Finally, systems with better performance tend to rely on more costly hardware that are more intrusive in

terms of privacy.

In summary, although most of the presence detection techniques have the advantage of low-cost and low-complexity, they only provide a coarse estimate of people within a space. In contrast, most of the people-counting techniques are either more expensive in terms of cost and complexity, struggle to perform crowd estimation, or require large, labeled databases. Based on this large body of work, there are no existing frameworks that can perform wide-area people counting with a single cost-effective and versatile sensor. In recent experiments using reverberation [75], it is clear that given a particular room geometry, audience absorption follows relatively distinct curves that make it a powerful feature for occupancy estimation. In this dissertation, we present one of the first end-to-end systems where ultrasound has been used to directly estimate occupancy.

2.3.2 Room Geometry Sensing and Acoustic Model Reconstruction

Geometrical room acoustic modeling has a long research history that involves many related topics, such as data acquisition, measurement uncertainty, signal processing techniques, acoustic design, and geometry reconstruction. Each of these topics has a large body of research, so for the scope of this dissertation, our focus is on the methods that can be used for estimating sound response in three-dimensional (3D) spaces and/or the reconstruction of their geometries. These two problems are similar in that the main contributing components of the impulse response, early reflections, and reverberation are by and large determined by the geometry of the room.

Acoustic geometry reconstruction typically assumes a set of microphones and speakers with known locations. These transmitter and receiver pairs measure the

RIR in order to estimate the location of reflectors and obstacles with respect to their own positions. While many approaches formulate the reflection localization problem in different ways, the relative positions between speakers, microphones, and walls can either be characterized by time of arrival (TOA) [27, 35, 60, 88, 110, 112, 113, 127], time difference of arrival (TDOA) [6, 88] of impulses, or the direction of arrival (DOA) [18, 19, 113]. Note that in some approaches, more than one of these characteristics are utilized in deriving the solution. Earlier work in this area has been focusing on two-dimensional (2D) reconstruction where the speakers, microphones, and reflectors lie on the same plane [6, 18, 19, 36, 88]. Lately, many of these approaches have been extended into 3D reconstruction. For example, the work in [6] and [5] has been improved respectively in [41, 111] and [93] to accommodate generalized 3D scenarios.

Direct Estimation Approach

One of the earliest approaches to room geometry reconstruction focused on direct localization of sound reflectors. For example, [74] used an approach similar to seismic exploration and underwater imaging to capture an image of the reflecting objects from reflected energy. The imaging process is based on inverse wavefield extrapolation from the receiver to the object's position. One major limitation of this approach is that it requires a planar array of microphones and assumes a specific spatial relationship between the microphone array and reflectors.

In [6, 41, 93, 112], the authors modeled walls as planar surfaces tangent to the ellipsoid defined by the echo distance between transmitter/receiver pairs. To find the overlaps among multiple ellipsoids derived from noisy measurements, most techniques adopt the Hough transform [38] or RANSAC process [42] to reliably and

efficiently refine the solution in the presence of outliers. However, these approaches often require a microphone array with known positions and the localization of the sound sources using DOA. In addition, in order to prevent "ghost" walls/reflectors caused by higher-order reflections, several studies imposed restrictions on the dimensions of the room [41, 93].

Image Source Approach

A more recent approach to room geometry reconstruction relies on the Image Source (IS) model [2] to indirectly locate the reflectors. The IS model defines imaginary sound sources by mirroring the true sound source against the reflecting surfaces. From the receiver's perspective, any multipath reflections can be treated as LOS signals from the image sources, which helps to describe how the sound waves propagate and to greatly reduce the computational complexity of locating the reflectors. This solution is exact, assuming the reflecting surface is rigid and the wave incidence is spherical. In most scenarios, TOAs are first converted into distances to determine the locations of the image sources, and in turn, determine the locations of the reflectors using the IS model.

The main challenge of the image source formulation is that echoes reflected from different walls can arrive at the receiver in an arbitrary order, and it is not trivial to sort them based on the number of times they have been reflected, nor to label them with the correct image sources. Many approaches alleviate this problem by making assumptions about the types of echoes or the order of echo arrival. In [27], the author considered the scenario with multiple sources and receivers, and proposed a reconstruction algorithm based on minimizing the delay ambiguity in echoes. However, the method assumed no higher-order reflections in

the received signal, and the sound sources cannot overlap in time. The same restriction on the order of RIR is assumed in [102], where a mobile node is used to recover a wide class of a polygonal shape geometry through only first-order RIRs. However, this approach may fail if the polygon has one or more pairs of parallel edges. In [88], the authors exploited the constraints on convex polyhedral room geometry imposed by the combination of first-order and second-order reflections and presented a method to reconstruct room geometry from a single channel impulse response. Unfortunately, this method requires prior knowledge of the labels for all TOAs and the detection of all first-order and second-order reflections, which is difficult in practice.

To relax the assumption on echo arrival, several works aim to solve the echo labeling problem directly. In [35], Dokmanica et al. proposed using the properties of Euclidean Distance Matrix (EDM) to solve the echo labeling problem by brute-forcing all combinations of echoes using multidimensional scaling (MDS). To alleviate the computational complexity, the system uses a microphone array of five microphones with known positions to reduce the number of echo combinations. In addition, the algorithm requires prior knowledge of the number of walls and the detection of all first-order echoes to eliminate higher-order echoes and correctly reconstruct room geometry. A follow-on work in [60] later transformed the echo labeling problem into a maximal independent set listing problem in graphs that can be solved more efficiently using an exponential space algorithm. They also used a rank-5 factorization method that was first proposed in [103] to directly compute the location of the transmitters and receivers in linear time complexity. While promising in simulation, this approach requires at least ten sound sources and five microphones. Another recent work further improves this graph-based approach

using subspace-based filtering to reduce the computational complexity [26]. However, similar to [60], this approach delivers the speedup by utilizing a larger number of microphones and sources.

More recent work leverages a mobile node to replace the need for multiple microphones [102, 142]. In [102], the author studied the possible room shapes that can be recovered using a mobile node, but assumed perfect localization and precise echo ranging. In [142], a commodity smartphone is used to achieve fine-grained reconstructions through short-range scanning. However, the method requires the user to walk a full loop closely to the internal room boundaries which is unsuitable for 3D reconstruction. To reliably measure the distance to walls, the smartphone also needs to be held in a specific position and follows a careful measurement gesture that is prone to error.

Based on this large body of work, we can conclude that most of the 3D reconstruction approaches either require a large array of transmitters/receivers and/or impose assumptions on the order of received echoes and the geometry of the room. In addition, few of them are evaluated in a real-world environment where missing/spurious echoes and measurement uncertainty have a huge impact on reconstruction stability and accuracy.

In this dissertation, we adopt the image source model and the EDM formulation as the building block of our algorithm, but assume no prior knowledge of the number of reflective surfaces nor the detection of all first-order echoes. We present a robust reconstruction algorithm that utilizes SDP to refine surfaces localization, combinatorial optimization to cope with measurement uncertainty, and a clustering algorithm with geometry properties to deal with missing and spurious echoes. We also present a searching algorithm to reduce overall

computational complexity. Our system requires only one speaker and a commercial off-the-shelf smartphone that samples at multiple random locations in the room with the help of Visual Inertial Odometry (VIO) tracking.

Chapter 3

Occupancy Estimation

3.1 Introduction

Being able to count the number of people accurately in a space has high utility for a number of applications. In building automation systems, knowing if a room is occupied or not can be used to control zone heating and cooling, or simply to disable unused lighting. Heating, Ventilation, and Air Conditioning (HVAC) of buildings represents about 17% of the total energy used domestically, equivalent to about 16.7 QBtu (quads) of energy annually. It has been shown that HVAC controls that are adaptive to fluctuations in occupancy density and distribution should allow optimization of air distribution and provide substantial energy savings in thermal and lighting control [32, 48, 69, 98, 138]. In the context of large facilities like conference centers or in the retail space, knowing how many people are in certain locations and how long they dwell can be used to value shelf-space or storefront locations and predict traffic flow. In architectural acoustics, knowing the number of audiences can improve our estimation of audience absorption and improve sound quality during live performances. These applications require a

sensor capable of counting how many occupants are within a space.

There are currently many approaches for measuring occupancy in spaces including passive infra-red (PIR) sensors, ultrasonic ranging sensors, microwave sensors, smart cameras, WiFi Access Point (AP), break beam sensors, and laser range-finders. These devices span across a wide spectrum of cost and performance. Lower-cost alternatives, like PIR and ultrasonic ranging sensors, are typically error-prone and usually only detect binary occupancy values rather than estimating load. More expensive sensors like smart camera systems tend to require sophisticated site-specific installation and calibration. They also require wall power, pose privacy risks, and are often hindered by obstructions.

In this chapter, we present Adaptive Ultrasonic Response Estimation Sensor (AURES) [120], a platform designed for low-power real-time sensing of the number of occupants in indoor spaces. AURES utilizes a small ultrasonic bandwidth just above human hearing range to sense occupancy silently. Figure 3.1 shows an overview of AURES where a tweeter transmits an ultrasonic signal into a room and a co-located microphone is used to receive the reflected signal. An electronics package is responsible for generating the signal, processing the reflected signal, and harvesting the required energy from nearby light sources.

AURES is equipped with an occupancy estimation algorithm based on the acoustic response of the environment over a range of ultrasonic frequencies. It is well known in the acoustics community that the number of people within a room impacts the reverberation of sound. Reverberation is typically defined by the RT_{60} time constant, which is measured as the amount of time it takes for a signal to decrease by $60dB$ [20] (in early experiments by Sabine at Harvard, this was the amount of sound decrease before organ pipes became inaudible). When designing

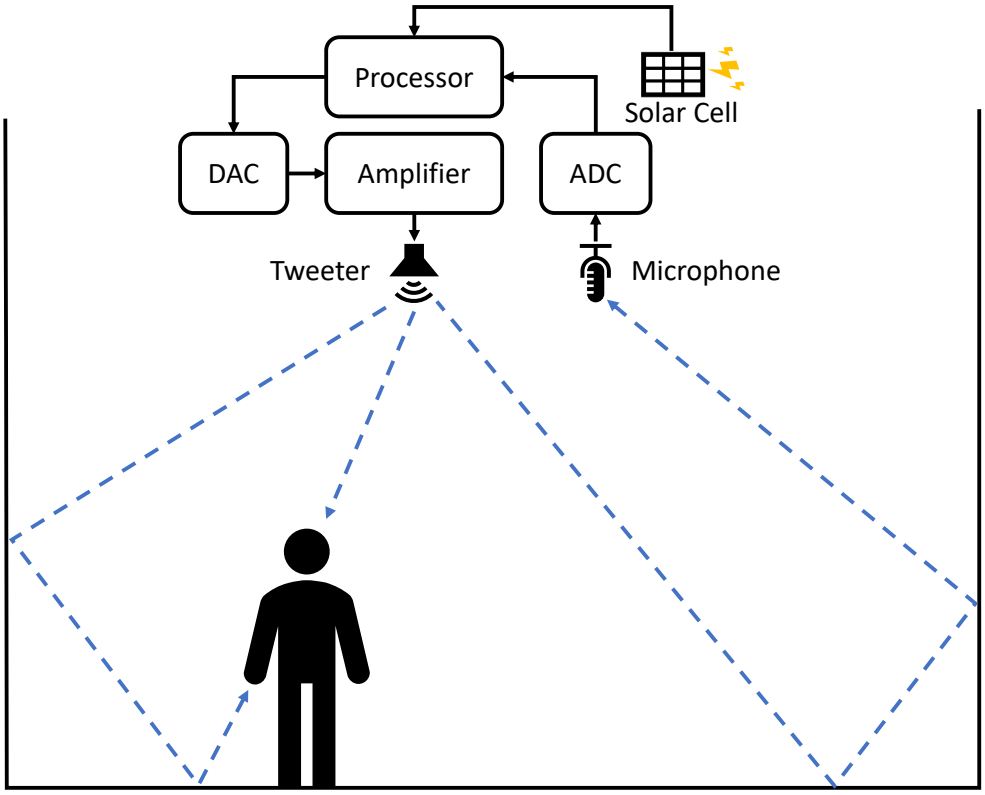


Figure 3.1: AURES system overview.

concert halls, musicians quickly realized that not only did the number of people in the audience significantly impact reverberation, but it was also frequency dependent. People in the audience act like sound absorbers which reduce the amplitude of reflections. As early as the 1890s, Sabine began to model the impact of people, frequency, and the absorbing surfaces of spaces on reverberation [114]. Extensive follow-on work has shown that the influence of audience on reverberation can vary depending on the geometry and area occupied by the audience. Many concert halls have been designed to sound their best when full of people and don't sound nearly as good when empty. AURES leverages the change in this reverberation phenomena in the ultrasonic frequency range as an inaudible way to accurately estimate occupancy.

To quickly measure the acoustic response of an environment, AURES transmits

a wide-band ultrasonic chirp (see Section 3.2) into a room and processes the superposition of the reflections recorded by a microphone (see Section 3.3). When a room is occupied, sound impulses dissipate faster over time and result in a shorter reverberation time. Since reverberation is frequency dependent, the dissipation time across multiple frequencies provides temporal and spectral features tied to the room occupancy level (see Section 3.3.2). By analyzing the frequency response over the chirp's bandwidth at different occupancy levels, we are able to extrapolate the response as the number of people in the room changes. We apply a semi-supervised machine learning approach that models the acoustic characteristics of the room under multiple loads with few labeled training data (see Section 3.4).

One of the key techniques for maintaining performance even when features of the environment change, like when furniture moves, is to let the system periodically recalibrate on a known occupancy level (see Section 3.5). This is achieved through two main phases of operation: presence detection and occupancy counting. In the first phase, AURES detects the presence of people using three different classifiers, and in the second phase, it estimates the number of occupants using the trained regression model. AURES uses multiple transmissions of a single frequency tone in order to measure Doppler shift, changes in signal amplitude, and changes in signal energy within a short time window. The presence detector combines these three classifiers to identify both sudden movements and static changes in the presence of occupants. These presence features are general enough to be used in different indoor environments without training on known data or assuming prior knowledge. If the room is classified as empty in the first phase, then the received signal in the second phase is used to recalibrate the trained model for occupancy estimation in

order to adapt to changes in the environment.

Another main challenge when installing occupancy sensors is the cost of running power and data cables. Many motion detectors can wirelessly transmit data to gateway nodes within a building. Some of these sensors can even operate for extended periods (years) off of batteries. Unfortunately, these systems only detect motion and cannot count the number of people in a space. More sophisticated occupancy estimation sensors like PIR arrays or smart cameras currently consume too much power to make prolonged battery operation feasible. Unlike PIR motion detectors, occupancy estimation sensors are significantly more difficult to aggressively duty-cycle, since they often resort to tracking or frame differencing, or have long warm-up and configuration times. The AURES platform is designed with an energy-harvesting sub-system that can power the system and charge onboard batteries using indoor light sources (see Section 3.6). A typical use-case is to place a solar panel inside a recessed lighting or fluorescent fixture and then run the low-voltage wire (which does not require a commercially certified electrician) to the main AURES module mounted nearby on the ceiling. In drop-down ceiling tile installations, the majority of the transducers can sit on the top of the tile, with the ultrasonic transducer protruding through the tile. In combination with our improved algorithm that can run on a microcontroller, this makes for an extremely effective, low-cost, and easy-to-install sensing package.

3.2 Impulse Signal

In order to efficiently collect the response of the environment over a range of frequencies, AURES utilizes a sinusoidal signal that linearly increases in frequency.

These types of signals are commonly known as *chirps*. Much like how the lens of a camera controls the quality of a photo, the characteristics of a chirp determine the acoustical information embedded in the received signal. In this section, we evaluate how these characteristics affect our system performance.

3.2.1 Ultrasonic Chirp

Chirps exhibit pulse compression, which is a common technique often used in SONAR and RADAR systems to improve the ranging resolution. By nature, chirps have a high correlation with themselves, and can be easily detected with an increased SNR at the receiver. Since the chirps naturally sweep across a frequency range, this allows us to conveniently collect the reverberation characteristics across a larger bandwidth in a single operation. In fact, the same approach can also be observed in nature. A number of bat species emit short but broadband ultrasonic signals in order to differentiate the texture of their prey or plants by the interference pattern reflected in echoes [95, 116, 130]. Our algorithm aims to mimic this instinctive behavior by leveraging ultrasonic chirps to learn the absorption patterns for human bodies. While bats can only recognize their targets within a short distance using the first reflection, we apply the same idea to the reverberation composed of dissipating multipath reflections and in turn extend the sensing range.

3.2.2 Bandwidth and Chirp Length

Since reverberation is a function of frequency, one would expect that a chirp's frequency and duration have a direct impact on the performance of the system.

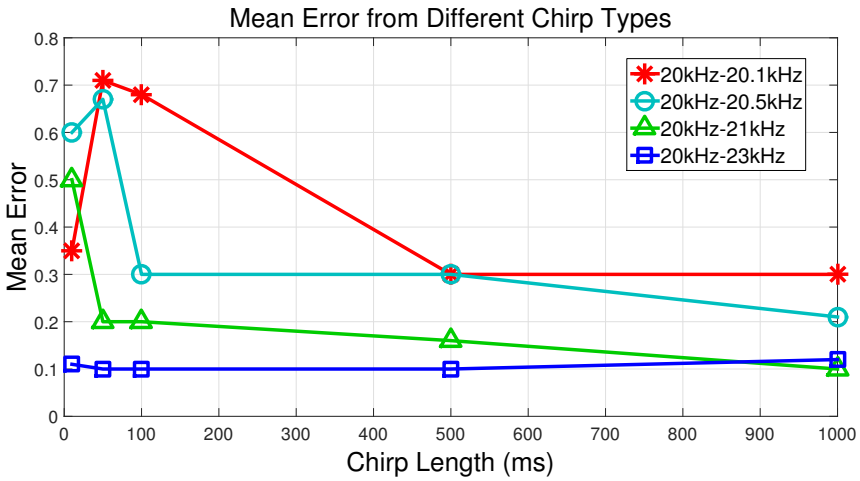


Figure 3.2: Impact of chirp length on classification error.

Given more bandwidth, we should be able to collect more reverberation characteristics as the signal dissipates. The length of the chirps define the resolution of the frequencies we can acquire given a fixed sampling rate. By design, we limited the frequency bandwidth to the ultrasonic range, such that it is inaudible to humans.

In order to test the impact of bandwidth and chirp length, we collected more than 100 points of data for 0–5 people at four different bandwidths and five different chirp lengths, for a total of 8000 samples. In Figure 3.2, we show the sensitivity of chirp length and bandwidth on our classifier. An important trend to see is that the performance is proportional to a bandwidth and time product. Picking the minimum length and bandwidth helps scope the hardware requirements and maximizes sensing rate. Based on our experiments, the chirps with a bandwidth of 20–23kHz and a length of greater than 300ms gives the best performance. Note that the upper bound of $23k$ is also considered the highest frequency most common (non-ultrasonic) speakers can support.

While larger bandwidth and longer chirp length are better, we use a rather

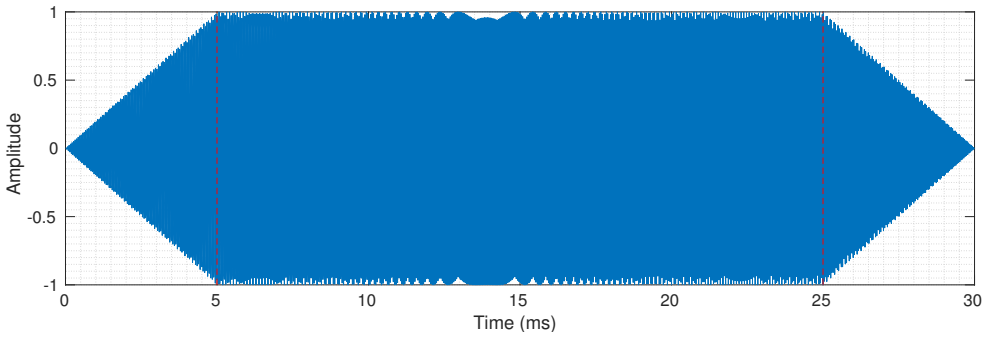


Figure 3.3: The transmitted chirp has a total length of 30ms including 5ms of fade-in time and 5ms of fade-out time.

short chirp length of $30ms$ instead in order to prevent crosstalk between a co-located transmitter and receiver on an embedded platform. We will discuss more details on how to compensate the resulting performance loss when constructing the training features in Section 3.3.2. As studied in [78], many tweeter speakers exhibit non-ideal impulse responses that can result in audible artifacts like clicking sounds. To alleviate this problem, we add $5ms$ of fade-in and fade-out time to the chirp's ramp up and ramp out time. The interval between each chirp is set to $500ms$, allowing the chirp to fully dissipate in the room. This is significantly longer than the reverberation time derived from the Sabine and Eyring equation [75]. We show the transmitted ultrasonic chirp in the time domain in Figure 3.3 and its spectrogram in Figure 3.4 respectively.

3.2.3 Sampling Rate

The minimum sampling rate to support the system is an important factor driving both the cost of the hardware components and the computational requirements of receiving the signal. Generally, normal commodity audio equipment designed for music only supports sampling rates up to $48kHz$. Also, the dispersion pattern of a lower ultrasonic frequency tends to be more omnidirectional. As shown in

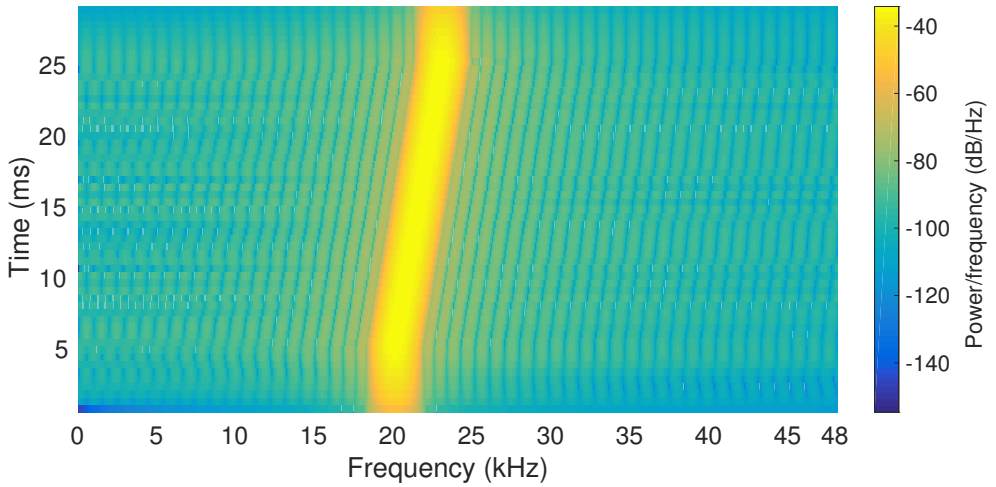


Figure 3.4: The linear chirp starts at 20kHz and crosses 23kHz at t=30msec.

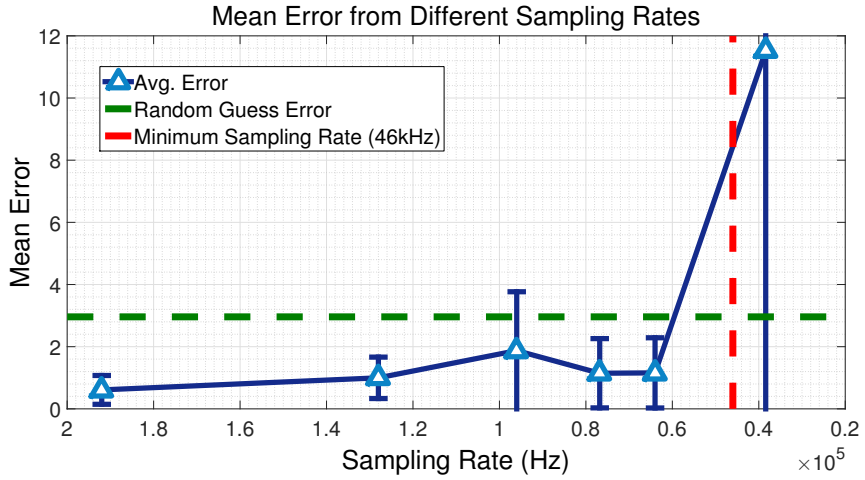


Figure 3.5: Impact of sampling rate on classification error.

Figure 3.5, a higher sampling rate has a slightly better overall performance, and large error is expected when the sampling rate drops below the Nyquist limit (the lower-bound sampling rate for alias-free signal sampling). The interesting point to note is that the performance does not significantly increase with much higher sampling rates than the input audio signal. This support the notion that our features are based on the decay within our frequency band (see Section 3.3.2).

3.3 Preprocessing

Before attempting to classify data, the raw signals are pre-processed to minimize noises caused by multipath or any audio sources to improve prediction accuracy (see Section 3.3.1). We then discuss how to properly generate the training features given the constraint of the co-located speaker and microphone (see Section 3.3.2).

3.3.1 Matched Filter

We assume that the transmitted signal goes through an additive white Gaussian noise (AWGN) channel while disseminating in the room. By this assumption, the matched filter is known to be the optimal receiver filter to increase the signal-to-noise ratio (SNR) of the received signal. In general, the signals can be represented as

$$y(t) = h(t) * x(t) + n(t) \tag{3.1}$$

where $y(t), x(t)$ is the received signal and the transmitted signal, $h(t)$ is the impulse response of the room, and $n(t)$ are the background noises. Since the transmitted signal is known and $h(t)$ is the target of interest, we matched filter the received signal with the original transmitted signal to maximize the SNR. A high SNR of the received signals is vital for the later analysis with machine learning techniques, which identify the most important characteristics in the frequency changes that differentiate the signals of different occupancy levels.

The matched-filtered signal is then transformed into the frequency domain using Fast Fourier Transform (FFT), and band-pass filtered to remove noise from other acoustic sources. The filter's bandwidth is exactly the same as the chirps'

sweeping bandwidth. Transforming into the frequency domain also helps to reduce the dimensions of the collected data and minimize the training time and complexity.

3.3.2 Training Features

After matched-filtering, the training features are directly extracted from the processed data capturing the frequency response of the chirp's bandwidth. As previously discussed in Section 3.2.2, we showed that the chirp's frequency and duration have a direct impact on the performance of the system, where increasing the chirp's frequency band and length improve the system performance. However, when building the platform, we found that if the transmitter and receiver are physically close, then the system suffers from crosstalk. Recording after playback, in turn, defines an upper bound of the chirp length which must now be much shorter (originally 300ms, now 30ms). To compensate for the performance loss, we split the received signal into segments that have the same length as the chirp to increase the temporal diversity of the features. Each segment is transformed into the frequency domain individually and later combined together to form the training features. This provides us with additional amplitude data across each segment. Since the chirp is much shorter than the recording, this approach generates features that not only better capture how the sound dissipates over time in amplitude, but also greatly reduce the memory required to perform the FFT. To prevent bias between features when performing WPCA (see Section 3.4.1), all features are later normalized and subtracted by their means.

Figure 3.6 provides a simple example of the types of features the algorithm is trying to identify. In the figure, we show the averaged spectrum over all segments

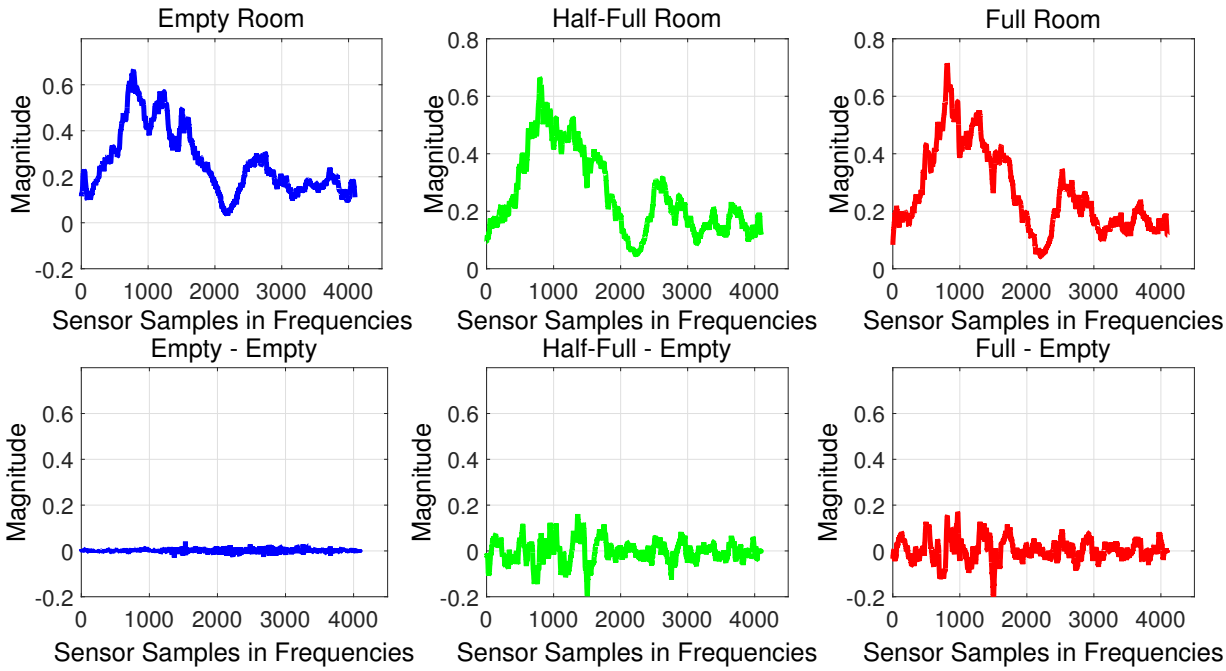


Figure 3.6: Raw features for empty, half-full, and full room scenarios.

for better visualization. The top row shows the filtered spectrum after matched filtering of an empty room, a half-full room, and a full room. The bottom row of the image shows the difference between each top image and the empty room sample. For example, an empty room shares little difference with another empty room and hence we find almost no changes in the signal. However, in the case of a half-full room and a full room, we see a significant difference. It is worth noting that the difference between a half-full room and a full room is much more subtle.

3.4 Occupancy Estimation Algorithm

The occupancy estimation algorithm is composed of two parts. In the first part, the Weighted Principal Component Analysis (WPCA) is performed on the training dataset that contains data points collected from different occupancy levels. It allows us to learn which of the principal components best characterize the

absorption pattern of human bodies and reduce the dimensionality of the received data. We assign lower weights to the empty room instances, which are identified by the presence detector later discussed in Section 3.5.1, so that WPCA is biased toward finding principal components that best explain the variance between different occupancy levels. Once we have decomposed the signal into weighted principal components, we use a DBSCAN clustering algorithm [83] to cluster each identifier in a low-dimensional space to reduce the impact of noise.

In the second part, a regression model is built based on the projected data in order to interpolate/extrapolate the occupancy beyond the training data. This improves scalability and eliminates the need for copious amounts of labeled training data. We derive the relationship between the number of people, which can be seen as the absorption material in a room, and the amplitude difference in frequency with the help of the Sabine equation and reverberation properties found in [114]. To obtain the best prediction function and the estimated occupancy level for each cluster, we design a loss function to be minimized based on several heuristics. In order to speed up the process and improve the performance of fitting, we assume the maximum capacity of the room is given and the data collected should contain instances of at least half of the maximum capacity. This can be achieved by setting up a data collecting period, such as a day, in the system for bootstrapping before running the estimator. The idea is to have a self-learning system that requires minimal training effort and is capable of training itself as more data is collected and learned over time.

3.4.1 WPCA

Before learning the human absorption pattern from the spectral features, we first need to reduce the correlation between the feature vectors. There are a few techniques that can be applied for this purpose, such as PCA, ICA, LDA, and autoencoder. After evaluating these methods, we determined PCA-based approaches to be the most suitable techniques for several reasons. First, since the training dataset is typically collected from a continuous period, the occupancy value follows a stochastic process with high dependency between the data instances, which makes techniques such as ICA a poor fit. Second, in order to minimize the training effort, algorithms such as LDA that rely on labeled data would be less suitable. In addition, we find autoencoder and other non-linear transformation methods less favorable due to the existence of numerous unknown latent variables in the dataset. For example, using autoencoder, in some datasets we find ourselves learning features corresponding to the geographical distribution of the occupants. However, with limited ground truth labels, it is difficult to identify the nature of the learned features, or rule out features that induce noise to occupancy estimation. On the other hand, people in a space significantly impact the reverberation and spectral amplitudes, which makes PCA-based approaches good candidates.

Another important property of PCA is that it determines the ranking of each independent component based on the magnitude of its corresponding eigenvalue. The ranking can then be used for dimensionality reduction that is useful for processing high-dimensional datasets while preserving as much variance between the data as possible. This allows us to build an efficient model in terms of computation and memory consumption. In addition, dimensionality reduction

helps in dealing with independent and identical Gaussian noise among the dataset. When applying PCA, we project the n-dimensional spectral features into an n'-dimensional space, where $n' \leq n$ and all variables in the new space are linearly uncorrelated with each other. The projection is achieved using the first n' principal components for transformation where the first principal component is defined as the variable that gives the maximum possible variance in the dataset. Since most variance is concentrated in the first few principal components, the effect of constant noise variance is proportionally less after the projection, allowing higher SNR in our spectral features.

However, when applying a vanilla PCA, we inherently assume the whole training dataset should be collected in background environments with almost identical reverberation characteristics. If the acoustic response of the environment changes dramatically during data collection, which is likely to happen in practice, then PCA can perform poorly. The resulting PCA can erroneously produce principal components that explain the changes in the environment, rather than the desired ones that differentiate the varying occupancy levels. To solve this problem, a weighted variation of PCA (WPCA) is adopted to target components that separate occupancy levels. While classical PCA is known to be sensitive to outliers and missing data, WPCA increases the robustness of the system to outliers by assigning different weights to data points based on their estimated relevancy. For our application, we borrow the same idea to minimize the influence of the changing environments by giving the empty room data points a lower weight.

We assume the training dataset is given by the matrix X, where each of the i rows represents a feature variable and each of the j columns represents an observation.

The goal of a classical PCA is to find a decomposition of the matrix

$$X = PC \tag{3.2}$$

where P is the orthogonal matrix of principal component and C is the principal coefficient matrix, such that the matrix D given by

$$D = P^T X X^T P = P^T \sigma^2 P \tag{3.3}$$

is diagonal and has its variance maximized. The diagonals of D are often rearranged in order such that $D_{ii} >= D_{jj}$, $\forall i < j$ so that the first column of P represents the first principal component that accounts for the most variance. Equivalent to maximizing the variance, the principal components allow us to minimize the reconstruction error $\|X - PC\|_2^2$ when the data is projected into a lower-dimensional space. Similarly, the goal of WPCA is to minimize the weighted reconstruction error given by

$$\|W(X - PC)\|_2^2 = \sum_{ij} W_{ij}^2 (X_{ij} - PC_{ij})^2 \tag{3.4}$$

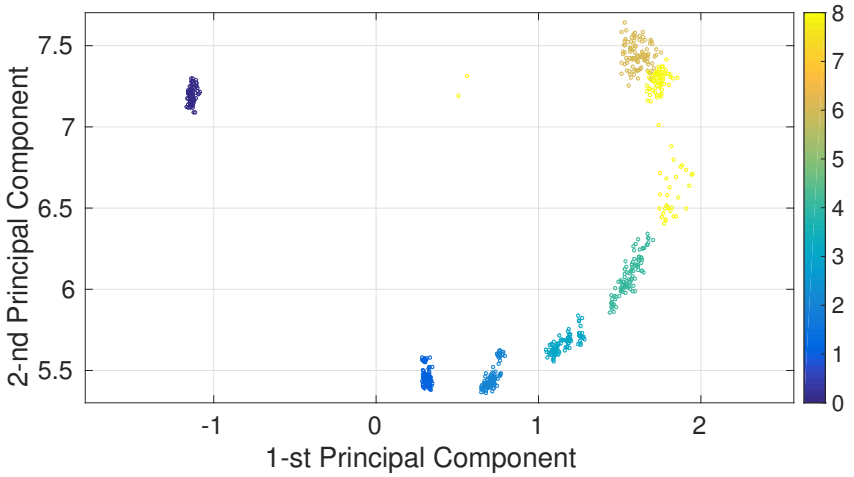
By assigning lower weights $w_j < 1$ column-wise to the empty room instances, which are identified by the presence detector (see Section 3.5.1), WPCA is biased toward finding principal components that best explain the variance between different occupancy levels. Other non-empty room instances are assigned with a fixed weight $w_j = 1$ to prevent bias. To center the dataset and calculate the

covariance matrix, the weighted mean to be subtracted is given by

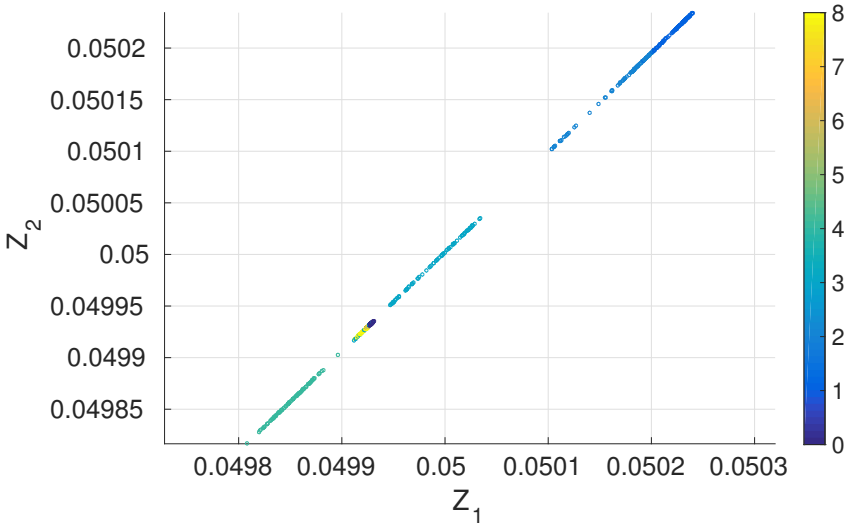
$$\bar{x} = \frac{\sum_j W_j X_j}{\sum_j W_j} \tag{3.5}$$

where X_j denotes the jth column of the dataset X. While lowering the dimensions of data reduces the overall complexity, more information is lost during the transformation process. Ideally, $n' \geq 5$ gives the best performance in clustering result (see Section 3.4.2) based on our empirical experiment, and the corresponding eigenvalue ratio representing the ratio of variance kept after transformation is around 25%. More evaluation on how to select a proper weight along with varying output volume is later discussed in Section 3.6.3.

In Figure 3.7, we show illustrations of the processed spectral features in 2D projection and compare the results derived from WPCA and autoencoder. This dataset is collected in a small room environment for 0–8 people and its colors reflect the occupancy levels. The autoencoder is composed of 3 hidden layers of size 100, 10, and 3 respectively and trained using a *sigmoid* activation function with regularized MSE loss function. We observe that while the autoencoder can obtain highly non-linear representations, the derived features may not correlate with the occupancy levels in a predictive manner. On the other hand, by maximizing the variance between the data, WPCA retains components that differentiate the occupancy levels. As a result, a better prediction accuracy is achieved when we adopt the linear analysis in this initial step, but introduce non-linearity to our algorithm later in the regression model (see Section 3.4.3).



(a) The WPCA projection using the first two principal components.



(b) The encoded data representation derived by an autoencoder of three hidden layers.

Figure 3.7: Visualization of spectral features processed by WPCA (Top) and autoencoder (Bottom).

3.4.2 Clustering

Once we have extracted the principal components of the signal, we cluster each identifier to reduce the impact of noise and outliers. Due to the projection of WPCA, each cluster of data can take arbitrary shape in the new space with varying density. The DBSCAN clustering algorithm [83] has been widely used in this

manner with high robustness to outliers and zero prior knowledge of the number of clusters. Moreover, we do not want to assume any prior distributions of people in the room, since the real distribution can vary from day-to-day and largely depends on the usage and functionality of the room. These properties of DBSCAN allow us to cope with noise caused by the different distribution of bodies in the room and successfully categorize the data with high accuracy. The DBSCAN algorithm can be summarized in the following steps:

1. For every point, connect all its neighbors within a given neighbor distance ϵ, and mark it as a core point if it has more neighbors than a given minimum neighborhood points.
2. For every core point, find the connected components on the neighbor graph (can be core or non-core) and form a cluster.
3. All points not reachable from any other point are outliers.

Note that these steps can be performed in iterations for one point at a time to save memory. Since each point will perform exactly one neighborhood query, which can be done in $\mathcal{O}(\log n)$, the overall average runtime complexity is given by $\mathcal{O}(n \log n)$.

One limitation of DBSCAN is that the clustering result is sensitive to the minimum neighborhood points and neighbor distance ϵ. In order to reduce the indeterministic outcomes and improve the quality of DBSCAN, each collected data point consists of multiple samples with a known number of chirps. Different neighborhood distances ϵ are also evaluated based on the intra-cluster distance derived from the training data, and the most frequent combination is selected as the clustering result.

One primary reason to cluster data before performing regression is to improve the prediction accuracy, especially for smaller room environments. In most of the

scenarios, the dataset is quite noisy in general and often overlaps with itself even in a high-dimensional space. By clustering the data into groups and removing outliers, the accuracy in regression is drastically increased, especially in cases with few people where we expect high granularity. Also, the computational complexity is greatly reduced since only the cluster representations are used in building the regression model instead of the raw dataset. The clustering algorithm also benefits from the chirps' physical characteristics. When using chirps with larger bandwidth, more reverberation information across the frequency band is learned in the training process. As a result, the density of each cluster is higher and inter-cluster distance is greatly increased in the observed data. Figure 3.8 shows the 2D WPCA projection with the clustering result. Each color and marker type reflects the clustering of different occupancy levels. Most of the clusters are correctly categorized except for a few points that are associated with the eight people case due to noise. In the figure, we can also see that as the number of people in the room increases, the dynamic distribution of people leads to a higher variance in the clusters. On the other hand, in larger rooms such as an auditorium, DBSCAN can be less successful in giving a conclusive clustering result due to excessive scattered data points. However, these scenarios are often the ones where the clustering algorithm will contribute the least to the results because the granularity of the estimation is relatively less important. The estimate will then rely mainly on the regression model, as discussed in Section 3.4.3.

3.4.3 Regression Model

In order to interpolate occupancy beyond the training data, we build a regression model based on only two labeled training points. One data point is when the room

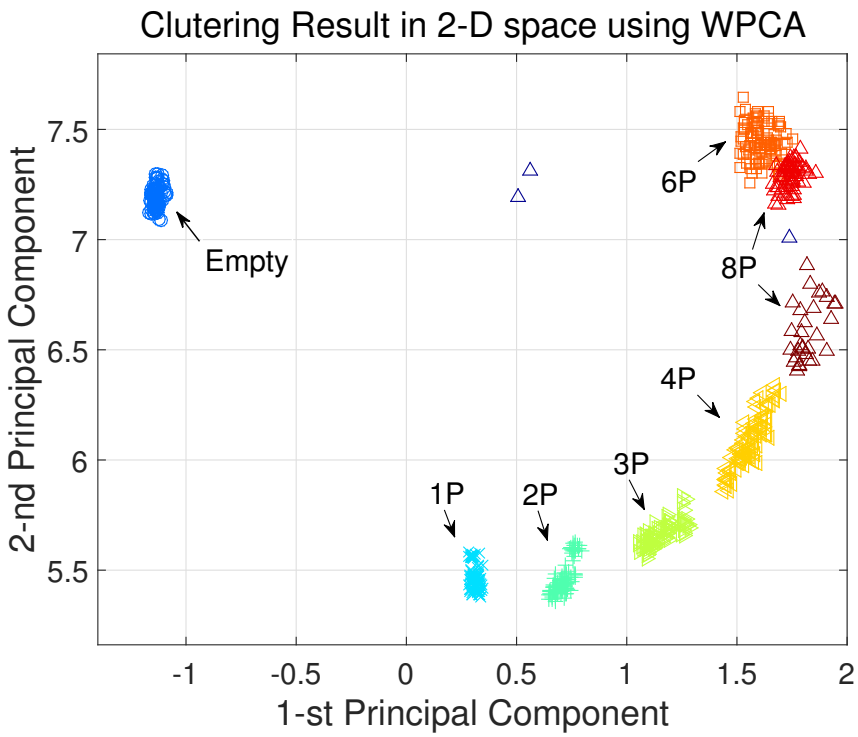


Figure 3.8: Clusters of different numbers of people in a small conference room shown in 2D principal component space.

is empty, while the other data point should be at a reasonable occupancy level ($\geq$ 10%). Here, we derive the relationship between the number of people, which can be seen as the absorption material in a room, and the amplitude difference in frequency with the help of the Sabine equation and reverberation properties found in [114]. As shown by the Sabine acoustic model (Equation 3.6), the duration of the audibility of the residual sound, namely the reverberation time (RT), follows a rectangular hyperbola curve against the total absorbing material. Here c_{20} is the speed of sound at 20 degrees Celsius, V is the volume (m^3) of the room, S is the total surface area (m^2) of a room, and a is the average absorption coefficient of room surface.

$$RT_{60} = \frac{24 \ln 10}{c_{20}} \frac{V}{Sa} \simeq 0.1611 \frac{V}{Sa} \tag{3.6}$$

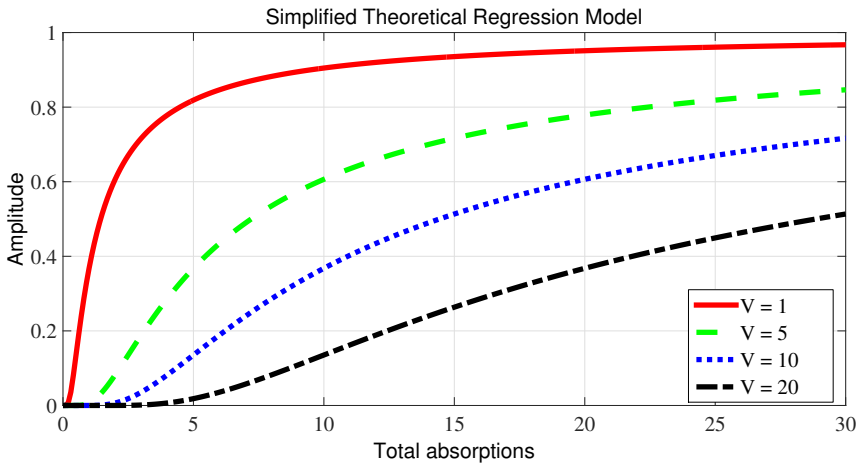


Figure 3.9: Theoretical regression trends with different room volumes based on Equation 3.8.

Since the RT is defined by the time for a signal to decay by a certain decibel (dB), we get

$$RT \propto \log(\frac{A_0}{A_m}) \tag{3.7}$$

where A_0 is the constant initial amplitude of the sound source and A_m is the measured amplitude after absorption. Combining Equation 3.6 and Equation 3.7, we obtain the relationship between the observed frequency amplitude and the number of people as

$$A_m \propto e^{\frac{-C_0 V}{Sa}} \tag{3.8}$$

As plotted in Figure 3.9, we can see that when the volume of the room is small, the curve tends to be similar to an exponential regression. However, as the volume of the room increases, the curve becomes smoother and more linear in regression. The size of the room can be estimated to help choose the best starting model. To calculate the amplitude difference, we first set the center of the empty room dataset as the new origin of the projected space, and for every cluster we calculate how far they are from the origin. We tested with multiple distance metrics and decided that

Chebyshev distance provided the best fit to the regression model shown across our overall data, which is defined as

$$D_{chebyshev}(a,b) = \underset{1\leq i\leq n}{max}(|a_i - b_i|) \tag{3.9}$$

where a, b are two arbitrary n-dimensional data points. The *unit distance* is further calculated based on the average of the pairwise-distance between the two training datasets, where the *unit distance* is namely the reference distance between N and $(N+1)$ people instance. Next, we estimate each cluster by fitting its distance to the origin to the regression model. By finding the variable that changes the most among all the data, which we note here is derived from a linear combination of all the variables in the original space, we capture the feature that differentiates the data the most and use it as a measurement to estimate the occupancy level.

Based on the observation from Equation 3.8, an exponential regression model (Equation 3.10) is adopted with the distance value as the function input. To adapt the model to varying estimation range, we define an exponential loss function to estimate the most likely capacity combination for each cluster. The loss function is defined in Equation 3.11 and Equation 3.12:

$$f(x) = \alpha e^{\beta x} \tag{3.10}$$

$$\hat{f} = \underset{\alpha,\beta}{argmin} \sum_{i=1}^{n} e^{W_i \phi(x_i)} \tag{3.11}$$

$$\phi(x) = f(x) - round(f(x)) \tag{3.12}$$

where n represents the total number of clusters, W_i is the weight of cluster i, and x_i is the distance between the cluster i and the origin (the empty room). The

weight of each cluster W_i is proportional to the number of data points in the cluster, and additional weights are assigned to the clusters with known training labels. This allows the curve to be fitted to the most important clusters and prevents overweight in outliers. Additionally, the function $\phi(x)$ tends to fit the curve in a way that the predicted number of people is close to an integer. By minimizing the loss function, we obtain the best prediction function $\hat{f}$ with corresponding parameter $\hat{\alpha}, \hat{\beta}$, and the estimated occupancy level for cluster i is assigned accordingly by $\hat{f}(x_i)$. To speed up the process and improve the performance of fitting, we assume the maximum capacity of the room is given and the data collected should contain instances of at least half of the maximum capacity. This can be achieved by setting up a data-collecting period, such as a day, in the system for bootstrapping before running the estimator. The idea is to have a self-learning system that requires minimal training effort and is capable of training itself as more data is collected and learned over time. It is worth mentioning that with more given training points, a more sophisticated regression model can be adopted to improve the accuracy of the prediction. However, one of our goals in this dissertation is to minimize the training effort from the user to improve the feasibility and scalability of the system.

In Figure 3.10, we show an example of the estimated occupancy made by our regression algorithms in a small room scenario. Each data point represents the estimation for an entire cluster consisting of at least 100 sample points. In general, the error slightly increases as the room size gets larger, but we are still able to achieve an error of fewer than 2 people from the average ground truth.

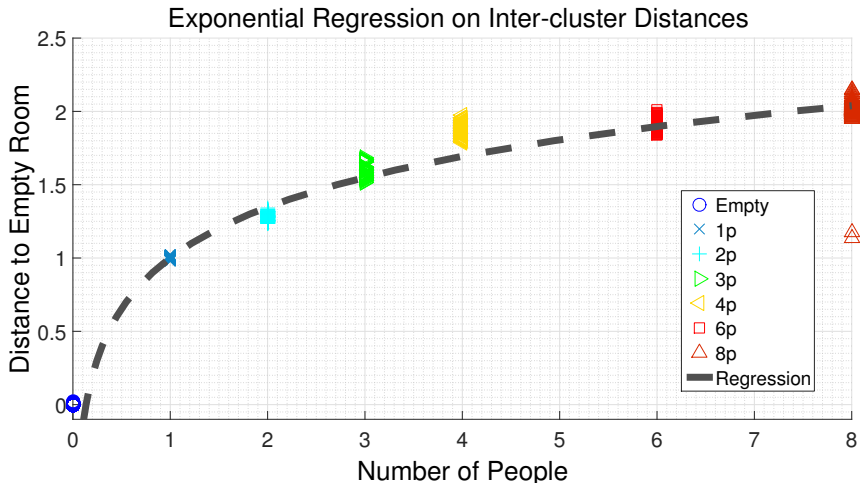


Figure 3.10: Adaptive exponential regression for occupancy estimation in small room scenario.

3.4.4 Training Point Selection

The selection of the second training point can also affect the result dramatically in certain cases. The training point consists of a single person or a group of a few people is typically ideal for small and medium room scenarios. However, as shown in Figure 3.11, using a small group of people as the training point in large rooms is likely to cause significant estimation error. The error comes from the fact that such changes in frequency magnitude are not strong enough to be fully captured. A training point of a group of eight people or more in a 150-person room gives a similar result with 5% error on average. Based on our experiments, training points of at least 10% of the maximum capacity work well.

3.5 Auto Recalibration

To prevent retraining from scratch every time the background environment slightly changes, the system requires a mechanism to slowly recalibrate itself over time. In

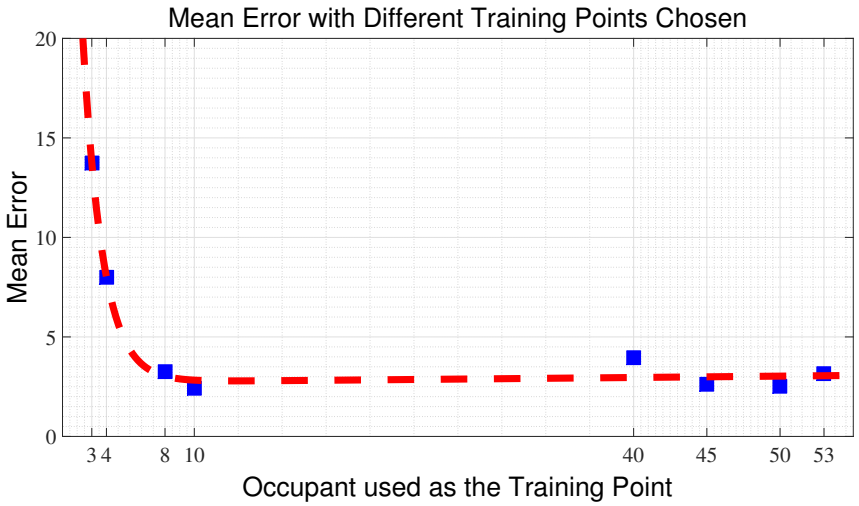


Figure 3.11: Mean estimation error based on different numbers of occupants used as the training point in a 150-person room.

this section, we propose an automatic recalibration mechanism that first detects the binary presence of occupants (see Section 3.5.1). If the room is identified as empty, the trained model is updated dynamically by analyzing the frequency response of the new environment and selecting the best principal components that represent it (see Section 3.5.2).

3.5.1 Presence Detection

The ability to detect whether a room is empty not only improves the quality of WPCA (see Section 3.4.1), but helps to determine when the system should recalibrate. To automate the recurring recalibration process, we proposed using a single tone instead of a chirp to facilitate the detection of Doppler shift. In each sensing period, the system transmits 5 consecutive tones with a delay of $300ms$ in between to allow echoes to fully dissipate. The received signals are first transformed into frequency domain and then filtered to remove out-of-band noise. The presence detector is composed of three binary classifiers (empty or

Param. / Sizes	Accuracy	FP	FN	Precision	Recall
Small room(s)	0.85	0.11	0.09	0.90	0.91
Medium room(s)	0.82	0.27	0.12	0.76	0.88
Large room(s)	0.75	0.29	0.21	0.72	0.79

Table 3.1: Presence detection performance with different room sizes.

non-empty), where each focuses on different features of the received signal. Note that since presence detection is now part of the people-counting algorithm, the features and mechanisms used in presence detection are independent from those used in the determination of occupancy level. The first classifier is a Doppler motion detector that detects Doppler shift caused by the movement of bodies or gestures. Even though Doppler detectors work well at detecting sudden movements, it is often difficult to detect static changes such as different postures of the occupants or slow motions. To improve performance, we apply two additional classifiers to calculate the variance of the spectral amplitude and the variance of the received signal energy, respectively. Tuning the threshold of each classifier allows us to control the ratio between false-negative rate (FNR) and false-positive rate (FPR). To prevent the system from recalibrating on non-empty data points, lowering the rate of getting a negative feedback while the room is occupied is critical. Recalibrating on FN instances offsets the baseline of the model and introduces substantial estimation error that would last until the next recalibration cycle. On the other hand, FP instances trigger the system to make an estimation on the new environment, which does not introduce much estimation error in comparison since the model is trained to detect human bodies that absorb more power. Therefore, the thresholds in all three detectors are tuned to be conservative, and the final decision is obtained by taking an OR operation between the three binary results in order to achieve a low FNR.

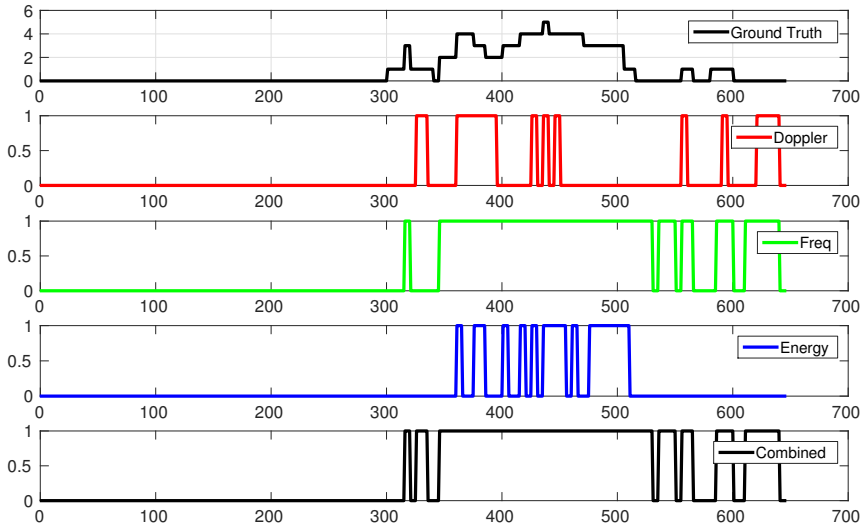


Figure 3.12: Presence detection result compared between the ground truth and the three classifiers with one day of empirical data.

In Figure 3.12, we show the classifiers' sensitivity to the room occupancy level. We can see that the Doppler-based classifier is sensitive to movements regardless of the number of occupants in the room, while the variance-based classifiers are more accurate when there are more occupants. The overall performance of the presence detector is summarized in Table 3.1, which includes the accuracy, false positive rate (FPR), false negative rate (FNR), precision, and recall. We see that the overall accuracy decreases as the size of room increases, which is not surprising since multipath reflections are much weaker and noisier in large spaces. In our ten different room environments, each classifier has an accuracy of 65–75% on average, but when combined, the overall accuracy increases to 80%. Since the detector is designed to reduce false positive instances, we are able to achieve a recall of 85%. For the remaining 15% false positive instances, we analyzed the distribution over the number of occupants in different room environments to see the negative impact on occupancy estimation. Figure 3.13 shows the FNR as the number of occupants increases. We see that the detector suffers the most from

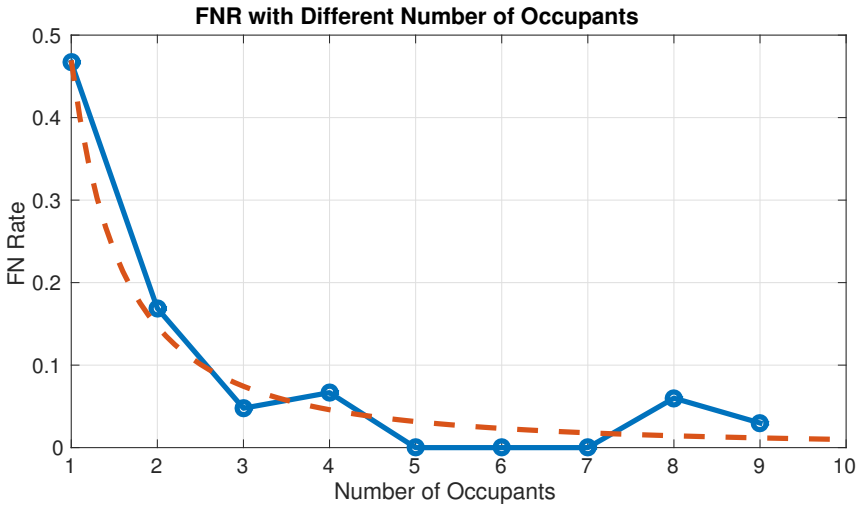


Figure 3.13: False negative rate in presence detection decreases exponentially as the number of occupants increases.

single person instances, especially in cases where the only person is still, as when typing or using a laptop. However, the false positive rate decreases exponentially as the number of occupants increases. This indicates the introduced error on the successive occupancy estimation is minimal, even if the system erroneously recalibrates on false negative instances. Also, it should be noted that in practice, we can further improve the detection accuracy by extending the sensing period and/or increasing the number of tones used for detection.

3.5.2 Recalibration Algorithm

Whenever the room is detected as empty, AURES begins collecting new data until a certain number of samples is reached or the room becomes occupied. The frequency response upon the new environment is then used to calibrate the model by updating the selection of the principal components. Given the old empty room dataset X and new empty room dataset X', we find the largest subset P' of the original principal components P such that the mean distance between the two

empty room datasets in the space spanned by P' is below a threshold, which can be represented as

$$\underset{|P|}{\text{argmax}} \qquad \sum |(X - X')P| \leq d \tag{3.13}$$

where the threshold d is determined based on the intra-clustered distance of the old empty room dataset. The idea behind the algorithm is to preserve as many principal components as possible, while minimizing the delta between the two environments. Since the projection may alter the magnitude of the raw data, the *unit distance* is scaled accordingly based on the eigenvalue ratio of P'. The new model will then have the origin XP' and the estimation can be made by applying the same regression model. In this manner, the system is able to retrain when the environment changes using only empty room training points.

To evaluate our automatic retraining technique, we collected three weeks of data in a noisy semi-opened laboratory environment (shown in Figure 3.24a), which frequently changed due to everyday use. We show the estimation traces of the first five days of the collected data in Figure 3.14, where the estimation model is trained using the first 500 samples with two labeled occupancy levels. Without periodic self-retraining, we see an offset of estimation error right after the lab is being used on the first day. Moreover, the error offset begins to accumulate over time and prevents the system from accurately estimating the occupancy levels for the following days. However, when the system retrains itself with presence sensing, it is able to re-zero the baseline according to the new environment sporadically and thus greatly reduce the estimation error. Using our presence detector, the system is able to reduce the mean error from 2 to 0.5 people despite a few occurrences of FP and FN events. In comparison, with a perfect presence detector, the estimation error can be further reduced to 0.3 people. As previously

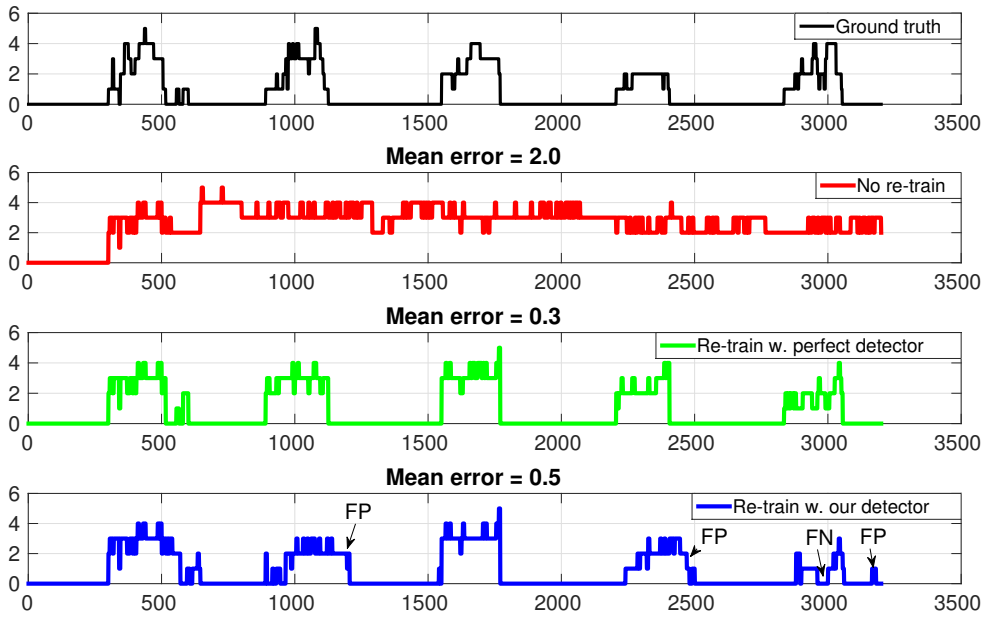


Figure 3.14: Occupancy estimation of five days of empirical data compared between (1) ground truth, (2) no retraining, (3) retraining with a perfect detector, and (4) retraining with our detector.

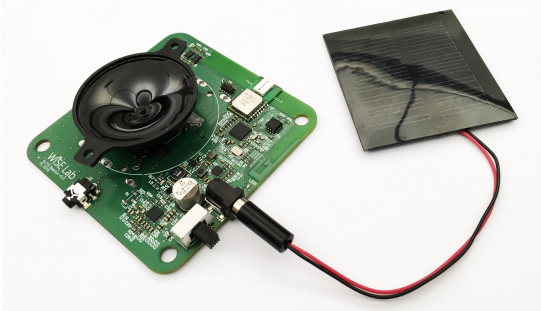
discussed in Section 3.5.1, the amount of improvement the presence detector provides depends mainly on its accuracy and the error distribution of the false negative cases.

3.6 Platform Implementation

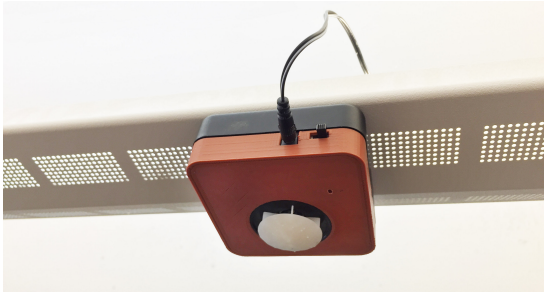
In this section, we discuss the hardware platform and the software processing workflow. This entails how data is captured and passed to a mobile device for installation and training.

3.6.1 Hardware Design

We developed an energy harvesting, embedded hardware platform for our ultrasound transceivers as shown in Figure 3.15a and Figure 3.15b. The platform



(a) Hardware PCB design with external solar panel.



(b) AURES mounted on hanging fluorescent light.

Figure 3.15: AURES hardware design.

was designed to have a low enough power consumption so that it can be powered using a $7 \times 5.5 cm$ solar cell harvesting energy from artificial or natural light sources. This allows for a flexible installation at a low cost, since the transceivers do not need to be connected to AC wall power, which is often difficult to access at ceiling mounting locations.

The hardware platform features a single PCB design, which uses a TI CC2650 multi-standard BLE and 802.15.4 SoC connected to a $192 kHz$ audio codec, a MEMS microphone and a piezo ultrasound speaker connected to a Class D piezo speaker amplifier to transmit and receive ultrasound signals. An ultrasonic horn as described in [80] is attached to the speaker to disperse the emitted ultrasound in an omnidirectional fashion. 2Mbits of onboard SRAM is used to store recorded waveforms before they are processed and the results are sent to a gateway using 802.15.4 or BLE. Figure 3.16 shows a block diagram of the primary components of the hardware platform. The total cost of our current hardware design is around $30 at quantity 1000, including the energy harvesting module.

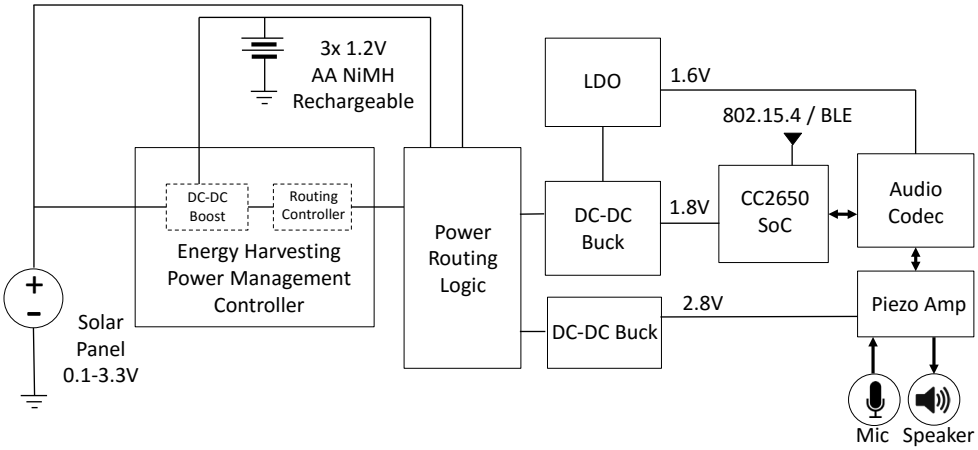


Figure 3.16: Block diagram of AURES main hardware components.

3.6.2 Processing Workflow

A general processing workflow of our system starts with an installation, a training process, and finally, a steady-state with retraining. All processing is performed onboard except the initial training, which is offloaded to a computer for processing due to memory constraint. An installer should first mount the AURES node to the ceiling in a central location with the solar panel near a lighting fixture. The installer can then configure the node using a BLE enabled device, like a smartphone, and bootstrap the volume configuration sequence on AURES where the transmitter profiles the room's SNR. After determining a sufficient volume threshold, the node periodically scans for presence followed by collecting an occupancy reading. Since initially there is no trained model, the node will store the output of the high-pass filtered spectrum response of the chirp in its flash memory as training data. This will be collected over an extended period and eventually all training data is transferred to a phone or computer to perform WPCA and regression. In cases where a gateway is available, this could also be done in a streaming fashion. During data collection, the installer should come back

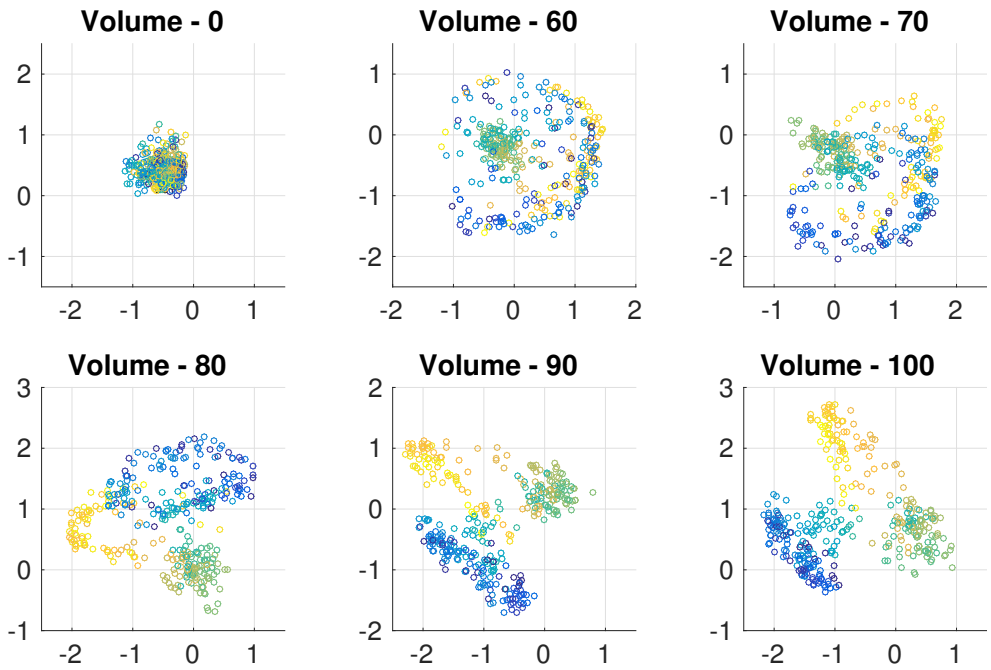


Figure 3.17: Effect of different speaker volumes on data clustering in 2D space derived by WPCA. Different colors reflect different occupancy levels.

periodically to label a subset of the room occupancy levels. In our experiments we used only two labels, but at least one point should be above 10% of the room's capacity, as discussed in Section 3.4.4. When collecting data once every 10 minutes, the AURES node has enough storage to hold two weeks of data in its 4Mbits of flash storage which requires up to 30 seconds to transfer to a phone. The resulting model ($\leq 4KB$) is then transferred back to the node over BLE at which point the system begins executing. The trained model is periodically updated afterward when the room is identified as empty.

3.6.3 Volume Control

Reducing the power consumption is key for building a self-sustained energy harvesting platform. Based on the energy footprint of the device, signal playback and recording are the most significant power-consuming operations. The power

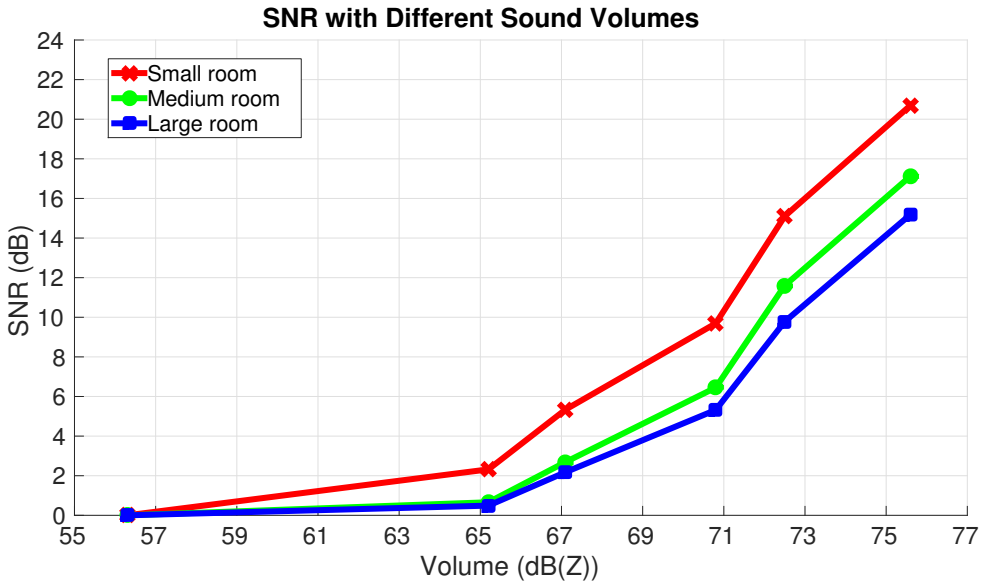


Figure 3.18: Received SNR plotted with different output volumes and room sizes.

consumption of recording is fixed, but the transmit power can be controlled by adjusting the speaker volume. We also generally want to decrease volume for scalability and to improve pet friendliness. Since the system relies on the amplitude of the received signal to estimate occupancy level, we observed a trade-off between the power consumption and the system performance. Figure 3.17 shows how volume impacts the clustering performance of WPCA in one of our test environments. For the purpose of visualization, the data are presented in 2D space using WPCA. Each data point represents an observation and its color reflects the occupancy level. We see that data collected at low volumes are more difficult to be separated by their occupancy levels, while data collected with higher signal strength can be easily clustered.

However, the ideal output power is both environment and room geometry dependent. To better understand how the volume affects the system performance in different environments, we also calculate their corresponding SNR. The duration of the received signal on which we calculate the SNR is an important

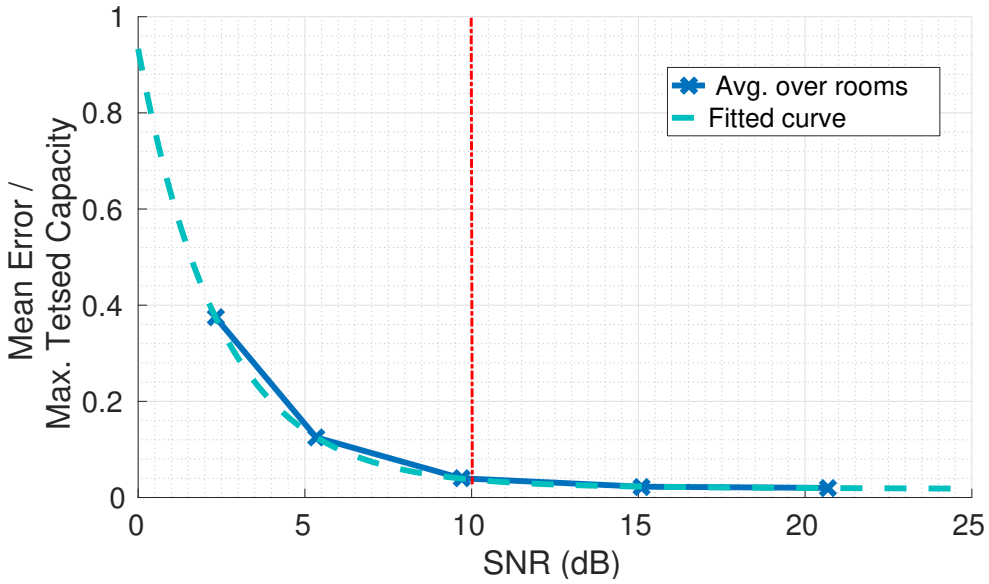


Figure 3.19: System performance with different SNR in small room environments.

factor, since the dissipation rate of the received signal is room geometry dependent. Based on our experiment results, we find that the features from the first two time segments of the received signal (i.e. the first reflection) are generally more significant in the generation of high-rank principal components, therefore we use them to define the SNR of the received signal. Figure 3.18 shows the average received SNR at different output volumes in different sizes of rooms. One could imagine using this property to estimate room size. We see received SNR increases exponentially with higher output volume, and the increasing rate is higher in smaller rooms. Figure 3.19 shows the overall system performance with varying SNR of the received signal in different environments. We see a positive correlation between the SNR and estimation accuracy, and we find that the mean error is greatly reduced once the received SNR pass the $10dB$ threshold. At installation, the system is designed to slowly increase the volume until this $10dB$ SNR threshold is reached.

In Figure 3.20, we show the system performance with varying SNR of the

received signal and weights assigned to the empty room instances. The assigned weights help the system cope with noisy environmental data in the training dataset. With a fixed SNR, we see that assigning overly high or low weights both negatively impact the system's performance. Assigning too much weight causes the WPCA to take into account the variance between different environments, and thus biases the estimator away from counting people. In contrast, an overly low weight would produce dominating principal components, poorly extrapolate the occupancy levels, and the estimator would overfit and often predict the room to be full or empty. This negative impact is more noticeable when the SNR decreases, which is not surprising since with a low SNR the amplitudes alone are not correctly estimating the occupancy level. At this point, increasing weights exacerbates the problem. Based on the experiment results, one should never use an overly low weight to prevent overfitting, and for our evaluation we choose weights equal to 0.5 since it works well in most configurations. During installation, the volume of the transmitter is slowly increased until a particular SNR threshold of the reflected signal is achieved.

3.6.4 Energy Harvesting and Consumption

The hardware platform uses a power management IC to charge three low-self-discharge cells to provide sufficient power for transmitting ultrasound, whether or not solar power is currently available. Figure 3.21 shows the typical power consumption of a transceiver waking up from sleep and activating its audio codec and piezo amplifier (1–2), transmitting a 40ms long ultrasound transmission (30ms chirp with 5ms fade-in and fade-out time to prevent audible artifacts) at maximum volume (86.5$dB(Z)$ at 1m) (2–3), recording for 300ms at a sampling rate

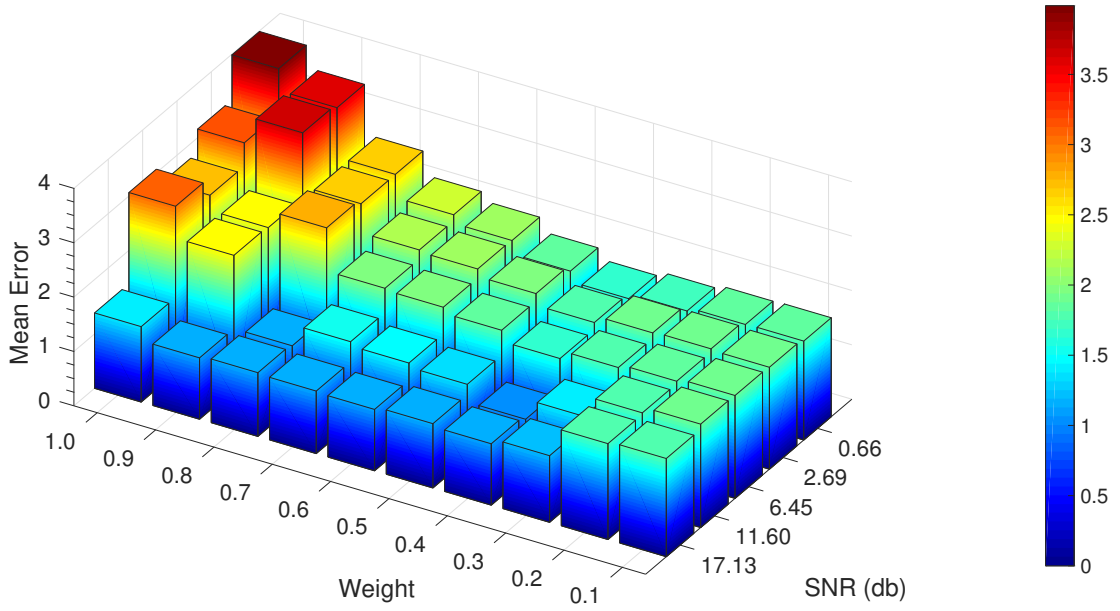


Figure 3.20: Mean estimation error with different received SNR and weights assigned to empty room instances in WPCA.

of 96kHz (3–4), processing the recording and sending the result over the radio (4–5) and then going back to sleep (5). This sequence of operations consumes a total of 18.56mWs. Figure 3.22 shows the power output at the maximum power point of our 7×5.5cm solar cell at various distances from a single 100W equivalent CFL bulb. Based on these numbers and a negligible sleep power consumption on the order of micro-watts, an update rate on the order of seconds is possible, while ambient light energy harvesting allows for an update rate on the order of tens of minutes.

3.6.5 Processing Microbenchmarks

The most CPU-demanding part of our system's operations is performing 10×2048 point FFTs on 10×30ms long chunks of the 300ms recording. Each segment is fetched from an external SRAM and then processed using ARM's CMSIS-DSP library. Benchmarking the time duration of this process using the

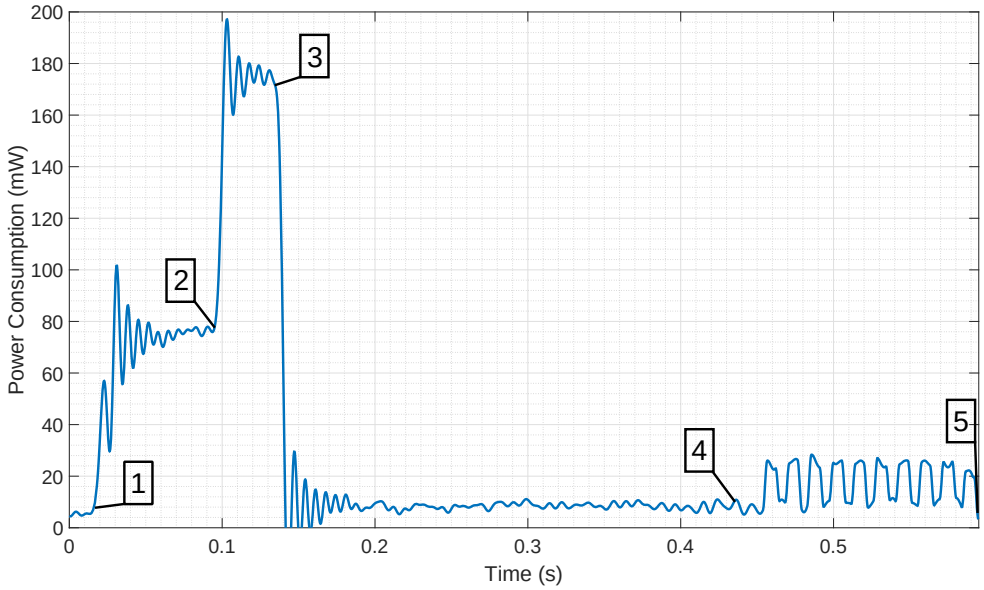


Figure 3.21: The power consumption of AURES at full volume.

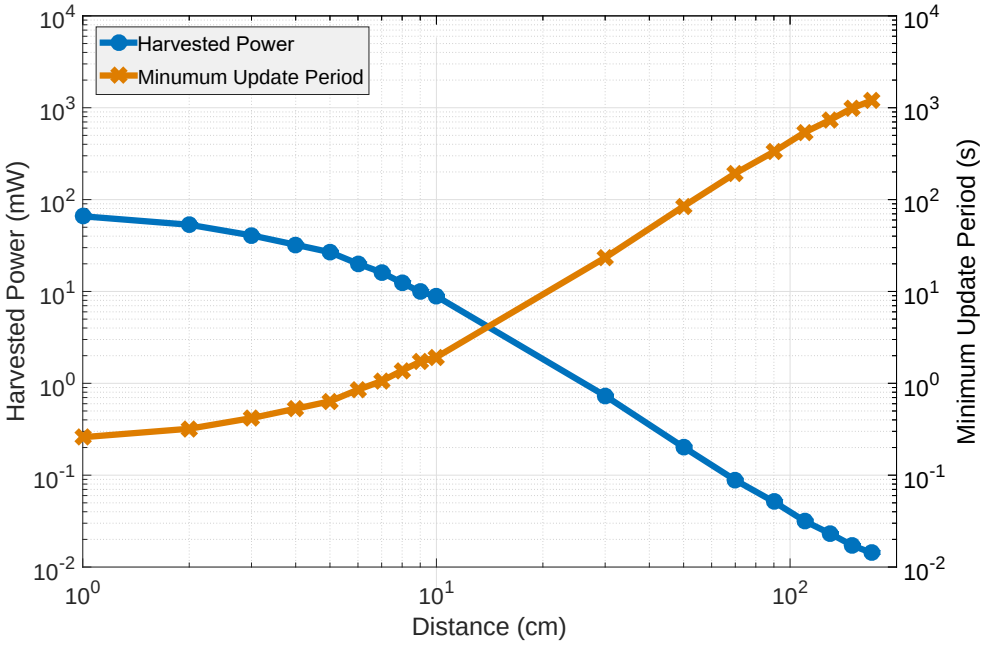


Figure 3.22: Power output from solar cell vs. distance to 100W equivalent CFL bulb vs. minimum update period.

microcontroller's clock, we see that this typically requires 144.45ms, of which 44.88ms are spent fetching the data and 99.56ms are spent calculating the FFTs. From each FFT result, 205 16bit samples from the frequency band of interest are

(a) Medium-size conference room in Collaborative Innovation Center.

(b) Medium-size classroom in Doherty Hall.

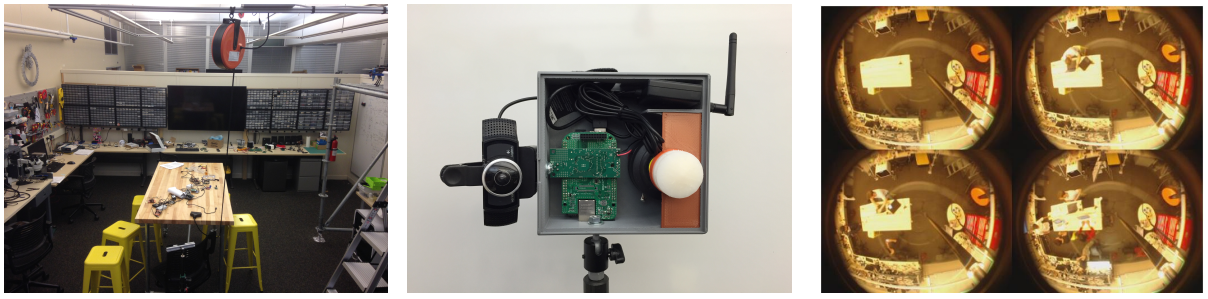
(c) Auditorium in Hamerschlag Hall.

Figure 3.23: Experiment environments for occupancy estimation.

sent back to a base station via RF. It takes approximately 595ms to transmit, record, process, and radio the result of an occupancy sample.

3.7 Real-world Performance

In this section, we discuss experimental results using data captured by our system. In order to collect raw waveform with ground truth, we connected the AURES transceiver to a BeagleBone Black Linux platform with a fish-eye camera. During the sensing period, our system starts the recording of 300ms right after each signal transmission and samples at a rate of 192kHz. The recording length is selected to be significantly longer than the time required for the chirp to dissipate fully in the room [114]. The ideal chirp length should be shorter than the acoustic round-trip time of the room. Assuming the smallest room of operation is $3m^2$, the maximum chirp's length thereby corresponds to 20ms. To prevent audible artifacts in low-cost speakers that could be detected by humans, as discussed in Section 3.2, we added an additional 5ms of fade-in and fade-out time to the chirp and ended up with a chirp length of 30ms. As discussed in Section 3.6.3, the system automatically adjusts its volume at runtime.



(a) (b) (c)

Figure 3.24: Experimental setup. (a) Lab with highly variable furniture and equipment positions. (b) AURES node connected to a BeagleBone Black with a fish-eye camera. (c) Ground truth camera snapshots of the lab at different occupancy levels.

3.7.1 Indoor Environment

We conducted experiments in ten environments of different room sizes over the campus[1]. Figure 3.23 shows example photographs of the three rooms where we ran our experiments. In each room, we mounted the system on the ceiling close to the center of the room to allow better coverage. The location of the transceiver has little impact on the system performance (see Table 3.2). A camera with a fish-eye lens (shown in Figure 3.24b) was installed next to the system and configured to take a low-resolution snapshot (shown in Figure 3.24c) right after each signal transmission to capture the ground truth. The system was configured to collect 5 samples for both the presence detection and occupancy estimation every 10 minutes throughout the day, which corresponded to ∼1300 samples per day. We collected data between 3–14 consecutive days in each room and periodically offloaded the collected data to a remote server. Once the data collection was completed, we trained a model using the data collected from the first day with two occupancy levels manually labeled. To generalize the evaluation

[1]Our IRB declared this data collection to be non-human subject research.

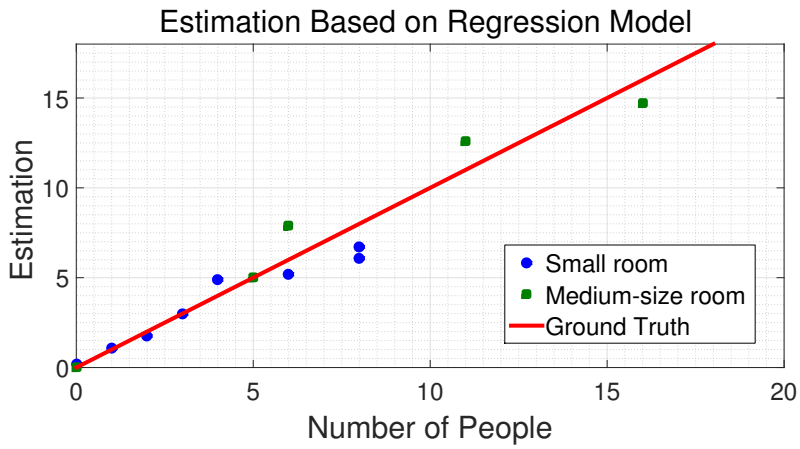


Figure 3.25: Estimation made by our algorithm compared to ground truth in small and medium-size rooms.

results, we classified these rooms into 3 categories based on their sizes. Rooms occupying less than $10m^2$ are classified as small rooms, rooms occupying between 10–$100m^2$ are classified as medium rooms, and rooms occupying more than $100m^2$ are classified as large rooms.

In Figure 3.25 and Figure 3.26, we show several estimation traces of our experiments. Figure 3.25 shows the occupancy estimation made by the regression algorithms respectively in small rooms and medium-size rooms. Each data point represents the estimation for an entire cluster, each of which consists of at least 100 sample points. We find the error slightly increases as the room size gets larger, but we are still able to achieve an error of fewer than 2 people from the average ground truth. In Figure 3.26, we show the traces of an experiment carried out in an auditorium before the start of a class. We periodically sample every 10 seconds while students enter the auditorium. Ground truth was captured with a camera that was hand annotated. As shown in the figure, the estimate tracks the ground truth quite well. Moreover, the system is responsive to rapid dynamics of the environment; the sudden boost in the estimated occupancy level happens right

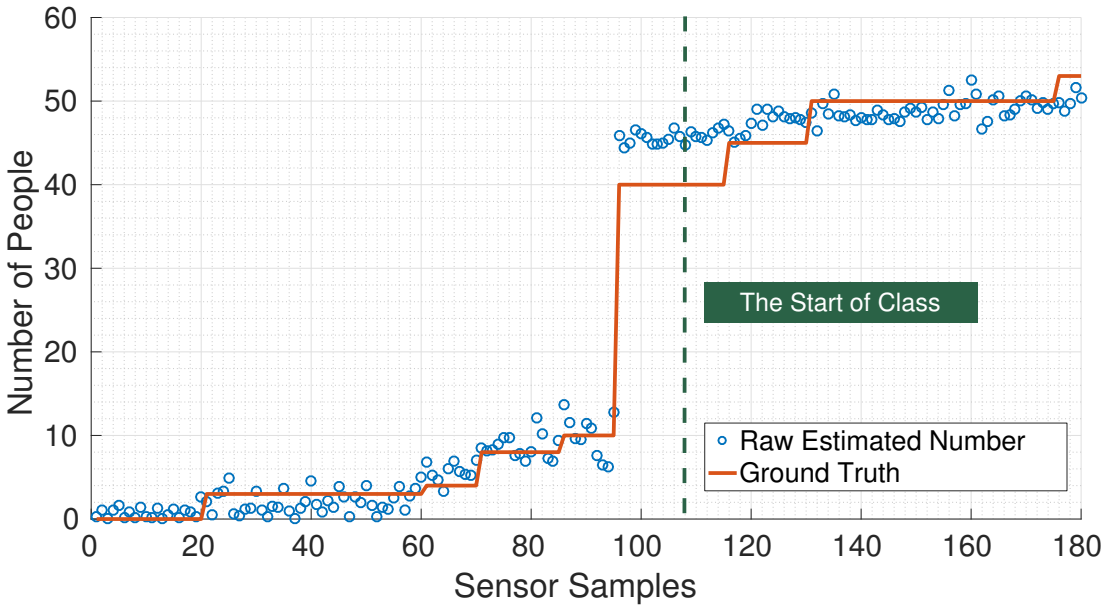


Figure 3.26: Estimation compared with the ground truth as students entered an auditorium.

Interference Type	**Increase in Avg. Error**
Door opened	1.63%
Windows opened	2.38%
Change volume	5.38%
Change position of the device	2.12%

Table 3.2: Impact of different interference sources in small room scenarios.

after a large group of students swarmed into the classroom.

In separate experiments, we evaluated how the system works in the presence of various error sources. This interference test was only evaluated in small room scenarios, since we believed this is where the interference would most significantly impact the result. We performed tests including opening the door to the room, opening windows in the room, changing the volume of the transmitter, and testing in the same room one week later. As shown in Table 3.2, the error was most affected by changes in volume and slightly by opening the windows. Error due to changes in volume is not surprising, since the regression model is built around magnitude changes in different frequencies.

Room Sizes	Avg. Tested / Max Capacity	Avg. Error	Error / Tested Capacity
Small	4/5	0.26	6.5%
Medium	10/20	0.94	9.4%
Large	21/100	2.36	11.2%

Table 3.3: AURES system performance in indoor environments based on room size.

Method	Proposed	[132]	[22]	[76]
Max. Counts	50	12	35	5
Avg. Error	1.6	0.4	1.3	0.7
Environ.	indoor	indoor	outdoor	indoor
Complexity	low	medium	high	medium
Cost	low	high	medium	low

Table 3.4: Comparison of system performance between multiple people counting approaches.

Finally, the overall system performance in different environments is summarized in Table 3.3, and the comparison with related approaches in people counting is shown in Table 3.4. The error is calculated by taking the absolute difference between our estimation and the actual number of people in the room. The overall error slightly increases with the room size since large rooms result in lower received signal strength and higher variance in multipath delay. On average, the absolute error is no more than 3 people across different room sizes, and the error in percentage to the tested number of the participated occupants is around 10%.

3.7.2 Outdoor Environment

Unlike in enclosed rooms, in open-air environments a large portion of the transmitted signal will be scattered away after the first reflection, and only a small amount of the signal can be captured by the receiver. To test the system's performance and sensing range in an open-air environment, we collected a dataset

Sensing Diameter	Error / # of Occupants	Accumulated Error
$<$ 6m	0.08	0.08
6m-10m	0.27	0.21
$>$ 10m	0.48	0.35

Table 3.5: System performance in open-air environment based on sensing range.

of people standing in lines and in clusters at different distances away from the transceiver. Table 3.5 shows that performance is good for occupants standing closer than $6m$ in diameter from the transceiver with an 8% estimation error. However, as occupants move farther away, the estimation error increases to 27% with a large performance drop-off beyond a $10m$ diameter. In our experiments, we also noticed several blind spots at certain transmission angles that have a shorter detection range, which is likely caused by the imperfect beam pattern of our horn speaker design. In comparison to enclosed environments, the system's performance in open-air environments is noticeably worse except at close range. This supports the notion that our training feature is based on the reverberation and the decay of many multipath reflections. This experiment does show that our sensor could be used for estimating occupants in smaller regions, even in open environments, which might be a powerful tool for estimating line length in a food court or detecting people in cubicle areas.

Chapter 4

Room Geometry Sensing and Acoustic Model Reconstruction

In the 1960s, Mark Kac asked the famous question, "Can you hear the shape of a drum?" What he meant was, can one solve the age-old physics problem of inferring a drum's shape based on the sound it makes [63]? More precisely, the central question is whether the shape can be uniquely predicted given a known set of vibrating frequencies. Mathematicians soon discovered that the vibrating frequencies can be reformulated as the eigenvalues of the Laplacian, and the answer turned out to be negative. We can mathematically produce different drum shapes with identical vibrating modes [1] [23, 47].

This problem shares many similarities with room geometry reconstruction, yet they can be quite different from several perspectives. First, sound dissipates considerably faster in a room due to larger dimensions and higher total absorption. Second, typical materials used to construct a room are more rigid compared to the

[1]It was later proved to be positive only if we impose restrictions to certain convex planar regions with analytic boundary (no corners) [137].

membrane of a drum. If we strike a room on its walls, the room will barely vibrate or resonate with the echoes. In other words, the resonate properties of a room reveal little about its shape. However, as the dimensions of space increase, the echoes' arrivals are more discrete in time, allowing one to process them individually. Since each of the echoes carries a piece of information about its reflector, this reveals a way to learn about the room's shape and absorption property, a way that is fundamental to improving sound quality within space.

The field of architectural acoustics is a branch of acoustic engineering that focuses on improving sound quality within buildings. Applications of architectural acoustics include enhancing speech clarity in an auditorium, reducing background noise in a restaurant, or simply improving the quality of music in a concert hall or recording studio. One of the main challenges in this field is to understand room impulse response (RIR) along with the location of various sound reflecting surfaces. This information can be exploited for a variety of applications ranging from audio forensics [84] to creating 3D spatial sound effects [145]. The interaction between sound and the environment can be used by smart speakers [3, 8, 46] to either improve music quality or tune beam-forming algorithms to enhance speech recognition [10, 39, 135]. In contrast to existing room mapping approaches like laser and depth sensors, acoustic sensing identifies the surfaces that have the most significant impact on sound performance in space. For example, glass reflects sound but allows light to pass easily through it, and certain materials like felt absorb sound but would be easily detected by vision or lasers.

Currently, when acoustic engineers optimize the sound properties of a space, they draw from a set of sound modification options like adding sound absorbers, adding structures to block noise, adjusting frequency levels, or leveraging

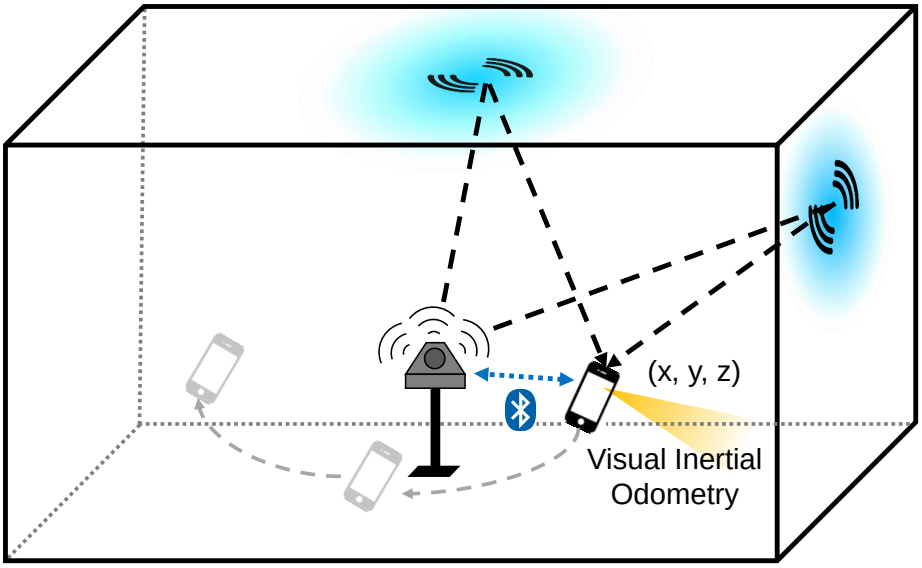


Figure 4.1: Synesthesia system overview.

electronic sound masking systems to combat various acoustic problems. For specialized listening areas like music halls, modeling tools can help optimize construction, but the fine tuning of real-world sound performance typically requires an arduous trial and error process where installers test various configurations of absorbers and reflectors, which they evaluate either with measurement microphones at specific points in space or with a well-trained ear. Smart speakers have the disadvantage of only being able to listen at a single point in space. The geometry of the space and the absorption coefficient of all surfaces play a large role in the space's overall acoustics. This makes it extremely difficult to optimize acoustic properties, especially with a limited number of sampling points.

In this chapter, we introduce Synesthesia[2], a system that takes the first steps towards providing acoustic engineers with the ability to accurately capture and visualize the reflection and absorption of sound within interior spaces through the

[2] Synesthesia is named after the phenomena where one sense in a person triggers a reaction in another sensing system (e.g. seeing sound).

use of a mobile phone as a receiver. With acoustic room geometry information (not just wall locations), smart speakers and high-end audio theater systems can better sense their environment to improve both their sound output quality as well as the ability to understand voice commands while playing music. They are currently limited with a single microphone that can only sense in a single fixed location. By using visual inertial odometry (VIO) on a smartphone (provided by platforms like ARKit [7]/ARCore [44]) we can precisely track a phone's relative location through space while simultaneously capturing a dense set of acoustic samples. Figure 4.1 shows an overview of Synesthesia that consists of a single fixed speaker array (i.e. a smart speaker) that generates a number of acoustic and ultrasonic chirps. The transmissions of chirps are synchronized with the mobile phone as the user walks around the space. Once a user has covered enough ground, our system can learn the RIR at each location to estimate the location of acoustic reflectors (like walls) based on echo arrival time and amplitude. After the acoustic room geometry has been reconstructed, this model can be passed on as information to an audio processing system, like a smart speaker, to improve sound quality. For example, Synesthesia creates a heat map of acoustic absorption at a range of test frequencies projected on each surface. Since the geometry is constructed relative to the VIO starting point of the phone, it is possible to overlay and visualize the final heat map using augmented reality. New versions of ARKit 2 support visual relocalization, so the phone can reload or share this information. This creates a powerful new way for users to explore the space, by seeing actual acoustic absorption mapped as colors in the environment. Though out of the scope of this dissertation, the same acoustic map may eventually be used to optimize smart speakers [109].

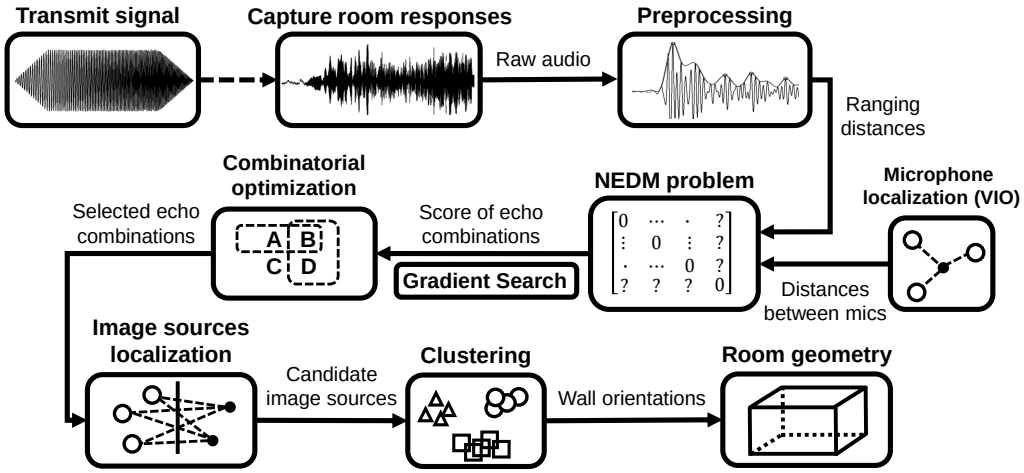


Figure 4.2: Overview of the room geometry reconstruction algorithm.

In the early 1900s, Sabine began to model the impact of people, frequency, and the geometry of spaces on acoustics [114]. Significant follow-on work has explored modeling sound in indoor spaces to the point where it is possible to use the arrival time of echoes reflected off of walls to reconstruct room geometry [6, 27, 35, 60, 88, 110, 112, 127]. These approaches leverage the RIR to find the most likely position of walls in space, using a set of speakers and microphones in a fixed and well-known configuration. For Synesthesia to achieve its goal of seamlessly allowing a phone to scan the space, we must relax a few of the key underlying assumptions from this body of previous work. First, our approach does not assume prior knowledge of the number of reflective sources (typically walls) found in the room. Second, we do not assume that we receive all of the first echoes reflected off of surfaces. Third, we assume that there are errors in the location estimates we received from our microphone placements provided by VIO.

In Figure 4.2, we show a flow diagram of our reconstruction algorithm. The system starts by periodically transmitting acoustic signals into a room with a loudspeaker, while the user captures echoes reflected back from the walls at

multiple locations using a mobile device. By the nature of sound propagation, the relative positions between the speaker, sampling locations, and the surrounding walls are embedded in the arrival time of the echoes. To retrieve this information, we first introduce the Image Source (IS) model (see Section 4.1) to help to formulate and refine the problem. Next, we extract the ranging measurements from the received signal at the sampling locations (see Section 4.2), and integrate them with their corresponding location information provided by VIO (see Section 4.3). Together, these distance measurements form the data input of our reconstruction algorithm (see Section 4.4).

We formulate the geometry reconstruction problem as a multi-layered optimization problem using Euclidean distance matrix (EDM) properties (see Section 4.4.2), and apply techniques including semi-definite programming (SDP) (see Section 4.4.3), mixed integer programming (MIP) (see Section 4.4.4), searching algorithm (see Section 4.4.5), and clustering (see Section 4.4.6) to tackle the problem. Using a dense sampling of chirp recordings with relative positioning, we can create a high-resolution 3D image of reflective and absorbing surfaces in any given space (see Section 4.6).

In our prototype, the audio signals are transmitted from a Bluetooth triggered piezo speaker and recorded by a smartphone (see Section 4.5). The smartphone is initialized from the speaker's position and is used to trigger transmission and recording of test waveforms, while annotating them with the visual odometry coordinates. Room geometry reconstruction and the absorption imaging are computed offline and then transmitted to an augmented reality phone application as a series of colored 3D translucent polygons.

Finally, with a change of perspective, we show how the entire system can be

used reversely to perform microphone localization once the room geometry is acquired (see Section 4.7). One of the most attractive applications of microphone tracking is indoor localization. The awareness of a user's location can enhance a variety of applications, including advertising, AR, and pervasive computing. Current state-of-the-art localization systems often require custom transmitter and receiver hardware, and rely on a large number of beacons to achieve fine-grained localization [54, 65, 86, 105, 108]. In comparison, by exploiting multipath reflections from the surrounding walls, our system requires only one speaker and one off-the-shelf mobile device to achieve accurate localization.

4.1 Image Source Model

To model the echo propagation in a room, we assume the room to be a K-faced convex polyhedron, and we adopt the image source (IS) model [2]. The main principle of the IS model is to replace a reflection path from a real source with a direct path from an image source. Assuming the location of the source is known and the echoes obey the law of reflection, the image sources are obtained by mirroring the real source to the walls. We refer to a received echo with n reflection as n^{th}-order echo and its corresponding image source an n^{th}-order image source. We show an example of a first and second-order image source in Figure 4.3. The IS model directly links the location of the image sources and the room geometry; knowing the location of an image source is equivalent to knowing the location of a wall. Using the IS model, we can convert the room geometry reconstruction problem into an image source localization problem, where typical indoor localization techniques can be applied. The main difference is that instead

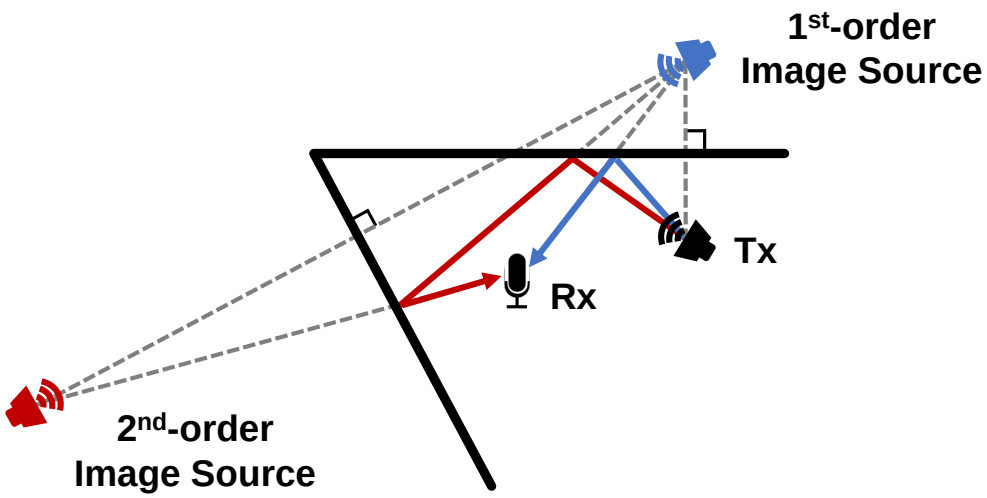


Figure 4.3: Illustration of the first and second-order image sources with their reflection paths.

of localizing the real source inside the room using line-of-sight (LOS) signals, we aim to localize multiple image sources outside the room simultaneously using the multipath reflections. To determine the location of an image source in 2D/3D, we obtain ranging measurements to the image source from at least 3/4 different locations (more locations will improve performance). This is achieved by measuring the RIR from the received signal and converting the arrival time of the echoes into ranging estimates, which we discuss in Section 4.2.

4.2 Acoustic Ranging

In Chapter 3 we showed how chirp pulse compression can be used to efficiently collect responses over a wide range of frequencies. But a more well-known property of pulse compression is its improved ranging resolution, which has been widely used in RADAR systems. There are several variations of chirp signal, which generally fall into two types: linear chirps and non-linear chirps. The main difference between them is that linear chirps tend to have more tolerance to Doppler shift, while non-

linear chirp can provide low time sidelobes. For our application, we use a linear chirp in order to reduce the impact of user movement.

A linear chirp signal $y(t)$ starts at $t = 0$ with a duration T and can be represented as

$$y(t) = \begin{cases} Ae^{j2\pi((f_0 - \frac{\Delta f}{2}) + \frac{\Delta f t}{2T})t} & \text{if } 0 \leq t < T \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

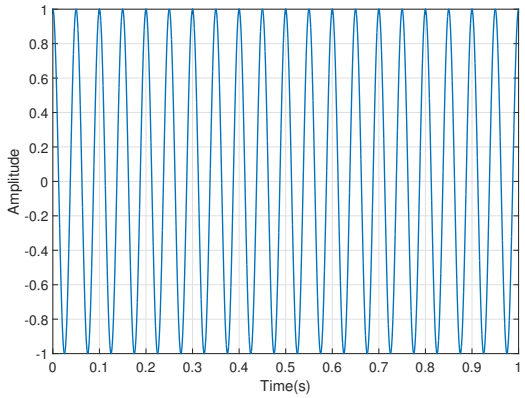
where A is the amplitude, f_0 is the center frequency, and Δf is the sweeping bandwidth. When a chirp cross-correlates with itself, the resulting auto-correlation can be represented as [55]

$$(y \star y)(t) = A^2 T \Lambda(\frac{t}{T}) sinc(\Delta f t \Lambda(\frac{t}{T})) e^{j2\pi f_0 t}$$

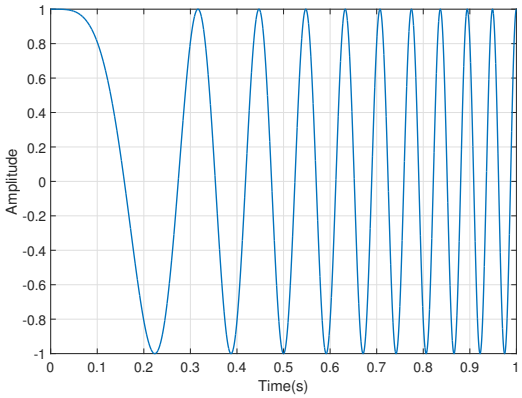
where Λ is the triangle function and $sinc(x) := sin(\pi x)/\pi x$ is the cardinal sine function. This auto-correlation function reaches its maximum at $t = 0$ and behaves as the $sinc$ function. Since the width of the main lobe of a $sinc$ function is much shorter, this results in better ranging resolution. For a linear frequency modulation chirp, its ranging resolution γ is inversely proportional to the sweeping bandwidth given by

$$\gamma = \frac{c}{2\Delta f}$$

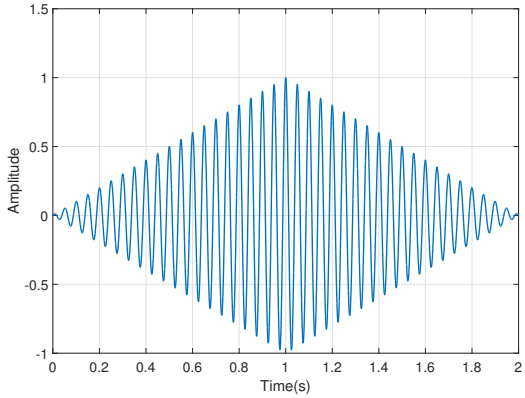
where c is the speed of sound (approximately $343 m/s$ at $20°$C). Moreover, as the energy of the signal does not change during pulse compression, the power concentrated in the main lobe in turn amplifies the received signal. This gain is inversely proportional to the width of the main lobe, and can be approximated as



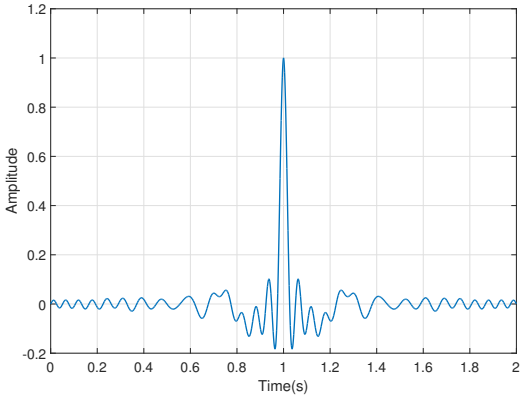
(a) A single tone of $20Hz$.



(b) A chirp sweeps from 0 to $20Hz$.



(c) The tone after matched filtering.



(d) The chirp after matched filtering.

Figure 4.4: An illustration of the chirp pulse compression technique.

$T\Delta f$. We show an example of this pulse compression technique compared to a single tone signal in Figure 4.4. The signals after matched filtering are shown in Figure 4.4c and Figure 4.4d respectively, and we can see how a chirp behaves like a $sinc$ and exhibits much higher peak-to-sidelobe ratio in comparison to a single tone signal.

In principle, to achieve the best signal reception and resolution, one would select a chirp with a long duration and a large sweeping bandwidth. In practice, however, we choose a chirp length of $300ms$ based on the RT_{60} reverberation time of a typical size room [119]. Limiting the chirp length to the reverberation time helps to maximize the SNR of the received signal without spending excessive

energy. Our chirp has a frequency sweeping range of 20–23kHz such that it is inaudible to humans while capturing room geometry. We can then lower the chirp's frequency into the audible range for capturing sound absorption in the audible frequencies that most acoustic engineers are concerned about. Since the sound wave becomes more directional at higher frequencies, we built a quad-sector speaker array to help improve the transmission range in all directions (Figure 4.14). As described in [78], many tweeter speakers exhibit non-ideal impulse responses that can result in audible artifacts similar to clicking sounds. In order to alleviate these artifacts, we add 10ms of fade-in and fade-out time to the chirp's ramp up and ramp out time.

In order to maximize the SNR of the received signal, we assume an additive white Gaussian noise (AWGN) model for the acoustic channel and apply a matched filter on the received signal. One side-effect of matched filtering a chirp signal is that it produces undesirable sidelobes around the main peaks, as previously shown in Figure 4.4d. This makes peak detection difficult when multipath reflections are present. In order to reduce the effect of sidelobes, we apply an additional envelope detector on the matched signal. We then search for the local maxima in the detected envelope and map them back to the nearest peaks in the correlated signal. We show an example of a real-world signal after applying a matched filter and an envelope detector, and selecting peaks in Figure 4.5.

4.3 Visual Inertial Odometry for Localization

One of the critical enablers for being able to perform rich sonic sensing of environments is the ability to collect recordings at known locations rapidly.

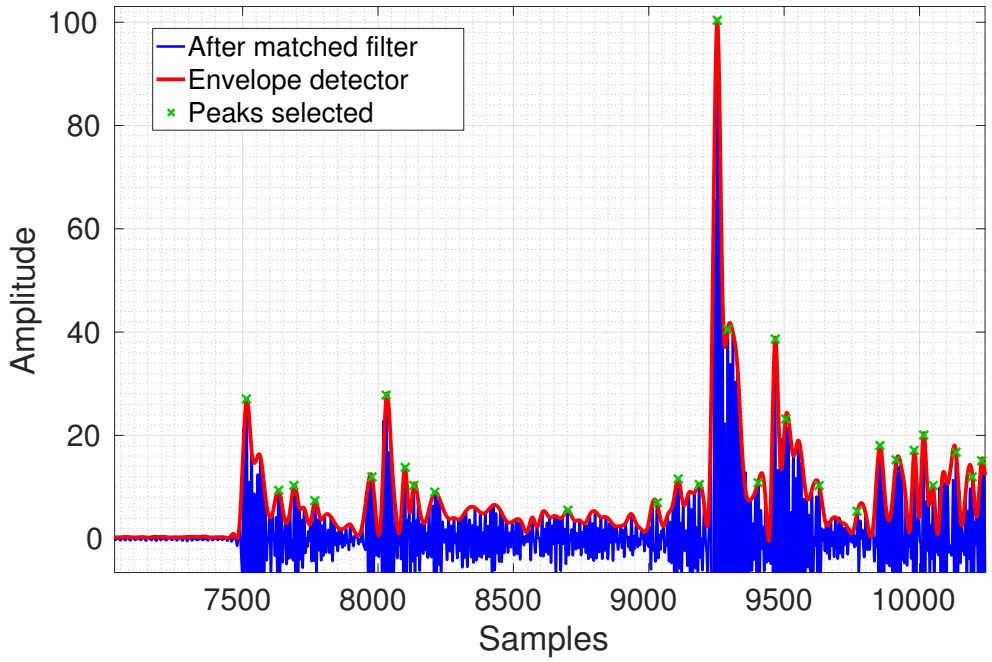


Figure 4.5: Example of the raw signal after matched filtering, the envelope detector, and the selection of peaks.

Recent advances in augmented reality (AR) [45, 87] have led to mobile phones that can precisely track their relative positions over multiple meters using visual odometry (VO) fused with onboard inertial measurement (IMU) data. The so-called VIO systems track the motion of a field of feature points across image frames to accurately estimate the device's motion path. Apple and Google have released ARKit and ARCore, respectively, both of which provide excellent VIO systems for mobile phones.

Due to acoustic reciprocity, it is conceptually possible to swap microphones and speakers at any pairwise recording locations. Using a fixed speaker and any number of microphones that can be localized moving through space, we can approximate arbitrarily dense sensing. In our prototype system, we use a single audio module to both transmit and record data in order to guarantee synchronization. The recording is performed by an external microphone placed near the mobile device, which is

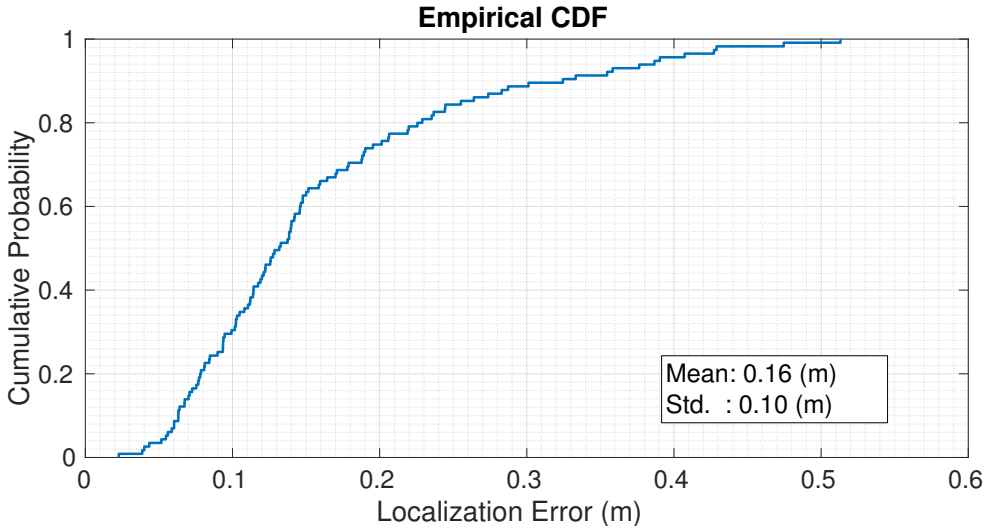


Figure 4.6: The cumulative distribution function of the localization error from ARKit.

constantly streaming its location back to MATLAB, which in turn choreographs the audio playback and recording. In practice, the mobile device would perform the recording locally and trigger the audio transmission either from a wire or over a BLE connection. We include a mount on the top of our speaker where a phone or tablet should be placed to maintain a constant starting origin coordinate. As we describe later in the evaluation section, this is useful for rendering objects like absorption heat-maps in their correct global coordinate frame for viewing with AR.

In our experiment, we used ARKit running on iOS 11 on an iPad Pro in order to evaluate the performance of currently available VIO systems on a mobile tablet. We collected ground-truth data at a set of 120 coordinates across our medium-size room shown in Figure 4.16c, by walking around the room while ARKit was streaming the tablet's coordinates at $30Hz$. Each time we reach a ground-truth marker, we pressed a way-point button on the screen. Figure 4.6 shows a cumulative density function of the localization error after the phone had moved more than 100 meters over a 20-minute period. We see that the average error was $16cm$ with a worst-case error of around $50cm$. In Section 4.6 we evaluate

the impact of this performance on our ability to reconstruct room geometry and perform acoustic sensing. One can imagine that as AR systems evolve with the addition of depth cameras and higher resolution VIO, this performance will only continue to improve.

4.4 Reconstruction Algorithm

Assuming an ideal scenario where ranging measurements are precise and each microphone recording captures echoes from all image sources, a minimum of 3/4 measurements would theoretically be enough to localize all image sources simultaneously in 2D/3D using trilateration. However, this would still require a mapping between the echoes to the image sources that produce them, which is referred to as the echo labeling problem (see Section 4.4.2). The main challenge of echo labeling is that the arrival time of the echoes is location-dependent, and higher-order echoes from a wall can arrive earlier than the first-order echoes from another wall.

Our solution adopts the same EDM formulation used in [35, 60] to select the best combinations of echoes that explain the ranging measurements. However, deriving an effective strategy to find the right combinations is non-trivial when using a single self-tracking microphone. This is because the received signal is often mixed with echoes from clutter in the environment, and the inaccuracies in ranging measurements and localization are large enough such that the EDM formulation is unreliable to determine the correct reconstruction. To solve this problem, we propose using a SDP-based method that is more robust against measurement uncertainty (see Section 4.4.3), and further refine our solution using

combinatorial optimization (see Section 4.4.4).

Another challenge with unconstrained microphone locations is the exponentially increased computational complexity. To reduce the computation time, we propose a searching algorithm that utilizes the convexity of EDM space to more efficiently determine the candidate combinations (see Section 4.4.5). Once we have mapped the echoes to their image sources, we can localize them and reconstruct the room geometry. In reality, using the minimum number of measurements is often insufficient to locate all image sources. One reason is that the possibility of receiving a reflection from a wall depends on both the measurement location and the room geometry. In addition, each propagation path can be individually blocked or badly attenuated by clutter and may not be captured by the microphone. To deal with missing and/or spurious echoes, we divide all microphone locations into subsets and derive sub-optimal solutions accordingly. We later consolidate the sub-optimal solutions using a clustering algorithm with geometric properties to precisely reconstruct the room geometry (see Section 4.4.6).

4.4.1 Preliminaries

We denote $\mathcal{S}^n$ as the space of $n \times n$ symmetric matrices and $\mathcal{E}^n$ as the space of $n \times n$ Euclidean distance matrices. A Euclidean distance matrix $D \in \mathcal{E}^n$ is defined by a set of n points $p_1, ..., p_n \in \mathbb{R}^r$ where

$$D_{ij} = \|p_i - p_j\|_2^2, \ \forall i, j = 1, ..., n \tag{4.2}$$

Let $\mathcal{S}_+^n$ denote the cone of positive semi-definite matrices in $\mathcal{S}^n$. We induce Löewner partial order $A \succeq B$ if $A - B \in \mathcal{S}_+^n$. We further denote the *hollow space* $\mathcal{S}_H^n := \{Y \in \mathcal{S}^n : \text{diag}(Y) = 0\}$ and the *centered space* $\mathcal{S}_C^n := \{Y \in \mathcal{S}^n : Y\boldsymbol{e} = 0\}$, where diag(·) is the operator taking the diagonal elements of a matrix and $\boldsymbol{e} \in \mathbb{R}^n$ is the vector of ones.

4.4.2 Echo Labeling and EDM

Correctly labeling the echoes we received to their corresponding walls is the key to recovering the location of the image sources. In our algorithm, we use the same EDM formulation that was first proposed in [35] as a building block, and we show how it can be used to solve the echo labeling problem. For now, we assume an ideal scenario where measurements are precise, and we show how to correctly label the echoes we received. We later relax this assumption by adding noise. Assuming we have K image sources in total and we collect echoes over N locations with known coordinates, we denote their corresponding TOF distance at each location as $d_n = [d_{n,1}, ..., d_{n,K}]$, $n = 1, ..., N$. Since the phone is tracked using VIO, we can also form the microphone EDM matrix $D_{mic} \in \mathcal{E}^N$ using the distances between the microphone locations. Then, by the definition of EDM, for each image source k there exists exactly one echo combination of squared distances, denoted by $c_k = [c_{k,1}^2, ..., c_{k,N}^2]$ for $c_{k,n} \in d_n$, such that the augmented matrix $\bar{D}_k$ given by

$$\bar{D}_k = \begin{pmatrix} [D_{mic}] & \begin{bmatrix} c_k^T \end{bmatrix} \\ [c_k] & 0 \end{pmatrix}$$

is also an EDM matrix in $\mathcal{E}^{N+1}$. For the rest of this dissertation, we denote an echo combination that corresponds to the same image source as a *good* combination, or otherwise a *bad* combination. With the EDM formulation, a naïve approach to finding all the good combinations is to exhaustively search through all possible combinations and verify whether their corresponding augmented matrices are EDMs. In practice, however, binary verification of EDM is uninformative because it is unlikely that any augmented EDM will be a real EDM due to ranging error and/or numerical inaccuracy. Our goal is then redirected to finding the augmented matrices that are the closest to real EDMs, which is referred to as the Nearest EDM (NEDM) problem discussed in Section 4.4.3.

4.4.3 Nearest EDM Problem

The delta between an augmented matrix and its nearest EDM provides an estimation of the goodness of a combination in a noisy environment, where good combinations are a sufficient condition for small delta values. Nevertheless, to ensure the necessity of the statement, the NEDM approach requires precise microphone locations to robustly recover from noisy distance measurements. In addition, since the total combinations grow exponentially with the number of microphone locations and the number of distance measurements extracted per location, the microphones need to be close enough together (e.g. using a microphone y) to effectively reduce the number of feasible combinations and determine a unique solution [35]. In this dissertation, we relax these constraints and allow users to take measurements at arbitrary locations tracked by VIO. This can potentially improve the accuracy in wall localization, due to geometric dilution of precision (GDOP) [100, 123], but has the trade-off of introducing additional

localization error and exponentially increased state space. Therefore, a robust and efficient approach to solving NEDM problems is vital to the uniqueness and correctness of our solution.

When the measurements are noisy, obtaining a set of points in low-dimensional space that satisfies the desired distances can be extremely difficult. In fact, solving NEDM problems with a low rank (low-dimension) constraint is non-convex and NP-hard [29]; most approaches rely on either heuristics or approximation. One popular approach to solving the NEDM problem is classical multidimensional scaling (cMDS), which is quite efficient in computational complexity. The cMDS starts by performing double centering on the augmented matrix and then projects the data into lower dimensions using the leading principal components. This method, however, inherits a similar drawback to principal component analysis (PCA) in terms of being sensitive to outliers. In addition, cMDS has several features that are undesirable when dealing with noisy data [21]. Instead of directly projecting a target matrix onto $\mathcal{E}^n$ to find its closest approximation, cMDS projects it onto the cone of $\mathcal{S}_+^n$ and maps it back to $\mathcal{E}^n$. This indirect mapping process makes the dissimilarities between the two matrices intractable and causes the result to be less robust. More generalized variations of MDS rely on distance scaling and direct approximation of target distances by minimizing stress based cost function. However, these iterative approaches do not guarantee a global optimum, especially when input distances are noisy.

Instead, we adopt the SDP approach we find to be more robust against noisy measurements in practice. SDP can be seen as a special case of conic optimization (a subfield of convex optimization) and can be solved efficiently using interior point methods [1].

More importantly, EDM-based problems can be mathematically transformed into SDP formulation by leveraging the close relationship between EDMs and semi-definite matrices [61, 71, 77, 106]. Next, we show how this is done for the NEDM problem. Given an EDM $D \in \mathcal{E}^n$, we can rewrite Equation 4.2 as

$$\begin{aligned} D_{ij} &= p_i^T p_i + p_j^T p_j - 2p_i^T p_j \\ &= Y_{ii} + Y_{jj} - 2Y_{ij} \end{aligned}$$

where $Y = p^T p$ is the Gram matrix of the point set that realizes the EDM. Note that each entry in a EDM^n is defined as the squared distance between points, since $\sqrt{\text{EDM}^n}$ is non-convex when $n > 3$ [29]. Since the Gram matrix is positive semi-definite, we observe a linear transformation $\mathcal{K}$ that maps $\mathcal{S}_+^n$ onto $\mathcal{E}^n$ ($\mathcal{K}(\mathcal{S}_+^n) = \mathcal{E}^n$) given by

$$\mathcal{K}(Y) := \text{diag}(Y)\boldsymbol{e}^T + \boldsymbol{e}\,\text{diag}(Y)^T - 2Y \tag{4.3}$$

And reversely, we can derive the Moore-Penrose generalized inverse $\mathcal{K}^\dagger$ of $\mathcal{K}$ ($\mathcal{K}\mathcal{K}^\dagger\mathcal{K} = \mathcal{K}$) given by

$$\mathcal{K}^\dagger(D) = -\frac{1}{2}V\text{offDiag}(D)V \tag{4.4}$$

where $V := I - \boldsymbol{e}\boldsymbol{e}^T/n$ is the geometric centering matrix and offDiag(D) $:= D -$ Diag(diag(D)) denotes the orthogonal projection onto the hollow matrices. This leads to a well-known result for the sufficiency of an EDM matrix originally presented by Schoenberg [117] and later independently found in [136]:

$$D \in \mathcal{E}^n \iff \begin{cases} -VDV \in \mathcal{S}_+^n \\ \\ D \in \mathcal{S}_H^n \end{cases} \tag{4.5}$$

From Equation 4.3, we can see an important property of the linear transformation $\mathcal{K}$: it is *translational invariant*, which means that the EDMs realized by a point set P and its infinite translational symmetries P' will be equivalent. To force the mapping $\mathcal{K}$ and $\mathcal{K}^\dagger$ to be bijective and prevent ambiguous solutions, we can restrict the transformation to subspace $\mathcal{S}_C^n$ and $\mathcal{S}_H^n$ respectively, and we have the mapping $\mathcal{K} : \mathcal{S}_C^n \to \mathcal{S}_H^n$ a bijection and $\mathcal{K}^\dagger : \mathcal{S}_H^n \to \mathcal{S}_C^n$ is its inverse. Furthermore, if we restrict $\mathcal{K}$ and $\mathcal{K}^\dagger$ to the convex cone $\mathcal{S}_C^n \bigcap \mathcal{S}_+^n$ and $\mathcal{E}^n$, we then have $\mathcal{K} : \mathcal{S}_C^n \bigcap \mathcal{S}_+^n \to \mathcal{E}^n$, a bijection, and $\mathcal{K}^\dagger : \mathcal{E}^n \to \mathcal{S}_C^n \bigcap \mathcal{S}_+^n$, its inverse. This result provides a key insight explaining the mapping between the convex cone of $\mathcal{E}^n$ and $\mathcal{S}_+^n$.

However, even though these two convex cones can be related, a direct mapping between the two sets does not exist under the same dimensionality [3]. In order to prevent unbounded optimal solutions [29], we can define a transformation $\mathcal{K}_V : \mathcal{S}^{n-1} \to \mathcal{S}^n$ given by

$$\mathcal{K}_V(X) := \mathcal{K}(V_n X V_n^T) \tag{4.6}$$

where $V_n \in \mathbb{R}^{n \times n-1}$ is the full rank skinny matrix such that $V_n^T \boldsymbol{e} = 0$. We then have $V_n X V_n^T$, the Gram matrix of the point set, and $\mathcal{K}_V(\mathcal{S}_+^{n-1}) = \mathcal{E}^n$. Based on this observation, we can derive a reformulation for the sufficiency of EDM matrices similar to Equation 4.5:

$$D \in \mathcal{E}^n \iff \begin{cases} -V_n^T D V_n \in \mathcal{S}_+^{n-1} \\ \\ \quad D \in \mathcal{S}_H^n \end{cases} \tag{4.7}$$

[3]This can be observed from $\mathcal{S}_C^n \bigcap \mathcal{S}_+^n = \emptyset$ or equivalently $\mathcal{E}^n \bigcap \mathcal{S}_+^n = \mathbf{0}$ (the origin).

With Equation 4.7, we can transform the NEDM problem into a SDP based norm minimization problem that can be generally modelled as

$$\begin{aligned} \underset{X}{\text{argmin}} \quad & \left\| W \circ (\mathcal{K}_V(X) - \bar{D}) \right\|_F^2 \\ & (V_n X V_n^T)\boldsymbol{e} = 0, \\ & X \succeq 0 \end{aligned} \tag{4.8}$$

where W is the weighted matrix that reflects the accuracy of the data, $\circ$ is the Hadamard product, and $\bar{D}$ is the target matrix we want to approximate. The first constraint ensures that the recovered matrix belongs to S_H^n, and the second constraint ensures it belongs to S_+^{n-1}. The hard rank constraint is dropped as a relaxation in order to prevent non-convexity. Note that X is solved in a lower dimension ($n-1$), and we can recover the optimal distance matrix by computing $\mathcal{K}_V(X)$. In rare cases where the rank constraint is not satisfied, the solution is rounded into lower dimensions based on eigenvalue decomposition. We choose the Frobenius norm for our objective function, since it naturally connects with the Euclidean distance space and it is strictly convex. We select the optimizer MOSEK [4] for solving the NEDM problem with SDP and the combinatorial optimization problem, which will be discussed in Section 4.4.4, with mixed integer programming (MIP).

A mini-benchmark on the NEDM performance in $\mathcal{E}^5$ is shown in Figure 4.7. We simulated the EDMs based on random microphone locations in rooms of various geometries and artificially added additional ranging error. We find SDP achieves lower NEDM error on average compared to classical MDS and s-stress MDS approaches, and the improvement grows more noticeable as ranging error

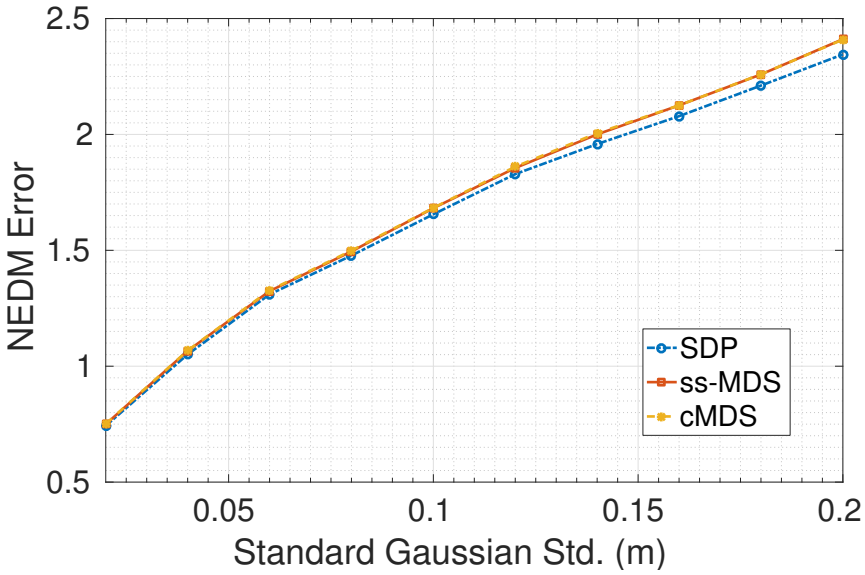


Figure 4.7: SDP achieves the same or lower NEDM error ($\mathcal{E}^5$) in various simulated room geometries, compared to classical MDS (cMDS) and s-stress MDS (ss-MDS).

increases. Although the improvement appears marginal, a small increment in the NEDM error will result in hundreds more ambiguous solutions. We will discuss this negative effect in detail in Section 4.4.4 and demonstrate its direct impact on reconstruction accuracy later in Section 4.6.

4.4.4 Combinatorial Optimization

By solving the NEDM problem, we are able to score all echo combinations based on their augmented matrices' proximity to the closest EDM. We determine their scores to be inversely proportional to their NEDM error reported in equation Equation 4.8. In an ideal scenario, we can find all the good combinations by selecting the ones with the highest scores/lowest errors. In reality, however, random bad combinations could potentially produce a lower NEDM error due to noisy measurement and inaccuracy in NEDM approximation. Moreover, this problem gets worse when microphone locations are unconstrained, since we are

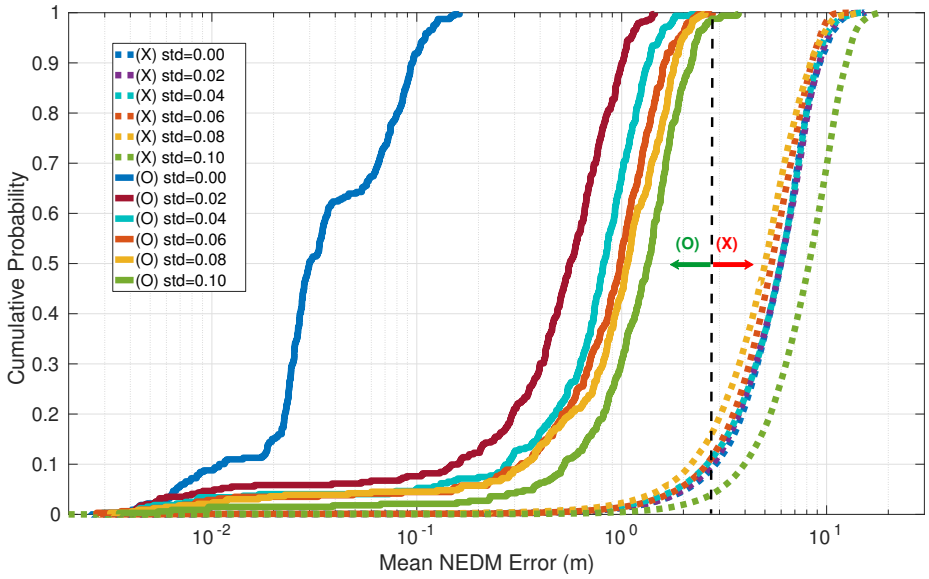


Figure 4.8: Mean NEDM error ($\mathcal{E}^5$) compared between the good combinations (O) and the bad combinations (X) with ranging error drawn from a standard Gaussian distribution with different standard deviations.

unable to use the distances between microphones to trim down the candidate combinations. To quantify the reliability of our NEDM approach in random room geometries with noisy measurements, we run simulations to compare the distribution of NEDM error between the good and bad combinations. The results shown in Figure 4.8 indicate that good combinations typically have an error below a meter, while bad combinations yield much higher error over a wider distribution. Despite their distinct error distribution, the number of bad combinations is orders of magnitude more than the number of good combinations that falls into the same error percentile. As an example shown in Figure 4.9, we find a hundred times more bad combinations that could have lower NEDM error than the good ones when the error standard deviation is $4cm$. Thus, selecting combinations with low NEDM error is insufficient to determine the correct solution.

In order to refine our solution, we expand our objective function to minimize the total NEDM error among multiple combinations while limiting the occurrence

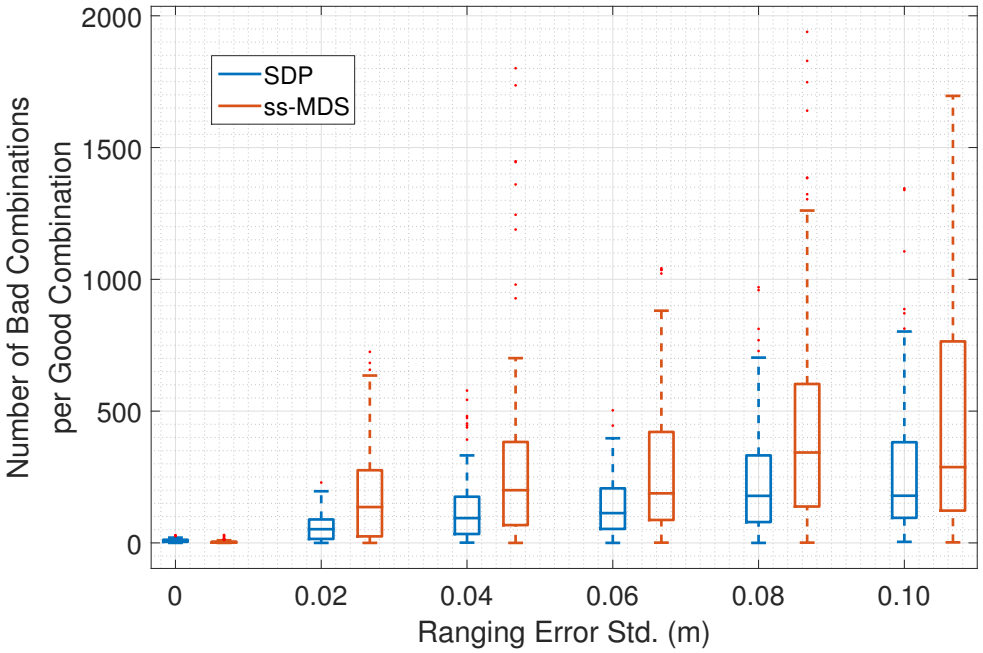


Figure 4.9: The average number of bad combinations with lower NEDM error per good combination increases with ranging error, and SDP achieves more robust result than s-stress MDS.

of each distance measurement across combinations. This in turn enforces constraints on our distance selection between combinations and greatly improves the chances of finding good combinations. The optimal selection is to find the set of combinations such that their combined score is the highest while satisfying all the constraints, which can be formulated as a combinatorial optimization problem. Suppose we compute the score s_i for each echo combination c_i by solving $s_i = \text{NEDM}(D_{mic}, c_i)$. Then, we can solve the following combinatorial optimization problem using mixed integer programming in the form of

$$\begin{aligned} \max \quad & s^T x \\ \text{subject to} \quad & A^T x \leq b \\ & x \in \{0,1\}^n \end{aligned}$$

where s is the vector of scores derived from solving NEDM problems, x is the binary vector indicating whether a combination is selected, b is the vector for constraining the occurrence, and A is the constraint matrix that limits the selection between combinations given by

$$A_{ij} = \begin{cases} 1, & \text{if } d_j \in c_i \\ 0, & \text{otherwise} \end{cases} \qquad \forall i = 1, ..., |c|,\ j = 1, ..., N$$

where $|c|$ is the total number of echo combinations. In our experiment, b is set as a vector of ones to achieve the best performance, but ideally the constraint can be more relaxed when ranging resolution is low and peaks are inseparable due to close arrival times. Since the total number of surfaces is unknown, we encourage the algorithm to find as many surfaces as possible by pruning the combinations using an error threshold, and to solve the combinatorial optimization problem with an objective function that maximizes the total score. This error threshold can be determined by simple heuristics or based on a prior estimation of the ranging error. While this greedy approach may allow some bad combinations to sneak through, it greatly improves the discovery of good combinations and benefits the following clustering algorithm (Section 4.4.6) and overall performance. The objective function is biased toward combinations with low error due to the non-linearity of the inverse proportion operation when calculating the scores. The intuition is to increase the likelihood of selecting good combinations since the ratio of bad combinations over good combinations decreases with NEDM error, as shown in Figure 4.8.

4.4.5 Gradient Search

Solving NEDMs for each combination inevitably becomes a computational bottleneck due to large combination space. In situations where computation power is limited and/or application is time sensitive, this indirectly impacts the reconstruction performance, since we have to be more conservative about peak selections.

In order to reduce the computation time, we implement an iterative gradient-based heuristic to directly search for combinations below a certain NEDM error threshold. Since the problem is essentially a combinatorial search, our goal is to find the majority of the target combinations in order to increase computational efficiency. Similar to iterative local search (ILS), our algorithm dynamically refines its search direction by exploring neighborhood candidates of the current solution. We exploit the convexity of EDM space and designed our neighborhood function based on gradient descent.

The search starts by randomly selecting a combination and solving the NEDM problem as described in equation Equation 4.8. At iteration t, we denote the selected combination as c_t and the resulting true EDM as $\mathcal{K}_V(X_t)$. To select the next combination c_{t+1}, we exploit $\mathcal{K}_V(X_t)$ and find the combination such that the gradient of the objective function with respect to the new augmented matrix $\bar{D}_{t+1}$ is closest to zero. Although the gradient does not guarantee an optimal searching direction, it can be computed efficiently and provide a good approximation by looking one step ahead. Given our objective function f, the gradient is given as

$$\bigtriangledown_{\bar{D}} f = -2(W \circ W \circ (\mathcal{K}_V(X) - \bar{D}))$$

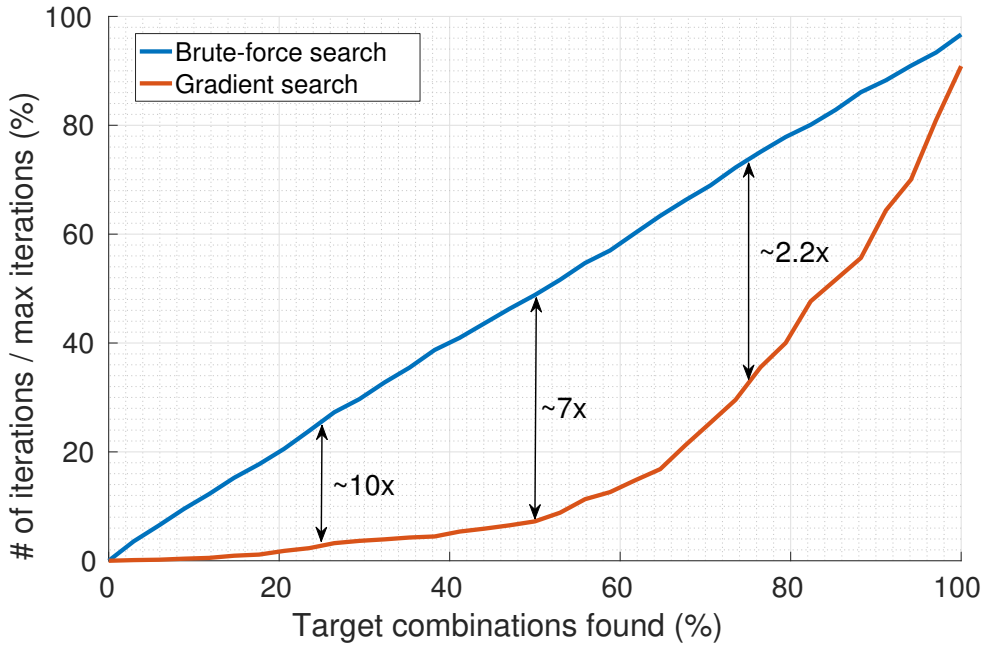


Figure 4.10: Improved performance with gradient-based local search compared to brute-force search.

In the process of iterative searching, we keep a history of the visited combinations and restart randomly when the gradient leads to a previously visited combination, in order to improve the exploration of our search. In addition, whenever the gradient reaches a local minimum and the NEDM error is below the given error threshold, we solve the combinatorial optimization previously mentioned in Section 4.4.4 to dynamically trim the search space to increase diversity. These constraints are removed when there are no feasible combinations left, and the algorithm restarts with the history of the visited combinations. The algorithm ends when a certain percentage of total iterations is reached or a certain number of target combinations is found.

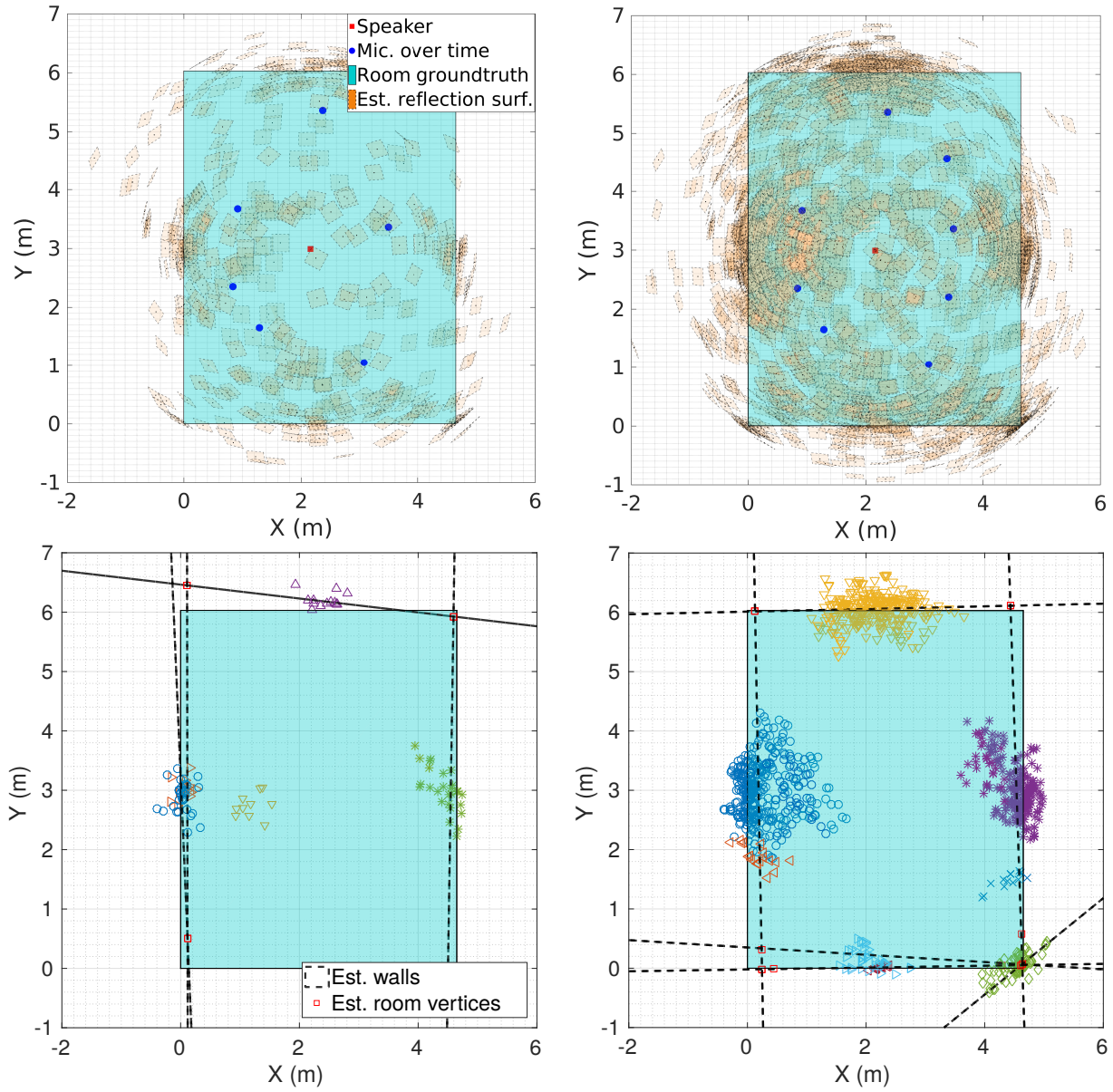
Since it is difficult to theoretically analyze the computational complexity for combinatorial search problems, especially when the performance depends heavily on room geometry/receiver locations, we use simulations to evaluate the

performance of our heuristic. In Figure 4.10, we show a mini-benchmark of our heuristic compared to brute force search in terms of the number of iterations spent on finding target combinations. The proposed heuristic is around 10 times faster in finding 25% of the target combinations and 7 times faster at the 50% mark. In general, the percentage of target combinations required for reconstruction could vary based on the ranging error and the number of microphone locations. Experimentally, we find that 50% of the combinations are good enough to properly reconstruct the room geometry.

4.4.6 Wall Estimation

In order to localize an unknown number of reflective surfaces within a reasonable amount of computation time, we accept sub-optimal solutions of inaccurate image sources and/or bad image sources. The final step of the algorithm is to eliminate the outliers and determine the true location of the good image sources. This is possible because bad image sources would be scattered due to the randomness of the combinations, while the good image sources would converge into clusters at their true locations. As shown in Figure 4.11, the corresponding surfaces, which are the bisectors of image sources and the speaker location based on the image source model, would also follow the same pattern. The echo combinations that grow exponentially with the number of measurements now improve accuracy; we can determine the true locations of the walls by clustering with fewer microphone locations.

In order to minimize the impact of the reflections from clutter in the environment, we recover the locations by selecting 4 random microphone locations at a time (minimum for 3D reconstruction) and iterate through different



(a) With 6 microphone locations.

(b) With 8 microphone locations.

Figure 4.11: Top-down view of the clustering process in 3D. The detected surfaces increase exponentially with measurements, improving the clustering accuracy and overall reconstruction accuracy.

combinations of microphone locations. During this process, we apply clustering on the combined results until the desired number of clusters are found. The algorithm therefore discovers larger reflectors (walls) first since they provide more consistent reflections and form clusters faster. To discover smaller reflectors, we

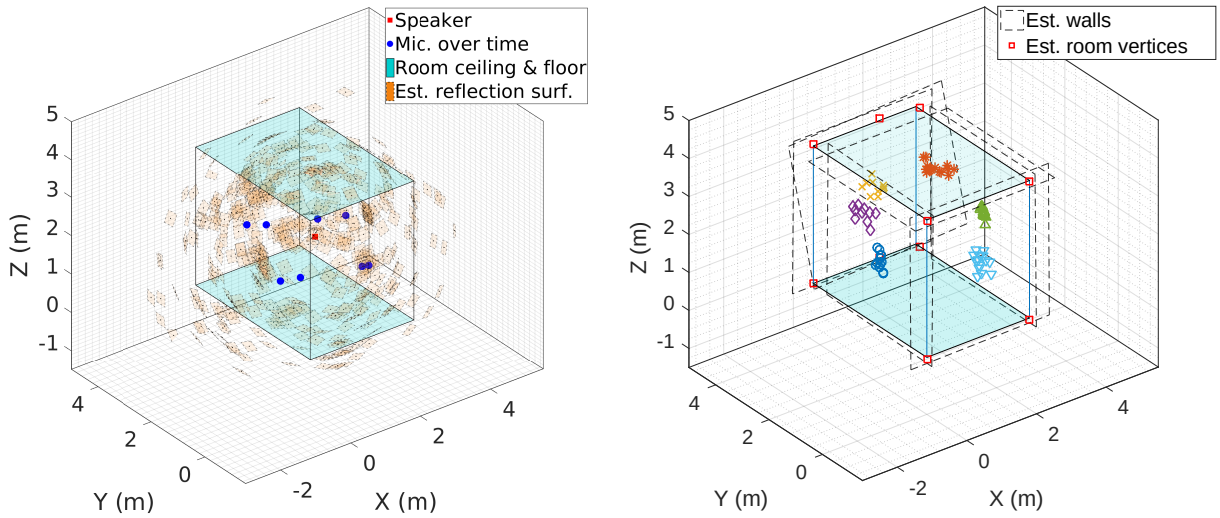


Figure 4.12: 3D view of the wall reconstruction rendered with the clustering result.

can iteratively remove the processed clusters and continue the clustering process with wider radius. In general, the resolution of features the system can detect as surfaces is a function of the number of measurements and compute time. Capturing the first dozen major features is quite feasible, but the complexity increases quickly for higher resolution maps. In this paper, we chose a density-based clustering algorithm DBSCAN [83], due to its robustness to outliers and zero prior knowledge of the number of clusters or density distribution. One drawback of the DBSCAN algorithm is that the clustering results are sensitive to the minimum neighborhood points and neighbor distance. Through experiments, we find the results to be the most stable when the neighborhood point is set to three times the reconstruction dimensionality, and the neighbor distance is determined based on our estimation of the ranging error. Once the clusters are found, each corresponding surface is determined as the plane passes through the geometrical center of the cluster with a normal vector pointing toward the speaker.

While clustering copes with the problems of missing echoes and having an

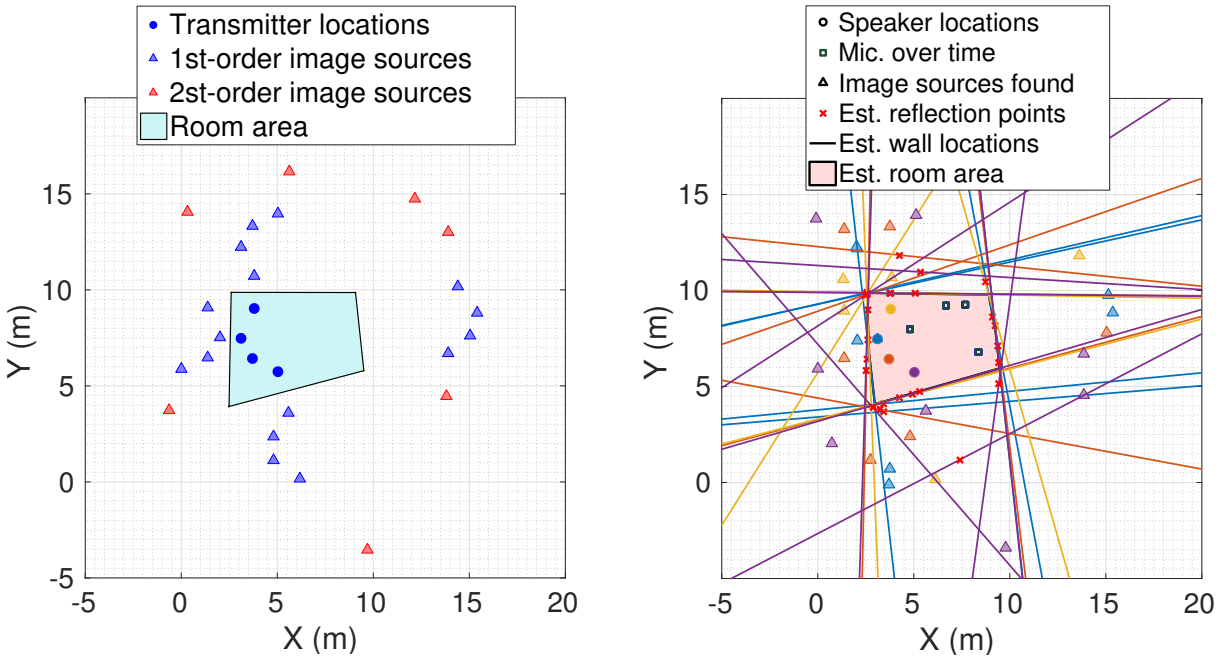


Figure 4.13: (Left) A simulated room geometry plotted with all possible first-order and second-order image sources. (Right) Overlays of the reconstruction results where each is computed using data from a single speaker. The ranging error is drawn from a standard Gaussian distribution with std=0.03m, and the overall reconstruction similarity is 93%.

unknown number of surfaces, the algorithm also captures the clusters from higher-order reflections. In order to bypass the process of identifying and eliminating higher-order image sources, we observe in Figure 4.13 that the virtual surfaces generated by second-order echoes from two adjacent surfaces will always cross the intersection of the surfaces. Similarly, if the reflected surfaces are not adjacent to each other, then the virtual surface crosses the intersection found by extending the surfaces. In fact, this result can be mathematically proven using geometry and holds for both 2D and 3D scenarios. Our algorithm leverages this geometry property to determine the room geometry as the smallest convex polyhedron within the virtual surfaces that bound all of the microphone locations. This heuristic ensures that the actual room geometry affected by second-order

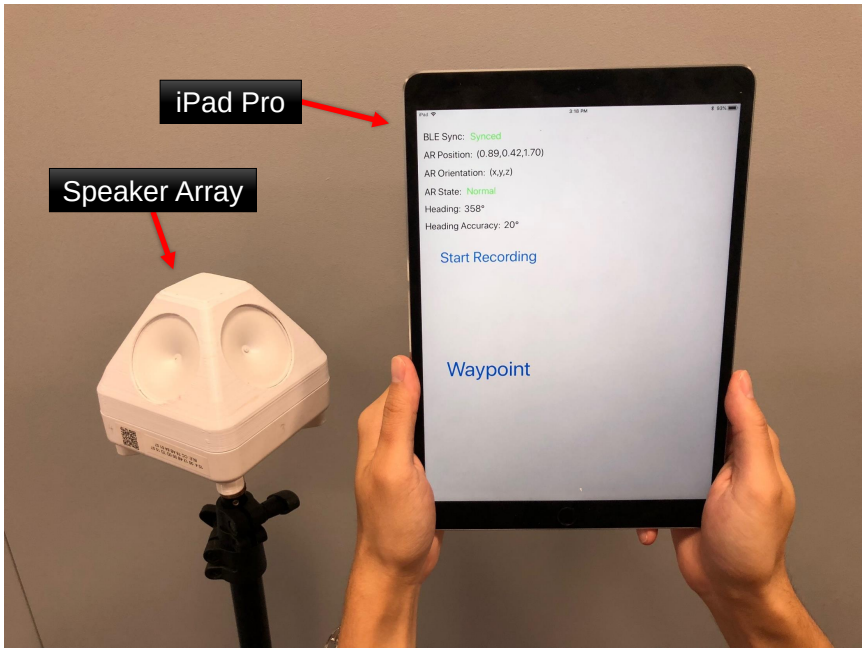


Figure 4.14: Synesthesia experimental setup.

reflections and missing first-order reflections is minimized in the presence of noise. In addition, this geometry property reveals an alternative way to reconstruct the room geometry with the help of higher order reflections when multiple transmitter locations are available.

4.5 Platform Implementation

An overview of our experimental setup is shown in Figure 4.14. Our prototype consists of an omnidirectional tweeter speaker with custom hardware and a mobile device. Our transmission signal is a linear frequency sweeping chirp from 20–23kHz with a sampling rate of 48kHz. Each of the four horns transmits the signal to distribute it uniformly through the space. The transmitter synchronizes using BLE with the mobile device while it records the room response. The synchronization error is less than 1ms where 95% is within $\pm$200μs, which results

in a ranging error of $\pm 6.8cm$. We based our transmitter design and time synchronization on the platform the authors described in [109]. In our version of the design, the speakers all transmit simultaneously (instead of cycling). We discuss the impact of ranging accuracy on system performance in Section 4.6.

4.6 Reconstruction Performance

In this section, we experimentally validate the performance in simulations and in a variety of rooms with real recordings. To quantitatively measure our performance, we defined an evaluation metric that captures the similarity between two arbitrary polyhedra (e.g. ground truth and our reconstruction) in 3D space (see Section 4.6.1). We conducted numerous simulations to evaluate the impact of room geometry, the number of transmitters/receivers, and ranging error on the reconstruction accuracy (see Section 4.6.2). We then validated the results in real-world environments, and compared our performance with state-of-the-art (see Section 4.6.3). Finally, we demonstrate its utility in estimating absorption coefficient of reflectors in an AR virtualization (see Section 4.6.4).

4.6.1 Evaluation Metric

To measure the similarity between the ground truth and our estimation of the room geometry, namely polyhedra A and B, we use the following criteria based on their overlapping volume and union volume, given by

$$\text{Similarity} = \frac{A \cap B}{A \cup B} \tag{4.9}$$

The similarity metric is strict since it reflects not only the ranging error, but also captures the translation and orientation for each wall. When computing the similarity in cases where some walls are missing and the estimated polyhedron is not bounded, we artificially added the ground-truth wall so the similarity can be determined, but also penalized by the percentage of the number of added walls to the total number of walls. If the estimated polyhedron is bounded despite missing walls, the same rule is applied when it results in better similarity to ensure a fair comparison. For example, in Figure 4.13 we showed a simulated reconstruction of a room from multiple speaker locations in 2D with small ranging error, in which we achieved a similarity of 93%. To the best of our knowledge, this is the first work that reports performance in 3D real-world environments with a normalized evaluation metric that captures the rotation, translation, and scaling of the room geometry at the same time.

4.6.2 Simulation

We randomly generated a set of room geometries with wall lengths from 5–10m with a minimum angle of 30 degrees between walls. Speaker and measurement locations are randomly selected in the room, each at least $50cm$ away from the walls, and the sound pressure level (SPL) is set to 65dB at 1 meter consistent with readily available commercial hardware. The wall absorption coefficient is set to 0.5 to simulate the absorption of common materials in the chirp's sweeping frequency [34]. When constructing the received signals, we add additional ranging bias for each impulse response. For each parameter configuration, we run at least 20 simulations. We use ray tracing to validate first and second-order image sources and to simulate path loss. Higher order reflections are dropped since they

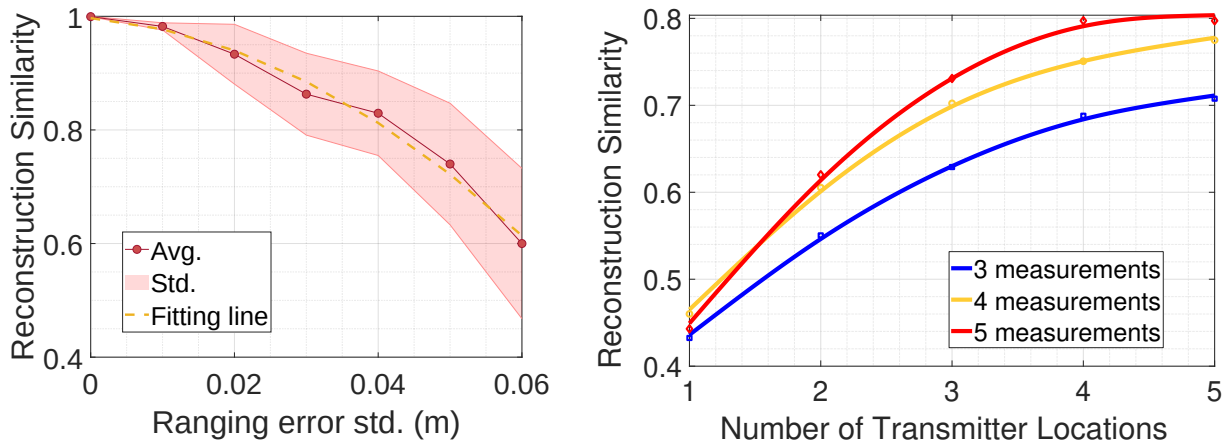


Figure 4.15: Simulated reconstruction similarity (2D) with (a) varying ranging errors and (b) a varying number of transmitter locations and measurements.

are rarely detected in reality due to attenuation. Since the estimated room geometry is both translational and rotational invariant, we use the Kabsch algorithm [62] to find the optimal rotation matrix that minimizes the Root Mean Squared (RMS) error on measurement locations to align the result with the global coordinates system for better visualization.

In Figure 4.15a, we simulated the overall reconstruction similarity with ranging errors drawn from a standard Gaussian distribution with varying standard deviations (i.e. a standard deviation $\sigma = 5cm$ implies a $\pm 10cm$ ranging error in around 95% of the time). Since each ranging measurement is sampled only once in simulation, the absolute ranging error can be seen as a folded standard Gaussian distribution where its mean is given by $\mu = \sigma\sqrt{2/\pi}$. The reconstruction similarity is found to be sensitive to the ranging accuracy and drops quickly as the ranging error increases. Still, we are able to achieve 75% reconstruction accuracy on average with a standard deviation of $5cm$. The relative positioning between the room, speaker, and microphone also have an impact on the reconstruction accuracy, especially when ranging error is high. Most of the reconstruction error

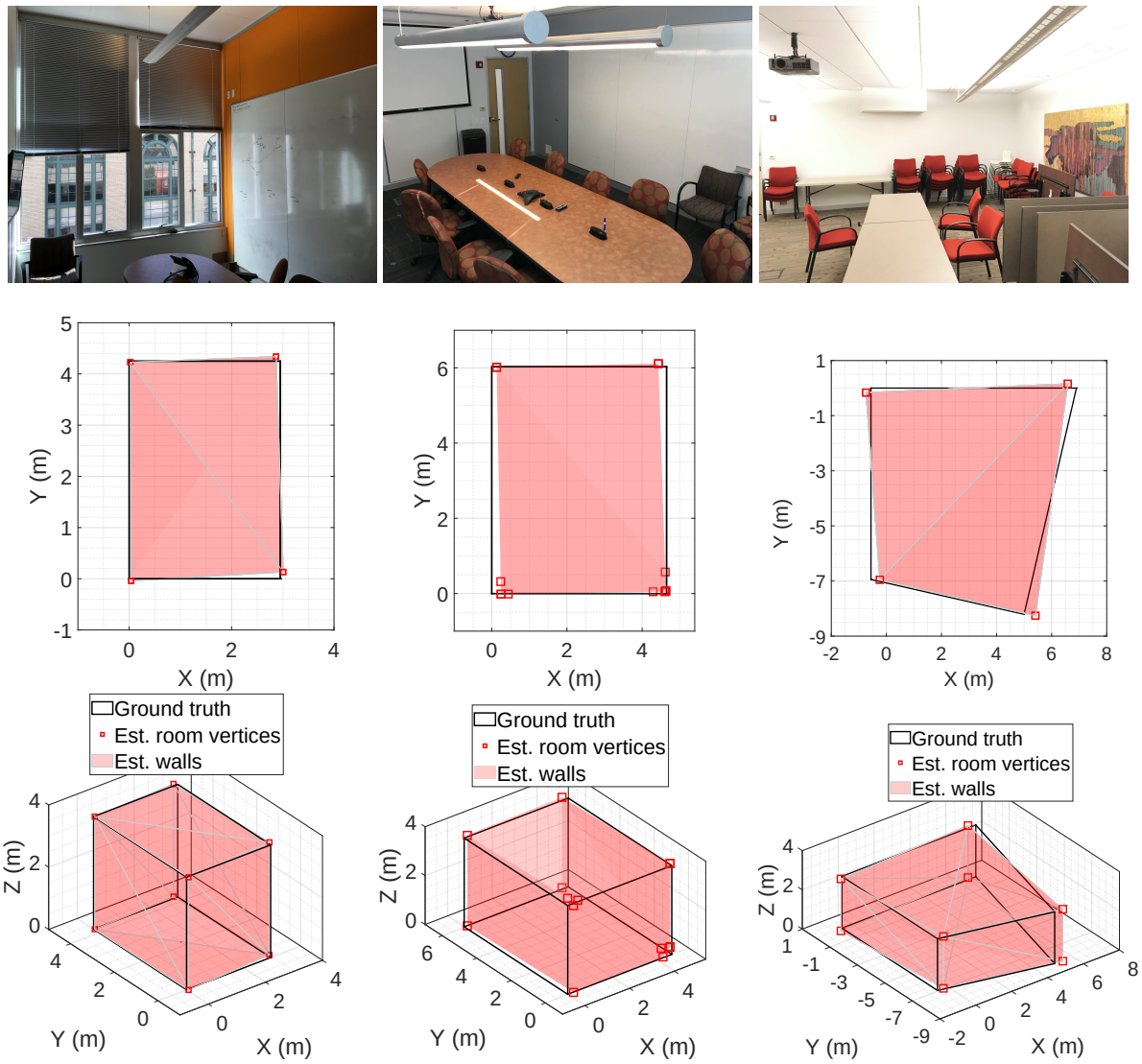
comes from peaks that arrive close in time, where the algorithm often fails to isolate the peaks or selects false peaks that increase the number of bad combinations.

In Figure 4.15b, we simulated how increasing the number of transmitter locations and microphone measurements can contribute to reconstruction similarity. Note that each reconstruction is independently computed using data from a single speaker. With more transmitter locations and/or measurements, we can effectively avoid scenarios where image sources cannot be localized due to geometry constraint, which provides a significant gain in performance when their numbers are low. Diminishing returns start when all walls are localizable, and adding more measurements only improves slightly on estimation accuracy.

4.6.3 Real-world Environment

In Figure 4.16, we show photographs of three experiment environments. Two of these environments are small and medium-size rooms with a shoe-box shape, and the third is a slightly larger breakout room with an irregular polygon. In our experiments, we placed the speaker close to the center of the room and collected data at 10 random microphone locations in the presence of clutter. Below each photograph, we show its reconstructed room geometry overlay on top of the ground truth.

An analytic comparison between the proposed method with related work is not trivial since most works have different assumptions and evaluation metrics. To ensure a fair comparison, we compared our approach to [35], which shares similar problem formulation. We ran the solver using the same collected dataset and preprocessing tool, but applied different optimization techniques accordingly. As



(a) Small conference room

(b) Medium-size conference room

(c) Medium-size breakout room

Figure 4.16: Experiment environments (Top) and their 3D reconstruction over ground truth from different views (Middle & Bottom).

shown in Table 4.1, reconstruction using the approach proposed in [35] is infeasible since the correct wall locations are overwhelmed by ambiguous solutions. When applied with the proposed optimization, more erroneous solutions are filtered, and the reconstruction accuracy gains 15% improvement. We also show that the robustness in NEDM approximation plays an important role

Room Size	Max. Similarity			Min. # of Locations		
	MDS [35]	MDS$^+$	SDP$^+$	≥70%	≥80%	≥90%
(a) 60.6 (m^3)	48.2%	63.5%	94.6%	6*	6*	7*
(b) 103.7 (m^3)	50.8%	68.4%	90.9%	6	8*	9*
(c) 132.2 (m^3)	46.0%	61.1%	90.5%	9	10*	11*

Table 4.1: Overall system performance and minimum microphone locations to achieve certain similarity threshold with SDP$^+$ ($^+$ with proposed optimization, * all walls are discovered).

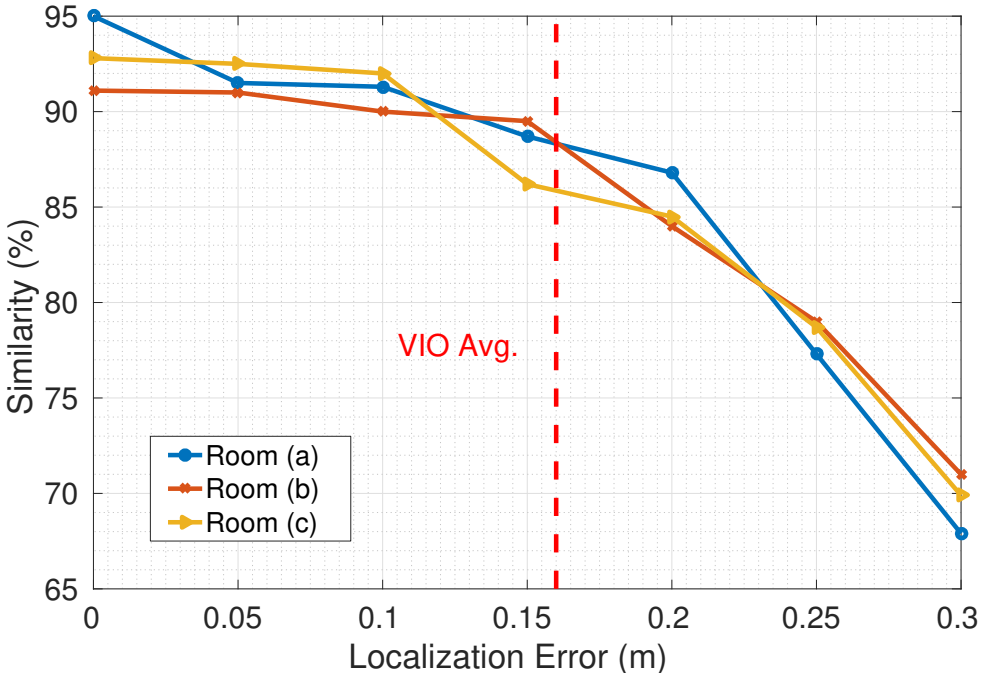


Figure 4.17: Impact of localization error to reconstruction similarity. VIO average highlighted.

in clustering accuracy. Switching the optimization technique from MDS to SDP further improves the overall reconstruction accuracy by 25%. On average, our approach is able to achieve more than 90% reconstruction similarity using a maximum of 10 locations in all room environments. Across different rooms, the number of locations required to achieve the same similarity level slightly increases with the size of the room. We believe this is mainly caused by the attenuation of the signal and can be compensated by proportionally increased output power. An

Summary	Proposed	Dokmanica et al.[35]	Jager et al.[60]	Moore et al.[88]	Zhou et al.[142]
# of speaker(s)	1	1	2	1	1
# of mic.(s)	1	5	5	1	2
Synchronized tx/rx	Yes	Yes	Yes	No	Yes
Known # of walls	No	Yes	Yes	Yes	No
Receive all 1^{st} order echoes	No	Yes	Yes	Yes	Yes
Method	EDM, SDP, combinatorial optimization, clustering, geometry properties, VIO	EDM, MDS	EDM, MDS, graph theory	Geometry properties	IMU, measurement gestures
Complexity	High	High	Medium	Medium	Low
Evaluation environment	Real-world	Real-world	Simulation	Simulation	Real-world
Cons	Require localization of the receiver.	Require careful calibration of the microphone array.		Assume 2D rectangular room shape.	Require localization of the receiver and additional user effort. Limited sensing range.

Table 4.2: System comparison with related work.

increased number of microphone locations can effectively reduce the impact of noise and spurious/missing echoes, which results in improved reconstruction accuracy. We show the minimum number of microphone locations required to achieve certain reconstruction accuracy.

In Figure 4.17, we show the impact of localization error on reconstruction similarity. This localization error is artificially added to the ground truth of the microphone locations in post-processing. We find the performance starts degrading when the localization error exceeds $0.2m$, but appears robust to the typical levels of noise we see from ARKit traces ($0.16m$). Finally, in Table 4.2 we summarize the assumptions and limitations of our proposed approach and related work.

4.6.4 AR Demonstration App

As a way to demonstrate the effectiveness of this sonic sensing approach, we developed an AR phone application that can visualize absorption on wall surfaces in a room. We ran Synesthesia in a small room and collected data from 20 microphone locations with sound-absorbing pads hanging on a wall, with one removed to increase reflectivity. After the model of the room was reconstructed, we derived the exact locations on the walls where echoes are reflected, along with

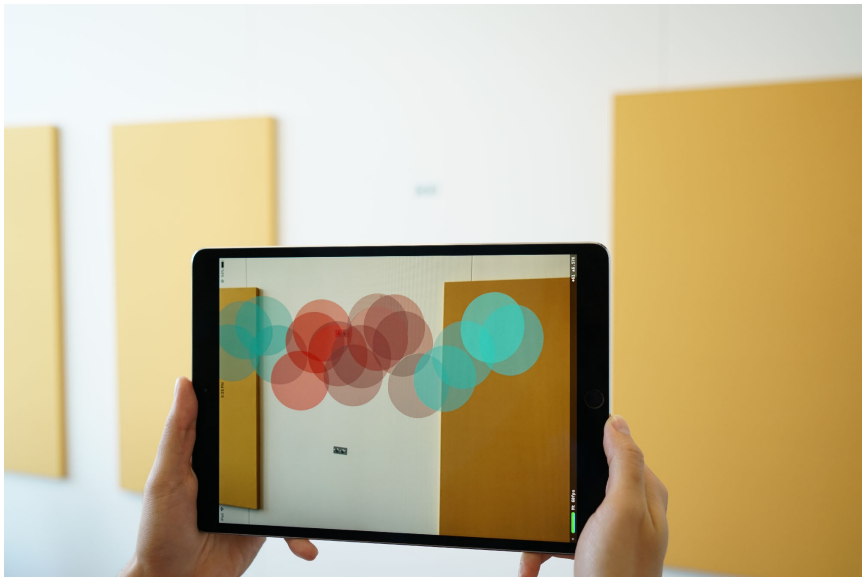


Figure 4.18: Visualization of sound absorption coefficient in AR. Red in color indicates less absorption while blue denotes more.

the echo propagation paths. Next, we computed the combined frequency response of the speaker and microphone using the intensity of the received LOS signal. Finally, we estimated each reflection surface's absorption coefficient based on the intensity of the reflected signal and its propagation path.

The result is registered as circles in the 3D environment in AR, allowing us to visualize the sound absorption. Figure 4.18 shows a photo of the AR app running, where the colored circles represented the absorption coefficient of the reflection surfaces. Each color is mapped to the absorption coefficient across a particular frequency range. To create a denser absorption map, a user simply needs to take additional measurements from more locations. Note that once the room model is obtained, this process is much faster, since we can effectively trim down the candidate combinations using the room geometry.

4.7 Microphone Localization

In Section 4.3 and Section 4.4, we have shown that the room geometry, speaker, and microphone locations can be expressed as the three variables of an equation written in echo distances, and given the speaker and microphone locations, we are able to accurately derive the room geometry. In other words, if we rewrite this equation switching one known variable with an unknown, then the entire system can be used in a reversed manner to localize a microphone given the room geometry and speaker location. In this section, we discuss how to use the same EDM and SDP formulation to perform microphone localization (see Section 4.7.1) and evaluate its performance in real-world environments (see Section 4.7.2).

4.7.1 Revisit the EDM

In Section 4.4, we show that the positioning between image sources and microphone locations can be described as pairwise distances between points in space, or equivalently, an EDM. When reconstructing the room geometry, the composition of the microphone EDM from known microphone locations is the key foundation to recovering noisy ranging measurements from image sources of unknown locations. In this reversed problem, we adopt the same idea, but instead compose an EDM using the image sources of the room geometry. Our goal is then to recover the best distance combination that agrees with the positioning between the image sources and an unknown microphone location.

Assuming the room geometry is a K-face convex polyhedron P in 3D space, it

can then be represented as the intersection of K half-spaces given by

$$P = \{x \in \mathbb{R}^3 : A^T x \leq b\} \tag{4.10}$$

where A is a $3 \times K$ matrix of which each column A_k represents the normal unit vector of a plane and $b_k \in \mathbb{R}$ is its translation. Given a fixed speaker location $s = (s_x, s_y, s_z)$, we can compute all of its first-order image sources I by mirroring s about each plane given by

$$I_k = s - 2A_k(A_k^T s - b_k),\ \forall k = 1, ..., K \tag{4.11}$$

These image sources serve as the anchor points in many TOF-based localization systems, and they are inherently synchronized based on the IS model. Higher-order image sources are discarded when constructing the EDM, since higher-order reflections are less likely to be detected due to reduced amplitude and geometry constraints. Similar to the EDM constructed in Section 4.4.2, we transform the distances between the image sources into an EDM $D_I \in \mathcal{E}^K$. Suppose M TOF distances are extracted from the received signal and denoted by $d = [d_1, ..., d_m]$. Then, in an ideal scenario, there exists a unique combination $c = [c_1^2, ..., c_K^2]$ for $c_k \in d$ such that the augmented matrix $\bar{D}_{I+}$ given by

$$\bar{D}_{I+} = \begin{pmatrix} [D_I] & [c^T] \\ [c] & 0 \end{pmatrix}$$

is also an EDM matrix in $\mathcal{E}^{K+1}$. In reality, we face the same challenges as in room reconstruction (such as measurement inaccuracy, missing/spurious echoes, and excessive computation time), but these challenges can be solved following the same optimization process as previously discussed in Section 4.4.3, Section 4.4.6,

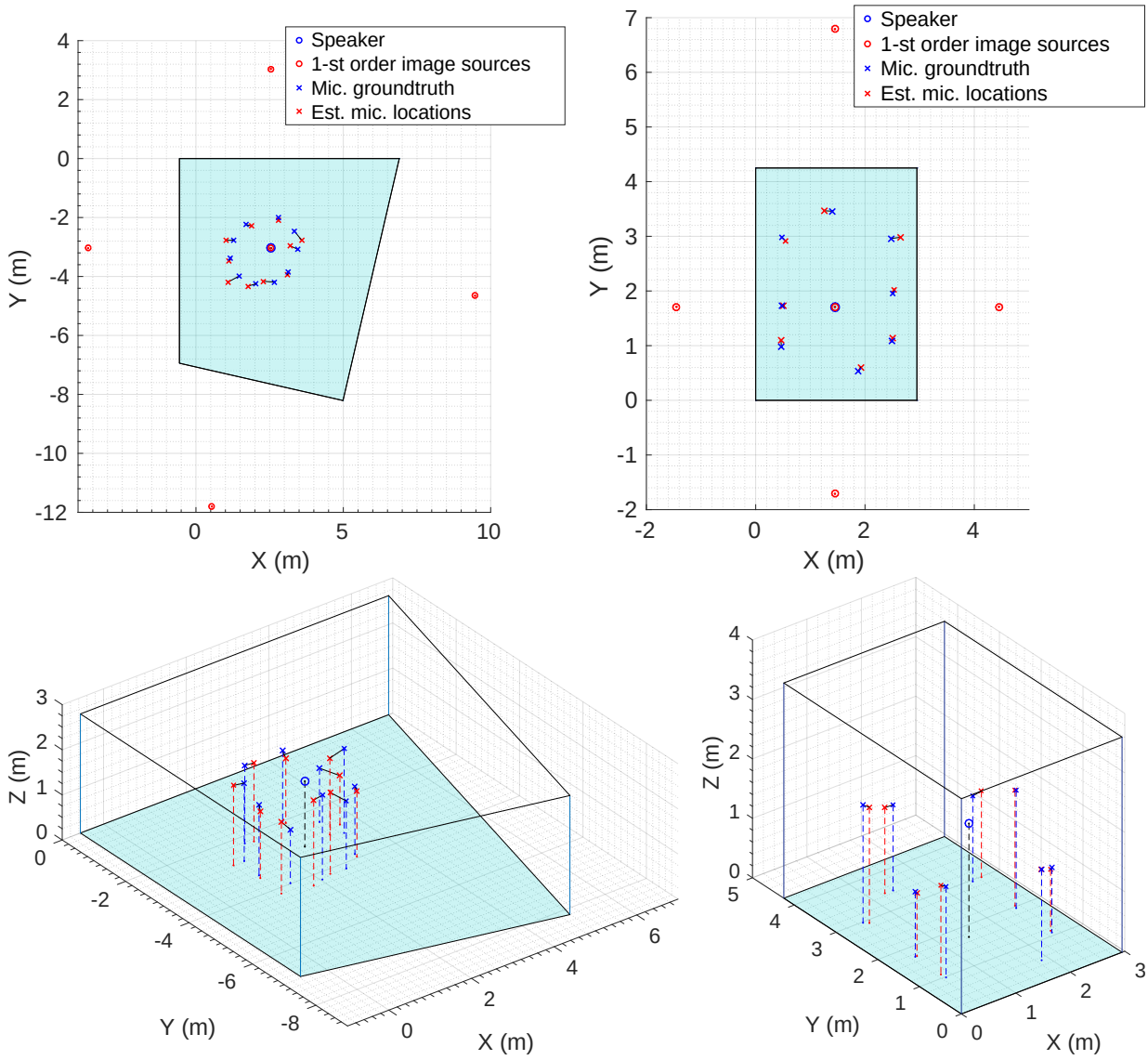
Room Size	2D (X,Y)		3D (X,Y,Z)	
	Mean (m)	Std. (m)	Mean (m)	Std. (m)
(a) 60.6 (m^3)	0.097	0.047	0.106	0.041
(b) 103.7 (m^3)	0.199	0.157	0.222	0.152
(c) 132.2 (m^3)	0.260	0.122	0.316	0.155
Overall	0.191	0.135	0.222	0.153

Table 4.3: Mean localization error in 2D and 3D.

and Section 4.4.5. In addition, we can further eliminate invalid solutions using the room geometry (Equation 4.10) as a constraint. One main difference, however, is that the estimation accuracy now depends on the fixed number of reflective walls in the space, rather than the number of sampling locations. More reflective surfaces give more potential ranging anchors and thus better localization accuracy and robustness against missing echoes.

4.7.2 Localization Performance

We empirically validate the localization performance in the same room environments (see Figure 4.16) with real recordings sampled at random locations. Note that each localization result is derived using one single recording without averaging over multiple samples. In Figure 4.19, we illustrate both 2D and 3D localization results along with the 1st order image sources we used to form the EDM. The average localization error is summarized in Table 4.3, and its overall cumulative distribution function is shown in Figure 4.20. On average, we are able to achieve less than 20cm of localization error in 2D with a worst case of 43cm, and less than 30cm of localization error in 3D with a worst case of 58cm. A clear trend can be seen from Table 4.3: the average localization error increases with room size. This performance loss in large rooms is expected, since the received signal contains more spurious multipath reflections from the environment and it is



(a) Medium-size breakout room

(b) Small conference room

Figure 4.19: Microphone localization results rendered in 2D (Top) and 3D (Bottom) viewpoint.

harder to isolate them from background noise accurately due to greater path loss. Since the NEDM optimization is quite sensitive to ranging error, we also observe a high variance in localization accuracy.

Another observation worth noting is that since most room geometries have vertical walls, their corresponding image sources will end up sitting on the same

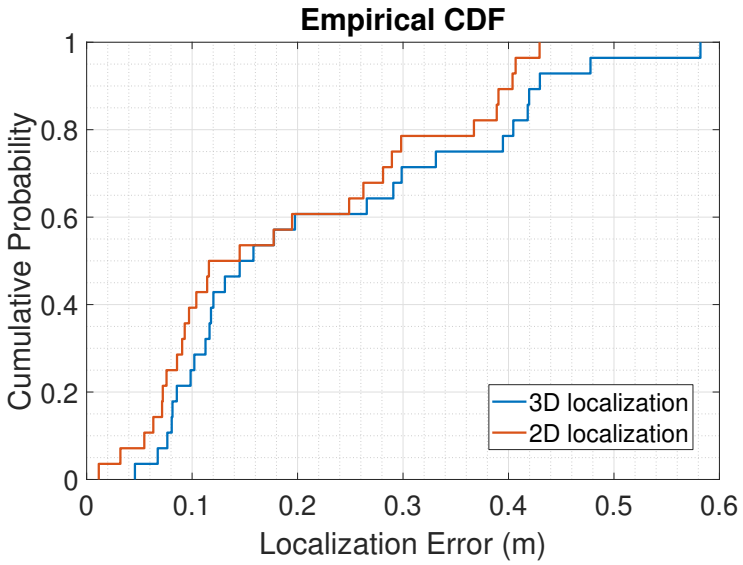


Figure 4.20: The overall cumulative distribution function of the localization error.

x-y plane. The lack of diversity in the z-axis, in turn, makes 3D localization more difficult, especially when reflections from the ceiling/floor are blocked. On the contrary, if the reflections from the ceiling/floor can be successfully captured, most localization error tends to concentrate in the 2D dimensions due to ranging inaccuracy induced by the obstacles in the environment.

Chapter 5

Conclusion and Future Work

In this dissertation, we explored the capabilities of enhancing active acoustic sensing using modern data-driven approaches. We utilized ultrasonic wide-band signals to capture various properties of our surrounding environment and nearby people. By using learning algorithms, we were able to extract enriched inferences from raw data waveforms and fully utilize the information carried in their reverberations. We demonstrated this ability in rigorous examples, including occupancy estimation, room geometry sensing, acoustic model reconstruction, and microphone localization. In summary, this dissertation makes the following contributions:

1. **Occupancy Estimation**

   We introduced an approach to estimate room occupancy by using reverberation across multiple frequencies. We evaluated numerous characteristics of the impulse signal and their impact on system performance. The estimation algorithm adopts a semi-supervised learning scheme to enable an effortless training procedure and minimize the model size. It is also robust against common environmental interference and able to automatically recalibrate its model when

the room environment changes over time.

We implemented the algorithm on our embedded platform AURES, which supports user-assisted training and labeling over BLE or 802.15.4 connectivity. To improve energy efficiency and scalability, we proposed a volume control mechanism and an energy-harvesting subsystem with benchmark test. We evaluated our system in 10 different rooms on campus with various sizes and geometries, and collected daily use data for 2 weeks, totaling over 60,000 data samples. Our results showed an average recall rate of 85% for presence detection, and less than 12% estimation error for people counting. In outdoor environments, AURES showed potential in line detection and achieved less than 10% estimation error with a sensing range of $6m$.

2. **Room Geometry Sensing, Acoustic Model Reconstruction, and Microphone Localization**

We proposed an approach for estimating the locations of reflective surfaces and forming a high-resolution image of the space, given a single acoustic source with multiple noisy microphone measurements. The same approach can also be used in reverse to perform microphone localization given a known room geometry.

Our algorithm utilizes a pipeline of optimization techniques to eliminate conventional assumptions on room geometry and detection of echoes. It is also robust against ranging error in recording data and missing/spurious echoes from the environment.

We presented the platform Synesthesia, which uses a single centrally located speaker and visual inertial odometry on a mobile phone for tracking. We showed through both simulation and experimentation that even with $20cm$ of uncertainty in the microphone locations, which is larger than the average VIO error found in

phones, we are still able to reconstruct the room geometry with more than 90% accuracy. We also demonstrated an augmented reality tool that can visualize an estimate of the sound absorption coefficient of materials in a room.

The same platform and algorithms can also be used to perform 3D localization given the geometry of the room. On average, our system achieved less than $20cm$ localization error in 2D and less than $30cm$ in 3D in various real-world environments.

5.1 Future Work

In this section, we discuss limitations and future improvements for our proposed systems and explore possibilities of extending their capabilities to broader applications.

5.1.1 Acoustic Impulse Signal

At its core, active acoustic sensing relies on the design of the impulse signal to gather useful information. In both Section 3.2 and Section 4.2, we have shown how temporal and spectral properties of the transmitted signal can impact our perception of the physical world. One additional dimension yet to be explored is the modulation on top of the signal. For example, in [79], rate adaptive Chirp Spread Spectrum (CSS) was used to support multiple access between concurrent localization beacons. The same technique can potentially be applied to multiple occupancy sensing systems working together to cover a larger ground without incurring interference. Another example can be found in [91], where OFDM was applied to help improve the ranging accuracy. We envision using this technique to improve the localization of acoustic reflectors and the reconstruction of room

geometry.

5.1.2 People Counting and Beyond

Our people-counting system has a few practical limitations. In Section 3.2, we chose our frequency range because it is supported by low-cost commercial audio codecs and it is the lowest inaudible frequency that attenuates significantly less than higher frequency narrow-band transducers. For this reason, the signal may be perceptible to service animals. Though our target duty-cycles and volume levels are designed to aggressively optimize energy and should be almost undetectable to most animals, more empirical testing is required. Aside from transducer cost, there is no reason why this approach cannot operate at higher frequencies to improve performance. At higher frequencies, sound becomes more directional, so further investigation would be required to determine if reverberation is still as sensitive to person count.

To achieve high performance in large spaces, our system will need a proportionally powerful transmitter that requires a larger amplifier and transducer. And as the space increases in size, the ability to finely distinguish the exact number of people diminishes. One possible improvement is to run multiple AURES nodes in the same space that work collaboratively to estimate the combined load. To improve the scalability of this scheme, we would require a mechanism to coordinate transmissions so that they do not experience cross-talk. Common multiple access schemes based on TDMA, FDMA, or coding could be utilized for this purpose, and peer-to-peer time synchronization could be achieved using on-board BLE or 802.15.4 connectivity [79].

Our current system also requires labeling of training data as mentioned in Section 3.4.4. While the proposed algorithm greatly reduces the amount of

labeling, and the mobile phone interface can simplify the training process, an installer still needs to capture a snapshot when the room has a reasonable (≥10%) occupancy level, which might be difficult in some cases. In the future, we intend to investigate using a DNN model to train a generalized model with more training data in a semi-supervised learning manner. We envision building a universal model that can be rapidly deployed in rooms of various sizes/geometries, and bootstrapped using just the empty room reverberation identified by our presence detector. This DNN-based solution will ultimately free the system from any manual inputs and eliminate the expensive labeling process carried out on a per-room basis. As more training data are collected from different room environments, the DNN model could fully utilize the combined result and generalize more precisely the key features that reflect the occupancy. Our geometry reconstruction algorithms may also be used to facilitate this process and improve our estimation accuracy, since we could better model and isolate reverberation coming from the room environment. The ability to isolate reverberation from specific objects may further be exploited to recognize gestures or ongoing activities along with people counting.

Aside from estimating occupancy in room environments, the same techniques can potentially be utilized in many different contexts as well. For example, in automobiles, we envision using in-car sound systems to estimate the number of passengers and their positions for air quality optimization. Our algorithm may also be trained to detect infants left in rear car seats for safety purposes. Instead of monitoring human bodies, the algorithm could also learn to identify animals, such as cats and dogs, to improve acoustic-based home security systems by reducing pet-triggered false alarms or installation difficulties.

5.1.3 Acoustic Imaging and Beyond

As previously discussed in Section 4.4.5, one of the main drawbacks of our acoustic imaging system is its long computation time. Even with the proposed searching optimization, it takes on average 1 hour to achieve more than 90% reconstruction similarity in MATLAB with a dual Intel Core i7 CPU. We expect to see a substantial speedup through improvements in peak selection (see Section 4.2) and searching heuristics (see Section 4.4.5). Peak selection would likely benefit from applying window functions to increase the peak-to-sidelobe ratio. However, since window functions often introduce drawbacks such as reduced overall gain and wider main-lobe, a trade-off between sensing range, ranging resolution, and computational complexity would be expected as a result. To improve our searching heuristic, relevant techniques from the literature of combinatorial search may help to further reduce the runtime complexity. We envision combining the local search with lookahead techniques, such as Monte Carlo Tree Search (MCTS), to quickly converge to an initial set of good combinations and then discover potential solutions in its neighborhood. On the other hand, we believe with more optimized implementation, parallelization, and GPU acceleration, it is possible to reduce the current computation time to a few minutes, or even seconds. In most applications, we envision users capturing an image, pushing the data to the cloud, and then retrieving it later for viewing.

We also plan to conduct more evaluations on our embedded mobile platform. This will involve additional sensitivity analysis on the frequency response of onboard microphones and a thorough comparison between different VIO techniques supported by these platforms. To improve real-world performance of VIO, user studies may be required to evaluate how the orientation of phones

impacts its tracking accuracy and how well it performs in a more adverse environment. A fusion of VIO with other mobile phone localization techniques (wireless and acoustic) should also be studied for potential improvements on localization accuracy. In addition, the quad-sector speaker array could potentially provide more spatial information if each sector transmits the signal in a time-division manner. While this approach may extend the time of data collection, correlating echoes based on the sector that transmits the signal may greatly improve the computation speed and reconstruction accuracy. Furthermore, advanced techniques such as beamforming could also be applied on the speaker array to enable finer control of the signal's direction.

It is also evident that our problem formulation shares a close relationship with Simultaneous Localization and Mapping (SLAM). For instance, the image sources we introduced can be treated as unique landmarks in SLAM, even though its quantity would be relatively limited compared to a typical SLAM problem. Well-studied probabilistic models in SLAM literature, such as the Kalman filter (KF) and particle filter, could potentially be utilized to help model missing/spurious echoes and improve estimation accuracy. Reversely, the geometrical information embedded in echoes could be exploited to perform Range-Only SLAM [14, 70]. This may provide an opportunity to alleviate the need for fine-grain sensors or transceiver arrays. The underlying EDM and SDP framework have also been studied in other research areas and applications such as facial reduction [37], Sensor Network Localization (SNL) [13], graph realization, and graph rigidity [121]. We believe relevant techniques, especially regularized SDP relaxation, may be applied to derive more accurate and robust solutions. We will be closely following these research areas in the future.

Bibliography

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5 (1):13–51, 1995. doi: 10.1137/0805002. URL https://doi.org/10.1137/0805002. 92

[2] J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(943), 1979. 25, 81

[3] Amazon Echo. https://www.amazon.com/echo, 2018. [Online; accessed 13-March-2018]. 76

[4] E. D. Andersen and K. D. Andersen. *The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm*, pages 197–232. Springer US, Boston, MA, 2000. ISBN 978-1-4757-3216-0. doi: 10.1007/978-1-4757-3216-0_8. URL http://dx.doi.org/10.1007/978-1-4757-3216-0_8. 95

[5] F. Antonacci, A. Sarti, and S. Tubaro. Geometric reconstruction of the environment from its response to multiple acoustic emissions. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2822–2825, March 2010. doi: 10.1109/ICASSP.2010.5496186. 24

[6] F. Antonacci, J. Filos, M. R. P. Thomas, E. A. P. Habets, A. Sarti, P. A. Naylor, and S. Tubaro. Inference of room geometry from acoustic impulse responses. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(10):2683 – 2695, July 2012. 10, 24, 79

[7] Apple ARKit. `https://developer.apple.com/arkit/`, 2018. [Online; accessed 13-March-2018]. 78

[8] Apple Homepod. `https://www.apple.com/homepod/`, 2018. [Online; accessed 13-March-2018]. 76

[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, Apr. 2010. ISSN 0001-0782. doi: 10.1145/1721654.1721672. URL `http://doi.acm.org/10.1145/1721654.1721672`. 13

[10] A. Asaei, M. Golbabaee, H. Bourlard, and V. Cevher. Structured sparsity models for reverberant speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(3):620–633, March 2014. ISSN 2329-9290. 76

[11] A. Balleri, K. Chetty, and K. Woodbridge. Classification of personnel targets by acoustic micro-doppler signatures. *IET Radar, Sonar Navigation*, 5(9):943–951, Dec 2011. ISSN 1751-8784. doi: 10.1049/iet-rsn.2011.0087. 11

[12] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. *Continuous Realtime Gesture Following and Recognition*, pages 73–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-12553-9. doi: 10.1007/978-3-642-12553-9_7. URL `https://doi.org/10.1007/978-3-642-12553-9_7`. 12

[13] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye. Semidefinite programming based

algorithms for sensor network localization. *ACM Trans. Sen. Netw.*, 2(2):188–220, May 2006. ISSN 1550-4859. doi: 10.1145/1149283.1149286. URL http://doi.acm.org/10.1145/1149283.1149286. 127

[14] J. L. Blanco, J. A. Fernandez-Madrigal, and J. Gonzalez. Efficient probabilistic range-only slam. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1017–1022, Sept 2008. doi: 10.1109/IROS.2008.4650650. 127

[15] C. A. Andree. The effect of position on the absorption of materials for the case of a cubical room. *Journal on the Acoustics Society of America*, 1932. 17

[16] C. C. Loy, S. Gong, and T. Xiang. From semi-supervised to transfer counting of crowds. In *International Conference on Computer Vision*, 2013. 17, 20, 21

[17] D. Caicedo and A. Pandharipande. Ultrasonic array sensor for indoor presence detection. In *Signal Processing Conference (EUSIPCO)*, 2012. 17, 18, 19

[18] A. Canclini, P. Annibale, F. Antonacci, A. Sarti, R. Rabenstein, and S. Tubaro. From direction of arrival estimates to localization of planar reflectors in a two dimensional geometry. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2620–2623, May 2011. doi: 10.1109/ICASSP.2011.5947022. 10, 24

[19] A. Canclini, F. Antonacci, M. R. P. Thomas, J. Filos, A. Sarti, P. A. Naylor, and S. Tubaro. Exact localization of acoustic reflectors from quadratic constraints. In *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 17–20, Oct 2011. doi: 10.1109/ASPAA.2011.6082277. 10, 24

[20] M. M. Carroll and C. F. Chien. Decay of reverberant sound in a spherical

enclosure. *The Journal of the Acoustical Society of America*, 62(6):1442–1446, 1977. 30

[21] L. Cayton and S. Dasgupta. Robust euclidean embedding. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 169–176, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143866. URL http://doi.acm.org/10.1145/1143844.1143866. 92

[22] A. B. Chan, C. La Jolla, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition*, 2008. 17, 20, 72

[23] S. J. Chapman. Drums that sound the same. *The American Mathematical Monthly*, 102(2):124–138, 1995. ISSN 00029890, 19300972. URL http://www.jstor.org/stable/2975346. 75

[24] J. Chen, A. H. Kam, J. Zhang, N. Liu, and L. Shue. *Bathroom Activity Monitoring Based on Sound*, pages 47–61. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-32034-0. doi: 10.1007/11428572_4. URL https://doi.org/10.1007/11428572_4. 12

[25] Y. Cherapanamjeri, P. Jain, and P. Netrapalli. Thresholding based efficient outlier robust PCA. *CoRR*, abs/1702.05571, 2017. URL http://arxiv.org/abs/1702.05571. 14

[26] M. Coutino, M. B. Møller, J. K. Nielsen, and R. Heusdens. Greedy alternative for room geometry estimation from acoustic echoes: A subspace-based method. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 366–370, March 2017. doi: 10.1109/ICASSP.2017.7952179. 27

[27] M. Crocco, A. Trucco, V. Murino, and A. D. Bue. Towards fully uncalibrated room reconstruction with sound. In *Signal Processing Conference (EUSIPCO), Proceedings of the 22nd European*, 2014. 10, 24, 25, 79

[28] D. Li, B. Balaji, Y. Jiang, and K. Singh. A wi-fi based occupancy sensing approach to smart energy in commercial office buildings. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '12, pages 197–198, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1170-0. 17, 18

[29] J. Dattorro. Convex optimization & euclidean distance geometry, 2005. 92, 93, 94

[30] W. J. Davies, Y. W. Lam, and R. J. Orlowski. Predicting theater chair absorption from reverberation chamber measurements. *Journal of the Acoustical Society of America*, 93(4):2238–2240, April 1993. 17

[31] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, Aug 1980. ISSN 0096-3518. doi: 10.1109/TASSP.1980.1163420. 12

[32] C. de Bakker, T. van de Voort, and A. Rosemann. The energy saving potential of occupancy-based lighting control strategies in open-plan offices: The influence of occupancy patterns. *Energies*, 11(1), 2018. ISSN 1996-1073. doi: 10.3390/en11010002. URL `http://www.mdpi.com/1996-1073/11/1/2`. 29

[33] L. Deng and X. Li. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5): 1060–1089, May 2013. ISSN 1558-7916. doi: 10.1109/TASL.2013.2244083. 12

[34] A. Dobrucki, B. Żółtogórski, P. Pruchnicki, and R. Bolejko. Sound-absorbing and insulating enclosures for ultrasonic range. *Archives of Acoustics*, 35(2):157 – 164, May 2010. 108

[35] I. Dokmanica, R. Parhizkara, A. Walthera, Y. M. Lub, and M. Vetterlia. Acoustic echoes reveal room shape. *Proceeding of the National Academy of Science of the United States of America*, 110(30), July 2013. 10, 24, 26, 79, 88, 90, 91, 110, 111, 112, 113

[36] I. Dokmanić, Y. Lu, and M. Vetterli. Can one hear the shape of a room: The 2-d polygonal case. In *2011 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011 - Proceedings*, pages 321–324, 2011. ISBN 9781457705397. doi: 10.1109/ICASSP.2011.5946405. 24

[37] D. Drusvyatskiy, N. Krislock, Y.-L. Voronin, and H. Wolkowicz. Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier. *ArXiv e-prints*, Oct. 2014. 127

[38] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, Jan. 1972. ISSN 0001-0782. doi: 10.1145/361237.361242. URL `http://doi.acm.org/10.1145/361237.361242`. 24

[39] Y. Fang, H. Feng, and Y. Chen. A robust interaural time differences estimation and dereverberation algorithm based on the coherence function. *Applied Acoustics*, 129:126 – 134, 2018. ISSN 0003-682X. URL `http://www.sciencedirect.com/science/article/pii/S0003682X17302852`. 76

[40] S. Fernández, A. Graves, and J. Schmidhuber. Sequence labelling in structured domains with hierarchical recurrent neural networks. In *IN PROC. 20TH INT. JOINT CONF. ON ARTIFICIAL INTELLIGENCE, IJCAI 2007*, pages 774–779, 2007.

13

[41] J. Filos, A. Canclini, F. Antonacci, A. Sarti, and P. A. Naylor. Localization of planar acoustic reflectors from the combination of linear estimates. In 2012 *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1019–1023, Aug 2012. 24, 25

[42] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL http://doi.acm.org/10.1145/358669.358692. 24

[43] S. K. Ghai, L. V. Thanayankizil, D. P. Seetharam, and D. Chakraborty. Occupancy detection in commercial buildings using opportunistic context sources. In 2012 *IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 463–466, March 2012. doi: 10.1109/PerComW.2012.6197536. 17, 21

[44] Google ARCore. https://developers.google.com/ar/, 2018. [Online; accessed 13-March-2018]. 78

[45] Google Glass. https://www.x.company/glass/, 2018. [Online; accessed 13-March-2018]. 86

[46] Google Home. https://store.google.com/product/google_home, 2018. [Online; accessed 13-March-2018]. 76

[47] C. Gordon, D. L. Webb, and S. Wolpert. One cannot hear the shape of a drum. *ArXiv Mathematics e-prints*, June 1992. 75

[48] S. Goyal, H. A. Ingley, and P. Barooah. Occupancy-based zone-climate control for energy-efficient buildings: Complexity vs. performance. *Applied*

Energy, 106:209 – 221, 2013. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2013.01.039. URL http://www.sciencedirect.com/science/article/pii/S0306261913000482. 29

[49] A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013. URL http://arxiv.org/abs/1303.5778. 13

[50] A. Gupta, C. G. S. Suggala, A. Gupta, H. Simhadri, B. Paranjape, A. Kumar, S. Goyal, R. Udupa, M. Varma, and P. Jain. Protonn: Compressed and accurate knn for resource-scarce devices. February 2017. URL https://www.microsoft.com/en-us/research/publication/protonn-compressed-accurate-knn-resource-scarce-devices/. 14

[51] E. Hailemariam, R. Goldstein, R. Attar, and A. Khan. Real-time occupancy detection using decision trees with multiple sensor types. In *Symposium on Simulation for Architecture and Urban Design*, 2011. 17, 19

[52] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR*, abs/1510.00149, 2015. URL http://arxiv.org/abs/1510.00149. 13

[53] M. Hazas and A. Ward. A high performance privacy-oriented location system. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, PERCOM '03, pages 216–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1893-1. URL http://dl.acm.org/citation.cfm?id=826025.826383. 10

[54] M. Hazas and A. Ward. A high performance privacy-oriented location system. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing*

and Communications (PERCOM '03), pages 216–223, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1893-1. URL http://dl.acm.org/citation.cfm?id=826025.826383. 81

[55] A. Hein. *Processing of SAR Data: Fundamentals, Signal Processing, Interferometry*. Springer Publishing Company, Incorporated, 1st edition, 2010. ISBN 3642057101, 9783642057106. 83

[56] T. Hidaka and N. Nishihara. Reverberation time, mean-free-path, and sound absorption in concert halls-numerical examination by computer simulation. *The Journal of the Acoustical Society of America*, 119(5):3430–3430, 2006. 17

[57] T. Hidaka, N. Nishihara, and L. L. Beranek. Relation of acoustical parameters with and without audiences in concert halls and a simple method for simulating the occupied state. *The Journal of the Acoustical Society of America*, 109, 2001. 17

[58] T. W. Hnat, E. Griths, R. Dawson, and K. Whitehouse. Doorjamb: Unobtrusive room-level tracking of people in homes using doorway sensors. In *ACM Conference on Embedded Network Sensor Systems*, 2012. 17, 21, 22

[59] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735. 13

[60] I. Jager, R. Heusdens, and N. D. Gaubitch. Room geometry estimation from acoustic echoes using graph-based echo labeling. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, March 2016. 10, 24, 26, 27, 79, 88, 113

[61] C. R. Johnson. Connections between the real positive semidefinite and

distance matrix completion problems. *Linear Algebra and its Applications*, 223 - 224:375 – 391, July 1995. 93

[62] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section* A, 32(5):922–923, 1976. doi: 10.1107/S0567739476001873. URL https://onlinelibrary.wiley.com/doi/abs/10.1107/S0567739476001873. 109

[63] M. Kac. Can one hear the shape of a drum? *The American Mathematical Monthly*, 73(4):1–23, 1966. ISSN 00029890, 19300972. URL http://www.jstor.org/stable/2313748. 75

[64] R. Kashyap, I. Singh, and S. S. Ram. Micro-doppler signatures of underwater vehicles using acoustic radar. In *2015 IEEE Radar Conference (RadarCon)*, pages 1222–1227, May 2015. doi: 10.1109/RADAR.2015.7131181. 11

[65] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta. Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 137–149. ACM, 2016. 81

[66] J.-W. Kim, K.-S. Choi, B.-D. Choi, and S.-J. Ko. Real-time vision-based people counting system for the security door. 2002. 17, 21

[67] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014. URL http://arxiv.org/abs/1406.5298. 13

[68] A. J. Kolarik, S. Cirstea, S. Pardhan, and B. C. Moore. A summary of research investigating echolocation abilities of blind and sighted humans. *Hearing Research*, 310:60 – 68, 2014. ISSN 0378-5955. doi: https://doi.org/10.1016/

j.heares.2014.01.010. URL http://www.sciencedirect.com/science/article/pii/S0378595514000185. 1

[69] C. D. Korkas, S. Baldi, I. Michailidis, and E. B. Kosmatopoulos. Occupancy-based demand response and thermal comfort optimization in microgrids with renewable energy sources and energy storage. *Applied Energy*, 163:93 – 104, 2016. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2015.10.140. URL http://www.sciencedirect.com/science/article/pii/S0306261915013823. 29

[70] M. Kreković, I. Dokmanić, and M. Vetterli. Echoslam: Simultaneous localization and mapping with acoustic echoes. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11–15, March 2016. doi: 10.1109/ICASSP.2016.7471627. 127

[71] N. Krislock and H. Wolkowicz. *Euclidean Distance Matrices and Applications*, pages 879–914. Springer US, Boston, MA, 2012. ISBN 978-1-4614-0769-0. doi: 10.1007/978-1-4614-0769-0_30. URL http://dx.doi.org/10.1007/978-1-4614-0769-0_30. 93

[72] A. Kumar, S. Goyal, and M. Varma. Resource-efficient machine learning in 2 kb ram for the internet of things. May 2017. URL https://www.microsoft.com/en-us/research/publication/resource-efficient-machine-learning-2-kb-ram-internet-things/. 14

[73] M. J. Kusner, S. Tyree, K. Weinberger, and K. Agrawal. Stochastic neighbor compression. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–622–II–630. JMLR.org, 2014. URL http://dl.acm.org/citation.cfm?id=

3044805.3044962. 13

[74] M. Kuster, D. de Vries, E. M. Hulsebos, and A. Gisolf. Acoustic imaging in enclosed spaces: Analysis of room geometry modifications on the impulse response. *The Journal of the Acoustical Society of America*, 116(4):2126–2137, 2004. doi: 10.1121/1.1785591. URL https://doi.org/10.1121/1.1785591. 24

[75] L. L. Beranek. Analysis of sabine and eyring equations and their application to concert hall audience and chair absorption. *The Journal of the Acoustical Society of America*, 2006. 23, 36

[76] K. P. Lam, M. Hoynck, B. Dong, B. Andrews, Y. shang Chiou, D. Benitez, and J. Choi. Occupancy detection through an extensive environmental sensor network in an open-plan office building. In *Proc. of Building Simulation 09, an IBPSA Conference*, 2009. 17, 21, 72

[77] M. Laurent. A connection between positive semidefinite and euclidean distance matrix completion problems. *Linear Algebra and its Application*, 273 (1 - 3):9 – 22, April 1998. 93

[78] P. Lazik and A. Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 99–112, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1169-4. doi: 10.1145/2426656.2426667. URL http://doi.acm.org/10.1145/2426656.2426667. 10, 36, 85

[79] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe. Alps: A bluetooth and ultrasound platform for mapping and localization. In *SenSys*, 2015. 123, 124

[80] P. Lazik, N. Rajagopal, B. Sinopoli, and A. Rowe. Ultrasonic time synchronization and ranging on smartphones. In *Proceedings of the 21st IEEE*

Real-Time and Embedded Technology and Applications Symposium (RTAS 2015), RTAS '15. IEEE, IEEE, 2015. 60

[81] S.-M. Lee, S.-H. Fang, J. weih Hung, and L.-S. Lee. Improved mfcc feature extraction by pca-optimized filter-bank for speech recognition. In *Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop on*, pages 49–52, 2001. doi: 10.1109/ASRU.2001.1034586. 12

[82] P. Lopes, R. Jota, and J. A. Jorge. Augmenting touch interaction through acoustic sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 53–56, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7. doi: 10.1145/2076354.2076364. URL `http://doi.acm.org/10.1145/2076354.2076364`. 11

[83] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*, 1996. 41, 46, 104

[84] H. Malik. Acoustic environment identification and its applications to audio forensics. *IEEE Transactions on Information Forensics and Security*, 8(11):1827–1837, Nov 2013. ISSN 1556-6013. 76

[85] O. Masoud and N. P. Papanikolopoulos. A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Transactions on Vehicular Technology*, 50(5):1267–1278, Sep 2001. ISSN 0018-9545. doi: 10.1109/25.950328. 17, 20

[86] M. McCarthy, P. Duff, H. L. Muller, and C. Randell. Accessible ultrasonic positioning. *IEEE Pervasive Computing*, 5(4):86–93, Oct. 2006. ISSN 1536-1268. doi: 10.1109/MPRV.2006.65. URL `http://dx.doi.org/10.1109/MPRV.2006.65`.

81

[87] Microsoft Hololens. https://www.microsoft.com/en-us/hololens/, 2018. [Online; accessed 13-March-2018]. 86

[88] A. H. Moore, M. Brookes, and P. A. Naylor. Room geometry estimation from a single channel acoustic impulse response. In *Signal Processing Conference (EUSIPCO), Proceedings of the 21st European*, 2013. 10, 24, 26, 79, 113

[89] S. Munir, R. S. Arora, C. Hesling, J. Li, J. Francis, C. Shelton, C. Martin, A. Rowe, and M. Berges. Real-time fine grained occupancy estimation using depth sensors on arm embedded platforms. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 295–306, April 2017. doi: 10.1109/RTAS.2017.8. 17, 21, 22

[90] F. Nan, J. Wang, and V. Saligrama. Pruning Random Forests for Prediction on a Budget. *ArXiv e-prints*, June 2016. 13

[91] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1515–1525, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858580. URL http://doi.acm.org/10.1145/2858036.2858580. 11, 123

[92] N. Nasir, K. Palani, A. Chugh, V. C. Prakash, U. Arote, A. P. Krishnan, and K. Ramamritham. Fusing sensors for occupancy sensing in smart buildings. In R. Natarajan, G. Barua, and M. R. Patra, editors, *Distributed Computing and Internet Technology*, pages 73–92, Cham, 2015. Springer International Publishing. ISBN 978-3-319-14977-6. 17, 21, 22

[93] E. Nastasia, F. Antonacci, A. Sarti, and S. Tubaro. Localization of planar acoustic

reflectors through emission of controlled stimuli. In *2011 19th European Signal Processing Conference*, pages 156–160, Aug 2011. 24, 25

[94] N. Nesa and I. Banerjee. Iot-based sensor data fusion for occupancy sensing using dempster -shafer evidence theory for smart buildings. *IEEE Internet of Things Journal*, 4(5):1563–1570, Oct 2017. doi: 10.1109/JIOT.2017.2723424. 17, 19

[95] G. Neuweiler. Auditory adaptations for prey capture in echolocating bats. *Physiological Reviews*, 70(3):615–641, 1990. doi: 10.1152/physrev.1990.70.3.615. URL `https://doi.org/10.1152/physrev.1990.70.3.615`. PMID: 2194220. 34

[96] T. A. Nguyen and M. Aiello. Beyond indoor presence monitering with simple sensors. In *2nd International Conference on Pervasive and Embedded Computing and Communication Systems*, 2012. 17, 19

[97] S. Novotney and C. Callison-Burch. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 207–215, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL `http://dl.acm.org/citation.cfm?id=1857999.1858023`. 15

[98] F. Oldewurtel, D. Sturzenegger, and M. Morari. Importance of occupancy information for building climate control. *Applied Energy*, 101:521 – 532, 2013. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2012.06.014. URL `http://www.sciencedirect.com/science/article/pii/S0306261912004564`. Sustainable Development of Energy, Water and Environment Systems. 29

[99] M. Ono, B. Shizuki, and J. Tanaka. Touch & activate: Adding interactivity

to existing objects using active acoustic sensing. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 31–40, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2501989. URL http://doi.acm.org/10.1145/2501988.2501989. 11

[100] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69, 2005. 91

[101] C. Peng, G. Shen, Z. Han, Y. Zhang, Y. Li, and K. Tan. A beepbeep ranging system on mobile phones. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 397–398, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-763-6. doi: 10.1145/1322263.1322313. URL http://doi.acm.org/10.1145/1322263.1322313. 10

[102] F. Peng, T. Wang, and B. Chen. Room shape reconstruction with a single mobile acoustic sensor. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1116–1120, Dec 2015. 26, 27

[103] M. Pollefeys and D. Nister. Direct computation of sound and microphone locations from time-difference-of-arrival data. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2008. 26

[104] N. B. Priyantha. *The Cricket Indoor Location System*. PhD thesis, Cambridge, MA, USA, 2005. AAI0808861. 10

[105] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (Mobicom '00)*, pages 32–43, New York, NY, USA, 2000. ACM. ISBN 1-58113-197-6. doi: 10.1145/345910.345917. URL

http://doi.acm.org/10.1145/345910.345917. 81

[106] H.-D. Qi and X. Yuan. Computing the nearest euclidean distance matrix with low embedding dimensions. *Mathematical Programming*, 147(1):351–389, 2014. ISSN 1436-4646. doi: 10.1007/s10107-013-0726-0. URL http://dx.doi.org/10.1007/s10107-013-0726-0. 93

[107] B. Raj, K. Kalgaonkar, C. Harrison, and P. Dietz. Ultrasonic doppler sensing in hci. *Pervasive Computing, IEEE*, 11(2):24–29, Feb 2012. ISSN 1536-1268. doi: 10.1109/MPRV.2012.17. 11

[108] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, IPSN '14, pages 249–260, Piscataway, NJ, USA, 2014. IEEE Press. ISBN 978-1-4799-3146-0. URL http://dl.acm.org/citation.cfm?id=2602339.2602367. 81

[109] N. Rajagopal, P. Lazik, N. Pereira, S. Chayapathy, B. Sinopoli, and A. Rowe. Enhancing indoor smartphone location acquisition using floor plans. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '18, pages 278–289, Piscataway, NJ, USA, 2018. IEEE Press. ISBN 978-1-5386-5298-5. doi: 10.1109/IPSN.2018.00056. URL https://doi.org/10.1109/IPSN.2018.00056. 78, 107

[110] T. Rajapaksha, X. Qiu, E. Cheng, and I. Burnett. Geometrical room geometry estimation from room impulse responses. In *In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016. 10, 24, 79

[111] L. Remaggi, P. J. B. Jackson, P. Coleman, and W. Wang. Room boundary estimation from acoustic room impulse responses. In *2014 Sensor Signal*

Processing for Defence (SSPD), pages 1–5, Sept 2014. doi: 10.1109/SSPD.2014.6943328. 24

[112] L. Remaggi, P. J. B. Jackson, W. Wang, and J. A. Chambers. A 3d model for room boundary estimation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 514–518, April 2015. 10, 24, 79

[113] L. Remaggi, P. J. B. Jackson, P. Coleman, and W. Wang. Acoustic reflector localization: Novel image source reversion and direct localization methods. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(2):296–309, Feb 2017. ISSN 2329-9290. doi: 10.1109/TASLP.2016.2633802. 10, 24

[114] W. C. Sabine. Collected papers on acoustics. *Harvard University Press*, 1923. 16, 31, 41, 49, 68, 79

[115] H. Sak, A. W. Senior, K. Rao, and F. Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *CoRR*, abs/1507.06947, 2015. URL `http://arxiv.org/abs/1507.06947`. 13

[116] S. Schmidt. Evidence for a spectral basis of texture perception in bat sonar. *Nature*, 331:617 EP –, Feb 1988. URL `http://dx.doi.org/10.1038/331617a0`. 34

[117] I. J. Schoenberg. Remarks to maurice frechet's article "sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de hilbert. *Annals of Mathematics*, 36(3):724–732, 1935. ISSN 0003486X. URL `http://www.jstor.org/stable/1968654`. 93

[118] M. R. Schroeder. Computer models for concert hall acoustics. *American Journal of Physics*, 41(4):461–471, 1973. 17

[119] O. Shih and A. Rowe. Occupancy estimation using ultrasonic chirps. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical*

Systems, ICCPS '15, pages 149–158, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3455-6. doi: 10.1145/2735960.2735969. URL http://doi.acm.org/10.1145/2735960.2735969. 84

[120] O. Shih, P. Lazik, and A. Rowe. Aures: A wide-band ultrasonic occupancy sensing platform. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, BuildSys '16, pages 157–166, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4264-3. doi: 10.1145/2993422.2993580. URL http://doi.acm.org/10.1145/2993422.2993580. 30

[121] A. So and Y. Ye. A semidefinite programming approach to tensegrity theory and realizability of graphs. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 766–775, 01 2006. 127

[122] P. Somervuo, B. Y. Chen, and Q. Zhu. Feature transformations and combinations for improving asr performance. In *INTERSPEECH*, 2003. 12

[123] M. Spirito et al. On the accuracy of cellular mobile station location estimation. *Vehicular Technology, IEEE Transactions on*, 50(3):674–685, 2001. 91

[124] R. Takashima, T. Takiguchi, and Y. Ariki. Hmm-based separation of acoustic transfer function for single-channel sound source localization. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2830–2833, 2010. 12

[125] B. Tan, J. Zhang, and L. Wang. Semi-supervised elastic net for pedestrian counting. *Pattern Recognition*, 44(10):2297 – 2304, 2011. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2010.10.002. URL http://www.sciencedirect.com/science/article/pii/S0031320310004863. Semi-Supervised Learning for Visual Content Analysis and Understanding. 21

[126] S. P. Tarzia, R. P. Dick, P. A. Dinda, and G. Memik. Sonar-based measurement of user presence and attention. *Ubicomp*, 2009. 17, 18

[127] S. Tervo and T. Tossavainen. 3d room geometry estimation from measured impulse responses. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2012. 10, 24, 79

[128] H. Trang, T. H. Loc, and H. B. H. Nam. Proposed combination of pca and mfcc feature extraction in speech recognition system. In *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, pages 697–702, Oct 2014. doi: 10.1109/ATC.2014.7043477. 12

[129] Z. Tüske, P. Golik, R. Schlüter, and H. Ney. Acoustic modeling with deep neural networks using raw time signal for lvcsr, 09 2014. 13

[130] D. von Helversen and O. von Helversen. Object recognition by echolocation: a nectar-feeding bat exploiting the flowers of a rain forest vine. *Journal of Comparative Physiology* A, 189(5):327–336, May 2003. ISSN 1432-1351. doi: 10.1007/s00359-003-0405-3. URL `https://doi.org/10.1007/s00359-003-0405-3`. 34

[131] W. Wang, C. Chen, W. Chen, P. Rai, and L. Carin. Deep metric learning with data summarization. In P. Frasconi, N. Landwehr, G. Manco, and J. Vreeken, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 777–794, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46128-1. 14

[132] D. B. Yang, H. H. Gonzalez-Banos, and L. J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *International Conference on Computer Vision*, 2003. 17, 20, 72

[133] L. Yang, K. Ting, and M. Srivastava. Inferring occupancy from opportunistically available sensor data. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 60–68, March 2014. doi: 10.1109/PerCom.2014.6813945. 17, 18

[134] T. Yokota and T. Hashida. Hand gesture and on-body touch recognition by active acoustic sensing throughout the human body. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16 Adjunct, pages 113–115, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4531-6. doi: 10.1145/2984751.2985721. URL http://doi.acm.org/10.1145/2984751.2985721. 11

[135] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann. Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition. *IEEE Signal Processing Magazine*, 29:114–126, 2012. 76

[136] G. Young and A. S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938. ISSN 1860-0980. doi: 10.1007/BF02287916. URL http://dx.doi.org/10.1007/BF02287916. 93

[137] S. Zelditch. Inverse spectral problem for analytic plane domains II: $\mathbb{Z}_2$-symmetric domains. *ArXiv Mathematics e-prints*, Nov. 2001. 75

[138] J. Zhang, R. G. Lutes, G. Liu, and M. R. Brambley. Energy savings for occupancy-based control (obc) of variable-air-volume (vav) systems. 1 2013. doi: 10.2172/1063080. 29

[139] Z. Zhang, P. Pouliquen, A. Waxman, and A. G. Andreou. Acoustic micro-doppler gait signatures of humans and animals. In *2007 41st Annual Conference on*

Information Sciences and Systems, pages 627–630, March 2007. doi: 10.1109/CISS.2007.4298383. 11

[140] X. Zhao, E. Delleandrea, and L. Chen. A people counting system based on face detection and tracking in a video. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 67–72, Sept 2009. doi: 10.1109/AVSS.2009.45. 17, 20

[141] K. Zhong, R. Guo, S. Kumar, B. Yan, D. Simcha, and I. Dhillon. Fast Classification with Binary Prototypes. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1255–1263, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. URL `http://proceedings.mlr.press/v54/zhong17a.html`. 14

[142] B. Zhou, M. Elbadry, R. Gao, and F. Ye. Batmapper: Acoustic sensing based indoor floor plan construction using smartphones. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '17, pages 42–55, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4928-4. doi: 10.1145/3081333.3081363. URL `http://doi.acm.org/10.1145/3081333.3081363`. 27, 113

[143] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. URL `http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf`. 15

[144] L. Zimmermann, R. Weigel, and G. Fischer. Fusion of non-intrusive environmental sensors for occupancy detection in smart homes. *IEEE Internet of Things Journal*, pages 1–1, 2017. doi: 10.1109/JIOT.2017.2752134. 17, 21

[145] D. N. Zotkin, R. Duraiswami, and L. S. Davis. Rendering localized spatial audio in a virtual auditory space. *IEEE Transactions on Multimedia*, 6(4):553–564, Aug 2004. ISSN 1520-9210. 76