The Dissertation Committee for Wei Sun
certifies that this is the approved version of the following dissertation:

From Active to Passive Spatial Acoustic Sensing and Applications

Committee:

Lili Qiu, Supervisor

Aloysius K. Mok

David Harwath

Sangki Yun

From Active to Passive Spatial Acoustic Sensing and Applications

by

Wei Sun

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2022

Dedicated to my wife Qianling.

Acknowledgments

This dissertation is accomplished with the support and guidance of many incredible mentors, colleagues, friends, and family.

First and foremost, I would like to thank my advisor Lili Qiu for the guidance and support during my research career. She showed the passion for research and dedication to work, which inspired my research a lot. She guided me to crystallize ideas into solid and fancy research works. Her spirit will continue to enlighten me on many aspects for the rest of my life

During my PhD career, I also want to thank many great people guiding my research and work. Aloysius Mok, David Harwath, and Sangki Yun have provided many helpful suggestions to improve the dissertation as my committee members. Yunxin Liu was my first external mentor during my internship at Microsoft Research in Asia in 2019 summer. I learned broad topics on deep learning methods and collaborative learning systems, which are fundamental for my further research. I was grateful to intern with Minghua Xu during VISA Research. He advised me to develop novel systems and collaborate with researchers and engineers from diverse teams. I also thank Luwei Cheng for being a wonderful mentor during my internship at Facebook. I am grateful to be supervised by Kristen Grauman and David Harwath on computer vision and speech processing in a collaboration project.

I feel genuinely lucky to have had many awesome collaborators at different stages over the past 5 years. Wenguang Mao, Swadhin Pradhan, Yi-Chao Chen, and Mei Wang taught me significant techniques and experience in wireless sensing and signal processing. Changan Chen led me to a new perspective on audio-visual learning and shared lots of deep learning techniques. The interdisciplinary knowledge and ideas broadened my views on the research areas and inspired me to find ignored but important research areas.

I feel very fortunate to have many awesome labmates to support each other. Apart from the people I have mentioned above, Jian He, Mubashir Qureshi, Ghufran Baig, Zhaoyuan He, Wangyang Li, Changhan Ge, Chenxi Yang, Shuozhe Li, Muhammad Muaz, Mikyung Han, Kyoungjun Park, Aashish Gottipati have had lots of support and joy time together with me. There are many friends in UT having a great time at different stages of my PhD: Zhipeng Jia, Wei Shi, Xingyi Zhou, Jiacheng Zhuo, Jiacheng Xu, Jifan Chen, Hao Yao, Wei Li, Yan Han, Fengyu Deng, Qifan Gu, Weiyu Zhu, Zhigang Wei, Wenbo Zhang and many others.

I have been helped by many other professors and staff members at UT as well, apart from the ones mentioned above. I want to thank Shyamal Mitra, Raymond Mooney, Simon Peter, Sarah Abraham and Todd E. Humphreys for their guidance in my PhD career. I want to thank Katie Traughber for her significant help as our graduate program coordinator.

Finally, I am so blessed to have had my wife Qianling Ye during these years. She gave me the most love to support me to achieve my goal. My

wife and my Parents, parents-in-law, uncles are always the backbone of my life from the past to the future.

From Active to Passive Spatial Acoustic Sensing and Applications

Publication No. _____

Wei Sun, Ph.D.
The University of Texas at Austin, 2022

Supervisor: Lili Qiu

The active acoustic sensing system emits modulated acoustic waves and analyzes reflection signals. It is dominant in acoustic spatial sensing. On the other side, the passive acoustic sensing system receives and investigates nature sounds directly. It is good at semantic tasks but has weak performance on spatial sensing. In this dissertation, we manage to bridge three gaps in existing systems. They are the gap between the assumption of signal processing algorithms and the real acoustic environment, the gap between powerful active spatial sensing and limited passive spatial sensing, and the gap between the semantic features and spatial information. We evolve the acoustic sensing system design and extend the functionalities by three novel systems.

First, we develop a fully active spatial sensing system *DeepRange* which can adapt to the real environment easily. We develop an effective mechanism to generate synthetic training data that captures noise, speaker/mic distortion,

and interference in the signals. It removes the need of collecting a large volume of data. We then design a deep range neural network (DRNet) to estimate the distance from raw acoustic signals. It is inspired by signal processing that an ultra-long convolution kernel size helps to combat noise and interference. The model is fully trained over synthetic data, but it can achieve sub-centimeter error robustly in real data despite various environments, background noise, interference, and mobile phone models.

Second, we develop a fused active and passive spatial sensing system for speech separation noted as Spatial Aware Multi-task learning-based Separation (*SAMS*). We leverage both active sensing and passive sensing to improve AoA estimation and jointly optimize the semantic task and the spatial task. *SAMS* estimates the spatial location and extracts speech for the target user during teleconferencing simultaneously. We first generate fine-grained spatial embeddings from the user's voice and inaudible tracking sound, which contains the user's position and rich multipath information. Furthermore, we develop a deep neural network with multi-task learning to jointly optimize source separation and location. We significantly speed up inference to provide a real-time guarantee.

Finally, we deeply fuse the semantic features and spatial cues to combat the interference and noise in the real environment as well as enable depth sensing in a fully passive setup. Inspired by the "flash-to-bang" phenomenon (*i.e.*hearing the thunder after seeing the lightning), we propose *FBDepth* to measure the depth of the sound source. We formulate the problem as an audio-

visual event localization task for collision events. Specifically, FBDepth first aligns correspondence between the video track and audio track to locate the target object and target sound in a coarse granularity. Based on the observation of moving objects' trajectories, it proposes to estimate the intersection of optical flow before and after the collision to locate video events in time. It feeds the estimated timestamp of the video event and the other modalities for the final depth estimation. We use a mobile phone to collect the 3.6K+ video clips involving 24 different objects at up to $60m$. FBDepth shows superior performance especially at a long range compared to monocular and stereo methods.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

There are two types of acoustic sensing systems. The active sensing system emits modulated acoustic waves and analyzes reflection signals. It studies acoustic wave propagation in space and detects the correlation between the reflection signal and reference signal. It has timestamps for transmitting and receiving signals. Therefore, it can be used for spatial applications such as acoustic tracking and acoustic imaging. On the other side, the passive sensing system receives and investigates nature sounds directly. It needs to deal with arbitrary sounds generated by specific objects and motions. It is commonly used for semantic tasks such as sound event classification and speech recognition. The lack of reference signals and transmitting timestamps makes it challenging to invest spatial location in the passive sensing setup. The traditional design of the acoustic sensing system can only leverage either spatial cues or semantic information, which restricts applicable scenarios.

In this dissertation, we evolve the design of acoustic sensing systems to boost the performance of typical applications with acoustic spatial cues and well-designed neural networks.

1.1 Motivation

Sensing is the key to interacting with the environment and understanding the world. It applies various mediums such as electromagnetic waves, mechanical waves, and smells to transmit information about the unknown object. Sensing can be classified as active sensing and passive sensing.

Active sensing emits self-generated energy and observes the reflection from the environment. An example of human perception is to tough the object to feed the hardness. Active sensing is dominant in the wireless system. It enables significant wireless applications such as Global Positioning System (GPS) by radar and points cloud generation by lidar. Modulated signals are transmitted to sense the spatial information that is further used for diverse spatial applications.

Acoustic active sensing has been attractive to researchers from the wireless community in recent years. They explore the physical properties of sound waves and focus on the characteristics of propagation in space to enable spatial sensing. Wireless researchers adjust signal processing methods for radio frequency technology to fit the new scenarios under the assumptions of sound waves. Many breakthrough results have been achieved in acoustic ranging, acoustic localization, acoustic tracking, and acoustic imaging with only acoustic signals. However, these methods are still vulnerable in a comprehensive environment with background noise, multipath interference, and hardware distortion.

Compared to other wireless mediums, the most attractive feature of acoustic is the low cost of deployment and power consumption. A set of a speaker and a microphone is widely equipped on various smart devices, such as mobile phones, smart speakers, unmanned aerial vehicles, smart furniture, smart glasses, and CCTV cameras. Besides, the velocity of acoustic is slow so the wavelength is quite small at the inaudible band(*i.e.*more than 18khz). A small wavelength is critical to an accurate estimation. Furthermore, the microphone array has been a trend to be equipped on advanced devices to enable spatial functionalities. Table 1.1 lists emerging novel devices equipped with microphone arrays in recent years. Developers are pushing more effort to enable spatial computing with novel microphone arrays and algorithms. This trend enables more interest to fully take advantage of the rich spatial information from acoustic waves.

Device	Category	Microphone Channels	Array Shape
Amazon Echo Studio	Smart Speaker	7	Circular
Apple MacBook Pro 16	Laptop	3	Triangular
Huawei MateBook 14s	Laptop	4	Linear
Azure Kinect DK[182]	Smart Sensor	7	Circular
Meta VR Glass[34]	Smart Glass	6	Head-mounted

Table 1.1: Novel devices equip diverse microphone arrays for spatial computing, far-field speech capture, speech separation, etc.

Passive sensing collects the received information from the environment directly. Human vision is a typical example to sense the world passively. Pas-

sive sensing is often applied to understand semantic representation, such as object detection and segmentation by visible light and temperature measurement from infrared light. It can also be used for sensing spatial information such as depth estimation on monocular images or stereo images, though the performance is worse than active sensing.

Acoustic passive sensing has been well studied by the speech community. They pay attention to semantic features of the sound generated in nature, such as human voice, impact sounds, urban sound, and music. They have developed excellent models and systems for automatic speech recognition, speech separation, audio classification, etc. Acoustic passive spatial sensing is investigated in two fashions as well. First researchers apply the microphone array to analyze the phase difference across multiple channels and derive a fair Angle of Arrival(AoA) of the sound source. Besides, they investigate room impulse response and direct-to-reverberant ratio to estimate the room size and the position of the sound source[21]. However, both cannot propose a fundamental formulation to map the distance to the acoustic features directly. Hence, the performance of acoustic passive spatial sensing is low and applicable scenarios are limited.

In all, there are several gaps in current acoustic sensing systems. The first gap is between the assumptions of signal processing methods and unexpected interference and noise, which degrades the performance in all metrics. The second gap is the different capabilities of spatial sensing between active methods and passive methods. The third gap comes from the individual re-

search on semantic features or spatial cues of the sound. There are few works to leverage both characteristics of the sound. In this dissertation, we manage to bridge these gaps progressively to enable novel acoustic sensing systems with superior performance and robustness.

1.2 Challenges

Several fundamental challenges lie in acoustic spatial sensing and applications. First, although many acoustic sensing algorithms have been developed, the performance degrades significantly under strong noise, interference, and hardware limitations. Many existing approaches exploit various signal processing techniques, but their assumptions can be broken up under low signal-to-noise ratio(SNR) and multipath propagation. For example, Frequency-modulated continuous wave(FMCW) based acoustic ranging algorithm can estimate the distance accurately in an ideal environment. However, low SNR can submerge the target estimation with many noisy estimations. Multipath interference causes a shift of the estimation to the nearby objects. The complexity of real environments makes it challenging to design a robust acoustic sensing system.

Second, passive sensing naturally lacks access to the reference of clean sound and the reference emitting timestamp, which are critical to active sensing. For example, active sensing can emit a reference FMCW and record the local clock. Then it compares reflection FMCW with the reference FMCW and records the receiving clock, which enables to estimate ToF and AoA. But

passive sensing always receives the target sound together with interference and noise. Meanwhile, it has no idea of when the sound is produced. It is more challenging to analyze the spatial location from the noisy mixture sound without the reference signal and the reference timestamp.

Third, the larger wavelength of natural sound in passive sensing results in lower resolution compared to high-frequency signals used in active sensing approaches. For example, the frequency of voice is less than 4kHz commonly. The wavelength is more than 8 cm. Active sensing uses signals whose frequencies are higher than 18 kHz. The wavelength is less than 2 cm. Larger wavelength results in less phase difference among channels in the same microphone array. It is challenging to leverage phase information to estimate AoA accurately.

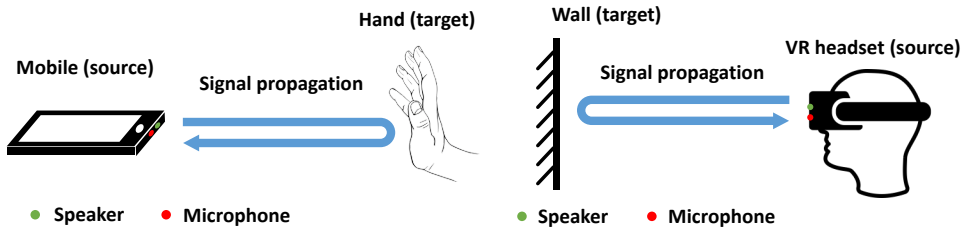
1.3 Proposed Techniques and Applications

1.3.1 Fully Active Sensing for Acoustic Ranging

Motivation: We revisit the fundamental building block of spatial sensing in the active sensing setup, ranging. We manage to figure out the insight on the outstanding performance of acoustic active sensing. Ranging is the technique to estimate the distance from a signal source to a target object. It is fundamental for localization, motion tracking, gesture, and activity recognition, which have a wide variety of applications. For example, it enables gesture-based interfaces to remotely control smart appliances, virtual reality (VR), augmented reality (AR), and gaming. It offers an effective way of sensing environments

for wireless optimization (*e.g.*, beamforming, AP selection), habitat monitoring, disaster recovery, user tracking, health monitoring, and context-aware applications.

We consider emitting the inaudible signal actively and ranging with the reflected signals. As shown in Figure 1.1, the source sends and receives signals to measure the round trip propagation delay from the target (*e.g.*, a user's hand) and compute the distance.



(a) Motion tracking or gesture recognition (b) Indoor mapping or obstacle detection

Figure 1.1: Applications for passive acoustic ranging.

Active ranging is useful for many applications, such as gesture recognition, indoor mapping, virtual reality (VR), and augmented reality (AR). For example, we can use ranging techniques to create a map for indoor environments by measuring the distance to walls and furniture. We can also estimate the distance to nearby obstacles for safety applications.

A number of ranging approaches have been developed, based on different signals, such as sound [176, 85, 153, 94, 177, 134, 148], radio frequency (RF) [58, 145, 123, 166, 62, 111, 83, 81], and infrared lights [54]. Compared to other signals, acoustic ranging has the following advantages. First, the slow

propagation speed of acoustic signals is beneficial to achieve high accuracy in distance estimation. Second, most devices are already equipped with speakers and microphones, and we do not need to deploy extra hardware (*e.g.*, RFID readers and tags, millimeter wave antennas, and depth cameras) for ranging. Third, the sampling rate of acoustic signals is low so that the processing can be done in software. This makes acoustic ranging easily available on commodity devices (*e.g.*, smartphones and VR/AR headsets).

Existing ranging algorithms are designed by domain experts. In this paper, we explore the following interesting questions: Can a deep neural network automatically learn the features in the received acoustic signals to estimate the distance? Can it outperform traditional signal processing algorithms designed by domain experts?

This direction is interesting for several significant reasons. Scientifically, it is interesting to understand the feasibility of the machine learning approach in automatically learning features. This learning task seems challenging since unlike images or videos, where humans can easily determine the correct answers (*e.g.*, image label), human is not good at estimating the distance from acoustic signals. Practically, if successful, the resulting approach can improve the accuracy of ranging and benefit a wide range of applications. Moreover, it can also shed light on the limitations and potential of signal processing versus machine learning approaches. Such insights will help us design new algorithms that achieve the best of both worlds.

Approach: Our work is inspired by the tremendous success of deep neural

network (DNN) and its advantage in nonlinear problems. Motivated by its success in vision and speech recognition communities, we have seen applications of neural networks to ranging and tracking. For example, RF-Echo [29] applies a neural network with a single hidden layer to estimate the propagation delay of RF signals based on the correlation profile. RF-Pose [186, 187] develops a convolutional neural network (CNN) to estimate a user's pose based on the heatmap generated by applying FMCW to RF signals. Different from these works, which use the features designed by domain experts as the neural network input, we aim to automatically learn the features from raw acoustic signals. Moreover, unlike the existing works, which require training data from real testbeds and can be time-consuming and labor intensive, we aim to automatically generate the training data.

In order to apply DNN to the received signals, we need to address two major challenges. First, DNN requires a large volume of training data to work well, and it is important to have an efficient way of generating lots of training data. Second, we need to design a DNN that works well for distance estimation.

To address the first challenge, we develop a simulator that models how acoustic signals propagate through the environment. Through extensive trials, we find that not only noise and multipath (*i.e.*, reflections from objects other than the target) affect the learner performance, but also self interference (i.e., the signal directly going from the transmitter to the receiver) and speaker/microphone distortion have a significant impact. Therefore, our simu-

lator captures all these factors. To derive a general model, we add randomness when generating signals to prevent the network from overfitting specific values or patterns. For example, not only the noise in our synthetic signals is a random Gaussian variable, but its standard deviation is also randomly chosen. Similarly, we randomly synthesize a piece-wise polynomial function to capture the frequency responses of speakers and microphones. In this way, our simulator achieves simplicity, generality, and realism. We only use the data generated from our simulator to train DNNs for ranging and show that they work well for real signals using 11 different smartphones, 10 users, different targets, and 4 different locations.

To address the second challenge, we start with a generic multi-layer fully connected neural network. Interestingly, we find the weights connected to each neuron in the first layer have a high correlation with the transmission signals at different shifts. This insight motivates us to develop a CNN. We first try the traditional CNNs, such as AlexNet [65] and VGG [126], but find they do not work well because their filters have too short kernel size (*e.g.*, 3×3 or 11×11) and fail to detect local patterns under noise and interference. To robustly capture signal patterns, we develop a CNN with filter sizes comparable to the length of transmission signals.

Moreover, we find the network weights converge to different values during different runs. Intuitively, these weights correspond to different ways of feature extraction for ranging. Therefore, instead of training a single network, we train a set of networks and combine them using an ensemble model. To

maximize the effectiveness of ensemble learning, we add randomness during training by using random initialization, applying dropout, and using different sets of training data.

We evaluate our approach by training DRNet using synthetic data generated from our simulator and testing on the acoustic signals collected from 11 phones with different brands, 10 users, different targets, and 4 environments including a lab, a conference room, a corridor, and a cubic area. The evaluation results show that our network generalizes well to different scenarios. Compared to three baseline approaches that use FMCW, correlation, and phases, our learning-based approach achieves up to 5 times improvement on the ranging accuracy when the SNR is low and the interference is severe.

1.3.2 Fused Active and Passive Sensing for Speech Separation

Motivation: In this project, we further explore how to use active methods to improve the AoA estimation of passive sensing and how to leverage the semantic features and spatial cues together to improve semantic task and spatial sensing simultaneously. We choose the indispensable online meetings as the applicable scenarios. During Covid-19, the online meeting is the only means to connect with many people. Kids depend on it for education, adults rely on it for work, and friends count on it for socialization. Moreover, its importance will likely go well beyond Covid-19 as many companies will continue to allow work-from-home, and online classes will likely be the future trend due to convenience and broad reach.

Over 70% US households have two or more people [129] and the average household size across the world is 4 [110]. While one is participating in an online meeting or taking an online class, other house members may generate sound. Unlike videos, voice signals can travel across rooms and result in significant interference unintentionally. After all, not everyone has enough room at home to create a separate home office space. This both degrades the audio quality and raises serious privacy issues.

There has been significant work on signal source separation. Earlier works use signal processing, such as PCA and ICA. More recent works use machine learning (ML) to further improve separation accuracy. Videos can also be used to improve source separation [118, 36] since the camera captures mouth position and movement. However, video requires good lighting conditions, has a limited field of view, and raises significant privacy concerns. [2] also reports turning off a camera during teleconferencing reduces the environmental footprint of a meeting by 96%.

Despite considerable work, existing works primarily focus on using raw audio samples for source separation. User location can have a significant impact on source separation but has not been explicitly considered until recently. Some recent works use the ground truth location for source separation and report significant benefits (*e.g.*, [24, 183, 189]). But they rely on external modality to provide the location. The speech itself contains the spatial cues to extract by signal processing. Besides, the specific multipath profile provides more unique spatial information.

There is a strong inter-dependency between the source separation and the AoA estimation. Considering the traditional beamforming algorithm, the accurate AoA can yield source separation of high quality spatially. On the other side, the reference signal can be applied to optimize the search of AoA from the mixed signals. In the blind speech separation task, both AoAs and target speech are unknown. Therefore, we propose to learn the speech separation and AoA estimation jointly from the received acoustic signals.

Approach: To preserve privacy and achieve efficiency, we seek an audio-only solution to explicitly estimate the user's spatial information including position and multipath profiles for source separation. Since the spatial information contains non-negligible errors, we need to explicitly consider the localization error in source separation.

Consider a user uses a computer to join an online meeting while other people are talking in the background. Since the user is close to the computer during the meeting, a small movement can result in a large difference in the Angle of Arrival (AoA). Our measurements in Figure 1.2 show that there is a frequent head movement that yields over a 20-degree change in the AoA and a 20 cm change in the distance when a user speaks spontaneously. Therefore it is useful to track the user's location and use it for source separation.

As shown in Figure 1.3, we develop a system that automatically removes the interference by explicitly estimating the user location and multipath, and using the estimated spatial embeddings to enhance source separation accuracy.

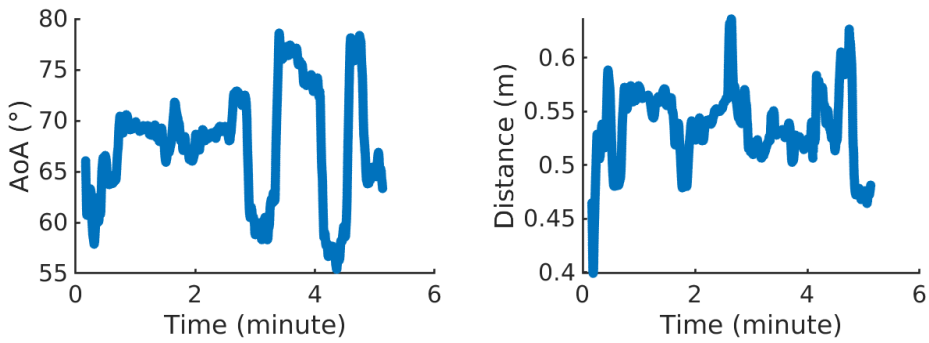


Figure 1.2: Speaker movement during spontaneous speech. We use a depth camera to collect the movements.

While there has been existing work on localizing a user using inaudible signals (*e.g.*, [85, 93]) or audible signals (*e.g.*, [124, 152]), our work is the first that leverages both audible and inaudible signals to achieve high localization accuracy. More specifically, we extract masks in the Time-Frequency (TF) domain from audible signals and use the mask to select TF bins dominated by the target user's voice to improve the localization accuracy of the voice signal under interference. Meanwhile, we let the computer generate inaudible acoustic chirps to track user position. We feed location profiles from both audible and inaudible signals to the 3D convolutional layer to generate spatial embeddings, which will serve as the input to source separation. These spatial embeddings contain not only the user's location but also rich multipath information and play an important role in source separation.

We then leverage the user's location to improve the source separation by developing a novel multi-task learning framework to jointly learn the source separation and location. Instead of treating the estimated position as an input

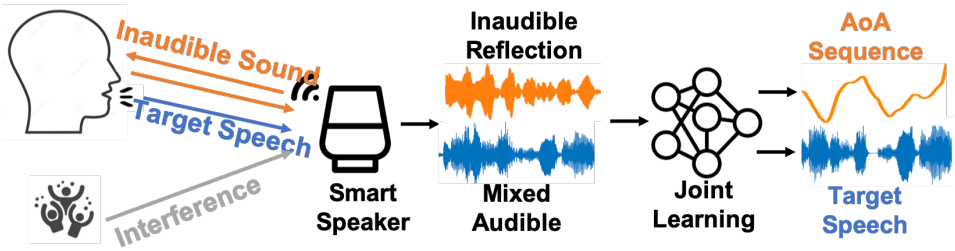


Figure 1.3: SAMS automatically removes acoustic interference and ambient noise for online meetings.

feature to the separator, we explicitly take into account of the position estimation error in the multi-task loss function and guide the network to jointly learn the separation and localization by establishing the consistency between the target user speech and position.

In order to achieve real-time processing during an online meeting, we make the following enhancements: (i) leveraging casual convolution (*i.e.*, using only past samples) and non-casual convolution (*i.e.*, using future samples) with a small look-ahead window to reduce response time, (ii) cache previous intermediate results in the neural network and reuse whenever possible, and (iii) further optimize the computation graph with Microsoft Onnxruntime [9]. According to [115], 150 ms is a recommended one-way latency. SAMS processes audio every 90 ms within 42 ms on a laptop without GPU. So the total latency is 132 ms, well below 150 ms. We implement our approach on a laptop with dual speakers and a microphone array. We evaluate different users' performance in different environments with various SNR and interference sources. Our results show that SAMS achieves 10.71 - 13.61 dB Scale-invariant Signal-to-Noise Ratio (SiSNR) under a varying number of interfering users. This

is a 3.4-5.0 dB improvement over Conv-TasNet [80], and a 1.4-9.58 dB improvement over PHASEN [174]. The multi-task learning with novel spatial embedding fusion also helps SAMS to achieve a 1.2-3.18 dB improvement over other multichannel separation strategies based on MVDR pre-filtering or AoA-based pre-mask.

1.3.3 Fully Passive Sensing for Sound Source Depth Estimation

Motivation: We manage to find the explicit fundamental formulation to map the depth to explicit physical variables in a pair of fully passive sensing sensors(*i.e.*one camera and one microphone).

Depth estimation has been a popular topic owing to many important applications. It is the fundamental functionality to enable 3D perception and manipulation. Although there have been significant efforts on developing depth estimation methods with various sensors. current depth estimation schemes fail to achieve a good balance on multiple basic metrics including accuracy, range, angular resolution, cost, and power consumption.

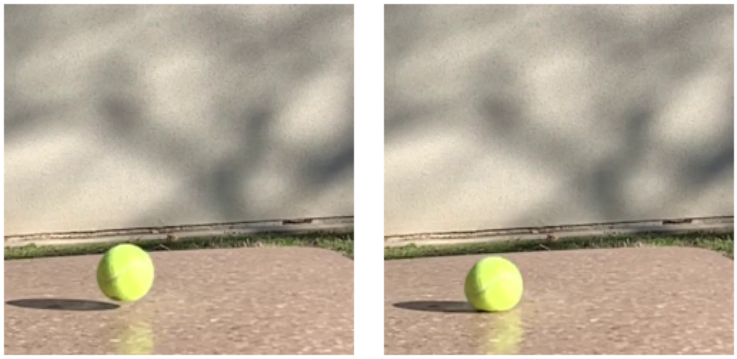
Active depth sensing methods actively emit signals, such as LiDAR [18], structured-light [182], mmWave [12], ultrasound [85], WiFi [146]. They compare the reflected signal with the reference signal to derive time-of-flight (ToF), phase change, or Doppler shift to estimate the depth. Active methods can achieve high accuracy because the modulated sensing signal is well designed. Lidar is the most attractive active senor due to its large sensing range and dense point cloud. However, the density is not sufficient enough to enable a

small angular resolution. Therefore, the points are too sparse to be recognized at a long distance. Besides, the prohibitive cost and power consumption limit the availability of Lidar on general sensing devices.

Passive depth sensing takes signals from the environment for sensing directly. It commonly uses RGB monocular camera [15, 66], stereo camera [130, 25], thermal camera [76], or multi-view cameras [72]. These sensors can achieve pixel-wise angular resolution and consume pretty less energy due to omitting the signal emitting. Among them, stereo matching can effectively estimate the disparity and infer a dense depth map. The baseline of the stereo camera determines the effective range and accuracy. Therefore, the dimension of the stereo camera is placed as the critical trade-off with sensing metrics. Thanks to the advance in deep learning, the cheap monocular depth estimation keeps on improving performance with new network structures and high-quality datasets. However, the accuracy is still not satisfactory especially at a long range because it can only regress depth based on the implicitly visual cues. Besides, it is highly fitting to the training dataset so it is pretty challenging to adapt the out-of-domain scenarios and calibrate the unseen camera intrinsics [68].

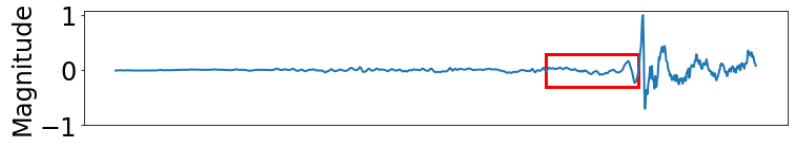
Approach: We propose to add only one microphone to enable explicit physical depth measurement and boost the performance of a single RGB camera. It does not rely on the camera's intrinsic and implicit visual cues. We develop a novel passive depth estimation scheme, called Flash-to-Bang Depth (FBDepth). It is inspired by a well-known phenomenon – 'Flash-to-Bang',

which is used to estimate the distance to the lightning strike according to the difference between the arrival time of a lightning flash and a thunder crack. This works because light travels a million times faster than sound. When the sound source is several miles away, the delay is large enough to be perceptible. Applying it to our context, FBDepth can estimate the depth of a collision that triggers audio-visual events. The collision event has been explored for navigation and physical search in [39], but our work is the first that uses the collision for depth estimation. Collisions are common and can arise when a ball bounces on the ground, a person takes a step, or a musician hits a drum. We identify and exploit several unique properties related to various collisions in the wild. First, the duration of a collision is short and collision events are sparse. Thus, there are few overlapped collisions. Second, though the motion of objects changes dramatically after the collision, they are almost static at the collision moment. Third, the impact sound is loud enough to propagate to a long range.

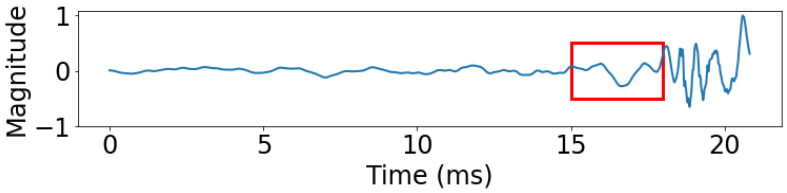


(a) frame before a collision (b) frame after a collision

Figure 1.4: Consecutive frames in the 240 FPS video. It does not capture the moment to start to touch the surface without the deformation.



(a) Impact sound of a wooden brick



(b) Impact sound of a tennis ball

Figure 1.5: Two different impact sounds. It is impossible to distinguish which sample is the start sample of the impact sound.

While the Flash-to-Bang phenomenon is intuitive, using it for accurate depth estimation poses several significant challenges: (i) It is challenging to get the ground truth collision time from both video and audio. Video only offers up to 240 frames per second, and may not capture the exact instance when the collision occurs as shown in Figure 1.4. Audio has a higher sampling rate but it is hard to detect the start of a collision solely based on the collision sound due to different sound patterns arising from collisions as well as ambient noise. In Figure 1.5, we can estimate a coarse duration of impact sounds but we have no idea which sample is the first impact sample even though the noise is low. (ii) We need highly accurate collision time. 1 ms error can result in a depth error of 34 cm. (iii) Noise present in both audio and video further exacerbate the problem.

To realize our idea, we formulate the sound source depth estimation as the audio-visual localization task. Whereas existing work [164, 165] still fo-

cuses on 1-second-segment level localization. FBdepth performs event-level localization by aligning correspondence between the audio and the video. Apart from audio-visual semantic features as input in existing work [141, 20], we incorporate optical flow to exclude static objects with similar visual appearances. Furthermore, FBDepth applies the impulse change of optical flow to locate collision moments at the frame level. Finally, we formulate the ms-level estimation as an optimization problem of video interpolations. FBDepth succeeds to interpolate the best collision moment by maximizing the intersection between extrapolations of before-collision and after-collision flows.

With the estimated timestamp of visual collision, we regress the sound source depth with the audio clip and visual features. FBdepth avoids the requirement to know the timestamp of audio collision. Besides, different objects have subtle differences in audio-visual temporal alignment. For example, a rigid body generates the sound peak once it touches another body. But an elastic body produces little sound during the initial collision and takes several ms to produce the peak with the maximum deformation. We feed semantic features to enable the network aware of the material, size, etc.

1.4 Summary of Contributions

The major contributions of this dissertation are summarized as follows:

- We design a novel deep learning framework to enhance the robustness and performance of the active acoustic ranging task. It includes a spe-

cial DRNet that takes the raw received acoustic signals without feature extraction as the input. The framework also includes a sensing data simulator that captures noise, speaker/mic distortion, and interference as well as generates diverse enough training data. The DRNet trained by the simulation data can significantly outperform the existing signal processing algorithms designed by domain experts on the real sensing data collected from a variety of scenarios. Furthermore, We analyze the DRNet structure and identify several important findings that can potentially help improve existing signal processing methods.

- We propose a first multi-task learning framework to fuse active acoustic sensing and passive acoustic sensing in the speech separation task. We show the inherent relationship between source location and separation. Thus, SAMS actively emits the inaudible signal to sense human motion. With a novel pre-mask learning network, SAMS can jointly learn source separation and location by leveraging the spatial embeddings from both audible speech and inaudible reflections. We implement our approach and significantly speed up the inference time to provide real-time guarantees. We demonstrate its significant performance benefits over existing approaches.

- We develop the first passive audio-visual depth estimation with one camera and one microphone. We bring the physical propagation property to audio-visual learning as well as fuse the semantic features and spatial

cues deeply. To formulate our novel depth estimation task, we introduce the ms-level audio-visual localization task. We propose a novel coarse-grained to fine-grained method to improve temporal localization resolution by leveraging the unique properties of collisions. We incorporate multiple modalities including RGB image, optical flow, and audio clip to predict the depth. Our evaluation using 3.6K+ audio-visual samples across 24 different objects in the wild shows that FBDepth achieves 0.64m absolute error (AbsErr) and 2.98% absolute relative error (AbsRel) across a wide range of scenarios from 2 m to 60 m. The benefit of FBDepth is especially high in long-range scenarios.

1.5 Dissertation Outline

We discuss related work in Chapter 2. We describe the active acoustic ranging system in Chapter 3; spatial aware based speech separation in Chapter 4; visual assisted sound source depth estimation system in Chapter 5

Chapter 2

Related Work

2.1 Motion Tracking

2.1.1 Acoustic Motion Tracking

A number of systems have been developed for motion tracking using acoustic signals. BeepBeep [108] measures the distance between two mobiles by correlating the pseudo-random sequence. Based on BeepBeep, SwordFight [181] further improves the efficiency and supports the interaction for mobile motion games. ApneaApp [93] uses FMCW to measure the chest and abdomen movements for apnea detection. AAMouse [176] relies on Doppler shifts of the signals to capture the distance changes over time. CAT [85] develops a distributed FMCW to estimate the distance between separated speakers and microphones. LLAP [153] leverages the phase changes of raw acoustic signals to determine the distance changes, while Strata [177] and VSkin [134] use the phases of channel taps. FingerIO [94] estimates the distance based on correlation and uses the properties of OFDM symbols to refine the estimation. DroneTrack [88] develops an approach to estimate the distance between a drone and a user based on MUSIC algorithm. MilliSonic [148] combines FMCW and phase measurements to estimate the distance. It achieves impressive estimation accuracy at high SNR, but does not work well under low SNR

because it relies on the assumption that the estimation error is always less than the wavelength (*e.g.*, about 2 cm for 17 KHz acoustic signals) to avoid phase ambiguity. These algorithms are developed by human experts. DeepRange complements the existing work by exploring the possibility of applying DNN to raw acoustic signals for distance estimation. The insights gained from DRNet can potentially help improve the existing signal processing methods.

2.1.2 RF Based Motion Tracking

Many motion tracking systems use radio frequency signals, such as WiFi, RFID, and millimeter-wave transceivers. ArrayTrack [167] estimates AoAs to different access points based on an array of antennas. RF-IDraw [150] uses the phase difference between a pair of RFID tags to estimate the incoming angles of the signals. Tagoram [173] estimates the locations of RFID tags by generating holograms with an array of RFID readers. TurboTrack [81] exploits the physical properties of RFID tags to emulate large bandwidth for accurate motion tracking. MTrack [158] measures the phases of 60 GHz waves for motion tracking with highly directional and steerable antennas. WiTrack [4, 5] leverages FMCW sweeping from 5.5 GHz to 7.2 GHz to estimate the positions of multiple people in the room. The above approaches require access to raw signals. When raw signals are not available, channel state information (CSI) is used to infer the position of a target. CUPID [123] and Splicer [166] derive the power delay profiles for the paths from the transmitter to the receiver by applying IFFT on CSI measurements. Widar [111] constructs the model be-

tween CSI and target motion and uses it for tracking. SpotFi [62] applies 2D MUSIC to jointly estimate the distance and the AoA of a target based on CSI measurements. Chronos [145] combines the CSI measurements at different bands to improve the tracking accuracy. WiDeo [58] determines the propagation delays of all paths by finding the best match between the CSI calculated according to these paths and measured values. WiDraw [135] uses CSI to estimate the AoAs of a target and relies on multiple WiFi APs for localization. The CSI-based approaches work on commodity devices (*e.g.*, WiFi APs and smartphones) and do not require any special hardware (*e.g.*, RFID tags and steerable antennas). However, these approaches only achieve decimeter-level tracking accuracy, which is insufficient for fine-grained tracking applications.

2.1.3 Neural Network Based Tracking

RF-Echo [29] applies a neural network with a single hidden layer to estimate the propagation delay of RF signals based on the correlation profile. RF-Pose [186] develops a convolutional neural network to estimate a user's poses based on heat maps generated by RF signals. RF-Pose3D [187] further extends above approaches to 3D pose estimation. Other works [77, 19] apply the recurrent neural network to determine a user's indoor location using the received signal strength of RF signals. RF-Finger [149] identifies the multi-touch gestures by applying CNN.

WordRecorder [35] extracts the spectrogram feature from acoustic signals and combines it with CNN for handwriting classification. A multi-LSTM

neural network is designed in [67] to fingerprint mobile device sensors, instead of using handcrafted features. RTrack [86] develops an RNN to automatically learn the mapping between the 2D profile and target position to exploit the temporal locality. All these works need features extracted from the received signals before applying neural networks, while our network directly takes the received signals as the input. Using the raw signals is beneficial to achieve better performance as demonstrated by our experiments. In addition, these approaches require collecting lots of data to train the networks, which can be time-consuming and labor-intensive.

Recently, there have been many CNN-based object detectors improving the accuracy of detecting sonar images. [161] first implements CNN over synthetic aperture sonar image and argument data by mirroring mugshots. [60] makes use of the efficient YOLO model on forward-looking sonar images for real-time detection. [190] extracts target features by AlexNet and classifies objects by applying SVM to side-scan sonar images. [154] further proposes an adaptive weights CNN to fuse the generated weights of the deep belief network and normalize the adaptive weights by local response normalization. They directly apply detectors for optical images to sonar images and ignore the inherent differentiation. [82] designs a Noise Adversarial Network as the sideway network to introduce perturbation with specific noise to sonar images during training to generalize the object detector in sonar images. These works apply deep learning to features extracted from acoustic signals by signal processing. In comparison, DeepRange directly feeds raw acoustic signals to DNN. This

is more challenging but can achieve higher gains since post-processing using existing signal processing methods may already reduce accuracy, which can be hard to recover at a later stage. Besides, DRNet does not require real data to train. It is trained only with well-designed synthetic data and generalizes well on diverse real data.

2.2 Speech Separation

2.2.1 Single-Channel Speech Separation

When the received signal only has one channel, researchers usually exploit the inherent vocal features and speech content to extract the independent components from mixture signals. Most approaches (*e.g.*, [116, 103, 139, 74, 170]) use the TF representation of the speech signal, computed by the short-time Fourier transform (STFT), also noted as a spectrogram. Then a mask matrix can be estimated for the target source and then multiplied with the mixture spectrogram to recover the clean spectrogram.

Early approaches tried to estimate the clean spectrogram by training a nonlinear model [74, 170] directly. Then embeddings [48, 23] are used to encode TF bins to higher dimensions for clustering. PHASEN [174] develops a two-branch learning framework to improve phase estimation. Meanwhile, new structures use encoder and decoder structures to perform the speech separation in the time domain [80, 104, 79, 22]. They also use LSTM or Temporal Convolution Network (TCN) to learn the context in the time domain. Dual-Path RNN (DPRNN) [79] further exploits extremely long sequences by splitting the

input sequence into multiple chunks that are processed with different RNNs by the local path and the global path. DPTNet [22] integrates a transformer into DPRNN. SepFormer [131] further integrates multi-head self-attention. These approaches have shown good performance on the WSJ0MIX synthetic dataset. However, the single-channel input is limited to exploiting crucial spatial information.

2.2.2 Multiple-Channel Speech Separation

Animals have developed multiple ears through millions of years of evolution. Similarly, more receiver channels can significantly improve source separation performance in theory and practice. Independent component analysis (ICA) [121] separates the mixture into additive subcomponents. However, it requires the source signals to be non-Gaussian and the number of sources to be smaller than the number of channels. Beamforming algorithms [107, 128, 49, 38] can leverage spatial information to strengthen the signal in the target directions and null the interference from unwanted directions. [24] proposes to iteratively run classic beamforming and separation for several rounds to guide the network focus on the appropriate direction. Recent work (*e.g.*, [44, 171]) uses a neural beamformer, which fuses the single channel spectrogram and internal phase difference to directly learn the target speech. [183, 172] develop end-to-end learning of complex covariance matrix to predict spatial complex filter. These works require the source position as another input for speech separation explicitly. The user position contains important infor-

mation for interference cancellation. It can be used for beamforming toward the target user. [32, 84] iterate localization and source separation using expectation maximization (EM). [57] formulates the separation and localization problem as the Bayesian inference problem. D-ASR [132] trains the localization network together with Automatic Speech Recognition (ASR) network to explicitly recognize the speech content of each speaker for source separation. CoS [56] integrates a binary search of the azimuth direction with separation. SAMS jointly learn the separation and location with a deep spatial fusion.

2.2.3 Multi-Modal Speech Separation

There has been a research interest to involve another coupled modality to help separate the source speech from the mixture. The modality is highly relevant to the source so that it can perform as a side channel to help recover the source speech. Ultrasound is proposed to use for source separation in [11]. It applies an ultrasound transceiver pair to capture frequency shifts caused by a talker's mouth movements and uses the semi-supervised nonnegative matrix factorization (NMF) for speech separation. UltraSE [133] implements this idea on mobile phones. It focuses on inaudible sound to the mouth to measure the Doppler shift of lip movement and applies contrastive learning to pair the ultrasound features and source speech features to enhance separation quality. In order to track lip movement using reflected inaudible sound, it requires the mouth-to-mic distance to be within 20 cm and a good facing angle towards the mic. [168, 71] use mmWave radar to sense vocal vibration which distorts the

skin-reflected mmWave signal. They build up the correlation model between source speech and mmWave reflection and design fusion network to perform speech separation and recognization. But mmWave deployment is less widely applicable than speakers and microphones.

Some recent works exploit audio and visual information together to enhance audio quality[37, 6, 40, 185]. Visual appearance can capture the important features of the speaker, such as gender, age, nationality, and body weight, which can impact the tones, pitch, and timbre of the voice signals. Moreover, it can track dynamic features of speech such as lip movement. [37] introduces a new audio-visual speech dataset AVSpeech and a multi-stream architecture to fuse visual features and audio features for separation. [185, 184] uses the natural synchronization of the visual and audio modalities. They can locate regions of interest in images that produce sounds and separate sound mixture into a set of components of each pixel from unlabeled videos. VisualVoice [40] explicitly leverages lip movements and the speaker's facial appearance simultaneously to isolate the corresponding speech. It jointly learns audio-visual speech separation and cross-modal speaker embeddings from unlabeled video.

2.3 Multi-Modality Depth Estimation

Recent work on depth estimation has shown the benefits of fusing cameras and other active sensors. [112, 53] recover dense depth maps from sparse Lidar point clouds and a single image. [73] associate pixels with pretty sparse radar points to achieve superior accuracy. The effective range can be increased

as well by Lidar-camera [180] or Radar-camera [178]. However, these methods are still expensive in cost and power consumption.

[41, 105] emit audio chirps and learn the depth map implicitly with audio reflections and a single image. However, these methods require many nearby acoustic reflectors to produce effective echos so the setup is limited in rooms. Besides, they are evaluated in an audio-visual simulator. FBDepth only uses one extra microphone to perceive natural sounds directly. It keeps the passive design of the audio but applies the physical measurement explicitly. The one-path sound propagation has a longer effective range than echoes.

2.4 Sound Source Localization

Previous systems localize sound sources with microphone arrays [144, 114] or one microphone with a camera [47]. They intend to estimate the direction of arrival(DOA) or the distance. The DOA is inferred by the subtle difference in arrival time from the sound source to each microphone[87, 136] or by semantic matching with the visual appearance if given images[141, 8]. The distance can be estimated by triangulation methods with multiple DOAs and room structures[151, 124]. It can be fully learning-based to predict depth with room acoustic features [21].

2.5 Audio-Visual Event Localization

It aims to detect and localize events in videos. [141] first propose the task that detects events that are both audible and visible. They build up the audio-visual event (AVE) dataset and apply an audio-guided visual attention mechanism to learn visual regions with the related sounding object or motions. Recent works develop plenty of fusion strategies to leverage the global features on this task, such as dual-modality sequence-sequence framework [70] and dual attention matching mechanism [164]. However, the temporal event boundary is second-level in AVE dataset so it is split as 1s-long segments. We study the instant collision event and come across the coarse boundary problem as well. Thanks to the unique properties of collisions and the coarse-to-fine strategy, we can achieve ms-level temporal resolution without the ground truth timestamp.

[39] uses an embodied robot agent to localize a dropped object in 3D virtual rooms. They integrate asynchronous vision and audition and navigate to the object with imitation learning, reinforcement learning, and modular planning. Asynchronism comes from the invisibility of the object. Even though their simulator has been pretty vivid enough for semantic tasks, it has a gap in real-world collision for the ms-level formulation. Falling objects dataset[63], TbD dataset[64] and TbD-3D dataset[120] explores falling motions and fast movings but they do not have audio and depth information.

2.6 Video Frame Interpolation

It aims to synthesize intermediate frames between existing ones of a video. Most state-of-the-art approaches explicitly or implicitly assume a simplistic linear motion. Warping-based methods [10, 106] apply optical flow and forward warping to shift pixels to intermediate frames linearly. Phase-based methods [90, 89] combine the phase information across different scales but the phase is modeled as a linear function of time. Recent methods are developed to approximate non-linear motion, such as kernel-based methods [97, 96], quadratic interpolation [169], cubic motion modeling [26], etc. However, they still fail to complex non-linear motions because precise motion dynamics cannot be captured in the blind time between keyframes. Unfortunately, collisions are super non-linear and instant. Given two keyframes before and after the collision, it is ambiguous to decide whether there is a collision. Hence, these methods are not applicable.

Chapter 3

DeepRange: Acoustic Ranging via Deep Learning

3.1 Background of Acoustic Ranging

[1] There are a number of acoustic ranging algorithms, such as frequency-modulated continuous-wave (FMCW) [176, 85], correlation with known sequence [108, 181, 94], and monitoring phase changes [153, 177, 148]. According to our experiments, FMCW and correlation-based methods outperform the phase-based method under low SNR and/or strong multipath. In this section, we briefly describe how FMCW works and use it to illustrate why existing approaches do not work well under strong noise and multipath.

To estimate the distance propagated by the signals, we let the speaker periodically send chirps whose frequency linearly increases over time as shown in Figure 3.1. Upon receiving the chirp reflected by the target, we perform a mixing operation (*i.e.*, multiply the received chirp with the transmission signal) and apply a low-pass filter. It can be shown that the mixed signal

[1]The work in this chapter was supervised by Prof. Lili Qiu. I was the co-primary author and made contributions to designing research, performing research, analyzing data, and writing the paper. It was originally published in: Mao, Wenguang, Wei Sun, Mei Wang, and Lili Qiu. DeepRange: Acoustic ranging via deep learning. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4, no. 4 (2020): 1-23.

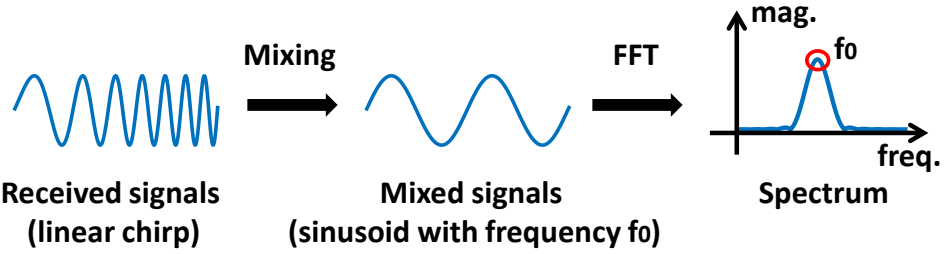
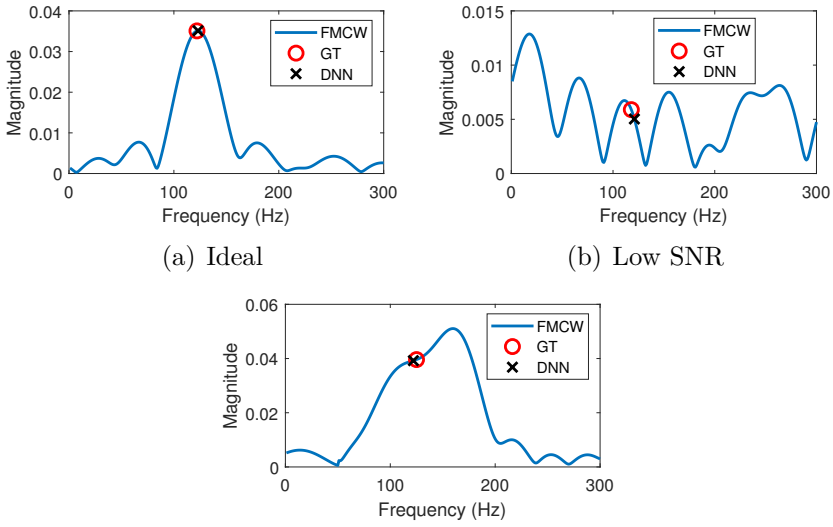


Figure 3.1: FMCW processing stages.

is a sinusoid with the frequency proportional to the signal propagation delay [4, 85]. To determine the delay, we estimate the frequency of the mixed signal by applying Fast Fourier Transform (FFT) on the mixed signal and finding the peak frequency in the spectrum. Figure 3.2(a) shows an example for the spectrum derived by FMCW, where the ground truth frequency is labeled by a red circle.

With other ranging techniques, the performance of FMCW degrades significantly when the SNR is low. In this case, the spectrum derived above becomes very noisy as shown in Figure 3.2(b), and it is difficult to locate the target because the target location does not correspond to the highest peak in the spectrum. The performance of FMCW also degrades under severe multipath. For example, when we place another object next to the target, the microphone will receive reflections from both the target and the nearby object. The peaks corresponding to both the target and object may interfere and merge, which makes it difficult to locate the target as shown in Figure 3.2(c). However, while it is challenging to design a hand-crafted heuristic to select a peak corresponding to the target, it may be possible for deep learning to



(a) Ideal (b) Low SNR

(c) Interference

Figure 3.2: The spectrum derived by FMCW under ideal, low SNR, and severe interference scenarios.

automatically learn the pattern by mining a large amount of data. For example, the shape of a real peak may be quite different from the noise. Similarly, by analyzing the shape of a merged peak, it seems possible to locate the first peak if the target can be assumed to be the closest object after interference cancellation.

We also evaluate the impact of speaker and microphone frequency response and find it does not have a significant impact on FMCW performance. However, learning-based approaches require training traces with realistic speaker/microphone frequency response; otherwise, there is significant performance degradation as shown in Section 4.4.

In summary, the existing ranging algorithms are developed based on

a solid theoretical foundation. However, they face challenges arising from low SNR, multipath interference, and speaker/microphone frequency response. Deep neural network (DNN) has the potential to address these challenging scenarios by automating feature extraction. In fact, our DNN can accurately estimate the distance to the targets in the examples shown in Figure 3.2 by automatically learning from labeled data.

3.2 Approach

In this section, we develop a DNN to estimate the distance based on received signals. To minimize the overhead of collecting training data, we develop a simulator to synthetically generate training data that captures noise (*e.g.*, ambient sounds or random acoustic noise), interference (*e.g.*, the signals propagated from the direct path and reflections from non-target objects), and speaker/mic distortion (*e.g.*, uneven frequency response of the speakers and microphones). We further develop a convolutional neural network (CNN) with long kernel sizes to achieve high accuracy and outperform both classic CNN and fully connected neural networks. We also propose an ensemble method to further enhance the performance.

3.2.1 Signal Generation

Basic model: We use $x(t)$ to denote the transmission signal over time t, which is a chirp as discussed in Section 3.1, where $0 \leq t \leq T$, and T is the transmission period. We use $y(t)$ to represent the received signal and it is what

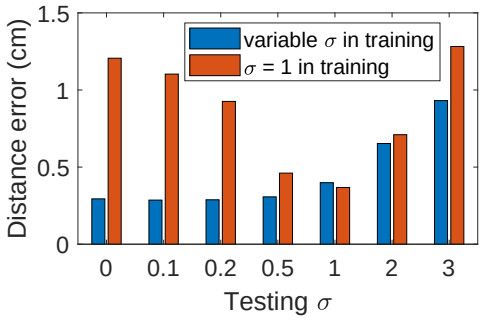


Figure 3.3: The median errors with fixed and variable noise strength.

we need to generate. Ideally, the received signal is the transmission signal after certain attenuation and delay. Therefore, one can generate received signals as follows:

$$y(t) = a_0 x(t - t_0) \tag{3.1}$$

where a_0 is the attenuation coefficient and t_0 is the delay. We train a neural network based on signals generated in this way, and test it using real data (refer Section 3.3 for details about our neural network and testing data). It achieves a 1.9 cm median distance estimation error, which is much higher than 0.75 cm achieved by FMCW.

To achieve better performance, we should generate training data that is more similar to real data and captures various factors in real environments.

Noise: First, we add a noise term $n(t)$ to our received signal as

$$y(t) = a_0 x(t - t_0) + n(t) \tag{3.2}$$

We generate $n(t)$ following Gaussian distributions with a zero mean and standard deviation σ. σ has a significant impact on learning performance. If σ

is too small, the learner cannot gain knowledge about how to deal with low SNR. If σ is too large, the signals are too noisy to learn features from them. Moreover, we cannot use a single value for σ, since the learner may overfit the particular noise level and does not generalize well to other situations. This is shown in Figure 3.3: when we use the synthetic data with a fixed σ to train a neural network, it does not work well even for the test cases with less noise. Note that the testing data in this experiment is generated using our simulator so that we can control the noise level. For all other experiments in this section, we use the testing data collected from real environments as described in Section 3.3.

Therefore, our approach uses σ from a range $[0, 4]$. As a reference, the magnitude of signals reflected by the target at 0.3 m is set to 1. The upper bound of the range is tuned based on the experiments to give the best performance. It covers low SNR scenarios but prevents signals from getting too noisy to provide information. For each transmission period, we randomly choose a value from that range and generate noise following a Gaussian distribution. After taking into account the noise, the distance estimation error of our neural network reduces to 1.1 cm, but is still much higher than FMCW.

Multipath: Multipath is a common phenomenon in wireless signal propagation [143], where the signal generated from the transmitter takes multiple paths to reach the receiver. The signal along each path is a delayed and attenuated version of the transmission signal. The final received signal is the superposition of signals along all the paths. To capture this effect, our gener-

ation model becomes

$$y(t) = \sum_{i=0}^{L} a_i x(t - t_i) + n(t), \tag{3.3}$$

where t_i is the propagation delay sorted in ascending order. The parameter we need to decide is how many reflection paths (*i.e.*, L) should be added into signals. While there could be many reflection paths in practice, most of them are static (*e.g.*, reflection paths from furniture and walls) and can be removed using interference cancellation [92, 27]. The main idea is to record the reflection from these objects in advance when the target is absent and then remove them when collecting the signals to estimate the distance to the target. After interference cancellation, only a few reflection paths remain. In our approach, we generate data to emulate signals after interference cancellation and use them as the neural network input. This not only reduces the number of reflection paths, but also removes the environmental dependency because most reflections from the environment are removed. Based on our experiments, we only need to generate 0–4 reflections (excluding the one coming from the target) with random propagation delay in the training signals and our neural network generalizes well to the cases with more reflections. Further increasing the number of reflection paths leads to little additional improvement.

For the attenuation coefficient a_i, we first set its value inversely proportional to the propagation delay of the i-th path so that the signal energy on that path follows the inverse-square law [143]. Then we multiply the above value with a random number in the range $[1/2, 2]$ to take into account the other

factors that affect the reflection strength, such as object materials and sizes. The multiplier larger than 1 indicates a stronger reflector with a larger reflection area or less absorption. Note that choosing another range (*e.g.*, $[1/3, 3]$) may achieve similar performance. The key point here is to add randomness to prevent the neural network from relying on only the signal strength. After incorporating multipath in the training data, our neural network outperforms FMCW and achieves a 0.58 cm median distance estimation error.

Self interference: In addition to the target reflection, the transmission signals also propagate directly from the speaker to the microphone, which we call self interference. Since the relative position between the speaker and microphone on mobile is fixed, the signals through the direct path can be removed by self interference cancellation. However, the transmission signals sent at different times are slightly different in practice due to variation in device temperature, power supply, and self interference channel. Therefore self interference cancellation is not perfect. Since self interference is orders of magnitude higher than the target reflection due to a small separation between the speaker and microphone, the residual self interference can still be relatively large compared with the target reflection and should be taken it into account in our signal generation.

To model the residual self interference, we use $[1+\epsilon(t)]x(t)$ to represent the transmission signal, where $\epsilon(t)$ captures the variation. Without loss of generality, we generate $\epsilon(t)$ using random splines whose magnitude is within ϵ_{max}. According to our observations, the transmission signal variations are

usually on the order of 10^{-3}. Therefore, we set the upper bound ϵ_{max} as 0.01.

By applying interference cancellation, the residual self interference becomes

$$a_s\epsilon(t-t_s)x(t-t_s), \tag{3.4}$$

where a_s and t_s are the attenuation coefficient and propagation delay of the self interference. Since the signal on the direct path always has the shortest propagation delay, it corresponds to the first term in Equation 3.3. Based on the above discussion, our signal generation model becomes

$$y(t) = a_0\epsilon(t-t_0)x(t-t_0) + \sum_{i=1}^{L} a_i x(t-t_i) + n(t). \tag{3.5}$$

By incorporating the self interference, the median ranging error of our neural network reduces to 0.51 cm.

Transceiver frequency response: Another important factor needed to be taken into account is the frequency response[109] of the speaker and microphone. Ideally, they should have the same gain across the entire frequency. However, real speakers and microphones have different gains across different frequencies, especially for those above 18 KHz (used by our transmission signals), as they are hardly audible and not optimized. To emulate this effect, we let our synthetic signals pass through a digital filter with uneven frequency gains. For each transmission period, we use different filters to cover various possibilities. The frequency responses of these filters are generated using ran-

dom splines. Thus, our signal generation model becomes

$$y(t) = a_0\epsilon(t - t_0)\tilde{x}(t - t_0) + \sum_{i=1}^{L} a_i\tilde{x}(t - t_i) + n(t), \tag{3.6}$$

where $\tilde{x}$ stands for the signals distorted by uneven frequency response. After incorporating this effect, the median distance estimation error of our neural network reduces to 0.42 cm.

As we will show, our synthetic data generated in this way are both realistic and diverse by capturing the important real-world effects, such as multipath, noise, and speaker/microphone distortions. So we transform the target distance x to the received signal y as $y = f(x)$. However, it is challenging to infer x based on y since $f()$ is non-linear, unknown in advance, and varies over time and across environments and speakers/microphones. The neural network is an effective way to model a non-linear and complex relationship between the input (*i.e.*, the received signals) and output (*i.e.*, the target distance). In the following sections, we develop a DNN to estimate the distance based on the raw acoustic signals.

3.2.2 Deep Neural Network for Ranging

A neural network includes three elements: input, output, and network structure. The output of our neural network is the distance between the transceiver and target. Since there could be a few objects whose reflections are not removed by interference cancellation [92], we assume that our target is the one closest to the transceiver to avoid ambiguity. In our signal model in

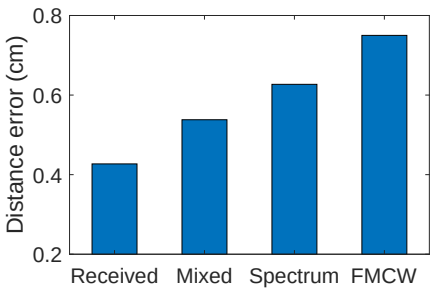


Figure 3.4: The median errors using signals at various stages.

Equation 3.6, the target reflection corresponds to $a_1\tilde{x}(t - t_1)$. Note that any passive-ranging technique needs certain assumptions to distinguish the target reflection from others. We assume the target is the first reflection after static background cancellation since it holds more often than alternative assumptions (*e.g.*, the target is the largest reflection). For example, when a user puts his hand toward mobile for tracking, the hand is closer to the phone than the arm and body, but the body reflection may be stronger due to the larger reflection area.

There is an interesting trade-off regarding which input to use for training. Using the signals at later stages as the input means relying more on feature extraction and less on machine learning. Since feature extraction can reduce the input dimension and make the relationship between the input and output more clear, training becomes easier. On the other side, since feature extraction may also remove some useful information, using the signals at earlier stages might potentially achieve better performance. Figure 3.4 shows the performance of the DNNs trained with signals at different stages. Refer to Section 3.3 for the details about the neural network and testing data.

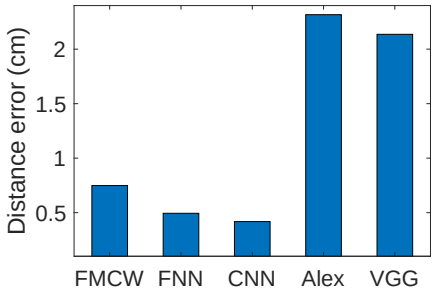
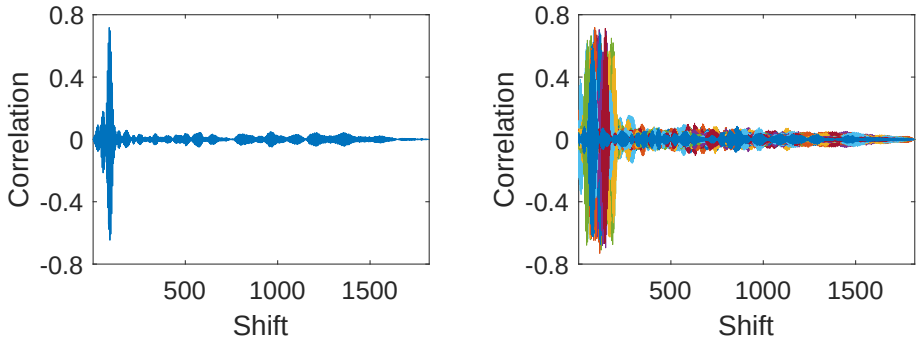


Figure 3.5: The median errors with different network structures.

We see that using received signals after pre-processing (*i.e.*, band-pass filtering and interference cancellation) provides the best performance. Since the pre-processing only removes unwanted artifacts from signals (*e.g.*, out-of-band noise and background reflections), it is beneficial to apply pre-processing to get a cleaner version of received signals. Therefore, we choose them as our network input.

For the network structure, we start with a multi-layer fully-connected neural network (FNN), which is the most general structure. Using 6 hidden layers, 50 neurons in each layer, a 1920-element vector as the input (representing the received signal duration each period), and 200 K synthetic training samples, FNN can achieve much lower distance estimation error than FMCW, as shown in Figure 3.5. The median errors of FMCW and FNN are 0.75 cm and 0.49 cm, respectively.

To further improve the network structure, we examine the weights in our FNN. Interestingly, the weights for the first layer have high correlation with the delayed versions of the transmission signal. To illustrate that, we



(a) Single neuron (b) All neurons

Figure 3.6: The correlation between shifted transmission signals and the weights in the first layer of our FNN.

take one neuron from the first layer and use $\mathbf{w}$ to denote the weights connected to it. Let $\mathbf{x}$ represent the discretized transmission signal. We calculate the cross-correlation between $\mathbf{c}$, $\mathbf{w}$ and $\mathbf{x}$ as

$$c[i] = \frac{< \mathbf{w}, \mathbf{x_i} >}{|\mathbf{w}| \cdot |\mathbf{x_i}|} \tag{3.7}$$

where $< \cdot, \cdot >$ represents the inner product, $|\cdot|$ stands for the L2 norm, and $\mathbf{x_i}$ denotes $\mathbf{x}$ delayed by i elements. If $c[i]$ is close to one, $\mathbf{w}$ has high similarity to $\mathbf{x_i}$. Figure 3.6 plots $\mathbf{c}$ for all neurons in the first layer. We see that the neuron weights have a high correlation with the transmission signal shifted by different amounts. This reminds us CNN, where the same set of weights are shifted by different amounts and applied to different portions of the input. Based on the above observation, we next develop a CNN for ranging.

We first investigate if traditional CNNs can be used for ranging. We train AlexNet and VGG networks customized to fit our application. The number of layers and the structure of these networks remain the same. The filters

are changed to one-dimensional kernel (*e.g.*, a 3×3 filter becomes a 3×1 filter) because our inputs are one-dimensional. The number of filters in convolutional layers and the neurons in FC layers are tuned to have the same amount of parameters as our final model in Section 3.3. Further increasing the model size requires more training data and longer training time with only marginal improvement. As shown in Figure 3.5, both networks perform significantly worse than FMCW. In fact, traditional CNNs rely on convolutional filters with small kernel sizes (*e.g.*, from 3×3 to 11×11) to capture local patterns. Based on the local patterns, CNNs gradually construct a global view of the input at upper network layers. This does not work well in our case since a short convolution is very sensitive to noise and interference, which are common in acoustic signals. If the low-level pattern detection is erroneous, it is challenging for upper layers to mitigate these errors.

To tolerate noise and interference, we develop a CNN using convolutional filters with long kernel sizes. Intuitively, a convolutional filter is used to detect a specific pattern in the signals. When the pattern is longer, it is less likely for noise or interference to resemble the specific pattern. Therefore long kernel sizes up to the transmission signals are more robust for detecting patterns. However, patterns longer than the transmission signal does not help improve the performance. Therefore, we use convolutional filters in the first layer to have similar length to that of the transmission signal. Figure 3.7 shows the performance of neural networks with one convolution layer and 5 fully connected layers but using different filter sizes. We see that the distance

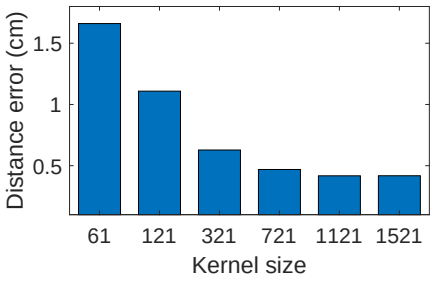


Figure 3.7: The performance of CNNs with various kernel sizes.

estimation errors first reduce, and then taper off as the filter size gets close to the length of our transmission signal (*i.e.*, 1440). Note that we use a grid search to determine the hyper-parameters in DRNet . To illustrate how a particular parameter affects the performance, we show figures by varying one parameter while keeping the other parameters the same.

The final design for DRNet is described in Section 3.3. Although it has a slightly smaller number of weights than our FNN, it achieves 20% lower distance estimation error as shown in Figure 3.5. As discussed, both the first layers in our FNN and CNN are used to detect certain patterns with different shifts, but the convolutional layer is more effective in capturing the pattern and achieves better accuracy.

3.2.3 Ensemble

A major advantage of using synthetic data is that it is easy to generate an arbitrary amount of training data. One way to leverage this benefit is to train larger networks with more data to improve the performance. Figure 3.8 shows the performance of CNNs with different sizes, measured by numbers

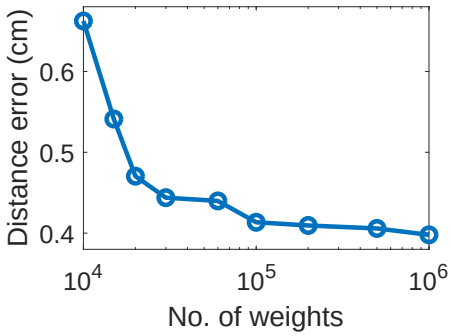


Figure 3.8: The performance of CNNs with various network sizes.

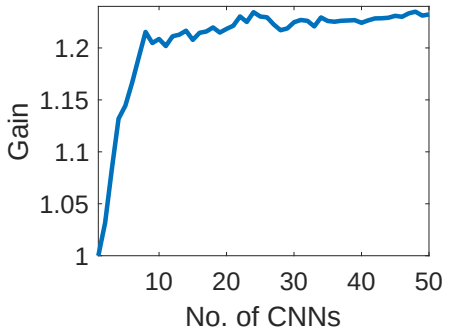


Figure 3.9: The gain with various no. of CNNs for ensemble.

of weights used by them. We change the network size by scaling the number of neurons in each layer of our default CNN and adjust the training data proportionally. We see the performance improvement tapers off when the number of weights in the network is larger than 100 K, which is the size for our FNN and CNN. As shown in Figure 3.17, further increasing the network depth does not help.

Another way to exploit a large amount of data is to train multiple CNNs and apply the ensemble method. [100] shows that a bagging ensemble nearly always outperforms a single classifier. The key observation is that our net-

work converges to different local minimums in different runs, and these local minimums lead to comparable distance estimation errors when applied to real signals. Intuitively, different converging points indicate that the networks capture different features from the signals for distance estimation. These features respond differently to the noise and interference in the signals. By combining these networks (*e.g.*, using the median of outputs from all networks), we can average out the impact of noise and interference and potentially improve the ranging accuracy. A similar idea is explained in [45]. Figure 3.9 shows the performance gain of ensemble learning. For our application, it is interesting to see that using an ensemble of multiple CNNs is more effective than using a larger CNN.

To maximize the effectiveness of ensemble learning, we try to increase the diversity across the networks by using 1) random initialization, 2) a different set of synthetic data to train each network, and 3) random dropout with the probability equal to 0.95 at the input layers. The last strategy not only helps increase the randomness during training but also improves the generalization of networks.

3.2.4 Observations from CNN

In this section, we use visualizations to better understand DRNet . Instead of figuring out exactly how the neural network works, which remains an open challenge, we would like to gain insights about why CNN performs well and what we can learn from CNN to improve ranging algorithms.

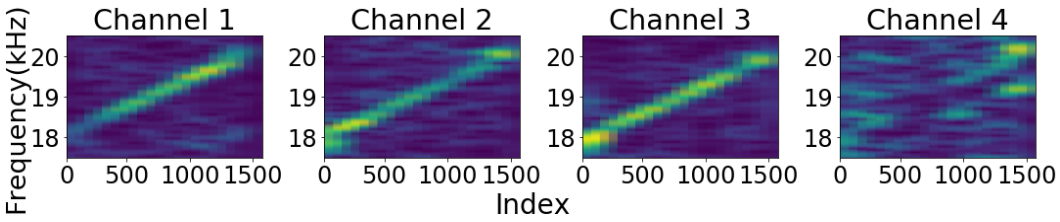


Figure 3.10: The spectrograms of CNN filters.

Observation 1: Three convolution filters are chirps with different energy distributions across frequencies. Figure 3.10 shows the spectrograms of filter coefficients for four channels in the convolutional layers of DRNet . In this figure, the x-axis represents the index of filter coefficients in each channel, and the y-axis stands for the frequency. The color indicates the strength of a specific frequency at a certain portion of the coefficient sequence. As we can see, the spectrograms of the first three filters show the pattern of chirps from 18 KHz to 20 KHz. However, different from a standard linear chirp used by FMCW, these filter coefficients have different energy distributions across frequencies. For example, the filter in the first channel has more energy at the end of the filter sequence since we see a spot there, while the second filter has more energy at the beginning. The third filter has relatively uniform energy across the whole band. This structure may be helpful to handle uneven frequency response caused by speaker/mic distortion. Based on this observation, interesting future work is to explore chirps with non-uniform energy distribution across frequencies for FMCW.

Observation 2: Combining multiple FMCW with transmission chirps shifted by different amounts is helpful to improve the performance.

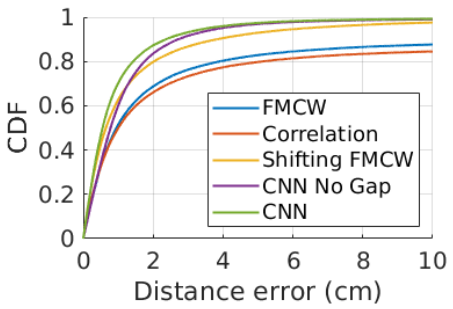


Figure 3.11: Performance of different methods.

The first three filters in DRNet have similar patterns to the chirps, and received signals are multiplied by these filters multiple times with different shifts. In contrast, traditional FMCW only multiplies received signals with the transmitted chirp once with no additional shift.

Inspired by CNN, we explore whether it is beneficial to mix the received signals with the transmission signals with different shifts. More specifically, we shift the transmitted chirps by different numbers of samples so that each one corresponds to a new propagation delay. For each shifted transmitted chirp, we multiply it with the received signals and apply FMCW techniques to estimate the propagation delay. It has an offset from the true propagation delay due to the additional shift introduced to the transmitted chirp. We compensate for the offset and then average the estimation after compensation. As shown in Figure 3.11, this approach (denoted by "shifting FMCW") significantly outperforms traditional FMCW, though its ranging errors are still larger than DRNet . The improvement can be because using different shifts smooths out the errors arising from noise, speaker/mic distortion and multipath.

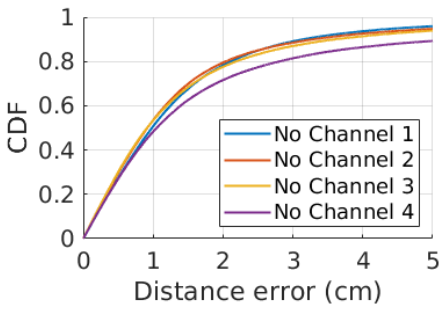


Figure 3.12: Performance of removing one channel information.

We also observe from Figure 3.11 that DRNet outperforms correlation based approach that selects the offset with the maximal correlation coefficient. This is likely because correlation only takes the peak but DRNet uses different filters to remove outliers and improve estimation.

Observation 3: One convolution filter is not a chirp but has a strong impact on the performance. The fourth filter has a very different pattern from the others but has the most significant impact on the performance. Figure 3.12 shows the performance of removing one convolutional channel by setting the corresponding channel outputs as zeros. It shows that removing the fourth channel degrades performance more than removing any other channels.

To better understand the role of the fourth filter, we find that it has more energy near the head and tail of the filters. We expect it to leverage the knowledge of noise, speaker/mic distortion and interference hidden in the non-chirp component of received signals. More specifically, the transmission signal is composed of two parts, a chirp and trailing zeros. Traditional FMCW only processes the chirp part of the receiving signal to compute the beat frequency.

The part with zeros can provide information about noise and interference. The fourth filter leverages this part to improve the ranging performance. To verify that, we replace the received signals in this part with zeros, and the error increases by 41%, as indicated by Figure 3.11 (denoted by "CNN No Gap").

3.3 Implementation

3.3.1 Acoustic Signals

To test our deep learning based acoustic ranging, we use the built-in speaker a smartphone to send chirp signals. The chirp frequency sweeps from 18 KHz to 20 KHz. According to [137], the absolute threshold of hearing (ATH) increases rapidly beyond 10 KHz. For example, human can hear the sound of 1 KHz at 0 dB sound pressure level (SPL), but over 75 dB SPL for sound beyond 17 KHz (10,000x). Our sound level at 18 KHz is 35 dB at 0.5 m from the speaker, well below ATH.

The chirp duration is 30 ms and the transmission period is 40 ms. We use the microphone on the same smartphone to receive the signals reflected from the target. The sampling rate of acoustic signals is 48 KHz so that the number of samples in a transmission period is 1920. We use the samples in one period as the input to our network for estimating the distance between the smartphone and target.

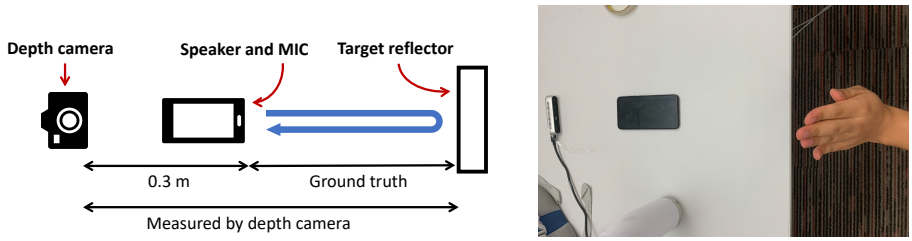
3.3.2 Training

For training, we generate synthetic data based on the signal parameters mentioned above and the model parameters discussed in Section 4.2. We generate ground truth distance and signals for 200 K transmission periods to train each CNN.

To tune the hyper parameters for DRNet , we generate synthetic testing signals for 100 K transmission periods. By applying the grid search, the final design is described as follows. DRNet has one convolutional layer with 4 filters. The kernel length for each filter is 1521. The convolution layer is followed by a max pooling layer with the stride and window size equal to 4. Then there are 5 fully connected layers with 200, 100, 50, 50, 50 neurons, respectively. We train our network in PyTorch [3] using Adam [61] optimization algorithm. The loss function is the mean square error. The batch size is 200. The initial learning rate is 0.001, which decays by a factor of 0.2 every 50 epochs. In total, we train 25 CNNs for ensemble learning.

3.3.3 Ground Truth

To evaluate the CNNs trained by synthetic data, we use them to estimate the distance between the mobile and target as shown in Figure 4.7 based on the signals recorded by the mobile. In the user study, the target is a user's hand. In other evaluations, the target is a 10 cm$\times$10 cm white cardboard. The distance between the mobile and target varies from 0.2 m to 1.2 m in all experiments except the long range one (i.e., Figure 3.19), where the testing



(a) Illustration

(b) Experiment at the office cubic

Figure 3.13: Top down view of our experiment setup.

range is extended to 5 m.

To get the ground truth distance, we use the Intel RealSense D415 camera system [54]. Its accuracy is 1mm when the distance is between 0.3 m and 1.5 m based on our calibration. It comes with an RGB camera and a depth camera. We use the RGB image to find our target and read the depth from corresponding pixels in the depth image. Since the depth camera does not work when the distance is less than 0.3 m, we put it 0.3 m behind the phone as shown in Figure 4.7 so that the minimum distance between the camera and the target is larger than 0.3 m. The accuracy of our depth camera significantly degrades when the distance is larger than 1.5 m. For long range experiments, we manually calibrate the distance between the phone and target.

3.3.4 Testing

As Figure 4.7 shows, we collect testing traces in the following steps: 1) place the camera and smartphone as described above; 2) send the chirp signals with the smartphone speaker; 3) let the microphone record the signals for one second when the target is not present to capture the background reflection,

which is used for interference cancellation [92]; 4) place the target and move it in front of the phone for one minute; 5) use the depth camera to get the ground truth distance; 6) use the microphone to receive the signals; 7) perform pre-processing on received signals, including band-pass filtering and interference cancellation [92]. Note that Step 5 is only used to quantify the accuracy but not required by our approach.

To demonstrate that our approach generalizes well, we use 11 smartphones with different brands and models to collect data, including Samsung S9 plus, Samsung S7 Active, Samsung S7 international version, Samsung S7 US version, iPhone X, iPhone 6, iPhone 5S, Huawei Mate 9, Huawei Honor 8, Xiaomi 8, and Google Pixel. The speakers and microphones on these phones have very different acoustic characteristics (*e.g.*, frequency responses) as shown in Figure 3.14. In this figure, the y-axis represents the normalized speaker and microphone amplification on signals at a specific frequency. We use the approach developed in [42] to measure the frequency response for a mobile. The testing traces are collected from 4 real environments, including a lab, a corridor, a meeting room, and a cube area. There are furniture and walls in all these locations. Besides static objects, there is also dynamic inference in our testing environments. For example, there are other people walking by our experiment setup. Moreover, the user's body and arm also exhibit some movements and introduce non-static interference.

We collected 119 testing traces. Each trace has 1375 transmission periods. In each period, we take the received signals as the input and the cor-

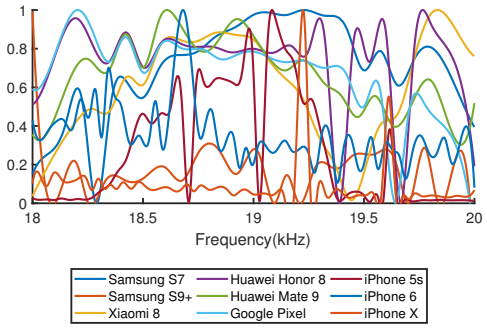


Figure 3.14: Combined frequency response of speakers and mics for different phones at 18–20 KHz.

responding ground truth distance as the output label. The traces are divided into four groups, where all groups except the last one track a board. (i) 33 traces in ideal scenarios where the SNR is around 10 dB and there is no object near the target; (ii) 38 traces where SNR falls into -15–5 dB due to low speaker volume or large separation between the mobile and target, (iii) 28 traces with SNR around 10 dB and an object (a 10 cm×10 cm cardboard) behind the target (0–15 cm) to introduce severe interference. (iv) 20 traces for tracking different users' hands.

3.4 Evaluation

In this section, we evaluate the performance of our ranging approach. We use the median and cumulative distribution functions (CDF) of distance estimation errors as our performance metrics. Except for the results shown in Figure 3.16, all evaluation uses the testing data collected from real environments as described in Section 3.3.

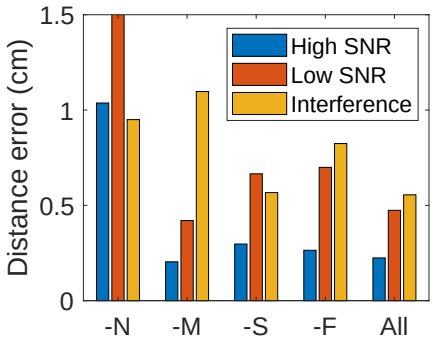
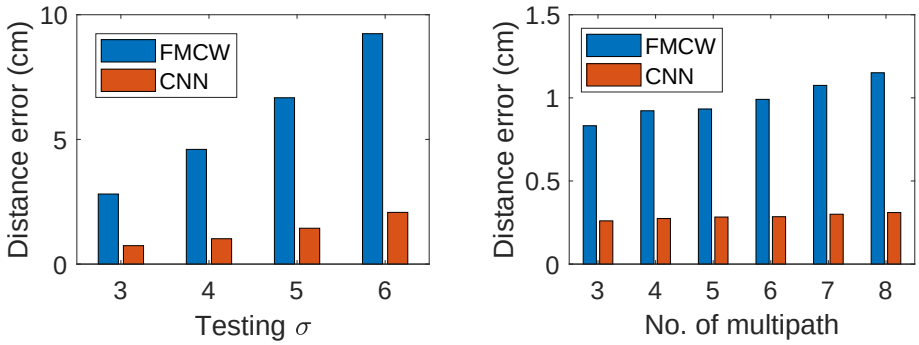


Figure 3.15: The impact of signal generation.

3.4.1 Micro Benchmark

We first evaluate various aspects of our approach, including signal generation, network generalization, network depth, and inference time. We report the ranging performance using a single CNN, and compare it with FMCW.

Signal generation: We evaluate the impact of key components in our signal generation. For this purpose, we generate training signals without noise (denoted by -N), without multipath (-M), without self interference (-S), or without frequency response (-F), respectively, and evaluate the performance of the CNN trained using signals without certain component. Figure 3.15 shows the testing performance on different testing data. We see that without adding noise, the group with low SNR has the largest median ranging errors, *i.e.*, 5.3 cm. Without adding multipath, the CNN does not work well for the group with strong multipath interference. Neglecting self interference and frequency response also degrades the performance. DRNet is trained using signals with all these components to achieve good performance across a wide



(a) Noise (b) Multipath

Figure 3.16: Generalization of our CNN.

range of scenarios.

Network generalization: We evaluate the performance of our network for the cases not covered by the training data. As mentioned in Section 4.2, σ (*i.e.*, standard deviation of noise) varies from 0 to 4 in our training data, and the number of multipath (excluding the target reflection) is randomly chosen from 0 to 4. To test if DRNet works when the noise and multipath go beyond these ranges, we generate two sets of synthetic data with 1) σ varying from 3 to 6 and zero multipath and 2) the number of multipath varying from 3 to 8 and σ equal to 1.

We use synthetic testing data in this experiment because we need to control the noise and multipath level. The testing performance on the two data sets is shown in Figure 3.16(a) and 3.16(b), respectively. We observe that the estimation errors of DRNet increase only marginally when the noise and multipath are outside the ranges covered by the training data, DRNet degrades slower than FMCW since it is more robust to noise and multipath. These

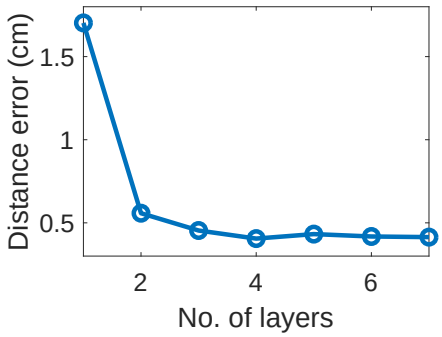


Figure 3.17: The impact of numbers of layers.

results demonstrate our model generalizes well to the uncovered scenarios.

Network depth: We evaluate the impact of the number of fully connected layers in our neural network. The deepest CNN we evaluate has 7 fully connected layers with 200, 100, 50, 50, 50, 50, 50 neurons, respectively. Each time we remove the last layer before the output and create a shallower CNN, until there is only one fully connected layer. As in Figure 3.17. the estimation error first reduces significantly and then tapers off. The result shows 5 fully connected layers are sufficient for our application.

Inference time: We run DRNet on a desktop with Ubuntu 16.04, Nvidia GTX 980 [98], and 6 GB VRAM. We also run it on Android 9, Snapdragon 845, and 6 GB RAM [113]. For the phone, we implement DRNet with Android NDK and uses Eigen [55] as matrix calculation library. The inference time for a single CNN is 0.36 ms on the desktop and 1.98 ms on the phone. The total time for 25-CNN ensemble learning is 7.1 ms on the desktop and 36.5 ms on the phone. Since the transmission period is 40 ms, our approach can support

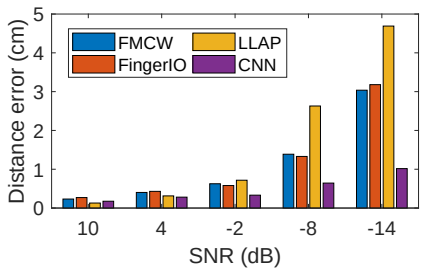


Figure 3.18: The median ranging errors at various SNR.

real time ranging even on the phone. In comparison, the running time for FMCW is 0.55 ms on the desktop and 4.7 ms on the phone.

3.4.2 Overall Performance

In this part, we evaluate the overall performance of our approach using an ensemble of 25 CNNs. We compare our method with FMCW, FingerIO [94], and LLAP [153]. FingIO uses correlation and LLAP uses phase to estimate the distance. We use 18-20KHz OFDM signals in FingerIO, and send five sinusoids at 18, 18.5, 19, 19.5, and 20 KHz in LLAP. All approaches use the same experimental setup.

Impact of SNR: We evaluate the performance of our learning based ranging under different SNRs. For this purpose, we measure the SNR of our testing data and show the ranging performance for the data for given SNR values (allowing ± 3 dB variance). As shown in Figure 3.18, when the SNR is around 10 dB, the phase-based approach (*i.e.*, LLAP) achieves the best performance – its median ranging error is 1.3 mm. It is followed by our approach with 1.7 mm median error, while FMCW and FingerIO have errors larger than 2 mm. As

the SNR reduces, the performance advantage of our learning based approach becomes more significant. At SNR of -14 dB, our approach reduces the median ranging error by a factor of 3 over FMCW and FingerIO. LLAP has the worst ranging accuracy in this case, indicating that phase-based approaches are most sensitive to noise.

Impact of range: We evaluate our approach under long distances. In this case, the propagation delay of reflected signals can be large (*e.g.*, 29 ms for a target at 5 m away). If we use the signals aligning with our transmission period for distance estimation, the target reflection is only present in the last few milliseconds of the signals. This has a negative impact on estimation accuracy. Instead, we choose a 40 ms window roughly aligning with the target reflection, and use the signals in this window for distance estimation. The delay estimated this way starts from the beginning of a selected window. We get the propagation delay by adding the offset between the starts of the transmission period and the selected window. Note that the rough knowledge about propagation delay of the target reflection is obtained by correlating with transmitted signals and detecting the second peak since the first peak is the direct transmission from speaker to microphone). We use the same correlation approach in all schemes for fair comparison.

The results are shown in Figure 3.19. As we can see, the distance estimation error of our approach is still within 1 cm at 4 m, while FMCW has the error close to 4 cm in this case. This experiment indicates that our approach has a larger working range.

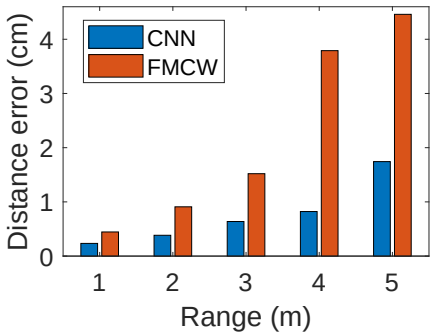
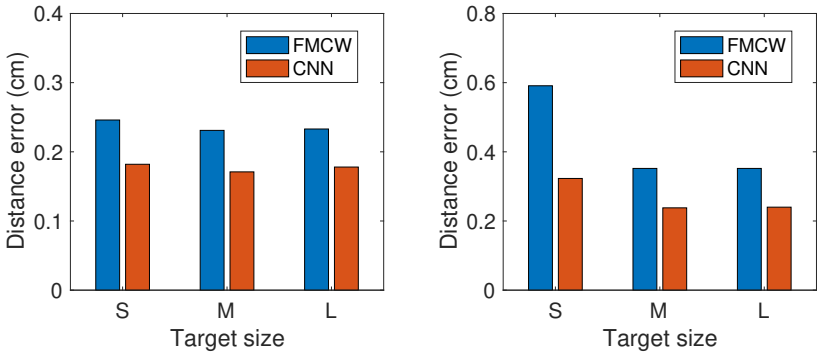


Figure 3.19: The median ranging errors at various ranges.



(a) High SNR (b) Low SNR

Figure 3.20: The ranging errors for target objects with different sizes

Target size: We evaluate the ranging errors for targets with different sizes: a 2 cm×2 cm cardboard (denoted by S), a 10 cm×10 cm one (denoted by M), and a 40 cm×40 cm one (denoted by L). We perform experiments under both high SNR and low SNR, where signals reflected by the 2 cm cardboard ranges between 0 and 10 dB SNR. A large cardboard tends to reflect more signals and yields a higher SNR. However, as the cardboard gets even larger, its gain becomes marginal since the regions far away from the perpendicular reflection point reflect little energy back to the microphone. We find the signals reflected

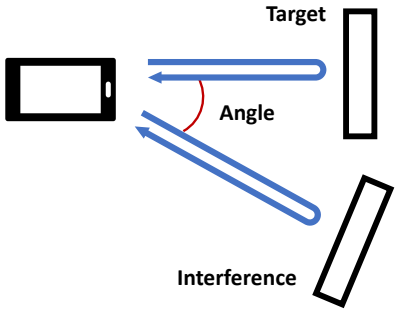


Figure 3.21: The interference at different angle.

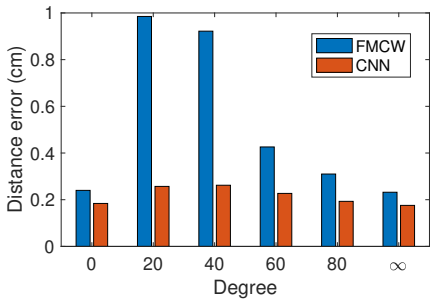


Figure 3.22: The interfering object at different angles.

by 10 cm cardboard are 6–7 dB stronger than those for 2 cm one. However, the SNRs for 10 cm cardboard and 40 cm one are similar. When the SNR is sufficiently high, the accuracy is comparable across all reflectors as shown in Figure 3.20(a). When the SNR is low, larger targets have higher SNR than smaller targets and experience smaller estimation errors as shown in Figure 3.20(b).

Impact of interference angle: We evaluate the impact when an interfering object (a cardboard with the same size as the target) is placed at different angles, as shown in Figure 3.21. The interfering object is 10 cm farther away from the mobile than the target so that the target is always the first reflector.

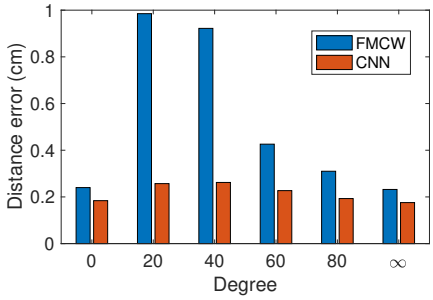


Figure 3.23: The median ranging errors with interference at various distance.

The results are illustrated in Figure 3.22. For comparison, we show the ranging performance without interfering objects (denoted by ∞). As we can see, when the angle is zero, the interfering object has no impact on ranging because its reflection has been occluded by the target. When the interfering object is placed at other angles, the impact of interference varies significantly because the speaker radiates different portions of energy across different directions, which can be characterized by the speaker's radiation pattern. For our speaker, the interference is maximized at 20 degrees while minimized at 80 degrees. Therefore, we use 20 degrees as the default angle for interference experiments in this paper.

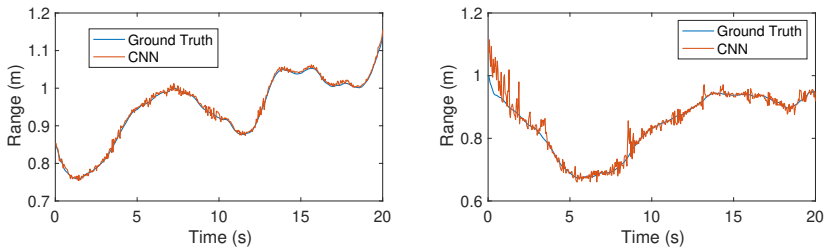
Interference distance: We evaluate the ranging performance under interference at different distances. For this purpose, we place the interfering object at a 20-degree angle from the target and 2.5-12 cm farther away from the mobile. Figure 3.23 shows the distance estimation errors with various separation between the target and interference reflection, where ∞ indicates no interference. Our approach achieves the best performance under multipath. It reduces the

ranging errors by a factor of 4.4 when the interference is 7.5 cm away, which is the most challenging cases. LLAP has the largest estimation errors under interference. We see that the errors first increase and then decrease, and reach the maximum when the interference reflection is 5–7 cm away from the target. This is because if two reflections have a similar distance, the interfering object does not affect the distance estimation. If the interference reflection is well separated from the target, they can be easily differentiated.

3.4.3 User Study

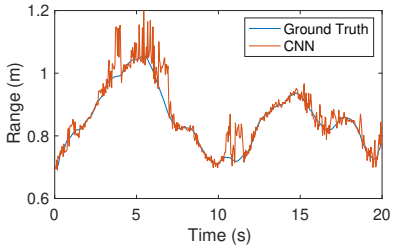
We use our ranging approach to track the distance between the user's hand and a smartphone, as shown in Figure 1.1(a). This is a key building block in the motion tracking. We conduct the experiment with 10 users including 4 women and 6 men. Their ages are between 20s and 50s. We let each user stand 1.2 meters in front of the phone, raise the hand to roughly the same height as the phone, and move the hand back and forth in an arbitrary pattern. The details of trace collection are described in Section 3.3.

Tracking samples: To provide intuition about how our approach performs for hand tracking, we show the raw traces with different performance. For this purpose, we sort all the traces based on the median ranging errors and plot the traces ranked at 20%, 50%, and 80%, as shown in Figure 3.24. The traces for our approach are directly generated from CNN outputs without any additional filtering. The median ranging errors of selected traces are 0.3 cm, 0.7 cm, and 1.2 cm, respectively.



(a) top 20%

(b) median



(c) bottom 20%

Figure 3.24: The sample user traces with different performance.

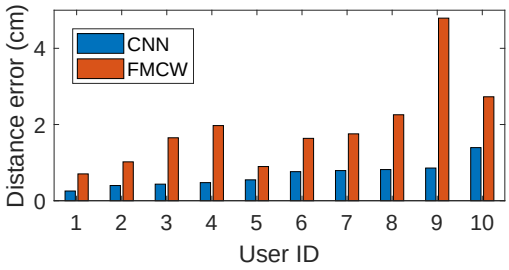


Figure 3.25: The median ranging performance for different users.

Performance for different users: Figure 3.25 and 3.26 show the median ranging errors and CDF across different users, respectively. We rank the users according to their median errors. For comparison, we also show the performance using FMCW. We see that our approach out-performs FMCW for all users, and the median errors are reduced by a factor of 1.6x - 5.6x. Moreover, we observe that there is a large performance variation across different

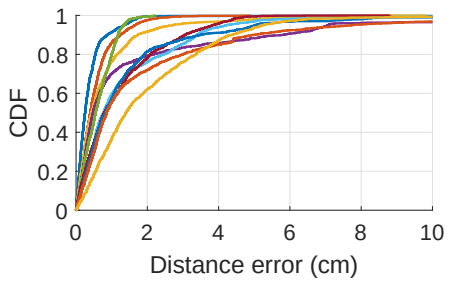


Figure 3.26: The CDF of ranging performance for different users.

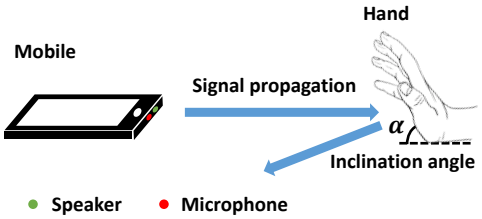


Figure 3.27: The side view of the user's hand.

users. User 1 achieves 0.255 cm median error and 0.5 cm 80-percent error, while User 10 has 1.39 cm median error and 3.4 cm 80-percent error. The performance variation mainly comes from different hand inclination and body posture. When the user's hand is not perpendicular to the signal propagation path, (*e.g.*, the inclination angle α in Figure 3.27 is less than 90 degrees), most of the reflection propagates downward, instead of returning to the phone, as shown in the figure. Different users use different inclination angles. The smaller inclination angle reduces SNR. In addition, the body posture affects the interference. When the separation between the body and hand is small and/or the area of the body facing the phone is larger, there is stronger interference. Since our approach is more robust to low SNR and strong interference, its performance benefit is higher in these traces.

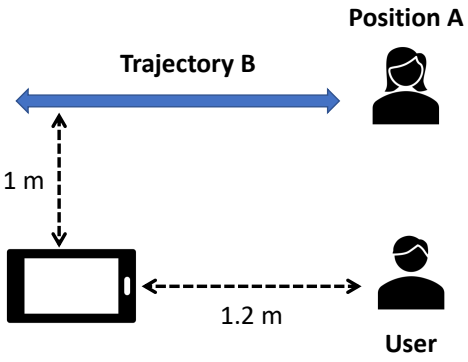


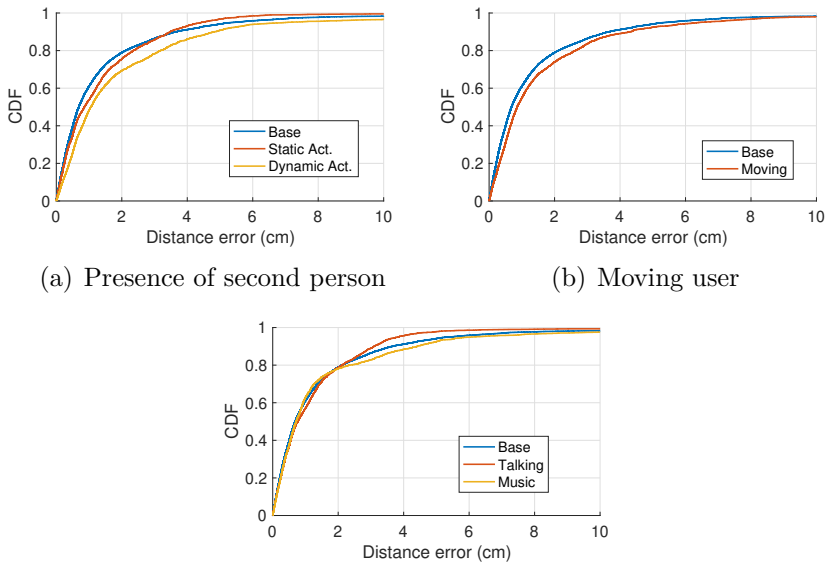
Figure 3.28: The setup with human interference.

Performance under practical situations: Furthermore, we evaluate our approach under the following practical situations:

- There is another person near the target user. We consider two cases. In the first case, the second person stands around the position A as shown in Figure 3.28 and performs semi-static activities like playing games on a mobile phone. In the second case, the second person walks roughly along the trajectory B shown in Figure 3.28.

- The user is allowed to walk back and forth towards the mobile. In this case, the tracked motion is the net effect of the user walking and the hand movement with respect to the user body.

- There are ambient sounds during the experiments. We consider two common types of sounds: music and voice. We set the music to the same volume as that of inaudible tracking signals, while the loudness of voice is the same as in normal conversion (*i.e.*, around 60 dB). The sound source is at 1.5 m away from the mobile.



(a) Presence of second person

(b) Moving user

(c) Ambient sounds

Figure 3.29: The ranging performance under various practical situations.

Figure 3.29(a) compares the tracking accuracy with and without the second person, where the latter is denoted as "Baseline". We observe that the second person with semi-static activities has little impact on the tracking performance, while the one with dynamic activities (*e.g.*, walking around) affects the ranging performance and the median error increases from 0.7 cm to 1.0 cm. This is expected since dynamic interference is harder to remove. Moreover, the second person may be temporarily closer than the user's hand. Under such challenging situations, our approach can still provide reasonable tracking performance with 1 cm median errors.

Figure 3.29(b) compares the performance with a semi-static and moving second user. As we can observe, our approach is fairly robust under the moving

user. The median ranging error is 0.8 cm.

Figure 3.29(c) compares the performance with and without ambient sounds, where the latter is denoted as Baseline. The ambient sounds, such as music and voice, has little impact on the accuracy since our tracking signals are at 18–20 KHz band while common ambient sounds like music and voice have little energy in such high frequency band.

Chapter 4

Spatial Aware Multi-Task Learning Based Speech Separation

4.1 Background

4.1.1 Multi-Task Learning

[1] Multi-task learning is a training paradigm to train ML models with data from multiple tasks at the same time. These tasks share similar semantic features. Thus, learning shared representations can increase data efficiency and potentially lead to faster learning speed for related tasks. Multi-task learning can improve multiple performance metrics by learning them jointly. It exploits commonalities and differences across tasks. Multi-task learning learns the shared representations at the early stage and then solves multiple downstream tasks at the same time in the later stage. It captures the natural learning process of human beings. For example, when a newborn baby starts to learn speech, it will recognize the sentence as well as understand the emotion from the speech simultaneously. The baby learns two tasks, i.e. speech recognition and emotion classification together.

[1]The work in this chapter was supervised by Prof. Lili Qiu. I was the first author and made contributions to designing research, performing research, analyzing data and writing the paper. It was originally published in: Sun, Wei, Mei Wang, and Lili Qiu. Spatial Aware Multi-Task Learning Based Speech Separation. arXiv preprint arXiv:2207.10229 (2022).

4.1.2 AoA Estimation

A number of algorithms have been developed for AoA estimation with a multi-channel receiver, including phase [16], beamforming [14], and MUSIC [122, 157]. The fundamental of AoA estimation is to figure out the subtle phase difference across multiple channels because of the spatial uniqueness of each channel. We introduce several typical AoA algorithms and conclude the insight of these methods.

MUSIC: MUSIC holds a high accuracy in theory. The key idea of MUSIC is that the steering vector of the unknown AoA $a(\theta_0, \phi_0)$ is highly correlated with the eigenspace of the auto-correlation matrix

$$R = x^H x \tag{4.1}$$

where x is the received signal and x^H is the conjugate transpose of x. θ and ϕ denote the azimuth and elevation angles, respectively. We then perform eigenvalue decomposition on R, and sort the eigenvectors in decreasing order of the corresponding eigenvalues. The signal space consists of the first M eigenvectors. The noise space, denoted as R_N, consists of the remaining eigenvectors. We derive the pseudo-spectrum of the mixed signals based on R as

$$P(\theta, \phi) = \frac{1}{a(\theta, \phi)^H R_N R_N^H a(\theta, \phi)} \tag{4.2}$$

where

$$R_N^H \cdot a(\theta_0, \phi_0) = 0 \tag{4.3}$$

In the free space, the AoA can be estimated by searching for a peak in the pseudo-spectrum. MUSIC can deal with M different AoAs of uncorrelated signals. However, Multipath reflections break this assumption and fuse the steer vectors of different AoA, which makes AoA estimation more challenging. Ambient noise and interference can further complicate the issues by adding false peaks. Therefore, it is hard to get a reliable AoA estimate in the real environment.

2D MUSIC jointly estimates the distance and AoA to increase the spatial resolution. The key idea of 2D MUSIC is to transform signals into a special 2D sinusoid with frequencies proportional to the distance and the distance and the AoA. It requires actively emitting modulated period signals and operating the reflection signals. Otherwise, it cannot sense the distance at all. We let a speaker on a computer transmit periodic FMCW chirps, whose frequency increases linearly from f_{min} to f_{max} during each period T. This yields a transmission chirp signal

$$u_t(t) = cos(2\pi f_{min} t' + \frac{\pi B t'^2}{T}), \tag{4.4}$$

where $t' = t - nT$ After going through the channel with the propagation delay t_d and attenuation α, the received signal becomes

$$u_r(t) = \alpha cos(2\pi f_{min}(t' - t_d) + \frac{\pi B(t' - t_d)^2}{T}) \tag{4.5}$$

Based on this formulation, we apply the MUSIC algorithm to the received signal to derive the pseudo spectrum. t_d is decided by the distance and

the AoA, so is the steering vector. We can generate a 2D profile $P(d, \theta)$, where d and θ denote the distance and azimuth angle, respectively.

GCC-PHAT: Generalized Cross Correlation with Phase Transform [16] is another popular AoA estimation algorithms. It is regarded as a basic phase transformation for many deep learning methods in recent works. We focus on the signal processing techniques and dig the more insight from its formulations. GCC-PHAT computes the cross-correlation between the received signal and the reference signal and estimates the arrival delay for all microphones. Then the delay can be applied to search for the best AoA. The correlation of GCC-PHAT is given as below:

$$\boldsymbol{GCCPHAT_m}(f) = \frac{\boldsymbol{Y_m}(f)\boldsymbol{Y_{ref}}(f)^*}{|\boldsymbol{Y_m}(f)\boldsymbol{Y_{ref}}(f)^*|} \tag{4.6}$$

where $\boldsymbol{Y_m}(f)$ is the noisy speech of the channel m in the frequency domain, $\boldsymbol{Y_{ref}}(f)$ is the reference target speech in the frequency domain, * denotes the complex conjugate. The AoA profile can be represented as

$$P(\theta, \phi) = \frac{1}{\sum_m ||2\pi f \tau_m(\theta, \phi) - GCCPHAT_m(f)||_2} \tag{4.7}$$

where $\tau_m(\theta, \phi)$ is the delay from the source at the azimuth θ and elevation ϕ to channel m. A clear reference signal can benefit estimating an effective phase difference a lot and searching for the AoA by more accurate delays.

4.1.3 Beamforming

Beamforming is the central technique used in sensor arrays for directional signal transmission or reception. It targets to find an optimal spa-

tial filter to combine signals in a multi-channel array to strengthen signals at particular directions and null signals in other directions. Minimum Variance Distortionless Response(MVDR) is one of the most classic beamforming algorithms. It takes advantage of the spatial position of the source signal and estimates the optimal weights to increase the signal energy from the given AoA. MVDR is an adaptive beamformer s to minimize the variance of the beamformer output and mitigates the effect of the noise. Assuming no correlation between the target speech and interference, the weights of MVDR are given as follows:

$$\boldsymbol{w}(f, AoA) = \frac{\boldsymbol{\Phi}_Y^{-1}(f)\boldsymbol{v}(f, AoA)}{\boldsymbol{v}^H(f, AoA)\boldsymbol{\Phi}_Y^{-1}(f)\boldsymbol{v}(f, AoA)} \tag{4.8}$$

where $\boldsymbol{\Phi}_Y^{-1}$ is the covariance matrix of sub-band noisy speech and $\boldsymbol{v}(f, AoA)$ is the steering vector of the target speech at the frequency band f with the given AoA. It shows that AoA plays an important role to construct the filter weights.

4.2 Approach

We consider a common scenario in teleconferencing. The user is using a computer equipped with either an internal or external microphone array and speaker to participate in an online meeting. We target to separate out the target speech from the noisy audio. The multi-channel audio alone can perform speech separation. We make two important observations about the user during the speech. First, the user will move dynamically in front of

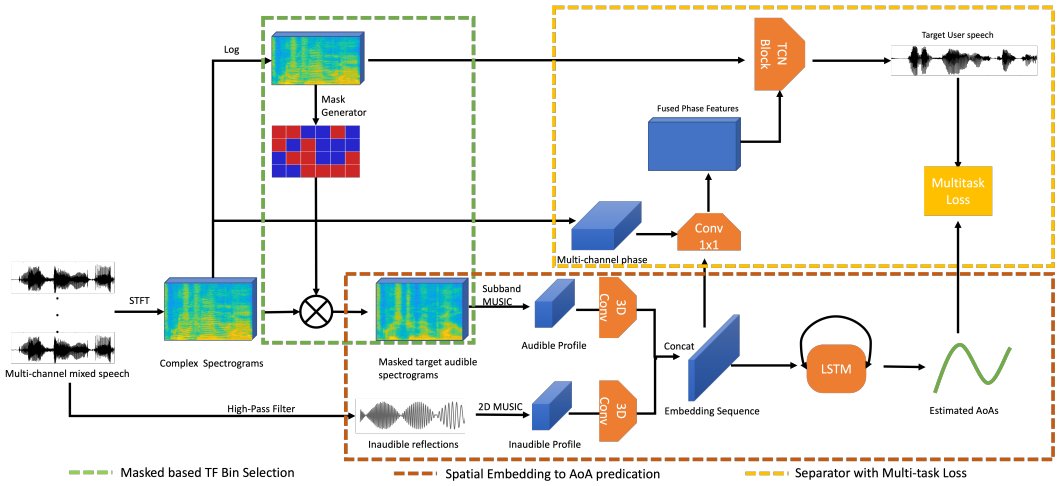


Figure 4.1: It shows the following components: (i) generating masks from the TF bins in audible signals and using the mask to generate MUSIC profiles from speech, (ii) generating spatial embeddings from inaudible tracking sound, and (iii) multi-task learning to jointly separate source and estimate AoA based on the spatial embedding

the devices. Second, the separation task and AoA estimation task are highly correlated. Thus, we propose to jointly learn the target speech and the AoA. Besides, the device emits an inaudible sound to detect human motion as the side channel to improve the AoA estimation.

Figure 4.1 shows the overall processing. It takes the acoustic signals from a microphone array with a sampling rate of 44.1 kHz and performs a low pass filter (*i.e.*, 0-8kHz) and a high pass filter (*i.e.*, 18-20 kHz). The outputs from the low-pass and high-pass filters contain audible speech signals and inaudible tracking signals, respectively. We use both signals to generate location embeddings, which will be used together with the audible speech signals under interference and ambient noise to extract clean target signals.

Our approach consists of four major steps: **(i)** generating location embeddings from audible signals, **(ii)** improving the location embeddings using inaudible signals, **(iii)** using multi-task learning for joint source separation and AoA estimation, and **(iv)** speeding up inference. Below we describe each of the steps.

4.2.1 Localization using Audible Signals

We first describe how to localize the user using audible signals. We take the audio signals recorded by a microphone array, and feed them to a low-pass filter (0-8kHz) since human speech is usually below 8 kHz. Then we downsample the audio signals to a 16 kHz sampling rate and apply STFT of 512 points to compute the complex spectrogram for different TF bins. We use a hop length of 10 ms in STFT and a Hamming window of length 32 ms. We then try to estimate the AoA for each bin using the MUSIC algorithm. However, some TF bins are dominated by interference and noise. They contribute a significant bias to the target AoA if we apply the MUSIC straightforwardly. We apply a novel masked-based AoA estimation to extract spatial features.

Mask Based TF bin selection: We improve the MUSIC accuracy by carefully selecting the frequency bins in the audible signals for aggregation to minimize the impact of noise and interference. Specifically, since MUSIC assumes narrowband signals, we apply STFT to the audible signals to generate TF bins, where each bin occupies 31.25 Hz and 10 ms. We then perform MUSIC on each TF bin and aggregate the MUSIC profiles across TF bins.

A natural approach to aggregate the MUSIC spectrum across different frequency bins is, to sum up the MUSIC profiles from all TF bins and select the angle corresponding to the highest peak. However, not all TF bins contain the target user's speech due to the sparsity of human speech over the frequency band [162]. Therefore, it is important to select the TF bins that contain strong Signal to Noise Ratios (SNR) from the target user. It is challenging to select the TF bins by just analyzing the power and phase because unlike tracking signals, human speech is out of our control and hard to predict. Moreover, some TF bins may contain significant ambient noise and interference, so we cannot simply select the TF bins solely based on the overall magnitude, but should select the TF bins with high SNR from the target user.

Mask generation: A number of approaches have been proposed to generate TF masks for speech enhancement. A few works estimate the amplitudes of the audio spectrogram (*e.g.*, real-valued ideal binary mask (IBM) [50, 51] and ideal ratio mask (IRM) [156]). [162, 174] develop DNN-based approaches to generate amplitude and phase masks. Most of these methods focus on combating noise.

We use IBM to select appropriate TF bins for the target user from the mixed noisy complex spectrogram. IBM is a method for speech separation based on deep neural network [51]. Even though IBM is not the best method for source separation, it is a good fit for selecting the TF bins dominated by the target speaker. Other mask-based methods apply linear translation to the original TF bins, which introduces phase distortion and degrades mask

generation.

IBM is based on the sparsity of human speech (*i.e.*, the number of non-silent TF bins from a speaker tends to be small). It determines a binary mask for each TF bin, where 1 means the target signal dominates interference and 0 otherwise. It takes a downsampled audio signal and decomposes it into 2D TF bins. Then it extracts several features, such as the autocorrelation of a filter response, the autocorrelation of an envelope of filter response, and cross-channel correlation. Next, it performs clustering based on these extracted features (*e.g.*, cluster into a target stream and an interference stream), and tags each TF bin with either target dominant ("1") or interference dominant ("0") based on similarity with the clean target signal (spoken at a different time), which is also an input. We use the clean target signal, which is location independent and can be collected only once during user account creation. Since an online meeting requires a user to sign in, it is reasonable to assume the target user is known.

We train the IBM mask estimator using Deep Cluster [48], which is a general and robust method to estimate the mask. It takes a log power spectrogram (LPS) (*i.e.*, the log power of received signal across 256 positive frequency bins) as the input to estimate an initial binary mask of the target user. We get the pre-trained model using the LibriMix [31] data along with our own testbed traces described in Section 4.4.1. The binary mask is a coarse estimate of effective TF bins, but it maintains phase information and is useful for selecting the TF bins for further analysis. We estimate binary masks for

all microphones. To minimize interference, we select the TF bins for AoA estimation only when the masks from all microphone channels are 1s. In this way, we effectively remove the TF bins with large interference and noise.

Applying a mask to MUSIC: We apply the MUSIC algorithm to the TF bins with masks. Then we concatenate the MUSIC spectrum from all frequencies together. The output profile is represented as a 2D matrix of size $M_f N_a$ across different frequencies, where M_f denotes the number of frequency bins and N_a denotes the number of angles. M_f is set to 103 frequency bins (equally spaced from 800Hz to 4KHz for human speech), and N_a is set to 181 (spanning 0 degrees to 180 degrees with 1 degree apart) in our evaluation. This will be further combined with the output from inaudible signals for generating location embeddings.

4.2.2 Leveraging Inaudible Sensing

Apart from the audible band, we leverage the inaudible acoustic signal to improve the robustness of spatial representation.

Motivation for using inaudible signals Estimating multipath profile solely based on speech has several limitations because of the audible band. First, audible signals may contain significant interference and ambient noise, which results in significant AoA errors. Figure 4.2 shows the AoA estimation from the audible target speech is fairly accurate without interference. However, when the interference is introduced to the audible band, the estimation deviates from the ground truth a lot. The deviation can be large and irregular due

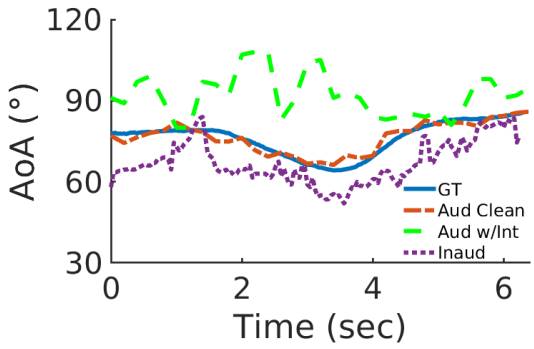


Figure 4.2: AoAs estimated from the audible band and inaudible band. GT refers to the ground truth AoA. Aud Clean refers to the AoA estimated by only target speech. Aud w/ Int refers to the AoA estimated by a mix of target speech and audible interference. Inaud estimates AoA with only inaudible reflections.

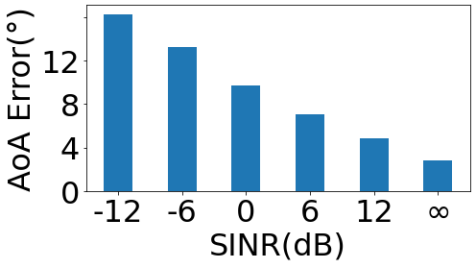


Figure 4.3: AoA error under different SINR

to unexpected speech from an unknown direction. To quantify the impact of interference, we apply MUSIC to the audible signals under different signal-to-interference and noise ratios (SINR) by scaling the magnitude of interference and mixing it up with target speech. As shown in Figure 4.5(b), adding interference increases the AoA error significantly. For example, the AoA error increases by 10.45° over no interference when SINR = -6 dB; the error increases by 13.53° when SINR = -12 dB. While using masks removes a significant amount of interference, the removal alone is insufficient to support accurate

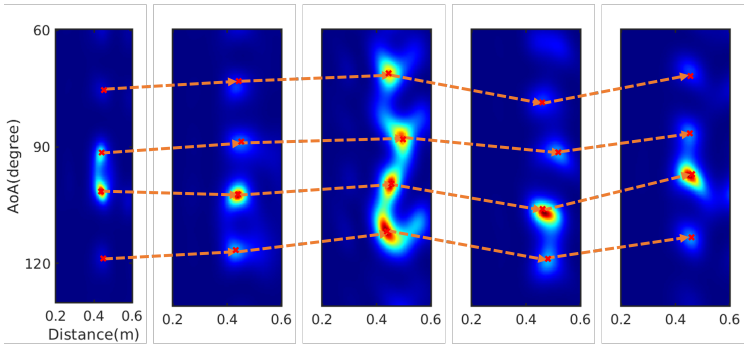


Figure 4.4: 2D MUSIC profiles to capture multiple reflections from the human body every 0.2s. Each peak corresponds to one reflection point. They have similar moving trends.

AoA estimation.

Besides, most energy in speech concentrates in low-frequency bands (*e.g.*, below 2 kHz), which have large wavelengths and lead to low AoA resolution [152]. Furthermore, typically a relatively large time window (*e.g.*, hundreds of ms) is used to analyze audible signals in order to ensure there is enough energy from the target speaker. This limits the update rate of multipath profile estimation. The temporal resolution of TF bins is 10 ms, but the AoA update rate cannot keep up with the TF bin update rate. We need more frequent multipath profiles encoded by the AoA estimation.

In comparison, inaudible signals are not affected by the interference and noise in the audible band. It can work as a side channel to detect and track the target speech. Meanwhile, inaudible signals have shorter wavelengths so that they can achieve higher AoA resolution. Finally, inaudible signals are modulated at the transmitter end. It can be designed as a small duration

to improve the estimation rate. Hence, inaudible signals enable tracking at a much higher frequency (*e.g.*, tens of ms), which is important to adapt more quickly to the changing user position.

However, in the typical usage scenarios where the speaker and microphones are on the desk, inaudible signals are mostly reflected by the user's body instead of the mouth. Therefore, inaudible signals mainly track body movement instead of lip movement. Besides, there are multiple reflections from the human body. It is challenging to distinguish which parts of the human body reflect the inaudible signal. Fortunately, body movement is highly correlated with mouth movement. As shown in Figure 4.2, the AoAs estimated using inaudible signals follow a similar moving trend to the ground truth even though its absolute AoA differs from the ground truth. Moreover, even though we may see reflections from different body parts, they tend to have a similar trend as the entire body moves together. Figure 4.4 shows multiple reflections from the human body over time. As we can see, these reflections share a similar moving pattern over time. Thus inaudible signals can capture the motion and help correct inaccurate AoA estimation from the noisy audible bands. There may be multiple AoA candidates from inaudible reflection (*e.g.*, due to reflection from multiple body parts), but they have similar movement trends, which are useful for tracking and source separation. We develop a neural network to automatically exploit the features extracted from both audible and inaudible signals.

Feature extraction with 2D MUSIC

We apply 2D MUSIC to the inaudible reflections to extract the dynamic motion of humans because of the high resolution of 2D MUSIC. In the free space, the peak in the 2D MUSIC profile indicates the target AoA and distance. In practice, signals traverse over multipath, which may introduce multiple peaks. If the two paths are too close, their resulting peaks can get merged and the highest point in the peak may not correspond to the target location. Noise and Doppler shifts resulting from movement complicates the 2D MUSIC profiles by adding false peaks and distorting the real peak.

Therefore, instead of selecting a single peak for AoA and distance estimation, which may introduce an error to propagate to further stages, we feed the complete 2D MUSIC profile from inaudible tracking along with the MUSIC profile from voice signals to generate location embeddings. The two profiles operate on different frequency bands. They do not affect each other. While the audible profile captures the voice source in a lower resolution due to the smaller operating frequency, the inaudible profile encodes the overall human motion in a higher resolution. Hence, both profiles can compensate for each other and benefit the final AoA estimation.

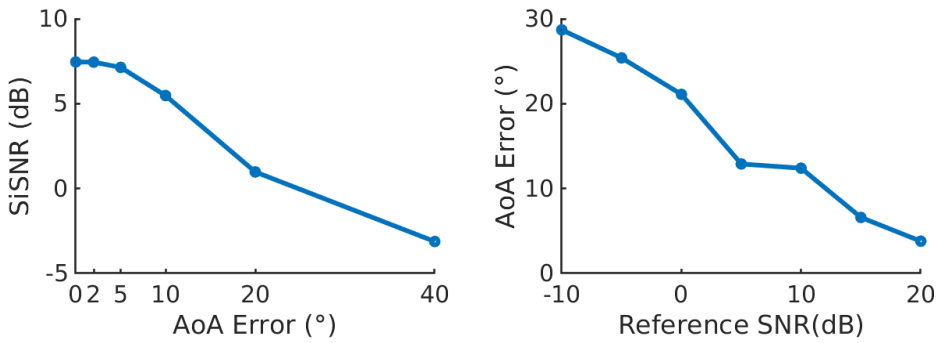
Generating Spatial Embeddings Each MUSIC profile can be considered as an image, and a sequence of MUSIC profiles can thus be treated as a video sequence Therefore, we can take advantage of state-of-the-art video analytics. We apply 3D convolution [142] with a $5 \times 7 \times 7$ kernel and ResNet-18 [46] to MUSIC profiles generated from audible and inaudible signals to effectively learn spatio-temporal features. The small temporal window of 3D kernels

helps filter out noisy patterns based on neighbor frames. It is followed by batch normalization and ReLu activation. Then output features are fed to ResNet18 to encode profile embeddings with 512 dimensions. The concatenated embeddings from audible and inaudible profiles will be directed to the source separation.

A sequence of embeddings is extracted from the MUSIC profile sequence. We feed them into both LSTM and source separation networks. LSTM takes these embeddings to estimate AoAs because they include important spatial information from audible and inaudible signals over the recent time window. The 3D convolutional layers use temporal information in small time windows at an early stage, while the LSTM can leverage a much longer sequence. Moreover, audible embedding and inaudible embedding complement each other due to the correlation between mouth movement and body movement. The memory unit in LSTM helps track long-term movement. Our objective is to minimize the L1 loss between the estimated AoA ($A\hat{o}A$) and ground truth AoA (AoA), denoted as

$$L_{AoA} = ||AoA - A\hat{o}A||_1 \tag{4.9}$$

We use LSTM to estimate the AoA based on the audible and inaudible location embeddings. Each LSTM cell is a fully connected network with 128 input nodes and 1 output node, which is the estimated AoA. There is 1 hidden layer with 64 ReLU nodes.



(a) The reduced AoA error improves the performance of MVDR beamformer

(b) The increased SNR of reference speech decreases the AoA error in GCC-PHAT

Figure 4.5: AoA and separation benefit each other in a signal processing view

4.2.3 Multi-task Learning for Source Separation

So far, we have focused on localizing the target user. Next, we consider how to leverage the location information for source separation. Due to the strong inter-dependency between tracking and source separation, we apply a multi-task learning framework to jointly estimate the location and separate the source.

A signal processing perspective on AoA estimation and speech separation We revisit two typical signal processing algorithms to show how the separation and the AoA can benefit each other.

Beamforming is a pure source separation method with spatial information. The goal of source separation is to minimize the difference between beamformed speech $\tilde{s}$ and target speech s. In the formula 4.8, AoA is one of the key factors to decide the beamforming weights. Figure 4.5(a) shows

that a more accurate AoA estimation yields better beamforming and source separation.

On the other side, separation can benefit the AoA estimation as well. As the formulation 4.6 shows, a clean reference signal can estimate a more reasonable cross-correlation, which results in a better delay estimation and infer a more accurate AoA. Figure 4.5(a) shows that an improved reference speech from the source separation reduces AoA estimation error in the end.

In the blind speech separation task, both AoAs and target speech are unknown. Therefore, we propose to learn the speech separation and AoA estimation jointly from the received acoustic signals. Our goal is to optimize both objectives to improve overall performance.

Learnable Pre-mask Previous works [24] develop a pre-mask to take into account of AoA θ. It first forms a steering vector $\boldsymbol{e}_\theta(f)$ based on the AoA, and computes the cosine distance between the steering vector and the complex values in each TF bin as follows:

$$A(t,f) = \sum_{k=2}^{M} \frac{\boldsymbol{e}_{\theta,k} \frac{\boldsymbol{Y}_k(t,f)}{\boldsymbol{Y}_1(t,f)}}{|\boldsymbol{e}_{\theta,k} \frac{\boldsymbol{Y}_k(t,f)}{\boldsymbol{Y}_1(t,f)}|} \tag{4.10}$$

where $\boldsymbol{Y}$ is the complex spectrogram, M is the number of microphone channels, and k is the microphone index starting from the second microphone as the steering vector is normalized to the first microphone. $A(t,f)$ represents the pre-mask value to a TF bin. The pre-mask indicates the probability of a TF bin dominated by the source coming from the given AoA. Intuitively, the pre-mask lets the network beamform towards a given direction. Pre-mask improves

over traditional linear beamformers (*e.g.*, MVDR [14] and LCMV [91]) by using a DNN-based non-linear filter, so it has better spatial discrimination and interference cancellation.

The pre-mask assumes the input AoA is accurate. In our context, the AoA estimation can be erroneous due to interference, ambient noise, and multipath propagation. Moreover, not only the direct path but also reflected paths are important for source separation because the overall received phase is the result of all multipath. A single AoA estimate does not provide complete spatial information of the target speaker. Therefore, we propose to fuse our spatial embeddings with the mixed phase from the complex spectrogram to learn a better spatial pre-mask.

For each microphone and TF bin, there is a 512-long embedding from audible profiles and another 512-long embedding from inaudible profiles. These embeddings are concatenated and processed by a 1x1 convolutional layer followed by a layer normalization and PReLU. The output of each TF contains a spatial feature map. It is concatenated with LPS and fed into Temporal Convolution Network (TCN) [80]. TCN outputs a mask, which can be applied to the mixture complex spectrogram to generate the target complex spectrogram. Then we perform an inverse short-term Fourier Transform to estimate the target signals.

Multi-task learning objective A common learning objective in the existing separation network is to maximize Scale-invariant Signal-To-Noise Ratio

(SiSNR) [80]. Let $\hat{x}$ denote the estimated signal and x denote the clean reference signal. We compute SiSNR as follow:

$$SiSNR = 10log_{10}\frac{||x_{target}||^2}{||e_{noise}||^2} \tag{4.11}$$

where

$$x_{target} = \frac{<\hat{x}, x> x}{||x||^2} \tag{4.12}$$

$$e_{noise} = \hat{x} - x_{target} \tag{4.13}$$

By normalizing x and $\hat{x}$ to zero mean, we ensure scale invariant. The loss function is defined as $L_{SiSNR} = -SiSNR$. Following the existing work (*e.g.*, [133]), the target and interference signals are measured separately and added up to simulate interference. Therefore, SiSNR can be computed based on their values.

Unlike the existing works that optimize only SiSNR, we develop a novel multi-task learning framework to jointly learn speech separation and AoA. Multi-task learning trains ML models for multiple tasks simultaneously using a shared structure. The idea of multi-task learning is that internal representations learned for one task can be helpful for the other tasks, and vice versa.

Our key observation is that spatial embeddings can benefit both speech separation and AoA estimation. An accurate embedding enables LSTM to accurately estimate the AoA. It also provides good hints for TF bins, which will be fused with spatial knowledge and target speaker direction.

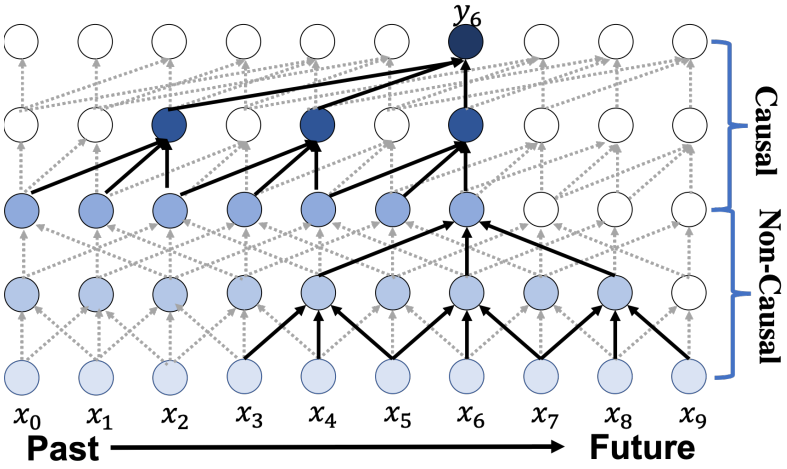


Figure 4.6: Stack causal and non causal convolution layers.

Another important insight is that jointly learning the separation and the AoA can reinforce the network to learn the AoA instead of treating the AoA as the fixed input, which prevents the gradient from propagating back and contributing to the training task. In comparison, when the AoA is set to be learnable together with separation, the mixed phase can contribute to the learning objective. By fusing the phase and embedding, the learned phase is more consistent with the separated source. The phase is represented as a 2D tensor, which represents the phase over different microphones and frequencies. We use more convolutional layers to learn the phase using the following objective for training:

$$L = L_{SiSNR} + \lambda L_{AoA}$$

where λ is a relative weighting factor and is set to 0.5 in our evaluation.

4.3 Implementation

Real-Time Model Design While several source separation schemes claim to achieve real-time inference because their processing time is shorter than the audio duration (*e.g.*, 1.71s processing time for 4s audio in Conv-TasNet[80]), this is insufficient to ensure real-time. Another requirement is that the output should be generated without much delay. This has significant implications for the neural network structure. In particular, existing works use non-casual convolutional layers, which do not distinguish between past and future input samples and require many future samples (*e.g.*, 1.28 seconds in Conv-TasNet).

In order to support real-time processing, we only use a small look-ahead window (*e.g.*, 90 ms in our implementation). We show that even a small look-ahead is sufficient to yield good performance. Moreover, we pay special attention to causal vs. non-casual convolution. The top 2 layers are causal convolution in Figure 4.6. They only require previous samples. The bottom two layers are non-causal convolution which is common in most network architecture. They need future samples to perform convolution operation. The stack of non-causal layers increases the perception field of future samples exponentially. In order to achieve high accuracy while limiting the latency, we configure the first 2 convolutional layers of each TCN block as non-causal with a small look-ahead and the other layers as causal. Figure 4.6 shows an example: at timestamp 6, the first two non-casual layers use samples up to timestamp 9 and the next two casual layers only use samples up to the current timestamp 6.

Model Implementation We implement the SAMS model in Pytorch [3]. We use the Adam optimizer with an initial learning rate of 0.0001. We apply a multi-step scheduler to drop the learning rate by 50% at epochs 40 and 75. The maximum training epoch is 150 but it will stop early if there is no more improvement for 10 epochs. SAMS has 20.2M parameters in total.

To provide a real-time guarantee, we cannot wait to accumulate a few seconds of audio before processing, but the process more frequently (*e.g.*, at least once every 150 ms). But the processing time does not reduce proportionally with the reduced input size due to the lack of batching opportunities. Hence how to achieve real-time ML-based source separation remained open.

To speed up processing, our system processes audio every 90 ms. We introduce a cache tensor for each block to cache the previously computed intermediate result in the neural network and reuse it in the next round. Moreover, we use Microsoft Onnxruntime [99] to significantly speed up the inference. Since certain operations are not supported in Onnxruntime, we replace these operations with similar but supported operations. Together, the resulting system achieves real-time processing – it processes 90 ms audio within 42 ms, which yields the total latency of $90 + 42 = 132$ ms (within the 150 ms real-time requirement).

Setup: As shown in Figure 4.7, we connect a laptop with a Bela platform [13] attached with a pair of speakers and four microphones. The microphones form a linear array spanning 8 cm with non-uniform space between them. Their

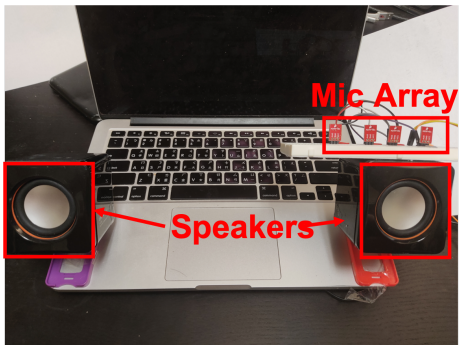


Figure 4.7: Platform setup

positions are [0, 3 cm, 5 cm, 8 cm]. A similar setup is used in [87]. Note that certain mobile devices have a similar setup.

For example, Apple Macbook Pro[7] places three mics on the right up of the keyboard. Huawei Matebook 14s[52] places a front-facing quad-mic array to improve voice quality. Thus, our prototype does not bring extra complexity to the hardware design. The processing pipeline is done on the laptop using Pytorch. We train the model on NVIDIA GTX 2080ti GPU. We run inference on both the desktop and Macbook.

Data collection: There are no open-source multi-channel recordings for our evaluation. Therefore, we collect the data on our own. We will release the data to the public. Our data include multi-channel inaudible and audible audio signals. We let dual speakers both transmit periodic FMCW chirps from 18-20 kHz with a period of 40 ms and a sampling rate of 44.1 kHz. To avoid interference, the two speakers transmit the same chirp with a 20 ms difference in the starting time. The volume of the speakers is set to a little

less than the maximum to avoid signal distortion. We set another headset microphone to record the reference clean speech.

We collect 20 users' speeches using our setup. Among them, there are 8 females and 12 males. There are two kids: 8 and 13 years old, and the rest are adults between 22 – 59 years old. Each user speaks for 10 minutes - 1 hour. We let the users present slides or read books or papers to mimic online conferences. This is an easy way for users to generate continuous speech. The target user is 0.2–0.7 m away from the microphone. The users move naturally during the trace collection. For example, they sometimes lean towards or away from the computer, move side to side, or turn their heads. We collect the data from different environments (*e.g.*, lab, living room, study room, cubicle, and conference room). The environments have different multipath, which affects both AoA estimation and source separation.

We also separately record interference by letting an external speaker play a random subset of speech (*e.g.*, around 5 hours) from Librispeech [102], which contains more than 1K speakers and 26K English sentences lasting 1000 hours. We place the interfering speaker inside or outside the room where the target user is located. When the interfering speaker is inside the room, (s)he is a couple of meters away. We augment the real traces by scaling the SNR of the target signals from -6 dB to 6 dB. Moreover, we use gpuRIR [33] to simulate realistic interference and noise in multi-channel scenarios by estimating and applying Room Impulse Responses (RIR) to clean speech from Librispeech and noise from WHAM! [159]. WHAM! collects ambient noise in non-stationary

environments, such as coffee shops, restaurants, and bars. These sounds are generated by humans, musical instruments, and vehicles. Adding such noise to the background interference makes it even more difficult to extract clean signals.

Dataset Preparation: We mix the audio segments containing the target speaker's speech and inaudible FMCW reflection with different types of interference and background noise. We add different interference and noise to each target user's speech. We vary the amount of interference and background noise according to the required SNR. The number of interfering users is uniformly distributed between 0 and 3, and the SNR is uniformly distributed between $[-6, 6]$ dB. In total, the training data is generated from 16 users' speech. It contains 30K segments of mixed audio signals, where each segment lasts for 4 seconds and the total training data lasts for 31 hours. The testing data contains 6K segments generated from 4 users. Following the common practice, we vary the user in the testing dataset and use the remaining user for training. Both training and testing have real recording samples from all environments. Interference and noise are from the training split and test split of LibriSpeech and WHAM! respectively. In addition, we also evaluate how our model generalizes to a new environment that is not present in the training traces.

4.4 Evaluation

In this section, we first present our evaluation methodology and then describe our performance results.

4.4.1 Evaluation Methodology

Performance metrics: Following the existing works (*e.g.*, [133]), we use several metrics to quantify the performance of source separation: (i) **SiSNR** prevents unfair impact of the rescaled signals [78]; (ii) **Short-time objective intelligibility measure (STOI)** quantifies intelligibility of speech [138]; (iii) **Perceptual Evaluation of Speech Quality (PESQ) [1]** is designed to quantify the quality of processed speech, and its score ranges from 1 to 5. Higher values in the above metrics indicate better speech quality.

In addition, we also report AoA estimation errors. We measure the ground truth AoA using the Intel RealSense L515 [59]. To ensure the RealSense gets accurate AoA, we place it in line with the microphone array and provide good lighting conditions.

Baselines: We compare SAMS with the following state-of-the-art approaches: (i) Conv-TasNet [80]: It is one of the best speech separation approaches using single-channel speech. It is also one of the most widely used baselines due to its open-source. (ii) PHASEN [174]: It is a denoising network using two streams to improve phase estimation. UltraSE [133] shows that [80] and [174] are the best baselines that only use speech for source separation. All schemes

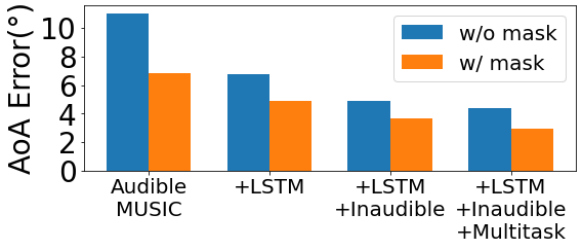


Figure 4.8: AoA Error of Different Variants .

are trained using the same data as our approach.

We did not compare with UltraSE [133], which targets phone users. We target computer users for online meetings, which is complementary to UltraSE and also more common than smartphone users (*e.g.*, [147] reports the majority of users use computers for online meetings). Moreover, UltraSE requires the phone's speaker/mic to face the user's mouth and be within 20 cm, which is even less common as most speakers/mics on the phone face bottom instead of the user. Meanwhile, our measurement shows that the headset Sennheiser DK-2750 improves SiSNR by 8.91 dB over the internal microphones of the laptop under interference. In comparison, our software-only solution provides higher SNR and hence is more attractive.

4.4.2 Results

We first present micro benchmarks, where we compare different variants of our own algorithm. In the micro-benchmark, we use one interfering user and background noise setup. Unless otherwise specified, we use the data collected from all environments, 20 users, and SNR range of -6dB to 6dB as the default

settings.

Impact of AoA estimation algorithms: We first compare different variants of our AoA estimation. Figure 4.8 plots the average AoA estimation error. The basic method is to apply standard wideband MUSIC to estimate the AoA (*i.e.*, applying MUSIC to each frequency band and summing up the results across all bands). We then augment the method with various enhancements. As it shows, each of our enhancements, namely mask, LSTM, inaudible tracking, and multi-task learning, helps improve the AoA error. Using a mask reduces the AoA error by $0.5° - 4.2°$ across different cases by removing the noisiest TF bins to prevent generating incorrect MUSIC spectrum. Using LSTM brings an additional 1.9° improvement over using the MUSIC profile in a single period since it leverages the inherent temporal locality in the movement. Using inaudible tracking further reduces the AoA error by 1.2° by overcoming audible noise and interference and updating the location more frequently. Finally, multi-task learning improves the AoA by another 0.7° through jointly optimizing the source separation and AoA estimation. Putting everything together, we achieve 3.8° AoA estimation error.

Impact of different separation algorithm: Next we compare the following source separation algorithms using the same set of data: (i) LPS [75]: It uses only a single channel LPS of raw mixed audio signals for source separation. (ii) MVDR [14]: We estimate the AoA by applying MUSIC to audible signals, and use MVDR to beamform towards the AoA. (iii) Est AoA: We estimate the AoA using both audible and inaudible AoA and use our source separation

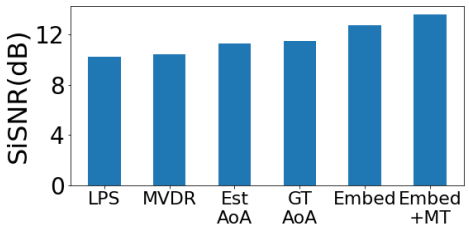


Figure 4.9: Different model structure

DNN to extract the target signals. It differs from our final algorithm in two aspects: (i) it uses AoA for source separation instead of location embeddings, and (ii) it disables multi-task learning. (iv) GT AoA: It applies our source separation network to the ground truth AoA. Like the above estimated AoA, it disables multi-task learning and uses AoA instead of location embeddings in source separation. (v) Embed: It uses spatial embeddings but disables multi-task learning in our final approach. (vi) MT: This is the final approach that uses both location embeddings and multi-task learning.

As shown in Figure 4.9, on average, LPS yields 10.26 dB SiSNR by leveraging the magnitude information to learn the acoustic model of speech. MVDR beamforming achieves 10.43 dB SiSNR by leveraging the AoA estimate from audible signals. The improvement is limited due to the limited accuracy of AoA estimation. By leveraging LSTM and inaudible signals, the estimated AoA yields 11.33 dB SiSNR. Using the ground truth AoA achieves 11.52 dB SiSNR. Using location embeddings improves SiSNR to 12.74 dB, which is 1.22 dB higher than using the ground truth AoA. This is because the ground truth AoA only provides information about the direct path, but multipath information is also useful for beamforming. Further incorporating multi-task

learning increases the SiSNR to 13.61 dB by jointly optimizing AoA and source separation. These results demonstrate each component in our system is useful and leveraging them all provides the best performance.

Impact of sensing using inaudible signals: Figure 4.8 shows that using inaudible signals together with audible signals decreases the AoA error by 1.8° and 1.2° over audible signal based sensing without mask and with mask, respectively. The reduced AoA error also translates into improved separation performance. Table 4.1 shows that SiSNR decreases 0.6dB, 0.83dB and 0.54db without inaudible information for three different noise and interference setups. While inaudible signal based sensing is useful, it alone (denoted as SAMS (w/o audible)) performs less well. These results confirm that combining audible and inaudible signals for sensing yields the best performance.

We compare the overall performance of SAMS with several existing source separation methods by varying the background interference, users, SNR, and environments.

Vary interference: Following UltraSE [133], we compare our algorithm with Conv-TasNet and PHASEN under different numbers of interfering speakers and noise. All schemes are trained using the same data. Note that SAMS and PHASEN only require the target signal for training, whereas Conv-TasNet requires both the target signal and interference for training. To support multiple interferers, it takes the total interference from all interferers as the ground truth output for training. Table 4.1 summarizes the performance in terms of

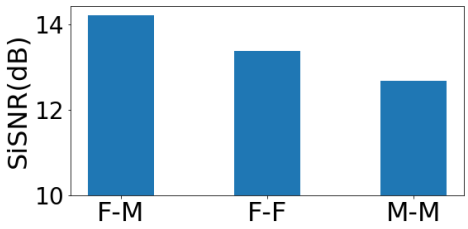


Figure 4.10: Different gender combinations

SiSNR, PESQ and STOI. As it shows, our algorithm improves over Conv-TasNet and PHASENby 5.00 dB, and 1.39 dB, respectively, under only ambient noise. The corresponding numbers become 3.39 dB, and 9.58 dB respectively, under 1 interfering speaker with ambient noise; and become 5.01 dB, and 8.10 dB, respectively, under 2 or more interfering speakers and ambient noise. The larger improvement over the existing approaches under interference is owing to the spatial embeddings learned from both audible and inaudible signals and multi-task learning. Conv-TasNet cannot perform well with only noise or more interference. PHASEN can deal with phase distortion caused by ambient noise, but cannot handle interference well. SAMS can outperform all of them even in their target scenarios. SAMS also achieves better PESQ and STOI as it reduces the phase distortion.

Vary gender pairs: Next, we consider the impact of different sets of users speaking at the same time. Conceptually, female and male voices are different and have distinct resonant frequencies, so they are easier to separate out. However, humans with the same gender can have a lot of similarities on the voice and accent. This is confirmed in our evaluation. As shown in Figure 4.10, the SiSNR of separating from the female-male(F-M) pair is 14.21 dB. The SiSNR

Environment	Model	SiSNR	PESQ	STOI
noise	SAMS	**10.71**	**2.21**	**0.76**
	SAMS(w/ GT AoA)	9.51	2.09	0.71
	SAMS(w/o inaudible)	10.11	2.15	0.74
	SAMS(w/o audible)	6.32	1.83	0.74
	Conv-TasNet	5.71	1.76	0.60
	PHASEN	9.32	2.09	0.70
1 interferer + noise	SAMS	**13.61**	**2.68**	**0.84**
	SAMS(w/ GT AoA)	11.52	2.49	0.79
	SAMS(w/o inaudible)	12.78	2.54	0.81
	SAMS(w/o audible)	9.93	2.14	0.77
	Conv-TasNet	10.22	2.19	0.76
	PHASEN	4.03	1.60	0.53
2 or 3 interferers + noise	SAMS	**12.21**	**2.44**	**0.78**
	SAMS(w/ GT AoA)	10.33	2.38	0.75
	SAMS(w/o inaudible)	11.67	2.41	0.77
	SAMS(w/o audible)	7.01	1.92	0.62
	Conv-TasNet	7.20	1.96	0.65
	PHASEN	4.11	1.69	0.53

Table 4.1: Performance across various interference and noise scenarios

of separating from the female-female(F-F) and male-male(M-M) is lower, but still quite high: 13.37 dB and 12.67 dB, respectively. These results show that even when the interference is similar to the target signals, SAMS can still separate out the signals by taking advantage of location embeddings.

Vary SNR: Then we vary the SNR of the target user from -6dB to 6dB by scaling the target signal. Each subset of a specific SNR includes all linear combinations of interference speech and noise. As shown in Figure 4.11, SAMS outperforms Conv-TasNet and PHASEN in all SNR scenarios by about 3dB and 5dB, respectively. Even for the low SNR case, SAMS can separate the weaker target speech and improve SISNR to 7.09dB, which is sufficient for

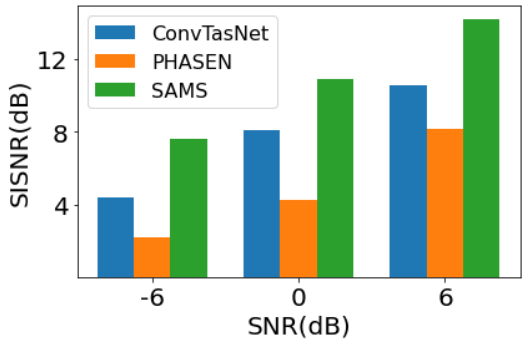


Figure 4.11: Different SNR

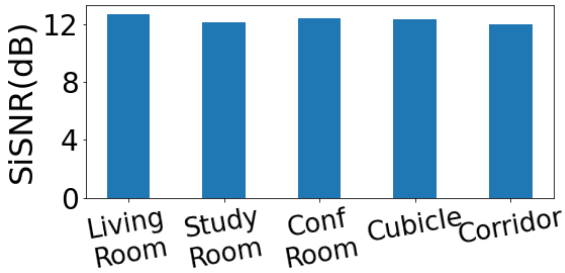


Figure 4.12: Different Environments

good audio quality in an online meeting.

Vary environments: The target user speeches are collected from 5 different environments: living room, study room, cubicle, conference room, and corridor. Different environments have different room structures and furniture, which results in various room impulse responses. It can affect the performance of both AoA estimation and source separation. We train our DNN using the data collected from all environments except the one used for testing. Figure 4.12 plots the source separation performance for the new environment that is not in the training. As it shows, SAMS generalizes well to the new en-

vironment and its performance is fairly stable across different environments. Because we incorporate the audible profile and the inaudible profile to combat the change in the environment, SAMS can be more robust than the audio-alone method.

Computation cost: We run inference on three platforms to quantify the computation cost: a desktop GPU (NVIDIA GTX 2080 Ti), a desktop CPU (Intel Core I7-8700K), and a laptop CPU (Intel Core I5-5257U). These platforms represent common devices for online meetings. Processing 4-second audio takes 61ms on the desktop with GPU, 0.41s on the desktop with CPU, and 1.31s on the laptop with CPU using Pytorch without any optimization. With optimization through caching intermediate results and Onnxruntime[99], our system can process 90-ms audio within 42 ms on the laptop with only a CPU. The total latency is 132ms, which is smaller than 150 ms (*i.e.*, the target latency requirement for VoIP), hence achieving real-time processing.

Chapter 5

Visual Timing For Sound Source Depth Estimation

5.1 Background of Depth Sensors

[1] We investigate a variety of depth sensors and put FBDepth into perspective by comparing them with the other approaches using several basic criteria. Due to a large number of depth sensors and methods, we only select the typical methods of each category of sensors and describe the fundamental strength and weaknesses.

We classify the existing approaches broadly into two main categories: active sensors in Table 5.1 and passive sensors in Table 5.2. The active sensor can actively emit modulated signals, such as LiDAR, structured light, radar, ultrasound, WiFi, RFID, or mmWave. They use time-of-flight (ToF), amplitude, phase, or Doppler shift to estimate the range because these physical measurements are directly determined by the target depth. Passive sensing uses signals from the environment for sensing. It commonly uses a monocular

[1]The work in this chapter was supervised by Prof. Lili Qiu. I was the first author and made contributions to designing research, performing research, analyzing data and writing the paper. It was originally published in: Sun, Wei, and Lili Qiu. Visual-Assisted Sound Source Depth Estimation in the Wild. arXiv preprint arXiv:2207.03074 (2022).

camera [15] or a stereo camera [130] by leveraging the implicit monocular or binocular depth clues from images. They commonly apply an indirect strategy to regress the depth.

Accuracy: Active sensing can achieve centimeter-level accuracy based method because their fundamental is based on the physical measurement and the wavelength of sensing signals is centimeter-level or even smaller. WiFi is an exception as its wavelength is decimeter-level.

Many passive sensing schemes use deep learning to achieve SOTA of depth estimation. However, the accuracy is not comparable to active sensing. Monocular camera based depth estimation [175, 69] can achieve 5.2% AbsRel in the Kitti dataset[43] and within 10% in NYU-Depth V2 [95]. However, its absolute error (AbsErr) increases linearly with the distance. The AbsErr can be several meters when objects are tens of meters away. Moreover, it relies on high-quality datasets and cannot handle arbitrary scenes and fields of view, which may not contain enough clues for depth estimation. Stereo depth performs superior to monocular depth because it transforms the spatial depth into pixel difference between left and right images. This triangulation transformation causes the accuracy to depend on the baseline between two cameras [25]. A larger baseline results in more pixel differences to the same depth. The triangulation makes the AbsRel increase with further depth.

Range: LiDAR and radar for autonomous driving can achieve up to hundreds of meters which depends on the signal design and frequency band. Other in-

Sensor	Device/Method	Accuracy	Range	Angular Resolution	Power	Cost
LiDAR	Velodyne HDL-32E [12]	2cm	100m	1.33°(V) 0.1°-0.4°(H)	10W	>$5K
structured light	Realsense D455	2%	6m	pixel-level	3.5W	$400
ToF camera	Azure Kinect [182]	< 1cm	6m	pixel-level	5.9 W	$600
mmWave	Navtech CTS350-X [12]	4.38cm	163m	1.8°	20w	>$500
inaudible sound	Rtrack [87]	2cm	5m	object-level	0.5W	<$10
WiFi	Chronos [146]	65–98 cm	$< 50m$	object-level	<10W	< $50

Table 5.1: Active sensors or signals. Among these metrics, it is better for accuracy, range, and resolution to be high while it is promising for power and cost to be low.

door estimation methods can achieve an effective range of about 6m. Monocular depth estimation is fully dependent on the depth range of the dataset. It works well indoors when it is trained by NYU-Depth V2 [95] and performs outdoors when trained by Kitti dataset[43]. Monocular depth achieves 1000 meters [117] with specific depth labeling. The baseline of the stereo camera determines the max depth as well because the view difference becomes tiny when the object is too far. FBDepth can support 60 m. Its large range comes from two major factors. First, it uses directly received audio signals instead of reflected signals. Besides, it uses audible frequencies, which have a slower decay and stronger frequency response than inaudible audio frequencies.

Angular Resolution: The resolution refers to the granularity that each depth measurement point corresponds to the 3D space. RGB-based methods,

Sensor	Device/Method	Accuracy	Range	Angular Resolution	Power	Cost
camera	NeWCRFs[175]	NYUv2[125]: 9.52% KITTI[43]: 5.20%	10m 80m	pixel-level	<1W	<$30
stereo camera	ZED 2i[130]	< 2% up to 10m; < 7% up to 30m	40m	pixel-level	2W	$450
camera + mic	FBDepth	overall 2.98%; > 30m: 2.41%	60m	obj-level	<1W	<$30

Table 5.2: Passive sensors or signals

ToF cameras, and structured light can achieve pixel-level estimation. It can be transformed to a numerical resolution by dividing the field of view by the image resolution. RF-based and acoustic-based solutions can only detect sparse reflection points. The LiDAR with multiple beams can generate a dense point cloud, the density of its point cloud decreases quadratically with the distance [163] and its point cloud becomes pretty sparse at a large distance. As a result, autonomous driving datasets commonly annotate point clouds up to 80m [18, 43]. Further points are difficult for humans to distinguish and label.

Power: Active sensing methods need to emit modulated signals, which tend to consume more power in order to get a reasonable SNR from the reflected signal over a large detection range. When they are applied to mobile phones, their performance is constrained by power. For example, iPhone Pro has a Lidar on the back of the phone with a max range of 5m and a ToF camera on the front with a max range of 50cm. In comparison, passive sensing does not require active transmission and consumes less power, which is desirable

for mobile devices.

Cost: Cameras and microphones are widely available. In comparison, LiDAR is much more expensive and not available on most mobile devices. mmWave also has limited deployment. While WiFi is popular, in order to achieve high accuracy, we need more spectrum and PHY layer information, both of which limit its availability on mobile devices. FBDepth only requires a camera and a microphone, which makes it possible for wide deployment.

Summary: There are many depth estimation technologies. Our FBDepth complements the existing solutions by adding a low-cost, easy-to-deploy, accurate, and long-range solution. It has higher accuracy than existing monocular solutions and a longer range than existing stereo solutions.

5.2 Approach

5.2.1 Audio-Visual Correlation

We identify the audio-visual correlation as the key component to building up the depth estimation framework. Audio-visual perception is important for humans to learn from and interact with the real world. Existing work shows more interest to incorporate audio and visual to boost enhanced semantic representation learning. Apart from existing audio-visual correlations, We identify novel audio-visual correlations to enable depth measurement in the wild.

Audio-visual Semantic Correlation Audio-visual learning has become a

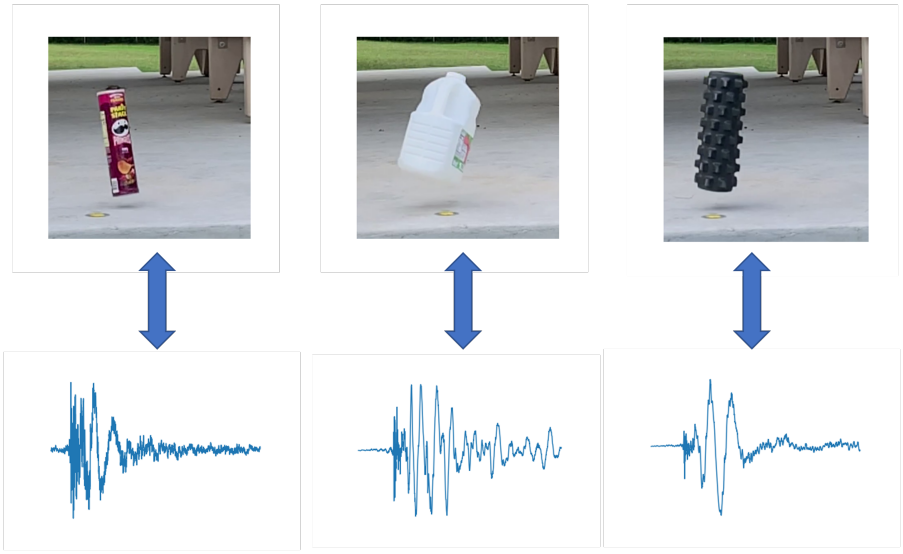


Figure 5.1: Audio-visual semantic correlation: Each object has the unique visual appearance and sound

popular trend in computer vision, speech, and robotics. The fundamental idea is to advantage of the coexistence of audio cues and video appearance of a specific event. In Figure 5.1, each object can generate the unique sound. Humans can quickly pair the image and sound by observing both modalities. This learning scheme has several attractive benefits. First, the coupled audio features and visual features can be paired and enriched to learn a more general and robust feature embedding compared to using either modality. Besides, it can combat the noise and interference from both modalities. Hence, audio-visual learning is applied to enhance the performance of analytic tasks and enable novel applications. by focusing on the semantic correlation between the two modalities and associating visual representations and audio representations.

In the audio-visual depth estimation, we are interested in a specific

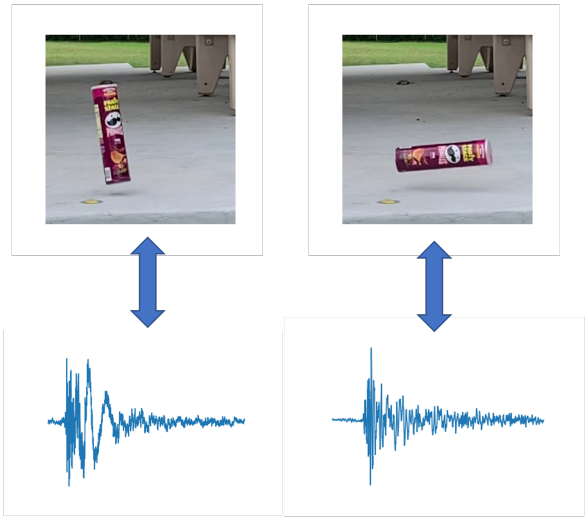


Figure 5.2: Audio-visual semantic correlation: Different motions yield various sound for same object

set of audio-visual events where the impact sound is triggered by a collision. These events are common in nature. For example, an object is freely falling or bounced by a racket or wall, or a person is stepping on the floor. Collisions follow a significant semantic correlation between audio modality and visual modality in two folds. The basic correlation is that each object can produce a unique sound by collision based on its size, material, shape, etc. The visual appearance can even be applied to synthesize the sound with one image[101].

Besides, we introduce a new type of audio-visual semantic correlation. The motion of an object is another key factor to generate the audio. The velocity of the object determines the magnitude of the sound. We also find that different motions result in various collision positions, which brings various sounds as shown in Figure 5.2. By observing the consecutive frames, we

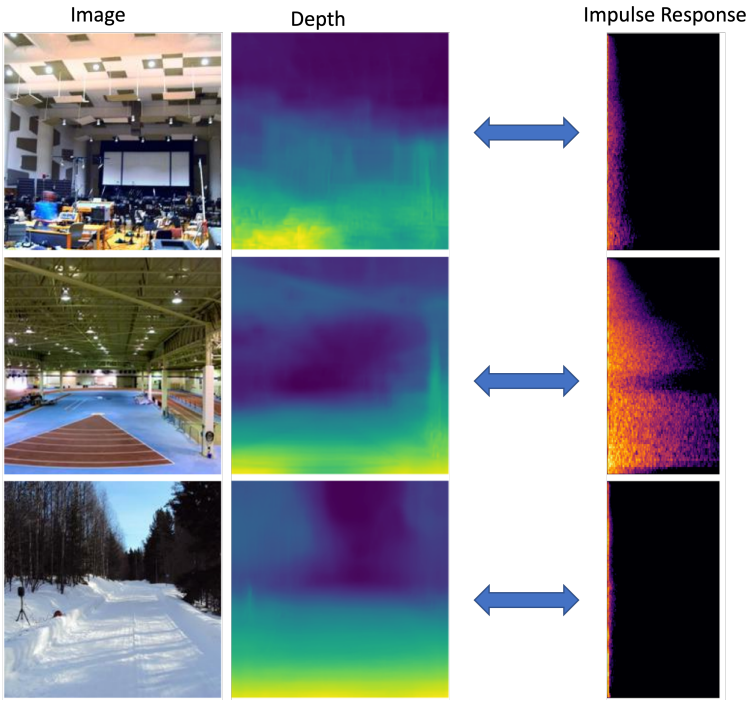


Figure 5.3: Audio-visual spatial correlation: different environments result in various impulse response. [127]

can capture the motion dynamics and predict a reasonable impact sound. Therefore, we leverage both the visual appearance and the motion from the video to correlate with sound for depth estimation.

Audio-visual Spatial Correlation: [41, 105, 21] introduces a sort of indirect spatial correlation. In Figure 5.3, they claim that impulse response(IR) varies a lot in different room structures and placement of furniture results. They try to infer the IR with the visual semantic context of the room, such as the structure, dimension, room types, materials, scatter reflectors, etc. However, it involves too many uncertain variables in the environment such as absorp-

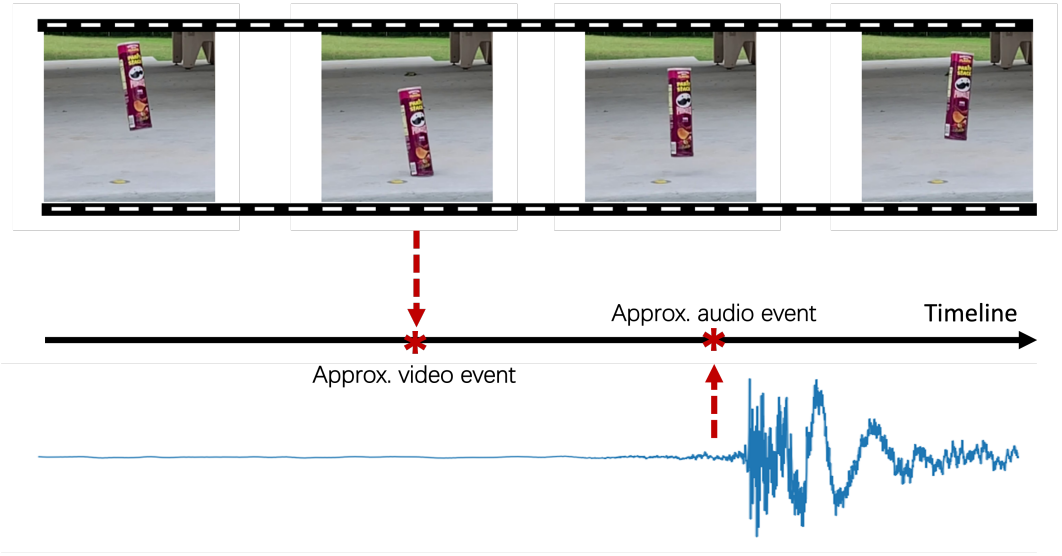


Figure 5.4: Audio-visual spatial correlation: significant propagation delay between light and sound

tion coefficients of materials, and the poses of reflectors. The formulation of this spatial correlation cannot be easily quantified. As a result, the indirect correlation is too weak to be learned with the controlled simulation data. The improvement of depth estimation is pretty limited.

We introduce a novel straightforward audio-visual spatial correlation in Figure 5.4. In a synchronized audio-visual recording, the collision is always observed earlier in the video than in the audio. It can be formulated by the physical laws easily as shown in Equation 5.1. We sense the delay between the audio event and the video event based on the difference between the propagation speeds of light and sound. The delay is perceptible if the event happens far away, such as a lightning strike, a flying plane, and a boom. However, it is hard for humans to estimate the range if the audio-visual event is close by.

Thanks to the advance in sensors, many mobile devices can support video recording with 240 FPS and even up to 960 FPS and audio recording with a 48 kHz sampling rate and up to 96 kHz. The high audio and video sampling rates make it possible to accurately measure the delay. We may figure out the approximate video frame and the potential audio sample where the collision starts to happen. The innovative sensing scheme can leverage audio-visual events to enable passive depth estimation. Further, it reveals the spatial uniqueness of sound and light, which enable further 3D perception more than with visual regression alone.

We observe that the duration of a collision is very short around 10 ms. Such a short collision makes it highly unlikely to have multiple collisions overlap in time. We analyze the video recording of the basketball court. Only two frames have double collisions among all 1208 frames and a total of 203 collisions when 7 basketballs are played during a 40-s duration. In this paper, we do not explicitly handle the rare overlapping collisions, but our evaluation does include cases involving overlapping collision sound and shows our scheme is fairly robust as long as the overlap is small. The most severe overlapping scenario is when two identical objects collide on the surface at the same time. FBDepth will overlook such scenarios and not estimate the doubtful depth.

Meanwhile, our system allows multiple objects that generate collisions to appear in the same video. This is achieved by audio-visual event localization to find all collisions in the video and then applying the further stages to each object for depth estimation.

5.2.2 Problem Formulation

We formulate the depth estimation by the physical law of wave propagation. We have:

$$\frac{d}{v} - \frac{d}{c} = T \tag{5.1}$$

where the depth of the sound source is d and the difference between the ToF of sound and light is T. c and v denote the propagation speeds of light and sound, respectively. We can estimate d based on $d = \frac{cvT}{c-v} \approx vT$ since $c \gg v$.

We observe $T = T_{audio} - T_{video} + T_{hardware}$, where T_{audio} and T_{video} denote the event time in the audio and video recordings, respectively, and $T_{hardware}$ denotes the start time difference in the audio and video recordings. It can be small as well as have a small variance with a well-designed media system such as the Apple AVFoundation framework. We regard it as a constant unknown bias to learn.

It is impossible to label the precise T_{video} and T_{audio} manually. T_{video} can be tagged at most frame-level. Even though many commercial cameras can support up to 240 FPS, it results in a 4-ms segment and 1.43m depth variation. Moreover, it is tough to determine the exact frame that is nearest to the collision in high FPS mode by a human being due to the constrained view of the camera. T_{audio} is challenging to recognize in the wild as well. Although the audio sampling rate is high enough, we can recognize the significant early peaks instead of the first sample triggered by the collision. The best effort of

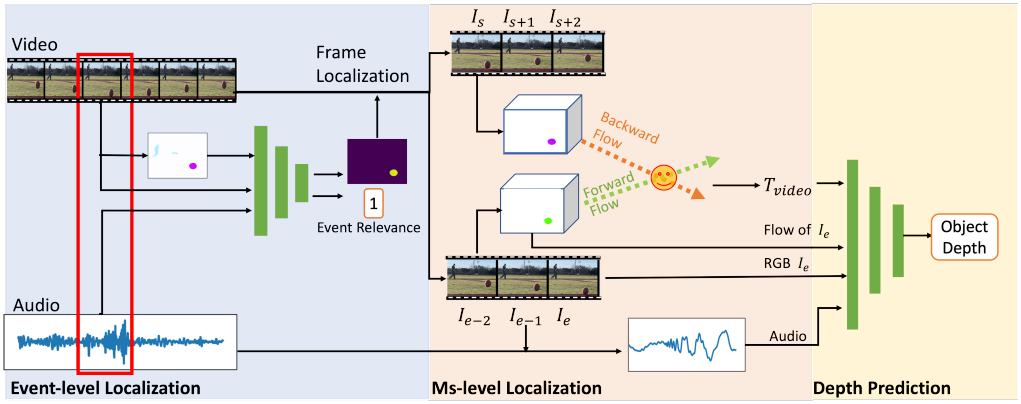


Figure 5.5: Model architecture. Our audio-visual depth estimation uses the video, audio, and optical flow to perform the event-level localization to retrieve the collision event. It analyzes the collision flow and estimates the collision timestamp in the video. It uses multiple modalities including RGB, flow, audio and the timestamp to estimate the depth.

segmentation is 10-ms level based on real data.

Hence, We cannot learn the timestamp with supervision. As figure 5.5 shows, we propose a 2-stage estimation framework. In the first stage, we target to localize the collision event in the video with the help of the audio and estimate the numerical T_{video}. We localize the audio-visual event in the stream and then take advantage of the unique optical flow of the collision to estimate T_{video} at ms-level. In the second stage, we regard the T_{video} as an anchor into the audio clip and direct regress the depth with extra visual appearance and optical flow under depth supervision. We make the network optimize T_{audio} automatically with knowledge of the T_{video}, the audio waveform and visual features.

We demonstrate a novel coarse-to-fine pipeline to localize the collision

with a super temporal resolution in the video. This method does not require annotations on ms-level, which is at least two orders of magnitude finer than previous approaches. They rely on the supervision of segment annotations, such as AVE dataset with 1-second segments [141], Lip Reading Sentences 2 dataset with word-level segments [28], BOBSL with sentence-level alignments [17].

5.2.3 Audio-Visual Coarse-to-Fine Localization

Primer on optical flow. The optical flow captures the movement of the brightness pattern, which is used to approximate the motion field. More specifically, for each pixel (x, y) in a video frame, the optical flow, denoted as $u(x, y)$ and $v(x, y)$, specifies the point's projection of motion in 2D. By assuming the brightness intensity constancy, we have

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) \tag{5.2}$$

Meanwhile, by applying a Taylor series expansion, we get

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\delta I}{\delta x}\Delta x + \frac{\delta I}{\delta y}\Delta y + \frac{\delta I}{\delta t}\Delta t. \tag{5.3}$$

Hence we arrive at

$$\Delta I v + I_t = 0 \tag{5.4}$$

where $v = (\Delta x, \Delta y)$ is the optical flow and $\Delta I = (I_x, I_y)$ is the spatial gradient, and I_t is the temporal gradient. By deriving ΔI and I_t from a pair of video frames, we can obtain the optical flow v.

Lucas-Kanade method is the most famous algorithm for computing the optical flow. It derives multiple constraints from the pixel along with its neighbors and uses the least square to estimate v. More recently, many deep learning algorithms have been developed to improve optical flow estimation. We choose to use the RAFT [140], which is one of the latest algorithms with excellent performance. It computes the optical flow in a multi-scale manner, where at each scale it uses a ConvGRU module to iteratively estimate the optical flow and move the pixel according to the derived flow.

5.2.3.1 Event-Level Localization

Audio-visual modeling for collisions. In this step, our goal is to localize the audio-visual event for the region and the period of interest. It is similar to [141], but the unique properties of collisions bring new opportunities to learning strategy to enable a high granularity localization.

Collisions have a significant motion than other sound sources. We can use the optical flow to inform the network of moving pixels. Besides, the impact sound is highly correlated to the rich information of objects [39], such as shape, materials, size, mass, etc. It makes audio-visual cross-matching easier than general audio-visual events so that we do not need to apply a complex scheme to learn. Another fact is that collisions are pretty sparse temporally in the wild because the duration of collisions is extremely short. It is rare to come across overlapped collisions based on our empirical study on the basketball court. Only two frames have double collisions among all 1208

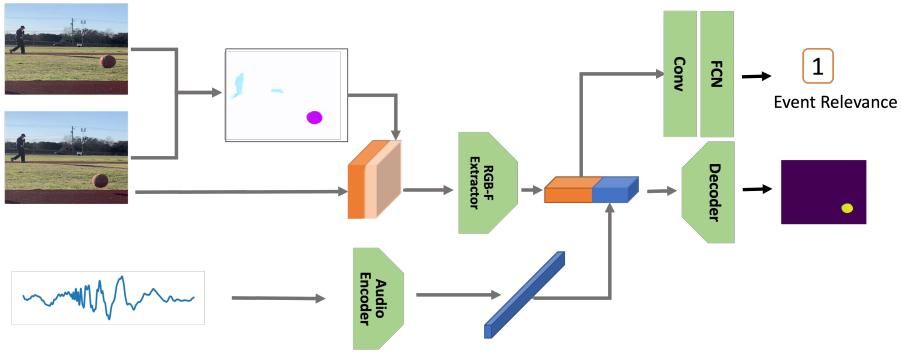


Figure 5.6: MVANet for audio-visual localization.

frames and a total of 203 collisions when 7 basketballs are played during a 40-s duration.

We propose a motion-guided audio-visual correspondence network (MAVNet). Figure 5.6 demonstrates the structure of MAVNet. Similar to [141, 164], MAVNet performs the cross-matching for the audio features and the RGB-F channels. Besides, it predicts the audio-visual segmentation to capture the whole pixels of the target object. It can achieve fine-grained audio-visual scene understanding [188]. We use the segmentation mask to filter flows of interest and perform high-resolution estimation in the next steps. MAVNet has two backbones to deal with RGB-F channels and audio clips respectively. A U-Net [119] style encoder is applied to extract the frame features conditioned by optical flows. It uses a series of convolution layers to extract visual features. Another branch is the audio encoder which takes in the time-domain signal. It has a 1D convolution layer to learn an STFT-like representation and a stack of 2D convolution layers with batch normalization to learn the semantic audio

features. We replicate the audio feature, tile them to match the visual feature dimension, and concatenate the audio and visual feature maps. MAVNet has two output heads as well. the U-Net decoder applies a series of up-convolutions and skip-connections from the RGB-F encoder to fused feature maps to learn the binary segmentation mask M. Meanwhile, the fused feature map is fed into a binary classification head consisting of convolution layers and linear layers to predict the audio-visual event relevance $y \in \{0, 1\}$.

Training We use the weighted sum Binary Cross Entropy (BCE) loss as the training objective for both segmentation and the cross-matching, We train all components to jointly optimize the location predictions and energy reconstruction. We minimize the total loss

$$\mathcal{L}_{total} = BCE(M, \hat{M}) + \lambda * BCE(y, \hat{y}) \tag{5.5}$$

where λ is the hypermeter to set.

Inference We demonstrate the inference on the video stream and audio stream in Algorithm 1. We only use low FPS to perform MAVNet to avoid dense inference at this stage. Moreover, we do not need to activate the segmentation head until the audio clip and the frame are highly matched. Finally, MAVNet uses this audio clip to retrieve a sequence of frames including the full collision procedure.

Algorithm 1: Audio-visual localization pseudo-code

```
V, A = VideoStream(), AudioStream()
while V, A is available do
    I_0, I_1, t = V.prevFrame, V.curFrame, V.curTime
    f = RAFT(I_0, I_1)
    a = A(t)
    cls, mask = MVANet(I_1, f, a)
    if cls is valid then
        track the object forward and backward a
        perform video event detection and depth estimation
    end
end
```

5.2.3.2 Frame-level Localization

Given a sequence of video frames, our goal is to split them into two sets: the frames before the collision $\mathbf{V_0}$ and the frames after the collision $\mathbf{V_1}$. This essentially requires us to determine the last frame I_e in $\mathbf{V_0}$ before the collision and the first frame I_s in $\mathbf{V_1}$ after the collision. Thus, we locate the collision between the frame I_e and I_s.

Based on the analysis of the physical motion, we make an important observation that can help determine I_e and I_s. The collision results in a significant acceleration change due to the strong impulse force. Let $a_t = v_t - v_{t-1}$ and $\delta a_t = a_t - a_{t-1}$ denote the acceleration and acceleration change of frame I_t, respectively. δa between I_e and I_s is large, while δa between adjacent frames before or after the collision is small. If the object stops moving immediately after the collision, we take the static frame I_{e+1} as I_s. Finally, we select the frames before I_e to generate $\mathbf{V_0}$, and select the frames after I_s

to generate $\mathbf{V_1}$.

We use the retrieved mask in the last stage to determine the object positions in the frames and calculate the velocity, acceleration, and acceleration change. We find the I_e and I_s at the low FPS and then replicate the procedure for frames between I_e and I_s at high FPS. Finally, we locate I_e and I_s in the high FPS mode efficiently.

5.2.3.3 Ms-level Localization

To further locate the exact moment of the collision, we try to interpolate frames between I_e and I_s to recover the skipped frame. Unfortunately, the common assumption of frame-based interpolation is fully broken down.

Motion consistency is fundamental for spatio-temporal video processing. If the motion of the object is temporally stable across several frames (*e.g.*, due to a constant force), the position and pose can be predicted in the future frames as well as be interpolated between two frames. We denote it as *motion first consistency*. However, the impact sound is caused by an impulse force, which results in a rapid change in the motion status. It breaks the motion continuity and consistency. When we observe I_e and I_s, we cannot determine whether a collision happens or the object just flies in the air.

Luckily, the collision moment retains a new form of motion consistency. We denote it as *motion second consistency*. It reveals that the motions before and after the collision share the same intersection position. Besides, they keep the *motion first consistency* separately. Therefore, we can extrapolate the

motions based on the *motion first consistency* and search for the most similar motion extrapolations by leveraging *motion second consistency*. Note that our final goal is to find the timestamp of the collision instead of the motion status at the shared position. [64, 120] try to recover the sub-frame motions and trajectories as well but they require the high FPS ground truth to guide the training. In our context, we care more about when the collision happens than what it looks like.

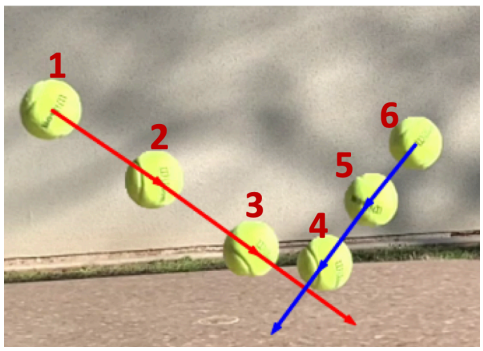


Figure 5.7: Interpolate trajectories before and after the collision to compute the intersection for collision detection.

Optical flow extrapolation Optical flow is widely used for frame prediction [155, 179] and interpolation [89, 97, 169] by warping the frame with the estimated optical flow. Because it can capture all motions of pixels and get a finer understanding of the object dynamics. The optical flow sequence is usually generated by adjacent video frames. However, it is not efficient for extrapolation. The drift of pixels in the flow requires extra iterative wrappings to align the corresponding pixels, which results in accumulation errors.

We can extrapolate the trajectories of $\mathbf{V_0}$ and $\mathbf{V_1}$ to locate the same

collision moment. A simple solution is to use the center of the ball to fit the 2D moving trajectory and use the intersection of two trajectories as the collision point. However, this method is not accurate for real experiments. The center point simplifies the shape and color of the object. It loses rich information of thousands of pixels from the object. For the example in Figure 5.7, the center point cannot model the pose or rotation of the object. Thus, it does not take advantage of all visual constraints. Moreover, the deviation of each center's position can lead to a considerable error because the data points are not sufficient.

We apply optical flow to enable a fine-grained robust video event estimation. Optical flow can provide pixel-to-pixel motion in the 2D frame projected from the 3D space. We can use all pixels to search for the optimal collision time. The optical flow sequence is usually generated by adjacent video frames. Considering a set of frames $\{I_0, I_1, ...I_n\}$, we can estimate a flow sequence $\{f_{0\to1}, f_{1\to2}, ...f_{n-1\to n}\}$. For a pixel $p_0(x, y)$ of the target object in the frame I_0, we can infer the new position of the pixel in the frame I_1, denoted as $p_0(x + \Delta x_1, y + \Delta y_1)$, where $(\Delta x_1, \Delta y_1) = f_{0\to1}(x, y)$. Based on the optical flow sequence, we can estimate the coordinate of this pixel in the next few frames as

$$\{(x, y), (x + \Delta x_1, y + \Delta y_1), ..., (x + \sum_{i=1}^{n} \Delta x_i, y + \sum_{i=1}^{n} \Delta y_i)\}$$

.

However, this method will result in error accumulation from the optical

flow and mismatch the pixels across the frames.

Therefore, we compute the optical flows from an anchor frame I_a to the frame sequence $\mathcal{V}$ as

$$\{I_0, I_1, ...I_n\}$$

We can estimate the flow sequence $\mathcal{F}_{a\to\mathcal{V}}$ as

$$\{f_{a\to 0}, f_{a\to 1}, ...f_{a\to n}\}$$

As $f_{a\to n}(x, y)$ represents the movement of the pixel $I_a(x, y)$ to I_n, $\mathcal{F}_{a\to\mathcal{V}}(x, y)$ describes how the pixel in $I_a(x, y)$ moves across the frame sequence $\mathcal{V}$. Hence, $\mathcal{F}_{a\to\mathcal{V}}$ tracks the global motion of each pixel without iterative warpings. With the historical positions of $I_a(x, y)$ from frame I_0 to I_n, we can regress the motion of this pixel and extrapolate the flow to $f_{a\to n+\delta t}$, which is the relative pixel position to $I_{n+\delta t}$ with an arbitrary δt.

In our context, We pick k consecutive frames before the collision $\mathcal{V}_{pre}$ as

$$\{I_{e-k+1}, I_{e-k+2}, ..., I_e\}$$

and after the collision $\mathcal{V}_{post}$ as $\{I_{s+k-1}, I_{s+k-2}, ..., I_s\}$. We select the frame I_e as the anchor frame. It is near the collision moment, so its motion to other frames is not dramatic and easy to be estimated. Hence, we can estimate the optical flow sequences $\mathcal{F}_{e\to\mathcal{V}_{pre}}$ and $\mathcal{F}_{e\to\mathcal{V}_{post}}$ Meanwhile, we apply the predicted segmentation mask of I_e to filter the pixels of the target object. In the last step, we build up regressors $\mathcal{R}$ for each pixel's motion individually and predict future locations in any sub-frame.

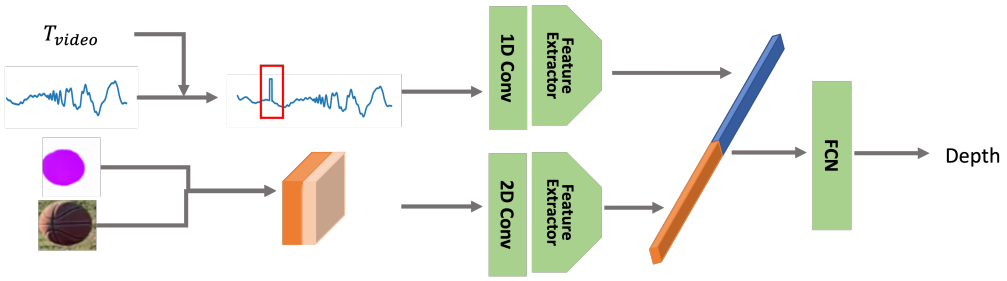


Figure 5.8: Structure of the depth predictor. It incorporates both RGB-F channels and visual timed waveform to regress the target depth

Optical flow interpolation We have construct pixel level regressors for $\mathcal{F}_{e\to\mathcal{V}_{pre}}$ and corresponding $\mathcal{F}_{e\to\mathcal{V}_{post}}$. They can extrapolate the flow $f_{e\to e+\delta t_0}$ and $f_{a\to s+\delta t_1}$, respectively. $\delta t_0, \delta t_1$ are extrapolation steps. The optimization goal is to

$$\min_{e-s\leq\delta t_1\leq 0\leq\delta t_0\leq s-e} ||f_{e\to e+\delta t_0}, f_{a\to s+\delta t_1}||_2, \text{s.t. } e+\delta t_0 < s+\delta t_1$$

The collision duration is $s+\delta t_1-(e+\delta t_0)$, which is always more than 0. $e+\delta t_0$ is the target ms-level localization $\hat{T}_{video}$. We can apply this interpolation methodology to search the intersection of the object's center trajectory or maximize the Intersection over the Union (IoU) of the object's bounding box. However, both only use several key points so they cannot achieve a fine granularity since the optical flow takes advantage of thousands of pixels.

5.2.4 Depth Regression

Based on the estimation $\hat{T}_{video}$, we directly regress the depth to fit the T_{audio} and the bias $T_{Hardware}$ with the supervision of ground truth depth. We observe that the sound generation procedure varies a lot across different

objects, materials, shapes, and motions. On one hand, the diverse waveforms make it impractical to measure the exact T_{audio} manually. On the other hand, each specific waveform has significant implications on what is the best T_{audio} corresponding to $\hat{T}_{video}$. To combat the background noise from other sources, we also feed the RGB-F crop of the target object from frame I_e to the depth predictor. It includes the semantic features of the object as well as the motion status just before the collision. These cues can guide the predictor to find the waveform pattern easily. The structure of the depth predictor is demonstrated in Figure 5.8.

We select a sequence of audio samples starting from I_e and label some anchor samples as 1 at $\hat{T}_{video}$. It informed the audio sequence about the timestamp of the visual collision directly. We feed the enriched sequence into the 1D convolution layer to extract a 2D representation. It is followed by two residual blocks to learn high-dimension features. Meanwhile, we use ResNet-18 [46] to extract the RGB-F features of the target object. We tile and concatenate the RGB-F features to the audio features along the channel dimension and append another two residual blocks to fuse the features. Finally, it is followed by a pooling layer and a fully connected layer to predict the depth. The output maps to depth by the 2D projection. We use Mean Square Error (MSE)

$$\mathcal{L}_{depth} = ||d, \hat{d}||_2 \tag{5.6}$$

as the learning objective where d and $\hat{d}$ are the target depth and the predicted depth.

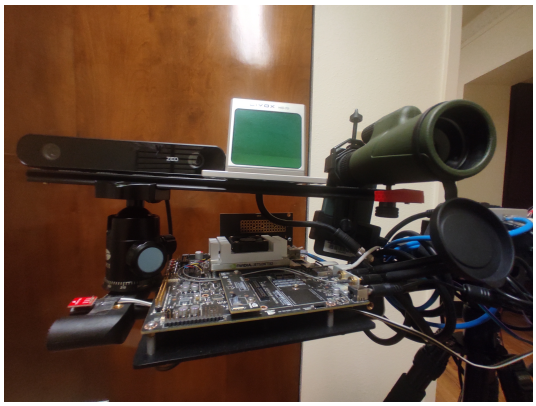


Figure 5.9: Data collection platform with multiple sensors

5.3 Implementation

5.3.1 Setup

Dataset platform and collection As Figure 5.9 shows, we build up a multi-sensor data collection platform to collect the depth data. We use an iPhone XR with a 240-fps slow-motion mode to collect the video with audio. The audio sampling rate is 48Khz. $T_{hardware}$ is small as 1 ms and has a small variance within 1 ms on the iPhone. Moreover, its frame rate is stable and does not have a frame rate drifting. Thus, we can transform the frame number to the timestamp accurately. The calibration of the audio-visual media framework is out of the scope of this work.

To capture the remote scene clearly, the telephoto lens has become indispensable in recent smartphones. Samsung Ultra 22 can support 10x optical zoom and 100x hybrid zoom, and Pixel 6 pro has 20x zoom in all. The iPhone XR is not equipped with a telephoto lens, so we mount an ARPBEST monocular telescope to enlarge the scene at a large distance. As shown in Figure 5.10,

(a) iPhone XR + telescope

(b) Pixel Pro 6

Figure 5.10: Compare the image quality captured by our telescope setup and the commercial telephoto lens on smartphones

the image quality of our setup is a bit worse than the one captured by Pixel 6 Pro's telephoto lens. Thus, our setup does not provide superior image quality compared to existing commercial camera modules on smartphones. The image taken by Pixel Pro 6 is sharp but noisy while the one taken by iPhone XR with the telescope is a bit blurred. We do not take advantage of the telescope from this perspective. We also try to use Pixel Pro 6 or other Android phones to perform the experiment. However, the A/V median framework of Android is not as robust as the IOS. The delay of audio-video recording is not only huge but also has a significant variance. Besides, the frame duration is not constant. It drifts by period, which introduces huge errors for timestamp estimation. Overall, our setup resembles the hardware available on commercial mobile phones.

We set a ZED 2 stereo camera with a 12 cm baseline and a Livox

Figure 5.11: A set of objects are used for experiments. They cover most common items and materials in daily life

Mid-70 lidar together with the iPhone. Due to the significant sparsity of the point cloud, it is almost inaccessible to measure the depth of a small object with the lidar, especially when it is far and moving. Thus, we use the lidar to measure the depth of an anchor position and throw objects to the anchor position precisely.

In Figure 5.11, We use 24 objects including various masses, sizes, shapes, and six common materials: wood, metal, foam, rubber, plastic, and paper. They keep the falling motion or hand-driven motion to produce the impact sound. We collect the collision from 2 meters to 60 meters in different environments. After data cleaning and annotation, we get 3.6K+ raw audio-visual sequences., including 280K+ frames. Each sequence has about 40 to 120 frames and a corresponding audio clip corresponding. We use the stereo camera to capture static images and use the lidar to capture static depth maps.

AVD Dataset We augment the raw audio-visual sequences to be a large

audio-visual depth(AVD) dataset. We generate the sequence with multi-collisions by cropping one moving object from a raw video sequence and augmenting it to another raw sequence with a random temporal location. Meanwhile, we add up the audio sequence with the same time shift as the video. We have 10K audio-visual sequences. For the event-level localization stage, we segment an audio clip of 66.7ms including the impact sound and sample 20 frames including visible objects from each sequence and pair them as positive pairs. Negative samples pair the frame with the audio clip without impact sounds or with irrelevant impact sounds. Finally, we generate around 400K audio-visual pairs. Besides, we augment the raw depth with a maximum 3% random change to diversify the depth and shift audio samples accordingly to the video timestamp. It can solve the problem of discrete anchor depths. The change cannot be significant because the impulse response of sound is also related to depth. It requires more transformation than just shifting audio samples. We also augment images with low light, flip and rotation, and audio with diverse background noise from WHAM![160].

5.3.2 Model Implementation

We use the pre-trained model from RAFT [140] to compute the optical flow. This model is pre-trained on a mix of synthetic data and real data [30]. It performs well on our current dataset. We pre-train a ResNet-18 [46] audio classifier on our own dataset. This model without the final layer will work as the audio feature extractor in the MVANet. It will output the audio feature

embedding with a size of 256. The weights are also updated using MVANet. The U-Net in MVANet has 5 down-convolution blocks to increase the number of channels to 16, 32, 64, 128, 256 and 5 up-convolution blocks to decrease the number of channels to 256, 128, 64, 32, 1. The single channel output is fed to the sigmoid layer to generate the segmentation mask. Meanwhile, the fused feature map is also input to 6 convolution layers, followed by a fully connected layer to classify the event relevance. The depth predictor first uses a 1D convolutional layer with a kernel of 32 and outputs 256 channels to encode the 1D signal into a 2D presentation. It is followed by a Resnet-like module to extract the audio embedding with a dimension of 1024. Meanwhile, it applies another Resnet-like extractor to the RGB-F input with 5 channels and outputs the embedding with a dimension of 1024. The embeddings are concatenated and fed into several full connected layer to regress the depth.

MAVNet and depth regression network are all implemented with Python and PyTorch. They are trained on the platform with an Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz CPU and an NVIDIA Quadro RTX 6000 GPU. The learning rate and batch size are set as 0.001 and 32 respectively.

5.4 Evaluation

Baselines We include three types of baseline for comparison. We compare to a monocular depth estimation method NeWCRFs [175], a state-of-the-art(SOTA) on multiple benchmarks. We also compare to stereo matching methods including the ZED built-in ultra depth estimation SDK and a SOTA

method LEAStereo [25]. We use dense depth maps collected by the lidar to finetune NeWCRFs. Stereo methods do not need finetune. Despite optical flow-based interpolation, we compare to interpolation using key points such as the trajectories of center or bounding boxes.

Metrics We use the mean absolute depth errors as

$$AbsErr = \frac{1}{n}\sum_{i=1}^{n}|d - \hat{d}| \tag{5.7}$$

, root mean square absolute relative errors

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d - \hat{d})^2} \tag{5.8}$$

, and absolute relative error

$$AbsRel = \frac{1}{n}\sum_{i=1}^{n}\frac{|d - \hat{d}|}{d} \tag{5.9}$$

as the end-to-end performance metrics.

FBDepth is a sparse depth estimation, which estimates the colliding objects. Meanwhile, monocular and stereo baselines have dense depth estimations for all pixels of the object. To be fair, we sample the median estimation depth from the estimated dense depth map. We compare it with the median depth from the ground-truth dense depth map collected by lidar. We provide the results over different distance ranges as close($\leq 10m$), mid($10m$-$30m$) and far($\geq 30m$). Intuitively, there is an upper bound for the temporal resolution due to various constraints such as hardware timestamp variance, limited FPS, etc. Hence, $AbsErr$ also has a corresponding lower bound of measurement.

Method	Input	FPS	AbsErr(m)	AbsRel(%)	RMSE(m)
NeWCRFs	V	-	1.68	9.48	3.49
ZED SDK	S	-	2.03	7.28	3.69
LEAStereo	S	-	0.88	4.09	2.95
FBDepth	A+V	30	0.95	4.26	1.51
FBDepth	A+V	60	0.72	3.34	1.27
FBDepth	A+V	120	0.67	3.11	1.09
FBDepth	A+V	240	**0.64**	**2.98**	**1.03**

Table 5.3: The comparison for different depth estimation approaches. V, S, A represent visual, stereo, audio respectively. We input video with different frame rates as well.

It results in a large *AbsRel* at close depths. However, the negative effect will relieve as the distance increases. We target to figure out the most suitable range for FBDepth.

5.4.1 Results

Table 5.3 shows the overall results on the depth estimation. Table 5.4 presents the performance over different distance. In all, FBDepth can achieve better performance on all metrics than baselines across different FPS. Several important trends can be observed. Stereo matching methods perform extraordinarily on close objects, where more clear view differences can be captured. The *AbsErr* and $RMSE$ increase dramatically as the targets become further because the limited baseline cannot resolve the view difference easily. On the other side, the *AbsErr* and $RMSE$ of FBDepth grow slowly with the increasing distance while the *AbsRel* decreases gradually. Intuitively, there is an upper bound for the temporal resolution due to the limited FPS, the lack of

Method	Input	FPS	AbsErr(m) close/mid/far	AbsRel(%) close/mid/far	RMSE(m) close/mid/far
NeWCRFs	V	-	0.553/1.09/3.27	11.1/6.74/8.64	0.895/1.51/5.82
ZED SDK	S	-	0.083/0.96/5.10	1.78/6.05/12.7	0.108/1.07/6.30
LEAStereo	S	-	**0.067**/0.66/2.47	1.48/4.24/5.98	**0.083**/0.76/5.08
FBDepth	A+V	30	0.485/0.83/1.33	10.9/5.20/3.32	0.731/1.01/2.29
FBDepth	A+V	60	0.418/0.70/1.11	8.94/4.33/2.79	0.597/0.83/1.86
FBDepth	A+V	120	0.392/0.61/0.98	8.42/3.79/2.49	0.534/0.75/1.68
FBDepth	A+V	240	0.337/**0.58**/**0.95**	7.25/**3.55**/**2.41**	0.476/**0.69**/**1.61**

Table 5.4: A detailed comparison of how different depth estimation approaches perform at various distances

an accurate timestamp, and the small disturbance of audio-video software. Thus FBDepth may not achieve the centimeter level easily. A Further depth can break the assumption of stereo-matching methods as well as monocular methods which has a fixed depth range of training data, but FBDepth still holds the physical propagation law in this condition.

FBdepth also shows the advantages on NeWCRFs. The monocular methods rely on the training set, which includes various scenarios and depths. Although we apply camera remapping with intrinsic matrix and finetuning, NeWCRFs still cannot achieve the best performance as the one in the pre-trained dataset. The implicit depth regression has difficulty in domain adaption. In the contrast, stereo methods can be directly applied to the new scenario and achieve awesome estimation because its fundamental is the explicit spatial view difference on stereo images. FBDepth applies explicit spatial delay and does not rely on the camera and scenarios heavily. It requires several learning models but these models can be applied to common cameras and

microphones. FBDepth can be more general with a more diverse dataset.

5.4.2 Ablation

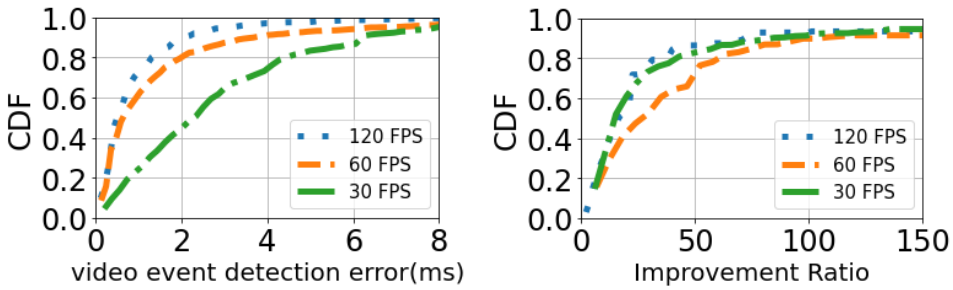
In the ablation study, we show how each stage contributes to the final results.

Event-level localization We invest in how the optical flow can help detect the collision event as well as contour the object mask in Table 5.5. We define recall and precision as the percentage of correct recognized audio-visual events in all audio-visual events and all recognized events with an IoU of more than 0.5, respectively. With the flow, both recall and precision improve as the flow can work as a pre-mask to guide the network. The main failures in recall come from weak collision sounds or rare simultaneous collisions. The incorrect recognition is mainly due to similar objects in the frame. The optical flow can work as a great pre-mask to guide the region of interest. Thus, It enables robust audio-visual segmentation. Besides, if the event is either too weak to detect or ambiguous to recognize, FBDepth will ignore the event detection and not try to estimate the depth.

Frame-level localization Frame rate is most related to frame-level stage. We observe that increasing the frame rate reduces the numerical error of FBDepth in Table 5.3. Especially, increasing 30 FPS to 60 FPS yields the largest improvement, and the benefit gradually tapers off with a further increase in the frame rate. We observe that 30 FPS is too slow to capture sudden movements and fast dynamics while 60 FPS is around the borderline. It is consistent with

Method	AbsErr(m)	AbsRel(%)	RMSE(m)	Recall	Precision
event loc w/o flow	-	-	-	87.3	93.7
FBDepth w/o interp	1.95	9.16	4.07	-	-
FBDepth w/ center	1.39	6.57	2.31	-	-
FBDepth w/ bbox	1.23	5.65	2.06	-	-
FBDepth w/o RGB-F	0.92	4.25	1.42	-	-
FBDepth	**0.64**	**2.98**	**1.03**	94.5	98.7

Table 5.5: Ablation study for FBDepth using different setups at each stage. The input is 240 FPS.



(a) Temporal error of the estimation of low FPS compared to 240 FPS

(b) Improvement ratio of temporal resolution

Figure 5.12: Effectiveness of the video event detection in the second stage

the trend to set 60 FPS as the default video recording and playing. The motion in 120FPS and 240 FPS is even slower so it is more difficult to distinguish the frame I_e. The frame error is no more than the one in the low FPS mode. Thus, 120 FPS and 240 FPS bring less improvement.

Ms-level localization We investigate our special interpolation from two perspectives. First, we need to verify whether this method works. However, there is no ground truth timestamp so we cannot directly quantify the accuracy. We set the estimation of 240 FPS as a baseline and compare the estimation of lower FPS with this baseline. Intuitively, high FPS can predict the collision

in the small frame duration and yield less error to the real collision timestamp. Hence, We compare the interpolation of low FPS to the high FPS to demonstrate how much temporal improvement. If it can get similar numerical results from independent input, which means the algorithm is reliable. In Figure 5.12, the median temporal error for 30, 60, and 120 FPS is 2.3ms, 0.65ms, and 0.5ms respectively. Considering the frame resolution, we can define the improvement ratio as

$$improvement = \frac{frame\_duration}{temporal\_error} \tag{5.10}$$

The 60 FPS has the largest 25x improvement over the frame duration. This is strong evidence that our ms-level localization is reasonable and robust.

Second, we compare the performance of depth estimation with different interpolation strategies in Table 5.5. We demonstrate how the optical flow based interpolation works better to estimate a more accurate timestamp. We use the result from frame localization to predict the depth when there is no interpolation. The error is large since this timestamp is ambiguous for the depth prediction. Interpolation with the traces of centers or bounding boxes does not work well. Both only use a few key points to track the overall motions, but they cannot capture dynamics in fine granularity. Center-based interpolation only tracks the geometry center of the object but the center can change dramatically as the object moves in 3D space. The bounding box enriches the representation of the object's motion but it still cannot describe the subtle motion of rotation. In fact, both methods can only fit the translation in the 2D plane and approximate the 3D motion.

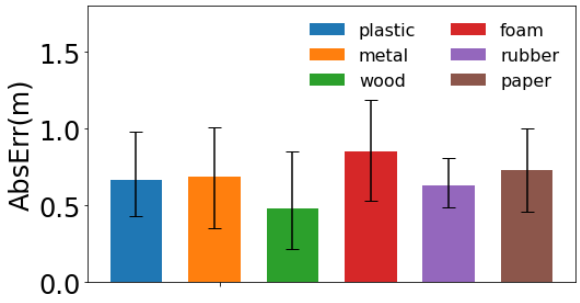


Figure 5.13: Depth estimation error across different materials

Depth regression Without the RGB-F channel of the target object in depth regression, the estimation will be less robust due to the ambient sound and the background noise as shown in Table 5.5

Impact of object materials We use the 24 diverse objects to perform the collision experiment. These objects have many common characteristics and unique features. Among them, we observe that materials can be one of the important features to decide the collision procedure and associate the visual features and audio features. On the video side, the material is highly correlated with the visual appearance and results in the motion after the collision. On the audio side, the material determines the elastic force directly and results in different frequencies and magnitudes. Our dataset includes six common materials. Figure 5.13 shows the performance of depth estimation on each material.

Among them, wood has the best performance. Its median AbsErr is 0.47m. We realize that wood always produces a loud sound and does not spin fast. The collision procedure of wood is smooth. Foam has the largest median

AbsErr of 0.81m. The impact sound is weak and the object is not rigid enough. the other materials have similar performance. Metal has a strong impact sound but it may spin fast. Rubber takes a lot of time to bounce but the sound is weak without a large velocity. In all, FBDepth can handle all these materials well.

Chapter 6

Conclusion

In this dissertation, we evolve the acoustic sensing system from a traditional active setup to a novel passive design. We introduce new modalities to acoustic sensing and identify the high correlation between them. With the novel components and effective design, we build up new applications based on the acoustic sensing system and demonstrate significant advantages over the existing approaches.

In Chapter 3, we introduce a novel deep learning module to the traditional active acoustic ranging system. To eliminate the need of collecting large volumes of training data, we generate synthetic signals by incorporating important factors in real environments, such as noise, multipath, self-interference, and transceiver frequency response. We develop a DNN that uses filters with long kernel sizes to detect signal patterns and applies the ensemble method to enhance the estimation accuracy. We evaluate DRNet on real data collected using 11 phones across 4 locations and show it achieves significant performance gain over FMCW.

In Chapter 4, we design a hybrid active and passive acoustic sensing system. we develop a novel system to combat acoustic interference for online

meetings. It advances state-of-the-art in acoustic-based tracking by leveraging both audible and inaudible signals. Moreover, it uses multi-task learning to jointly estimate the AoA and separate the source. Our evaluation shows that our system significantly improves over the state-of-the-art. We believe our work is an important step towards enabling online meetings and classes under interference and noise, which have already been playing a major role in our daily lives. Moving forward, we are interested in exploring other context information to further improve the performance of online meetings.

In Chapter 4, we propose a fully passive acoustic depth estimation system with the help of visual modality. Our novel method is based on the "flash-to-bang" phenomenon. We apply a coarse-to-fine estimation framework to achieve millisecond-level audio-visual event localization. We incorporate multiple modalities to estimate the depth in the wild without calibration or prior knowledge about the environment or target. Our extensive evaluation shows that our approach yields smaller errors across varying distances. In comparison, the errors of several existing methods increase rapidly with distance. Therefore, our method is particularly attractive for large distances. As part of our future work, we are interested in further enhancing the accuracy of our method, generalizing to more contexts, and using the estimated depth of the collided object to estimate the depth of other objects in the scene.

Bibliography

[1] Itu-t recommendation. perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. *Rec. ITU-T*, page 862, 2001.

[2] The overlooked environmental footprint of increasing internet use resources. Apr. 2021.

[3] Pytorch, 2019. https://pytorch.org/.

[4] Fadel Adib, Zach Kabelac, Dina Katabi, and Rob Miller. WiTrack: motion tracking via radio reflections off the body. In *Proc. of NSDI*, 2014.

[5] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via RF body reflections. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 279–292, 2015.

[6] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. My lips are concealed: Audio-visual speech enhancement through obstructions. In *Proc. of Interspeech*, 2019.

[7] Macbook tech specs, 2021. https://www.apple.com/macbook-pro-14-and-16/specs.

[8] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018.

[9] Junjie Bai, Fang Lu, Ke Zhang, et al. Onnx: Open neural network exchange. `https://github.com/onnx/onnx`, 2019.

[10] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011.

[11] Tom Barker, Tuomas Virtanen, and Olivier Delhomme. Ultrasound-coupled semi-supervised nonnegative matrix factorisation for speech enhancement. In *Proc. of IEEE ICASSP*, 2014.

[12] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020.

[13] Bela platform, 2017. https://bela.io.

[14] Jacob Benesty, Jingdong Chen, and Yiteng Huang. A generalized mvdr spectrum. *IEEE Signal Processing Letters*, 12(12):827–830, 2005.

[15] Amlaan Bhoi. Monocular depth estimation: A survey. *pre-print*, 2019. https://arxiv.org/pdf/1901.09402.pdf.

[16] Michael S Brandstein and Harvey F Silverman. A robust method for speech signal time-delay estimation in reverberant rooms. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 375–378. IEEE, 1997.

[17] Hannah Bull, Triantafyllos Afouras, Gül Varol, Samuel Albanie, Liliane Momeni, and Andrew Zisserman. Aligning subtitles in sign language videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11552–11561, 2021.

[18] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[19] Luis A Castro and Jesus Favela. Continuous tracking of user location in wlans using recurrent neural networks. In *null*, pages 174–181. IEEE, 2005.

[20] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15516–15525, 2021.

[21] Changan Chen, Wei Sun, David Harwath, and Kristen Grauman. Learning audio-visual dereverberation. *arXiv preprint arXiv:2106.07732*, 2021.

[22] Jingjing Chen, Qirong Mao, and Dong Liu. Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation. *arXiv preprint arXiv:2007.13975*, 2020.

[23] Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 246–250. IEEE, 2017.

[24] Zhuo Chen, Xiong Xiao, Takuya Yoshioka, Hakan Erdogan, Jinyu Li, and Yifan Gong. Multi-channel overlapped speech recognition with location guided speech extraction network. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 558–565. IEEE, 2018.

[25] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33, 2020.

[26] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *European Conference on Computer Vision*, pages 107–123. Springer, 2020.

[27] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. Achieving single channel, full duplex wireless communication. In

Proceedings of the sixteenth annual international conference on Mobile computing and networking, pages 1–12. ACM, 2010.

[28] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian conference on computer vision*, pages 87–103. Springer, 2016.

[29] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. Rf-echo: A non-line-of-sight indoor localization system using a low-power active rf reflector asic tag. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 222–234. ACM, 2017.

[30] MMFlow Contributors. MMFlow: Openmmlab optical flow toolbox and benchmark. `https://github.com/open-mmlab/mmflow`, 2021.

[31] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. Librimix: An open-source dataset for generalizable speech separation, 2020.

[32] Antoine Deleforge, Florence Forbes, and Radu Horaud. Acoustic space learning for sound-source separation and localization on binaural manifolds. *International journal of neural systems*, 25(01):1440003, 2015.

[33] David Diaz-Guerra, Antonio Miguel, and Jose R Beltran. gpurir: A python library for room impulse response simulation with gpu acceleration. *Multimedia Tools and Applications*, 80(4):5653–5671, 2021.

[34] Jacob Donley, Vladimir Tourbabin, Jung-Suk Lee, Mark Broyles, Hao Jiang, Jie Shen, Maja Pantic, Vamsi Krishna Ithapu, and Ravish Mehra. Easycom: An augmented reality dataset to support algorithms for easy communication in noisy environments. 2021.

[35] Haishi Du, Ping Li, Hao Zhou, Wei Gong, Gan Luo, and Panlong Yang. Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1448–1456. IEEE, 2018.

[36] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. In *In Proc. of ACM SIGGRAPH*, 2018.

[37] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. In *In Proceedings of ACM SIGGRAPH*, 2018.

[38] Hakan Erdogan, John R. Hershey, Shinji Watanabe, Michael I. Mandel, and Jonathan Le Roux. Improved MVDR Beamforming Using Single-Channel Mask Prediction Networks. In *Proc. Interspeech 2016*, pages 1981–1985, 2016.

[39] Chuang Gan, Yi Gu, Siyuan Zhou, Jeremy Schwartz, Seth Alter, James Traer, Dan Gutfreund, Joshua B Tenenbaum, Josh H McDermott, and Antonio Torralba. Finding fallen objects via asynchronous audio-visual integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10523–10533, 2022.

[40] R. Gao and K. Grauman. Visualvoice: Audio-visual speech separation with cross-modal consistency. In *Proc. of CVPR*, 2021.

[41] Ruohan Gao, Changan Chen, Ziad Al-Halah, Carl Schissler, and Kristen Grauman. Visualechoes: Spatial image representation learning through echolocation. In *European Conference on Computer Vision*, pages 658–676. Springer, 2020.

[42] Making gated-impulse frequency measurements using ARTA, 2012. https://pearl-hifi.com/06_Lit_Archive/15_Mfrs_Publications/ARTA%201.7/-FR%20_Measurement_Using_ARTA.pdf.

[43] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[44] Rongzhi Gu, Lianwu Chen, Shi-Xiong Zhang, Jimeng Zheng, Yong Xu, Meng Yu, Dan Su, Yuexian Zou, and Dong Yu. Neural spatial filter: Target speaker speech separation assisted with directional information. 2019.

[45] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[47] John Hershey and Javier Movellan. Audio vision: Using audio-visual synchrony to locate sounds. *Advances in neural information processing systems*, 12, 1999.

[48] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016.

[49] Jahn Heymann, Lukas Drude, and Reinhold Haeb-Umbach. Neural network based spectral mask estimation for acoustic beamforming. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 196–200. IEEE, 2016.

[50] G. Hu and Wang D. Speech segregation based on pitch tracking and amplitude modulation. In *In Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*, page 79–82, 2001.

[51] Guoning Hu and DeLiang Wang. Monaural speech segregation based on pitch tracking and amplitude modulation. *IEEE Transactions on Neural Networks*, 2004.

[52] Matebook 14s tech specs, 2021. https://consumer.huawei.com/en/laptops/matebook-14s.

[53] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2021.

[54] Intel realsense d415 camera, 2018. https://ark.intel.com/content/www/us/en/ark/products/128256/intel-realsense-depth-camera-d415.html.

[55] Benoit Jacob and Gael Guennebaud. Eigen, 2019. http://eigen.tuxfamily.org/index.php?title=Main_Page.

[56] Teerapat Jenrungrot, Vivek Jayaram, Steve Seitz, and Ira Kemelmacher-Shlizerman. The cone of silence: speech separation by localization. *arXiv preprint arXiv:2010.06007*, 2020.

[57] Daniel Johnson, Daniel Gorelik, Ross E Mawhorter, Kyle Suver, Weiqing Gu, Steven Xing, Cody Gabriel, and Peter Sankhagowit. Latent gaussian activity propagation: using smoothness and structure to separate and localize sounds in large noisy environments. *Advances in Neural Information Processing Systems*, 31, 2018.

[58] Kiran Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. Wideo: Fine-grained device-free motion tracing using RF backscatter. In *Proc. of NSDI*, 2015.

[59] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2017.

[60] Juhwan Kim and Son-Cheol Yu. Convolutional neural network-based real-time rov detection using forward-looking sonar image. In *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 396–400. IEEE, 2016.

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

[62] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using WiFi. In *ACM SIGCOMM Computer Communication Review*, volume 45(4), pages 269–282. ACM, 2015.

[63] Jan Kotera, Jiří Matas, and Filip Šroubek. Restoration of fast moving objects. *IEEE Transactions on Image Processing*, 29:8577–8589, 2020.

[64] Jan Kotera, Denys Rozumnyi, Filip Sroubek, and Jiri Matas. Intra-frame object tracking by deblatting. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision Workshops, pages 0–0, 2019.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *neural information processing systems*, 141(5):1097–1105, 2012.

[66] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *pre-print*, 2020. https://arxiv.org/pdf/2006.02535.pdf.

[67] Xiang-Yang Li, Huiqi Liu, Lan Zhang, Zhenan Wu, Yaochen Xie, Ge Chen, Chunxiao Wan, and Zhongwei Liang. Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint. *IEEE/ACM Transactions on Networking*, 27(5):1945–1958, 2019.

[68] Zhenyu Li, Zehui Chen, Ang Li, Liangji Fang, Qinhong Jiang, Xianming Liu, and Junjun Jiang. Unsupervised domain adaptation for monocular 3d object detection via self-training. *arXiv preprint arXiv:2204.11590*, 2022.

[69] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022.

[70] Yan-Bo Lin, Yu-Jhe Li, and Yu-Chiang Frank Wang. Dual-modality seq2seq network for audio-visual event localization. In *ICASSP 2019-*

2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2002–2006. IEEE, 2019.

[71] Tiantian Liu, Ming Gao, Feng Lin, Chao Wang, Zhongjie Ba, Jinsong Han, Wenyao Xu, and Kui Ren. Wavoice: A noise-resistant multi-modal speech recognition system fusing mmwave and audio signals. In *Proc. of ACM SenSys*, 2021.

[72] Xiaoxiao Long, Lingjie Liu, Wei Li, Christian Theobalt, and Wenping Wang. Multi-view depth estimation using epipolar spatio-temporal networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8258–8267, 2021.

[73] Yunfei Long, Daniel Morris, Xiaoming Liu, Marcos Castro, Punarjay Chakravarty, and Praveen Narayanan. Radar-camera pixel depth association for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12507–12516, 2021.

[74] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, volume 2013, pages 436–440, 2013.

[75] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, volume 2013, pages 436–440, 2013.

[76] Yawen Lu and Guoyu Lu. An alternative of lidar in nighttime: Unsupervised depth estimation based on single thermal image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3833–3843, 2021.

[77] Yuan Lukito and Antonius Rachmat Chrismanto. Recurrent neural networks model for wifi-based indoor positioning system. In *Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), 2017 International Conference on*, pages 121–125. IEEE, 2017.

[78] Y. Luo and N. Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *Proc. of ICASSP*, 2018.

[79] Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 46–50. IEEE, 2020.

[80] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 2019.

[81] Zhihong Luo, Qiping Zhang, Yunfei Ma, Manish Singh, and Fadel Adib. 3d backscatter localization for fine-grained robotics. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, pages 765–782, 2019.

[82] Qixiang Ma, Longyu Jiang, Wenxue Yu, Rui Jin, Zhixiang Wu, and Fangjin Xu. Training with noise adversarial network: A generalization method for object detection on sonar image. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 729–738, 2020.

[83] Yunfei Ma, Nicholas Selby, and Fadel Adib. Minding the billions: Ultra-wideband localization for deployed rfids. In *Proc. of ACM MobiCom*, 2017.

[84] Michael Mandel, Daniel Ellis, and Tony Jebara. An em algorithm for localizing multiple sound sources in reverberant environments. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

[85] Wenguang Mao, Jian He, and Lili Qiu. CAT: high-precision acoustic motion tracking. In *Proc. of ACM MobiCom*, 2016.

[86] Wenguang Mao, Mei Wang, Wei Sun, Lili Qiu, Swadhin Pradhan, and Yi-Chao Chen. Rnn-based room scale hand motion tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[87] Wenguang Mao, Mei Wang, Wei Sun, Lili Qiu, Swadhin Pradhan, and Yi-Chao Chen. Rnn-based room scale hand motion tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[88] Wenguang Mao, Zaiwei Zhang, Lili Qiu, Jian He, Yuchen Cui, and Sangki Yun. Indoor follow me drone. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 345–358. ACM, 2017.

[89] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 498–507, 2018.

[90] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1418, 2015.

[91] Paul Adrien Murice. Linearly constrained minimum variance beamforming. In J. Bourgeois and W. Minke, editors, *Time-Domain Beamforming and Blind Source Separation*, chapter 3. Clarendon Press, 1981.

[92] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, and Venkata N. Padmanabhan. Dhwani : Secure peer-to-peer acoustic nfc. In *Proc. of ACM SIGCOMM*, 2013.

[93] Rajalakshmi Nandakumar, Shyam Gollakota, and Nathaniel Watson. Contactless sleep apnea detection on smartphones. In *Proc. of ACM MobiSys*, 2015.

[94] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. FingerIO: Using active sonar for fine-grained finger tracking. In *Proc. of ACM CHI*, pages 1515–1525, 2016.

[95] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[96] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017.

[97] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017.

[98] Nvidia gtx 980, 2015. https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-980/specifications.

[99] Optimize and accelerate machine learning inferencing and training, 2021. https://onnxruntime.ai/.

[100] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

[101] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016.

[102] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[103] Ashutosh Pandey and DeLiang Wang. Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain. In *Proc. of ICASSP*, 2019.

[104] Ashutosh Pandey and DeLiang Wang. Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6875–6879. IEEE, 2019.

[105] Kranti Kumar Parida, Siddharth Srivastava, and Gaurav Sharma. Beyond image to depth: Improving depth prediction using echoes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8268–8277, 2021.

[106] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *European Conference on Computer Vision*, pages 109–125. Springer, 2020.

[107] Lucas C Parra and Christopher V Alvino. Geometric source separation: Merging convolutive source separation with geometric beamforming. *IEEE Transactions on Speech and Audio Processing*, 10(6):352–362, 2002.

[108] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. BeepBeep: a high accuracy acoustic ranging system using COTS mobile devices. In *Proc. of ACM SenSys*, 2007.

[109] Swadhin Pradhan, Wei Sun, Ghufran Baig, and Lili Qiu. Combating replay attacks against voice assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–26, 2019.

[110] PRB. International indicators: Average household size. `https://www.prb.org/international/indicator/hh-size-av/map/country/`.

[111] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. Widar2. 0: Passive human tracking with a single wi-fi link. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 350–361. ACM, 2018.

[112] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019.

[113] Snapdragon 845, 2018. https://www.qualcomm.com/products/snapdragon-845-mobile-platform.

[114] Caleb Rascon and Ivan Meza. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96:184–210, 2017.

[115] Acceptable jitter and latency for voip. `https://getvoip.com/blog/2018/12/20/acceptable-jitter-latency/`.

[116] Dario Rethage, Jordi Pons, and Xavier Serra. A wavenet for speech denoising. In *Proc. of IEEE ICASSP*, 2018.

[117] Md Alimoor Reza, Jana Kosecka, and Philip David. Farsight: Long-range depth estimation from outdoor images. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4751–4757. IEEE, 2018.

[118] Bertrand Rivet, Wenwu Wang, Syed Mohsen Naqvi, and Jonathon A Chambers. Audiovisual speech source separation: An overview of key methodologies. *IEEE Signal Processing Magazine*, 2014.

[119] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[120] Denys Rozumnyi, Jan Kotera, Filip Sroubek, and Jiri Matas. Sub-frame appearance and 6d pose estimation of fast moving objects. In *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6778–6786, 2020.

[121] Hiroshi Saruwatari, Satoshi Kurita, Kazuya Takeda, Fumitada Itakura, Tsuyoki Nishikawa, and Kiyohiro Shikano. Blind source separation combining independent component analysis and beamforming. *EURASIP Journal on Advances in Signal Processing*, 2003(11):1–12, 2003.

[122] Ralph Otto Schmidt. A signal subspace approach to multiple emitter location spectral estimation. *Ph. D. Thesis, Stanford University*, 1981.

[123] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding WiFi localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 249–262. ACM, 2013.

[124] Sheng Shen, Daguan Chen, Yu-Lin Wei, Zhijian Yang, and Romit Roy Choudhury. Voice localization using nearby wall reflections. In *Proc. of ACM MobiCom*, 2020.

[125] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[126] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *international conference on learning representations*, 2015.

[127] Nikhil Singh, Jeff Mentch, Jerry Ng, Matthew Beveridge, and Iddo Drori. Image2reverb: Cross-modal reverb impulse response synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 286–295, 2021.

[128] Mehrez Souden, Jacob Benesty, and Sofiène Affes. A study of the lcmv and mvdr noise reduction filters. *IEEE Transactions on Signal Processing*, 58(9):4925–4935, 2010.

[129] Statista. Distribution of households in the united states from 1970 to 2020, by household size. `https://www.statista.com/statistics/242189/disitribution-of-households-in-the-us-by-household-size/`.

[130] StereoLab. Zed 2 camera dataset. `https://www.stereolabs.com/assets/datasheets/zed2-camera-datasheet.pdf`, 2021.

[131] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE, 2021.

[132] Aswin Shanmugam Subramanian, Chao Weng, Shinji Watanabe, Meng Yu, Yong Xu, Shi-Xiong Zhang, and Dong Yu. Directional asr: A new paradigm for e2e multi-speaker speech recognition with source localization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8433–8437. IEEE, 2021.

[133] Ke Sun and Xinyu Zhang. Ultrase: Single-channel speech enhancement using ultrasound. In *Proc. of ACM MobiCom*, 2021.

[134] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 591–605. ACM, 2018.

[135] L. Sun, S. Sen, D. Koutsonikolas, and K. Kim. WiDraw: enabling hands-free drawing in the air on commodity WiFi devices. In *Proc. of ACM MobiCom*, 2015.

[136] Wei Sun, Mei Wang, and Lili Qiu. Spatial aware multi-task learning based speech separation. *arXiv preprint arXiv:2207.10229*, 2022.

[137] Yoiti Suzuki and Hisashi Takeshima. Equal-loudness-level contours for pure tones. *The Journal of the Acoustical Society of America*, 116(2):918–933, 2004.

[138] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *In Proc. of IEEE ICASSP*, 2010.

[139] Chuanxin Tang, Chong Luo, Zhiyuan Zhao, Wenxuan Xie, and Wenjun Zeng. Joint time-frequency and time domain learning for speech enhancement. In *Proc. of IJCAI*, 2020.

[140] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. of ECCV*, 2020.

[141] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 247–263, 2018.

[142] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[143] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[144] J-M Valin, François Michaud, Jean Rouat, and Dominic Létourneau. Robust sound source localization using a microphone array on a mobile robot. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 2, pages 1228–1233. IEEE, 2003.

[145] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single WiFi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, 2016.

[146] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016.

[147] Video conference statistics. https://www.lifesize.com/en/blog/vi-deo-conferencing-statistics/.

[148] Anran Wang and Shyamnath Gollakota. Millisonic: Pushing the limits of acoustic motion tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 18. ACM, 2019.

[149] Chuyu Wang, Jian Liu, Yingying Chen, Hongbo Liu, Lei Xie, Wei Wang, Bingbing He, and Sanglu Lu. Multi-touch in the air: Device-free finger tracking and gesture recognition via cots rfid. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1691–1699. IEEE, 2018.

[150] Jue Wang, Deepak Vasisht, and Dina Katabi. RF-IDraw: virtual touch screen in the air using RF signals. In *Proc. of ACM SIGCOMM*, 2014.

[151] Mei Wang, Wei Sun, and Lili Qiu. Localizing human voice. In *In Proc. of NSDI*, 2021.

[152] Mei Wang, Wei Sun, and Lili Qiu. {MAVL}: Multiresolution analysis of voice localization. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 845–858, 2021.

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

[154] Xingmei Wang, Jia Jiao, Jingwei Yin, Wensheng Zhao, Xiao Han, and Boxuan Sun. Underwater sonar image classification using adaptive weights convolutional neural network. *Applied Acoustics*, 146:145–154, 2019.

[155] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Longand, and L. Fei-Fei. "eidetic 3d lstm: A model for video prediction and beyond. In *Proc. of ICLR*, 2019.

[156] Y. Wang, A. Narayanan, and D. Wang. On training targets for supervised speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, page 1849–1858, 2014.

[157] Mati Wax, Tie-Jun Shan, and Thomas Kailath. Spatio-temporal spectral analysis by eigenstructure methods. *IEEE transactions on acoustics, speech, and signal processing*, 32(4):817–827, 1984.

[158] Teng Wei and Xinyu Zhang. mTrack: high precision passive tracking using millimeter wave radios. In *Proc. of ACM MobiCom*, 2015.

[159] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le

Roux. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*, 2019.

[160] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*, 2019.

[161] David P Williams. Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2497–2502. IEEE, 2016.

[162] D. S. Williamson and Y. Wang. Complex ratio masking for monaural speech separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, page 483–492, 2016.

[163] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*, 2021.

[164] Yu Wu, Linchao Zhu, Yan Yan, and Yi Yang. Dual attention matching for audio-visual event localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6292–6300, 2019.

[165] Yan Xia and Zhou Zhao. Cross-modal background suppression for audio-visual event localization. In *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 19989–19998, 2022.

[166] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise power delay profiling with commodity WiFi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 53–64. ACM, 2015.

[167] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, pages 71–84, 2013.

[168] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface. In *Proc. of MobiSys*, 2019.

[169] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances in Neural Information Processing Systems*, 32, 2019.

[170] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):7–19, 2014.

[171] Yong Xu, Meng Yu, Shi-Xiong Zhang, Lianwu Chen, Chao Weng, Jianming Liu, and Dong Yu. Neural spatio-temporal beamformer for target speech separation. *arXiv preprint arXiv:2005.03889*, 2020.

[172] Yong Xu, Zhuohuang Zhang, Meng Yu, Shi-Xiong Zhang, and Dong Yu. Generalized spatio-temporal rnn beamformer for target speech separation. *arXiv preprint arXiv:2101.01280*, 2021.

[173] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using cots devices. In *Proc. of ACM MobiCom*, 2014.

[174] Dacheng Yin, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Phasen: A phaseand-harmonics-aware speech enhancement network. In *In Proceedings of AAAI*, 2020.

[175] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Newcrfs: Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[176] Sangki Yun, Yi chao Chen, and Lili Qiu. Turning a mobile device into a mouse in the air. In *Proc. of ACM MobiSys*, May 2015.

[177] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. Strata: Fine-grained acoustic-based device-free tracking. In *Pro-*

ceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, pages 15–28. ACM, 2017.

[178] Diana Zhang, Akarsh Prabhakara, Sirajum Munir, Aswin Sankaranarayanan, and Swarun Kumar. A hybrid mmwave and camera system for long-range depth imaging. *arXiv preprint arXiv:2106.07856*, 2021.

[179] J. Zhang, Y. Wang, M. Long, W. Jianmin, and P. S. Yu. Z-order recurrent neural networks for video prediction. In *Proc. of ICME*, 2019.

[180] Kai Zhang, Jiaxin Xie, Noah Snavely, and Qifeng Chen. Depth sensing beyond lidar range. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2020.

[181] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proc. of ACM MobiSys*, 2012.

[182] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[183] Zhuohuang Zhang, Yong Xu, Meng Yu, Shi-Xiong Zhang, Lianwu Chen, and Dong Yu. Adl-mvdr: All deep learning mvdr beamformer for target speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6089–6093. IEEE, 2021.

[184] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1735–1744, 2019.

[185] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586, 2018.

[186] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[187] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. Rf-based 3d skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 267–281. ACM, 2018.

[188] Jinxing Zhou, Jianyuan Wang, Jiayi Zhang, Weixuan Sun, Jing Zhang, Stan Birchfield, Dan Guo, Lingpeng Kong, Meng Wang, and Yiran Zhong. Audio-visual segmentation. *arXiv preprint arXiv:2207.05042*, 2022.

[189] Hao Zhu, Man-Di Luo, Rui Wang, Ai-Hua Zheng, and Ran He. Deep audio-visual learning: A survey. *International Journal of Automation and Computing*, pages 1–26, 2021.

[190] Pingping Zhu, Jason Isaacs, Bo Fu, and Silvia Ferrari. Deep learning feature extraction for target recognition and classification in underwater sonar images. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2724–2731. IEEE, 2017.