박 사 학 위 논 문
Ph.D. Dissertation

로봇을 위한 새로운 음향 신호를 이용한 음원 위치 추적

Sound Source Localization with Novel Acoustic Cues for Robots

2023

안 인 규 (安 仁 珪 An, Inkyu)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

박 사 학 위 논 문

로봇을 위한 새로운 음향 신호를 이용한 음원 위치 추적

2023

안 인 규

한 국 과 학 기 술 원

전산학부

로봇을 위한 새로운 음향 신호를 이용한 음원 위치 추적

안 인 규

위 논문은 한국과학기술원 박사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2023년 5월 12일

심사위원장 윤 성 의 (인)
심 사 위 원 김 민 혁 (인)
심 사 위 원 안 성 진 (인)
심 사 위 원 명 현 (인)
심 사 위 원 최 정 우 (인)

Sound Source Localization with Novel Acoustic Cues for Robots

Inkyu An

Advisor: Sung-Eui Yoon

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science

Daejeon, Korea
May 12, 2023

Approved by

Sung-Eui Yoon
Professor of School of Computing

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

DCS 안인 규. 로봇을 위한 새로운 음향 신호를 이용한 음원 위치 추적. 전산학부 . 2023년. 69+iv 쪽. 지도교수: 윤성의. (영문 논문)
Inkyu An. Sound Source Localization with Novel Acoustic Cues for Robots. School of Computing . 2023. 69+iv pages. Advisor: Sung-Eui Yoon. (Text in English)

초 록

로봇 청각, 특히 음원 위치 추적은 로봇 개발의 중요한 부분입니다. 그러나 실세계 음원 위치 추적 알고리즘은 비가시 음원, 소음 간섭, 다중 소리 이벤트 등 많은 도전 과제에 직면합니다. 이러한 복잡성을 해결하기 위해, 저는 음향 광선, 역전파 신호, 확장형 마이크 쌍 훈련을 통한 강건한 도착 시간 차이 모델 (Robust-TDoA 모델)로 구성된 세 가지 **새로운 음향 신호**를 소개합니다. 음향 광선은 직접, 반사 및 회절된 전파 경로를 추정할 수 있어 비가시 음원의 위치를 추정하는데 효과적입니다. 역전파 신호는 마이크 어레이 오디오를 활용해 음원에서 로봇으로의 전파 경로로 전달된 소리 신호를 역추적하여 계산됩니다. 특히, 역전파 신호를 활용해 비가시 음원의 위치 추적 성능을 향상시킬 수 있습니다. 확장형 마이크 쌍 훈련을 통한 강건한 도착 시간 차이 모델 (Robust-TDoA 모델)은 다양한 유형의 마이크 어레이로부터 수집된 여러 데이터 세트가 포함하는 다양한 상황을 학습할 수 있습니다. 이 훈련 과정 후, Robust-TDoA 모델은 마이크 어레이 유형에 관계없이 음성 위치 추적 및 소리 이벤트 위치 추적 및 감지 (SELD) 작업과 같은 다양한 음원 위치 추적 잡업에 적용될 수 있습니다. 이 제안된 접근 방식의 효능을 도전적인 상황에서 검증했으며, 본 학위 논문에서 제시한 새로운 음향 단서 덕분에 만족스러운 성능을 보여주었습니다.

핵 심 낱 말 음원 위치 추적, 로봇 청각, 사람-로봇 상호작용, 음향 특징, 광선 추적, 주의집중 구조

Abstract

Robot audition, particularly Sound Source Localization (SSL), is a crucial aspect of robotic development. However, real-world SSL applications present numerous challenges, including non-line-of-sight (NLOS) sound sources, noise interference, and multiple concurrent sound events. To tackle these complexities, I introduces three **novel acoustic cues** consisting of Acoustic Rays, Back-propagation Signals, and a Robust-Time Difference of Arrival Model (Robust-TDoA Model) with Scalable Microphone Pair Training. Acoustic rays, capable of estimating direct, reflected, and diffracted propagation paths, offer an efficient tool for identifying NLOS sources. Back-propagation signals, calculated by reverse tracing sound signals through these propagation paths from a source to a robot, leverage microphone array audio data. These signals serve to enhance the localization performance, particularly for NLOS sources. The Robust-TDoA Model, in conjunction with Scalable Microphone Pair Training, facilitates learning from diverse situations presented in multiple datasets, collected through diverse types of microphone arrays. After this training process, the Robust-TDoA Model can adapt to various SSL tasks, including speech-oriented SSL and Sound Event Localization and Detection (SELD) tasks, regardless of the microphone array type. The efficacy of these proposed approaches was examined in challenging environments, demonstrating satisfactory performance due to these novel acoustic cues.

Keywords Sound source localization, Robot Audition, Human-Robot Interaction, Acoustic feature, Ray tracing, Attention mechanism

Contents

List of Tables

List of Figures

Chapter 1. Introduction

As mobile robots become increasingly commonplace in various fields, the development of innovative localization methods is attracting substantial interest. The main goal of these methods is to determine the current position of the mobile robot in relation to its environment, generally leveraging a pre-existing map and multiple sensors to estimate the robot's position and orientation. Frequently used sensors include GPS, charge-coupled devices (CCD), depth cameras, and acoustic ones.

The use of acoustic sensors for sound source localization (SSL), i.e., pinpointing active sound sources, has recently gained popularity. This trend is evident in the deployment of sonar signal processing for underwater localization and microphone arrays for indoor and outdoor settings. The recent use of smart microphones in commodity or IoT devices (e.g., Amazon Alexa) has triggered interest in better SSL methods [3, 4].

SSL in the context of robotics presents significant challenges due to the complex environments in which robots often operate (Figure. 1.1). These environments may contain numerous obstacles, such as walls or ceilings, resulting in a variety of propagation paths, like direct, reflection, and diffraction. Additionally, background noises, such as air conditioning or robot-generated noise, can interfere with SSL. Further complexity is added by moving sound sources obscured by obstacles (becoming non-line-of-sight (NLOS) sources) and simultaneous occurrence of various sound events like speech and clapping."

Numerous approaches leverage signal processing techniques to tackle the SSL problem in robotics, chiefly by estimating the direction of arrival (DoA) of sound. SSL research often utilizes the time difference of arrival (TDoA) at the receiver [5–7]. Techniques such as beamforming [8, 9] and subspace-based methods [10–12] have been employed for sound source localization.

Recent developments have aimed to pinpoint the locations of sound sources, going beyond merely estimating the DoA. Some methods localize positions under constraints by accumulating incoming sensor data, corresponding to the DoA of direct sound, measured from varied locations and orientations [13–15]. Others endeavor to localize moving sources with intermittent sound signals through a filtering process [1, 16, 17].

While these existing SSL strategies estimating DoAs and source positions have made significant progresses, they often fall short in challenging environments. These techniques mainly exploit the direct sound and its direction, i.e., the DoA at the receiver, without considering indirect sounds such as reflections and diffractions. For instance, when a moving sound source becomes an NLOS source as in Figure. 1.1, the contribution from direct sound may be minimal, leading to the possible deterioration of the accuracy of conventional SSL approaches.

Several deep learning (DL)-based methods have been introduced to tackle issues in complex environments, such as localizing simultaneous sources and distinguishing sound events in noisy conditions like those illustrated in Figure.1.1. These are challenges that existing signal processing-based methods have been unable to sufficiently resolve. Some techniques [18, 19] have proposed solutions for speech-oriented SSL in noisy environments. Others [20–23] have introduced sound event localization and detection (SELD) techniques, leveraging multi-label sound event datasets, such as speech and alarm.

However, these DL approaches were developed to operate with a specific type of microphone array, thereby facing scalability issues when different microphone array types are involved. Logically, these DL methods are not suited to work with various types of microphone arrays due to this scalability issue,

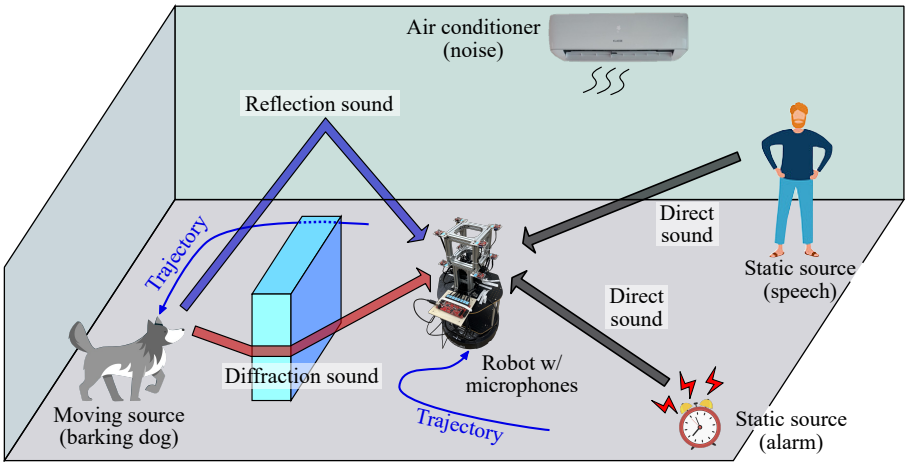


Figure 1.1: An example of a challenging environment of sound source localization (SSL) with a robot. The robot moves around an environment, and simultaneous moving and static sources of various sound events, e.g., alarm, speech, and dog bark, exist. The noises caused by the air conditioner and the movement of moving robots and sources can decrease the SSL performance. The moving sound source becomes a non-line-of-sight (NLOS) source when it is occluded by an obstacle.

which subsequently leads to several problems in robotics.

These DL methods cannot be trained with multiple datasets [18, 20, 24] collected using diverse types of microphone arrays. These multiple datasets are beneficial for challenging environments as they encapsulate various scenarios, such as static and dynamic sources, single and simultaneous sources, and different sound events. Furthermore, once these DL methods are trained on a specific type of microphone array, they cannot be employed with a different type of microphone array.

As robots are typically equipped with different types of microphone arrays, this limitation poses a significant issue. For instance, if I operate a robot with a particular type of microphone array and lack a corresponding dataset, I would have to gather a new dataset based on the existing DL methods. However, the collection of adequate data is both time-consuming and costly.

In this dissertation, I propose **novel acoustic cues**, specifically *Acoustic rays* (Chapter.2), *Back-propagation signals* (Chapter.3), and a *Robust-TDoA model with scalable microphone pair training* (Chapter.4). These are designed to handle the complexities in challenging environments (Figure.1.1), such as localizing non-line-of-sight (NLOS) sources, maintaining robustness against noise, and localizing simultaneous sources of sound events.

Initially, I propose unique acoustic rays considering reflection and diffraction (Chapter. 2) for localizing the NLOS source. In an NLOS scenario, direct sound is blocked by an obstacle, causing the sound to reach the robot via indirect propagation paths such as diffraction and reflection. These acoustic rays are generated to estimate propagation paths of direct, reflective, and diffractive sounds. Then, I use these acoustic rays of various propagation paths to identify the sound source location. Significantly, these acoustic rays, which take into account reflection and diffraction paths, aid in localizing NLOS sources.

Next, I propose a back-propagation signal for increased noise robustness (Chapter. 3). In the initial acoustic rays method, certain rays may be inaccurately generated due to ambient noises, which could negatively impact SSL performance. To mitigate this, it is essential to verify whether the Acoustic Rays originated from the actual sound source or were a result of noise. The back-propagation signal, a hypothetical signal that could be heard at a specific location on acoustic rays, serves this purpose.

Assuming the back-propagation signal is calculated at a point close to the actual source, it would bear similarity to the source signal, unlike a signal produced by erroneous acoustic rays caused by noise. By comparing these back-propagation signals, the method discerns whether the acoustic rays were generated by the actual sound source or noise, thereby enhancing SSL performance even amidst noise.

Lastly, I suggest a robust-TDoA model coupled with scalable microphone pair training to handle challenging environments featuring simultaneous sources of various sound events amidst noise (Chapter. 4). Although some DL-based methods have been proposed to localize simultaneous sound sources in noisy conditions, their successful operation in these environments hinges on ample training datasets. However, these methods face scalability issues based on the type of microphone array, as the training dataset must be gathered using a fixed array type. This limitation impedes the growth of the training dataset through multiple collections from different microphone array types. Furthermore, these DL-based methods may necessitate a new dataset if the dataset of robot-mounted microphone array is unavailable.

In contrast, the proposed scalable microphone pair training allows the model to train with multiple datasets recorded by various types of microphone arrays. These diverse datasets [18, 20, 24] encompass a range of challenging scenarios like simultaneous sources of sound events in noisy environments, which bolster SSL performance in these testing conditions. Post the scalable microphone pair training process, the proposed method can adapt to various microphone array types via an array geometry-aware training procedure, even in cases where the dataset of the desired microphone array is unavailable.

Chapter 2. Diffraction- and Reflection-Aware Sound Source Localization

2.1 Introduction

There have been efforts to model reflection sound in addition to direct sound based on ray tracing techniques [25–27]. They approximate the direct and reflection propagation paths of sound using acoustic rays generated by a ray tracing technique, and the source positions are estimated from computing intersections of acoustic rays. These methods are efficient in localizing the NLOS source by modeling reflection propagation paths. However, they assume that there is only one sound source, and thus additional process should be necessary to identify multiple source positions among many intersections. Moreover, the time to calculate the intersections increases as the number of acoustic rays increase; the time complexity of computing intersections is $O(N^2)$ where N is the number of acoustic rays. In the tests in this Chapter. 2, 75 acoustic rays have been produced in a single frame on average. It is difficult to compute every intersection of acoustic rays in a real-time.

The ray tracing technique, a type of geometric acoustic techniques [28–31], assumes the rectilinear propagation of sound waves and fits into high-frequency sounds; specular reflection is one of the high-frequency phenomena. They do not model many low-frequency phenomena such as diffraction, which is a type of scattering that occurs in the presence of obstacles whose sizes are of the same order of magnitude as the wavelength. In practice, diffraction is a fundamental mode of sound wave propagation and occurs frequently in building interiors, e.g., when the source is behind an obstacle or hidden by walls. These effects are more prominent for low-frequency sources, such as vowel sounds in human speech, industrial machinery, ventilation, air-conditioning units.

Main contribution. I present a novel sound localization algorithm that takes into account diffraction as well as reflection, even from non-light-of-sight sources and intermittent sound signals. A key aspect of the proposed work is that it models diffraction propagation paths of sounds by the acoustic rays and identifies multiple source locations from acoustic rays satisfying a real-time operation. The diffraction propagation paths are modeled by the Uniform Theory of Diffraction (UTD) [32] along the wedges and approximated by the diffraction acoustic rays. The proposed method efficiently identifies multiple source positions using Monte-Carlo localization, which finds out the convergence regions of rays, corresponding to intersections of rays.

The proposed approach supports the iterative computation and during every iteration can localize multiple dynamic sources as well as non-line-of-sight sources by modeling the reflection and diffraction of acoustic rays. Furthermore, The proposed approach can distinguish between active and inactive states of intermittent sound signals in addition to continuous sounds.

During the precomputation phase, I use SLAM and primitive fitting techniques to reconstruct the 3D map information of an indoor environment, specifically a 3D triangular mesh and wedges of obstacles. At runtime, I generate primary acoustic rays towards the incoming sound directions as computed by a DOA estimator. Once the acoustic ray hits the reconstructed mesh, I generate reflection rays (Chapter. 2.3.2). Furthermore, when acoustic rays satisfy the proposed diffraction-criterion, e.g., hitting on the edge of the wedge, I also generate diffraction acoustic rays (Chapter. 2.3.3). I estimate multiple source positions by performing Monte-Carlo localization to identify the ray convergence given generated acoustic rays

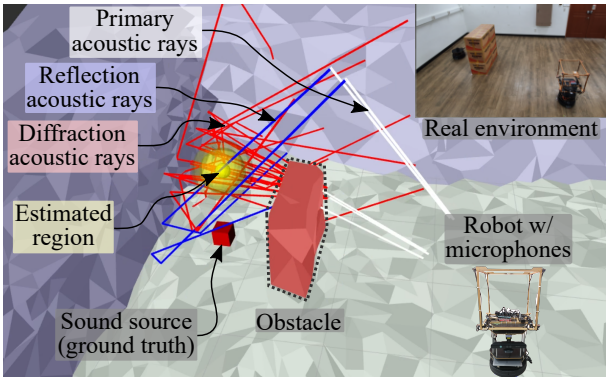


Figure 2.1: A robot, equipped with a cube-shaped microphone array, localizes a source position in a 3D space. The proposed formulation takes into account both direct and indirect sound propagation given its use of acoustic rays. The acoustic rays are initialized and propagated based on the proposed backward acoustic ray tracing algorithm that considers reflection and diffraction; primary, reflection, and diffraction acoustic rays are shown in white, blue, and red lines, respectively. The yellow disk, which is very close to the ground truth, represents a 95 % confidence ellipse with regard to the estimated sound source, as computed by the proposed approach.

(Chapter. 2.4).

I evaluated the proposed method in various scenarios in two indoor environments, one 7 m by 7 m in size and the other 7 m by 3.5 m in size and a height of 3 m. I also tested the proposed approach in different environmental or experimental setups and applied the proposed approach to the other task navigating to the NLOS source. Given these test environments, the proposed method achieves low average errors, e.g., 0.6159 m and 0.7364 m for clapping sound and human speech, respectively, even with a moving source and an obstacle occluding the line-of-sight between the listener and the source. Furthermore, the proposed method demonstrates high performance, in this case 0.5919 m and 0.5271 m, respectively, on clapping sounds and human speech with multiple stationary and dynamic sources. I also compared accuracies of the proposed method to the previous work [1] which does not consider indirect sound.

2.2 Related Works

Sound source localization (SSL) methods have been studied to overcome the difficulty encountered by a robot when attempting to identify a speaker such as a human, machine, or even another robots, in an real environment. I explain previous research on SSL (Chapter. 2.2.1), after which I introduce physical-based modeling techniques that enable realistic sound generation for simulators (Chapter. 2.2.2). The proposed approach is inspired by these physical-based modeling methods.

2.2.1 Sound source localization (SSL)

There have been many efforts to identify the sound source location and many of them have focused on estimating the direction of arrival (DoA).

For simple and fast DOA estimators in a 2D space, many methods have been proposed using microphone pair signals and their time difference of arrival (TDoA). The TDOA can be estimated by using a generalized cross correlation with phase transform (GCC-PHAT) [5, 6] and a difference singular value

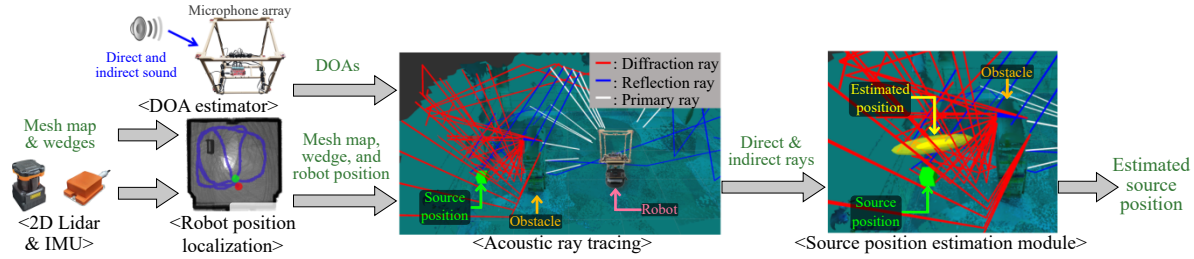


Figure 2.2: This figure demonstrates the run-time computations using acoustic ray tracing for sound source localization. Acoustic ray tracing is performed from DOAs, a mesh map containing wedges, and a robot position where a DOA estimator works on a cube-shaped eight-microphone array. The robot position is estimated by 2D SLAM from a 2D Lidar sensor, and the mesh map and wedges are generated during the precomputation phase. Source position estimation is performed by identifying ray convergence from the generated acoustic ray paths.

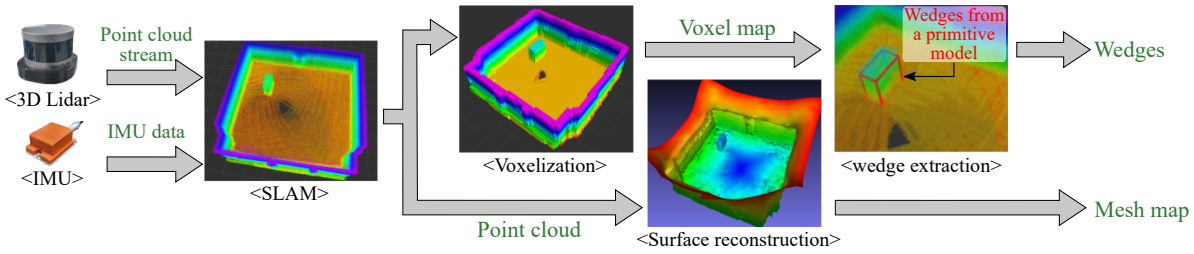


Figure 2.3: This figure shows the precomputation phase. I use SLAM to generate a point cloud of an indoor environment from IMU and 3D Lidar, and the mesh map is reconstructed via surface reconstruction techniques. To extract the wedge information, I utilize voxelization from the point cloud and fit a primitive model, e.g., a box model in this case, onto the voxel map. Wedges are then extracted from the fitted primitive model. The extracted wedges of the fitted box model are highlighted by the red line.

decomposition with phase transform (DSVD-PHAT) [7] from a microphone pair. Using microphone pair signals and their TDoA, Carlo *et al.* [33] propose a method for 2D sound source localization, i.e., DoA estimation, by considering echoes.

Beamforming or subspace-based methods have been suggested to estimate DoA in a system using multiple microphones, i.e., microphone array. A delay-and-sum beamformer was proposed for fast DOA estimation using the cross correlation operation with 8-microphones [34]. The fast and accurate speaker identification system for distributed meeting was suggested by using a minimum variance distortionless response (MVDR) beamforer [9]. Nakamura *et al.* [12] suggested a multiple signal classification based on generalized singular value decomposition (GSVD-MUSIC) algorithm for a robust and real-time localization method.

Recently, there have been demands for estimating the actual location of the sound source, not just DoA. Valin *et al.* [1] suggested a 3D sound source localization and tracking system based on a steered beamformer and particle filter. Portello *et al.* [17] presented an active source localization of an intermittent signal considering a motion of a moving binaural sensor. Nguyen *et al.* [16] presented a 2D

sound source localization method by updating sequential acoustic data using a mixture kalman filter. Sasaki *et al.* [13] and Su *et al.* [14] presented 3D sound source localization algorithms using a disk-shaped sound detector and a linear microphone array such as Kinect and PS3 Eye. Even *et al.* [25] presented a probabilistic 3D mapping algorithm of sound sources accumulating acoustic information of direct sound on an occupancy grid map. These approaches consider only direct sound, and thus are not designed for scenarios containing NLOS sources.

To localize the NLOS source, some methods have been presented based on a ray tracing technique. Kallakuri *et al.* [26] and Even *et al.* [27] suggested the NLOS source localization algorithm by modeling reflection. They produced reflected acoustic rays and computed the intersection of those rays corresponding to the source position. The proposed approach models diffraction as well as reflection. The proposed approach also computes the convergence of those rays based on the Monte Carlo localization to identify the source position; the Monte Carlo localization-based technique is more efficient and faster than computing the intersection of rays.

2.2.2 Interactive sound propagation

There has been considerable work in acoustics and physically-based modeling to develop fast and accurate sound simulators that can generate realistic sounds for computer-aided design and virtual environments. Geometry acoustic (GA) techniques have been widely utilized to simulate sound propagation efficiently using ray tracing techniques, and those ray tracing techniques are efficient to model sound propagations at high frequencies.

At high frequencies, the propagation of the sound waves can be approximated as traveling in straight and bouncing off the boundaries [28]. An estimation of the acoustic impulse response of high-frequency propagation between the source and the listener was performed using image-source-based ray tracing [29], Monte Carlo path tracing [30], or a hybrid combination of geometric and numeric techniques [31].

Low-frequency wave phenomenons, i.e. diffraction, need to be modeled separately since ray tracing algorithms are inappropriate for sound propagation models at low frequencies. Exact methods to model diffraction are based on directly solving the acoustic wave equation using numeric methods like boundary or finite element methods [35,36], the wave-geometric approximation method [37], the Kresnel-Kirchoff approximation method [38], or the BTM model [39] and its extension to higher order diffraction models [40].

Commonly used techniques to model diffraction with geometric acoustic methods are based on two models: the Uniform Theory of Diffraction (UTD) [41] and the Biot-Tolstoy-Medwin (BTM) model [39]. The BTM model is an accurate diffraction formulation that computes an integral of the diffracted sound along the finite edges in the time domain [36, 40, 42]. In practice, the BTM model is more accurate, but is limited to non-interactive applications. The UTD model approximates an infinite wedge as a secondary source of diffracted sounds, which can be reflected and diffracted again before reaching the listener. UTD based approaches have been effective for many real-time sound generation applications, especially in complex environments with occluding objects [30, 43–45].

The proposed approach, backward acoustic ray tracing, is motivated by these real-time simulation and proposes real-time source localization algorithm using ray tracing and UTD.

2.3 Acoustic Ray Tracing Handling Diffraction and Reflection

Motivation. After a source emits a sound, sound waves are propagated to free space and cause various interactions with obstacles; e.g., reflections occur after the sound wave hits obstacles, and diffractions arise at the boundary of the obstacles, such as an edge of wedges. While direct propagation paths are defined as paths propagating directly from a source to a listener without any interactions, a range of other interactions cause many types of indirect propagation paths of sound waves.

When the sound waves reach the microphone array through direct and indirect propagation paths, I can estimate the DoA (direction-of-arrival), Θ^*, of sound waves using the DAS beamformer. However, I cannot determine whether the DOA came from a direct or indirect sound propagation path. Many beamformers have focused on estimating DoAs came from direct propagation paths, but indirect, i.e., reflection and diffraction, propagation paths frequently occur. Especially, if the sound source becomes a non-line-of-sight source located in the invisible area for the microphone array, the indirect propagation path becomes a prominent path of sound propagation, and the beamformer cannot identify the DoA came from direct propagation paths. Furthermore, beamforming techniques do not localize the source position in environments but compute only the DoA. Thus, a new type of sound source localization algorithm is needed to identify 3D source positions.

Overview. I propose a novel sound source localization method that is a type of reflection and diffraction-aware sound source localization method. In indoor environments, there are many types of obstacles, e.g., walls, ceiling, and objects. They cause various interactions, i.e., direct, reflection, and diffraction, with sound waves, and a sequence of these interactions denotes a propagation path from a source to a measurement device. I want to estimate propagation paths considering reflection and diffraction using a ray tracing technique from signals measured by a microphone array, i.e., the eight-channel cube-shaped microphone array shown in Figure. 2.2. I then identify the positions of multiple sources based on the estimated propagation paths.

Before performing sound source localization at runtime, the proposed method reconstructs the structures of an indoor environment, i.e., the surfaces and wedges of objects, in order to handle the reflection and diffraction interactions of the sound waves (Figure. 2.3). Using a SLAM algorithm [2] with IMU and 3D Lidar, I extract a registered point set representing the indoor environment. I then generate a mesh map using surface reconstruction techniques, i.e., screened poisson surface reconstruction [46] or simplification [47], from the registered point cloud. The proposed method uses the mesh map to estimate the reflection of sound propagation. Furthermore, the proposed framework extracts wedges of objects to estimate the diffraction. I use a primitive fitting technique [48] to detect the wedges given a voxelization map of the point cloud. I then extract the edges from the wedges of the primitive having no contact with the floor.

A runtime overview of the proposed approach is shown in Figure. 2.2. First, I estimate incoming directions of the propagation paths using a DoA estimator (Chapter. 2.3.1). The proposed approach is basically built upon the delay-and-sum beamformer [8] of a cube-shaped eight-channel microphone array; it can be combined with different DoA estimators and microphone arrays (Chapter. 2.5.4) , e.g., an eigenbeam-minimum variance distortionless response (EB-MVDR) beamformer of a 32-channel spherical microphone array [49, 50]. I then generate the acoustic rays considering both reflection and diffraction based on ray tracing techniques. The proposed acoustic ray tracing algorithm initializes a primary acoustic ray from each estimated DoA and propagates it through free space. If the acoustic ray hits a surface of obstacle, I generate a reflection acoustic ray by considering specular reflection

(Chapter. 2.3.2). Additionally, when the acoustic ray satisfies the diffraction condition at a wedge, defined by the *diffractability*, the diffraction acoustic ray is generated based on a uniform theory of diffraction (UTD) model (Chapter. 2.3.3). Finally, the acoustic ray paths, a set of acoustic rays that originate from the same DoA, represent the estimated propagation paths of sound waves.

After generating the acoustic rays, I identify multiple source positions using acoustic ray paths (Chapter. 2.4). Because the propagation paths of sound waves propagate from the sources to the listener, i.e., the measurement device in this case, the estimated propagation path represented by the acoustic ray path should pass through sound source positions. Acoustic ray paths can therefore converge to each source position, and I find the convergence regions of acoustic rays and determine these locations as multiple-sound-source positions.

2.3.1 Estimating the Direction-of-Arrival (DoA) of Sound

Obstacles such as walls, ceiling and objects cause various propagation paths in indoor environments, and different paths caused by the same source can propagate to the measurement device from different DoAs. Given the measured sound pressures of L samples in a single frame, it is necessary to estimate multiple DoAs, as there can be more than one DoA. Given this problem, I utilize a beamforming algorithm to estimate tuples containing a DoA Θ_n^* and its average beamforming power β_n over angular frequencies:

$$[(\Theta_0^*, \beta_0), \cdots, (\Theta_N^*, \beta_N)] = \max_{\Theta}{}^{N} (\frac{1}{L} \sum_{\nu=0}^{L-1} \beta(\Theta, \omega_\nu)), \tag{2.1}$$

where $\max^N$ denotes the function of finding N tuples, (Θ_n^*, β_n) where $N = 4$, with large average beamforming powers, ω_ν is the ν-th angular frequency, and β is the beamforming power of the ν-th angular frequency at direction Θ; I refer beamforming formulas in [51]. I create 2562 points on the unit sphere from an icosahedral grid [52], and Θ is a specific direction $[\theta, \phi]$ corresponding to one of those points. I utilize a cube-shaped eight-channel microphone array with DAS beamformer [34]; however, the proposed approach works properly with different types of microphone arrays and other beamformers as well (Chapter. 2.5.4).

I initialize a primary acoustic ray from a tuple (Θ_n^*, β_n). The primary acoustic ray, r_n^0, is generated into the reverse direction of Θ_n^*:

$$r_n^0(l) = \hat{d}_n^0 \cdot l + \dot{o}, \tag{2.2}$$

where l is the ray length of a primary acoustic ray, $\hat{d}_n^0$ denotes the unit vector of the reverse direction of Θ_n^*, and $\dot{o}$ represents the origin of the microphone array. The superscript k of an acoustic ray, r_n^k, indicates the order of interactions, i.e., reflection or diffraction, along an acoustic ray path from the microphone array. For example, $r_n^0(l)$ indicates that there is no interaction, thus denotes a primary ray having the ray length l from the microphone array. All the other rays with a varying number of interactions, i.e., $k \geq 1$, are referred to as indirect acoustic rays with k-th order interactions.

When the primary acoustic ray r_n^0 is generated in Eq. 2.2, the primary ray is initialized with initial energy of β_n, which represents the incoming power from the n-th DoA. The energy of sound waves decreases with respect to the travel distance of the propagation path from the source to the listener and the absorption coefficient: $E(l) = E_0 \cdot 1/(1 + l^2) \cdot (1 - \alpha)^K$, where $E(l)$ is the energy when the sound wave propagates by distance l, E_0 denotes the initial energy of the sound waves at the sound source, and α is the constant absorption coefficient given the number of reflections, K. Actually, the absorption coefficients

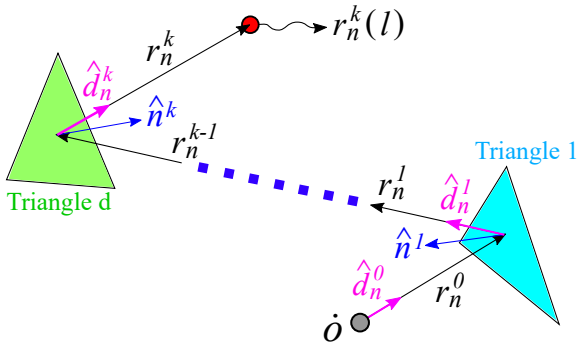


Figure 2.4: An example of propagating reflection acoustic rays. The acoustic ray path containing direction and reflection acoustic rays from r_n^0 to r_n^k is propagated from the origin $\dot{o}$ of the microphone array to the red point corresponding to $r_n^k(l)$. The summation of all ray lengths l of each acoustic ray from r_n^0 to r_n^k should be identical to l_{max}.

depend on the material properties, but I assume that all materials have a constant coefficient, i.e., 0.1, since the majority of sound materials in the experiment have low absorption coefficient (Chapter. 2.5.3).

Because I consider backward acoustic ray tracing from a microphone array (listener) to a source, the initial power β_n should be amplified with respect to the ray length, and I can determine the maximum travel distance, l_{max}, of an acoustic ray path as follows:

$$l_{max} = \sqrt{\frac{\beta_{th}}{\beta_n}(1-\alpha)^K - 1}. \tag{2.3}$$

The propagation of the acoustic ray terminates when the power of the ray exceeds a user-defined threshold for maximum energy, denoted as β_{th}, which is set by a reasonable power bound, in this case $10^{-4}W$, similar to the power of a loud alarm clock [53]: $0 \leq l \leq l_{max}$.

2.3.2 Acoustic ray tracing handling reflection

When an acoustic ray r_n^k hits an triangle of an object's mesh in the reconstructed environment, I need to simulate how the ray behaves at the hit point. Ideally, specular or diffuse reflection can occur with an energy absorption depending on the material type of the hitting surface. Since simulating all these types of interactions requires a prohibitive computation time, I support only a specular reflection in this work.

The decision not to support diffuse reflections is based on the following, two factors: 1) supporting diffuse reflections requires an expensive inverse simulation approach such as Monte Carlo simulation, which is unsuitable for real-time robotic applications, and 2) while there are many diffuse materials in rooms, each individual sound signal reflected from the diffuse material does not carry a high portion of the sound energy generated from the sound source. Therefore, when I choose high-energy directional data from the DoA estimator, the most sound signals reflected by the diffuse material are ignored automatically, and those with high energy are mostly from specular materials.

Note that the proposed work does not require all the materials to be specular. When some materials exhibit high energy reflectance near the specular direction, e.g., tex materials in the ceiling and finished wooden floors, the proposed method generates acoustic rays toward those specular reflection directions, and can identify the location of the sound source that generates those rays (Chapter. 2.5.3). As a result,

I focus on handling specular materials and treat each hit material as specular, and generate a reflection ray from the hit point.

The operation for specular reflection is defined as follows. Whenever an acoustic ray, r_n^k, hits the surface of the obstacle at the particular ray length, l_{hit}, I create a new, reflection acoustic ray, r_n^{k+1}, with the following direction:

$$r_n^{k+1}(l) = \hat{d}_n^{k+1} \cdot l + r_n^k(l_{hit}), \tag{2.4}$$

where $\hat{d}_n^{k+1}$ is the direction of the specular reflection of the ray r_n^{k+1}, and is analytically computed by $\hat{d}_n^{k+1} = \hat{d}_n^k - 2(\hat{d}_n^k \cdot \hat{n}^{k+1})\hat{n}^{k+1}$, where $\hat{n}^{k+1}$ is the normal vector at the surface hit point, $r_n^{k+1}(0)$. Its example is shown in Figure. 2.4. The primary acoustic ray r_n^0 is initialized at the origin $\dot{o}_t$ of the microphone array at a t frame and hits the triangle 1. The reflection acoustic ray, then, is generated into the $\hat{d}_n^1$ direction considering specular reflection. The acoustic ray path is propagated into the k-th order of reflection acoustic ray. The summation of all ray lengths of acoustic rays contained in the ray path should be same to l_{max} given the power bound.

The reflection acoustic ray that I create can be reflected further by getting another hit on other obstacles. While generating the acoustic rays of a path, I maintain them in a ray sequence, called a ray path, $R_n = [r_n^0, r_n^1, ...]$ generated for the n-th DoA.

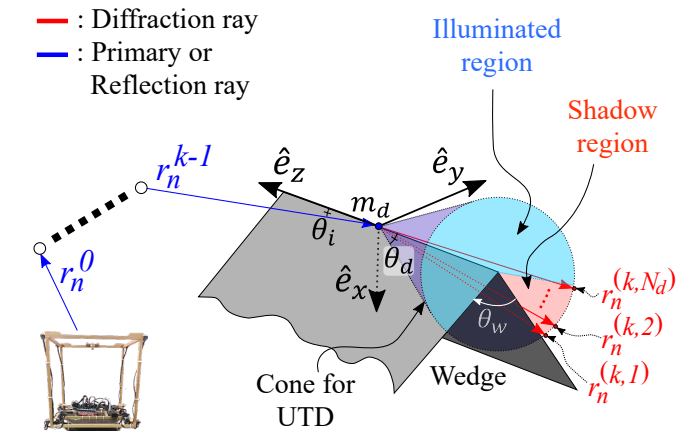
2.3.3 Acoustic ray tracing handling diffraction

I now explain the proposed algorithm to model diffraction efficiently within acoustic ray tracing. Since the goal is to achieve fast performance in localizing the sound source, I use the formulation based on Uniform Theory of Diffraction (UTD) [41]. The incoming sound signals collected by the microphone array consist of contributions from different propagation paths in the environment, including reflections and diffractions.

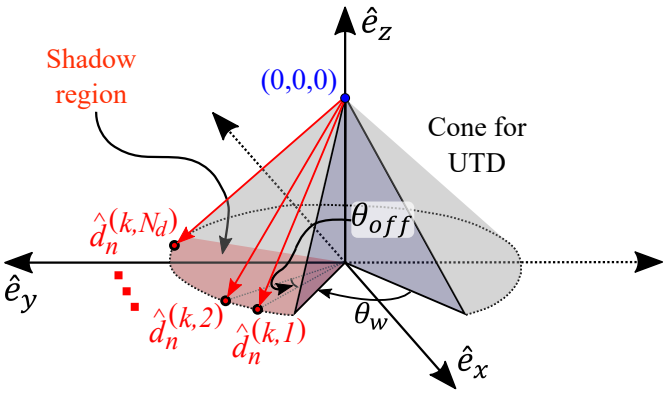
Edge diffraction occurs when an sound wave hits the edge of a wedge. In the context of forward acoustic ray tracing from a source, when an acoustic ray hits an edge of a wedge, the diffracted signal propagates into all possible directions from that edge. The UTD model assumes that the point on the edge causing the diffraction is an imaginary source generating a spherical wave [41].

In order to solve the problem of localizing the sound source, I simulate the process of backward ray tracing from the microphone array to the source. Suppose that an n-th DoA is generated by the diffraction at the point m_d on the wedge in Figure. 2.5(a). I generate the primary acoustic ray r_n^0 and perform backward acoustic ray tracing. In an ideal case, I can assume that the ray path R_n hits the point m_d on the edge of the wedge; for example, the ray r_n^{k-1} hits the point m_d and diffraction acoustic rays $r_n^{(k,\cdot)}$ must be generated in Figure. 2.5(a).

I assume that the point m_d causing the diffraction is an imaginary source generating the spherical wave based on the UTD model. Given the diffracted propagation path estimated by the ray r_n^{k-1} in Figure. 2.5(a), there might be an infinite number of candidates for incident propagation paths to the point m_d causing the diffraction. Given that it is difficult to determine the specific direction $\hat{d}_n^k$ corresponding to the direction of an incident propagation path, to generate the k-th order diffraction ray, I generate a set of N_d different diffraction rays that covers possible incident directions to the point m_d on the edge based on the UTD model. Intuitively, this set is generated based on the assumption that one of these generated rays may have the actual incident direction causing the diffraction, thus creating the subsequent ray r_n^{k-1}. When there are sufficient acoustic rays, including the primary, reflection, and diffraction rays, it is highly likely that those rays will pass through or close to the sound source location;



(a) Generating diffraction rays



(b) Computing outgoing directions of diffraction rays.

Figure 2.5: This figure illustrates the proposed acoustic ray tracing method devised to handle the diffraction effect. (a) Suppose that I have an acoustic ray r_n^{k-1} satisfying the diffraction condition, hitting or passing near the edge of a wedge. I then generate N_d diffraction rays covering the possible incoming directions (especially, in the shadow region) of rays that cause the diffraction. (b) An outgoing unit vector, $\hat{d}_n^{(k,p)}$, of a p-th diffraction ray is computed on local coordinates $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$, and used after transformation to the environment in runtime, where $\hat{e}_z$ fits on the edge of the wedge and $\hat{e}_x$ is set half-way between two triangles of the wedge.

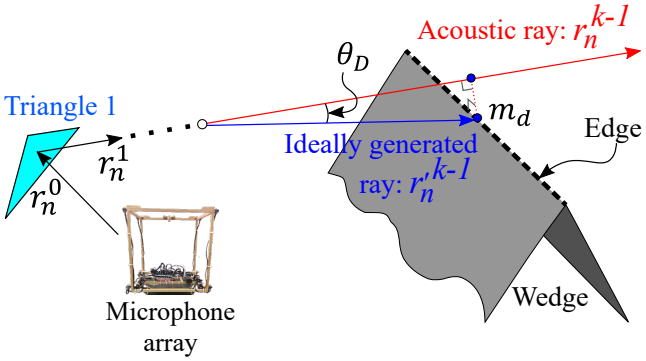


Figure 2.6: This figure illustrates the diffraction condition. When a ray r_n^{k-1} passes closely by an edge of a wedge, I consider the ray to be generated by edge diffraction. I measure and utilize the angle θ_D between the ray and its ideal generated ray that hits the edge exactly to verify the diffraction condition.

I choose a proper value of N_d, which is 5, by analyzing diffraction rays (Chapter. 2.5.2).

Given the n-th DoA caused by the diffraction, it is rare for acoustic rays of the n-th DoA to intersect an edge precisely because the proposed algorithm works in real environments containing various types of errors from sensor noise and resolution errors from the DoA estimator. In order to support various cases that arise in real environments, I propose the use of the simple yet effective notion of a *diffraction-condition* between a ray and a wedge. A diffraction-condition simply measures how closely the ray r_n^{k-1} passes by an edge of the wedge. Specifically, I define the *diffractability*, v_d, according to the angle θ_D between the acoustic ray r_n^{k-1} and its ideally generated ray, r'^{k-1}_n, for the diffraction with the wedge: i.e. $v_d = \cos(\theta_D)$, where the cos function is used to normalize the angle θ_D (Figure. 2.6).

Suppose that the n-th DoA is generated by the diffraction on the edge of the wedge, as highlighted by the dotted line in Figure. 2.6. In this case, the ray path R_n contains the diffraction propagation path, and I assume that the ideally generated ray r'^{k-1}_n represents the actual diffraction propagation path; the ray r_n^{k-1} does not hit the edge of the wedge due to the various errors that exist in real environments. I define the ideally generated ray r'^{k-1}_n as a ray touching the point m_d on the edge of the wedge and satisfying the smallest angle θ_D. To have the smallest angle θ_D, the distance from the point m_d to the ray r_n^{k-1} also becomes the smallest:

$$m_d = \underset{m'_d}{\text{argmin}}(\text{distance}(m'_d, r_n^{k-1})), \tag{2.5}$$

where m'_d is any point on the edge of the wedge, and the distance$(\cdot)$ denotes a minimum distance between the given point and line. The propagation direction of the ideally generated ray r'^{k-1}_n is identical to the vector from the origin of the ray r_n^{k-1} to the point m_d, and the angle θ_D can be computed using the inner product of the propagation directions of both rays, r_n^{k-1} and r'^{k-1}_n.

If the diffractabilty v_d is larger than a threshold value v_{th}, e.g., 0.984 in the tests in this Chapter. 2, the proposed algorithm determines that the acoustic ray is generated from the diffraction at the wedge, and I thus generate the secondary, diffraction ray at the wedge in the backward manner.

I now present how to generate the diffraction rays when an acoustic ray satisfies the diffraction-condition. The diffraction rays are generated along the surface of the cone (Figure. 2.5(a)), because the UTD model is based on the principle of Fermat [32]; the ray follows the shortest path from the source to the listener. The surface of the cone for the UTD model contains every set of shortest paths. When an acoustic ray r_n^{k-1} satisfies the diffraction-condition, I compute outgoing directions for those diffraction

rays. Those directions are the unit vectors generated on that cone and can be computed on a local domain as shown in Figure. 2.5(b):

$$\hat{d}_n^{(k,p)} = \begin{bmatrix} \cos\left(\theta_w/2 + p \cdot \theta_{off}\right) \sin\theta_d \\ \sin(\theta_w/2 + p \cdot \theta_{off}) \sin\theta_d \\ -\cos\theta_d \end{bmatrix}, \tag{2.6}$$

where $\hat{d}_n^{(k,p)}$ denotes the outgoing unit vector of a p-th diffraction ray among N_d different diffraction rays, θ_w is the angle between two triangles of the wedge, θ_d is the angle of the cone that is same as the angle between the outgoing diffraction rays and the edge on the wedge, and θ_{off} is the offset angle between two sequential diffraction rays, i.e. $\hat{d}_n^{(k,p)}$and $\hat{d}_n^{(k,p+1)}$, on the bottom circle of the cone.

Given a hit point m_d by an acoustic ray r_n^{k-1} on the wedge, I transform the outgoing directions in the local space to the world space by aligning their coordinates $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$. Based on those transformed outgoing directions, I then compute the outgoing diffraction rays, $\bar{r}_n^{(k)} = \{r_n^{(k,1)}, ..., r_n^{(k,N_d)}\}$, starting from the hit point m_d.

In order to accelerate the process, I only generate the diffraction rays in the shadow region, which is defined by the wedge; the outside of the shadow region is called the illuminated region. I focus on the shadow region because covering only the shadow region over the entire region generates minor errors for a simulation of the sound propagation [43].

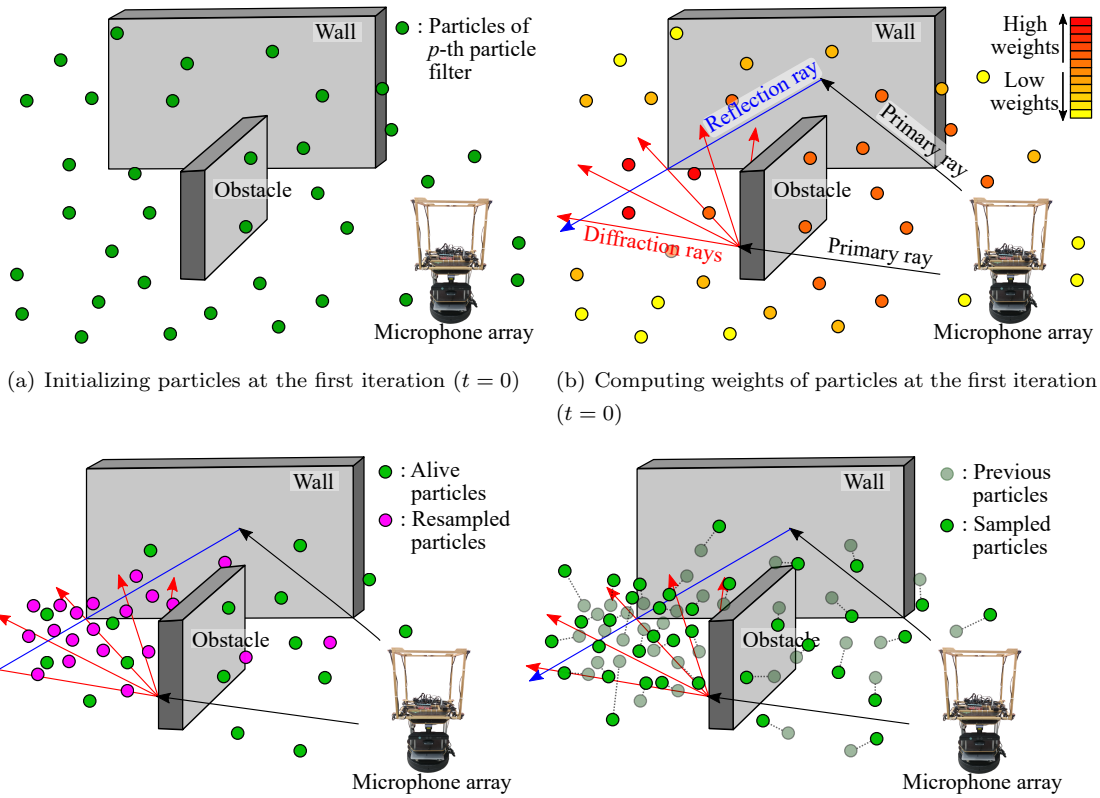
Given the new diffraction rays, I apply the proposed algorithm recursively and generate another order of reflection and diffraction rays. Given the n-th DoA, I generate acoustic rays, including direct, reflection, and diffraction rays and maintain the ray paths R_n in a tree data structure. The root of this tree represents the primary acoustic ray, starting from the microphones. The depth of the tree denotes the order of its associated rays. Note that I generate one child and N_d children for handling reflection and diffraction effects, respectively.

I maintain the ray path R_n for the fixed duration D_{ray}, one second, to accumulate a sufficient number of ray paths; the ray path is deleted after the duration D_{ray}. The duration D_{ray} is determined to maintain a ray path caused by an early reflection until a late reflection, i.e., reverberation. If the duration D_{ray} is too long, the proposed approach cannot properly reflect changes in the position of a moving sound source.

2.4 Monte Carlo localization for multiple sources

In the prior section, I generated primary, reflection, and diffraction acoustic rays starting from DoAs. Given those acoustic ray paths, I are ready to localize not only stationary sound sources, but also moving sound sources in 3D space; the proposed approach utilizes all ray paths created within the fixed time duration D_{ray}.

The generated acoustic ray paths represent the propagation paths of sound waves from sound sources to the microphone array. In an ideal case with multiple sources, it is sufficient to find points at which acoustic ray paths intersect and treat them as source positions. However, when I deal with real environments in practice, acoustic ray paths may not intersect precisely, as there are diverse types of noise from sensors, e.g., microphones, IMU sensors, and Lidars. I thus need a technique that is robust to these types of noise. I cast the problem as one involving the locating of regions where many such ray paths



(a) Initializing particles at the first iteration ($t = 0$)

(b) Computing weights of particles at the first iteration ($t = 0$)

(c) Resampling particles at the first iteration ($t = 0$)

(d) Sampling particles at the second iteration ($t = 1$)

Figure 2.7: An example of performing the p-th particle filter at the first and second iteration, i.e., $t = 0$ and $t = 1$. At the beginning of the proposed approach, i.e., $t = 0$, particles are initialized based on the uniform distribution in (a). In the weight computation part (b), weights of particles are computed given acoustic ray paths; particles have higher weights when they are located near the convergence region of ray paths. In the resampling path (c), particles with low weights are resampled close to particles with high weights. Thanks to the resampling part, particles can be moved to the convergence region of ray paths. After executing the part of allocating ray paths (Chapter. 2.4.4), the first iteration of the proposed approach is finished. At the second iteration, i.e., $t = 1$, the Monte Carlo localization starts with the sampling part, and particles are regenerated based on the Gaussian distribution in (d).

converge and treat the convergence regions as candidate regions containing sound sources. To achieve the goal, I propose the use of Monte Carlo localization (MCL) [13, 54], also known as a particle filter.

Sasaki *et al.* [13] proposed the source localization method based on a particle filter from estimating the convergence regions of the plane observation models, which contains direct sound information. I extend this approach to identify the convergence region of acoustic rays; the prior approach needs to satisfy some constraints, i.e., accumulating the observation models in different positions and orientations of a sound sensor, but the proposed approach does not require those constraints by considering indirect sound.

Assuming there are P sound sources, there can be P different convergence regions of ray paths. The p-th convergence region corresponds to the p-th sound source, and ray paths propagating to the p-th convergence region can be caused by the p-th sound source. In the acoustic ray tracing phase, it is difficult to determine what acoustic ray paths are generated by which sound sources. In every iteration of the proposed localization algorithm, therefore, I initially estimate the source positions using multiple

particle filters and then determine whether or not the ray paths are caused by estimated sources.

The proposed approach supports P different particle filters to localize P sound sources, and each particle filter can localize only a single source. Each particle filter consists of four parts and sequentially performs them every iteration. These are sampling, weight computation, resampling, and allocating ray paths. In the first three parts, particle filters identify the convergence regions of ray paths; Figure. 2.7 shows an example of executing three parts sequentially. In the sampling part, the proposed approach initializes (Figure. 2.7(a)) or regenerates the positions of particles randomly close to previous positions (Figure. 2.7(d)), to consider the movement of dynamic sources. The weights of particles are computed to ensure that the particles converge to the convergence region of the ray paths (Figure. 2.7(b)). In the resampling part, the proposed approach deletes particles located far from the convergence region and resamples new particles inside the convergence region (Figure. 2.7(c)).

In the last part of the proposed method, if there are convergence regions, acoustic ray paths are allocated to each convergence region into which they propagate. The ray paths generated from the ray tracing phase are initially not allocated to any convergence region; they have an initial state of -1; i.e., $S(R_n) = -1$, where R_n denotes the n-th ray path and $S(\cdot)$ is the function returning the state value of a given ray path. After acoustic ray paths are allocated to the convergence region of the p-th particle filter, it has a p state; i.e., $S(R_n) = p$.

2.4.1 Sampling

To identify the convergence region of ray paths, the p-th particle filter maintains a set of I particles, $X_t^p = [x_t^{(p,1)}, \cdots, x_t^{(p,I)}]$, which serves as hypothetical locations of a sound source; the number of particles, e.g., $I = 200$, should be sufficient to cover all 3D indoor environments. Those particles are spread out randomly in the 3D space based on the uniform distribution at the initial iteration, $t = 0$, and iteratively move to the convergence region of the ray paths at other iterations, $t \geq 1$. Also, if there is no ray path at a specific iteration, I treat that there does not exist any sound source, and perform the initialization process, i.e., spreading out randomly particles, again to quickly cover the entire 3D space.

To consider the movement of sound sources, a new set of particles, X_t, is incrementally created from the prior particles, X_{t-1}, for each iteration t other than the initial iteration. If I know a source position and its velocity, new particles can be created using the source velocity and the corresponding moving direction. However, in a sound source localization problem, I do not know the position or velocity of a source when the proposed approach begins. Therefore, I randomly create a new set of particles from the prior particles and then re-generate particles near the actual source position in the ensuing weight computation and resampling parts.

A new particle, $x_t^{(p,i)}$, of the p-th filter is generated by offsetting an previous one, $x_{t-1}^{(p,i)}$, in a random unit direction, $\hat{u}$, by an offset, δ:

$$\begin{aligned} x_t^{(p,i)} &= x_{t-1}^{(p,i)} + \delta \cdot \hat{u}, \\ \delta &= \| x_t^{(p,i)} - x_{t-1}^{(p,i)} \| \sim N(0, {\sigma_s}^2), \end{aligned} \tag{2.7}$$

where $N(\cdot)$ denotes a normal distribution with a zero mean and standard deviation. Actually, the random unit direction, $\hat{u}$, and the offset, δ, correspond to the unit vector of the velocity and the speed of the particle, respectively. Since the offset, δ, is sampled according to the normal distribution, Eq. 2.7 can cover the various movements of the stationary, constant velocity, and accelerated particles. The standard deviation, σ_s, is determined by the maximum speed of a moving sound source; e.g., $\sigma_s = 0.2$ m

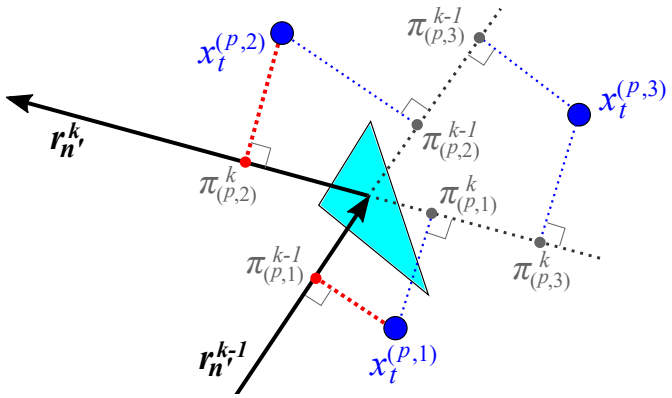


Figure 2.8: An example of computing weights of the p-th filter for particles against a ray path, $R_{n'} = [\cdots, r_{n'}^{k-1}, r_{n'}^{k}]$. The shortest distances for each particle over acoustic rays are shown in red and become the distances between the particles and the ray path.

in the experiments in this Chapter. 2. The proposed approach is designed to handle speeds up to 1 m/s, $= 0.2$ m/0.2 s, of moving sources, where the iteration period is 200 ms.

2.4.2 Weight computation

I associate a weight with each particle, and the weight indicates the importance of the particle, specifically encoding how closely the particle is located to a convergence region of ray paths. Suppose that ray paths are converged in a region containing the source position. In this case, the distances from any point inside the convergence region to the ray paths must be small; in an ideal case, ray paths intersect at a certain point, and distance between an intersecting point to a ray path should be zero. Therefore, when a particle is located inside the convergence region of ray paths, the distances between the particle and the ray paths are generally short. Based on these distances, I design the weight of the particle to have a higher value when it is located inside the convergence region.

The weight at the t iteration is also updated from the previous iteration, $t-1$, assuming that the sound source is active from $t-1$ to t iterations; how the proposed method handles intermittent sources is discussed later in this section. During the weight computation phase, the p-th particle filter only considers ray paths in the -1 or p state. The -1 state means that the ray path is not yet allocated to any estimated source position. The p state indicates that the ray path propagated close to the estimated source position in a prior iteration and was therefore allocated to the sound source estimated by the p-th particle filter. I ignore the remaining ray paths with other states. Therefore, in the p-th particle filter, the weight of the particle is computed based on the observations, o_t^p, consisting of the acoustic ray paths, $R_{n'}$, in only the -1 or p state.

The distance between a particle and an acoustic ray can be computed by calculating the distance from a point to a line segment; the distance between a point to a line segment corresponds to the distance from a point to a perpendicular foot on a line segment. I define the distance, $dist(\cdot,\cdot)$, between a particle and a ray path $R_{n'}$ as the shortest distance among the distances from the particle to the rays of $R_{n'}$:

$$dist(x_t^{(p,i)}, R_{n'}) = \min_k(\|x_t^{(p,i)} - \pi_{(p,i)}^k\| \times F(x_t^{(p,i)}, r_{n'}^k)), \tag{2.8}$$

where $\pi(x_t^{(p,i)}, r_{n'}^k)$, in short, $\pi_{(p,i)}^k$, defines the perpendicular foot of the particle $x_t^{(p,i)}$ to the ray $r_{n'}^k$ (Figure. 2.8), and $\|\cdot\|$ denotes the L2 norm. F is a filter function returning infinity to exclude irrelevant

cases when the perpendicular foot is outside of the ray segment $r^k_{n'}$, e.g., $\pi^k_{(p,1)}$, $\pi^{k-1}_{(p,2)}$, $\pi^{k-1}_{(p,3)}$, and $\pi^k_{(p,3)}$ in Figure. 2.8. Otherwise, the filter function returns one.

Based on the distance between the particle $x_t^{(p,i)}$ and the ray path $R_{n'}$, I define the probability density $P(R_{n'}|x_t^{(p,i)})$:

$$P(R_{n'}|x_t^{(p,i)}) = N(dist(x_t^{(p,i)}, R_{n'}) \mid 0, \sigma_w^2), \tag{2.9}$$

where $N(\cdot\,|0, \sigma_w^2)$ indicates a normal distribution with a zero mean and standard deviation, representing a parameter that controls how many particles converge to the estimated source position; a smaller standard deviation makes particles converge to a smaller area, meaning that the estimated convergence region of ray paths also becomes smaller. The standard deviation of σ_w only has a minor effect on the accuracy. I use $\sigma_w = 0.1$, corresponding to 0.1 m in the space, for the tests in this Chapter. 2.

Simply speaking, the probability density $P(R_{n'}|x_t^{(p,i)})$ becomes higher if the i-th particle $x_t^{(p,i)}$ is closer to the ray path $R_{n'}$, and $P(R_{n'}|x_t^{(p,i)})$ has the highest value if the particle lies on the ray path. From the probability density $P(R_{n'}|x_t^{(p,i)})$, I design the likelihood $P(o_t^p|x_t^{(p,i)})$ of the particle $x_t^{(p,i)}$, where the observation o_t^p consists of N' different ray paths having -1 and p states: $o_t^p = [R_1, \cdots, R_{N'}]$. I define the likelihood as the average of $P(R_{n'}|x_t^{(p,i)})$ over all ray paths:

$$P(o_t^p|x_t^{(p,i)}) = \frac{1}{N'} \sum_{n'=1}^{N'} P(R_{n'}|x_t^{(p,i)}). \tag{2.10}$$

The likelihood $P(o_t^p|x_t^{(p,i)})$ indicates that how much the particle $x_t^{(p,i)}$ is close to ray paths contained in the observation o_t^p.

I define the weight, $w_t^{(p,i)}$, at the t iteration based on the likelihood $P(o_t^p|x_t^{(p,i)})$:

$$w_t^{(p,i)} = \frac{P(o_t^p|x_t^{(p,i)}) w_{t-1}^{(p,i)}}{n_c}, \tag{2.11}$$

where n_c denotes the normalization constant and $w_{t-1}^{(p,i)}$ is the weight at the previous iteration $t-1$. The weight $w_{t-1}^{(p,i)}$ helps to consider the convergence region at the previous iteration $t-1$. When the particle $x_{t-1}^{(p,i)}$ is close to the convergence region at the previous iteration $t-1$, the weight $w_{t-1}^{(p,i)}$ should be large, causing the weight $w_t^{(p,i)}$ to increase. If there is no acoustic ray at iteration t, I set all weights to a uniform probability, i.e., $1/I$.

Suppose that an intermittent source is activated at iteration t, while it was inactive at the previous iteration $t-1$. At iteration $t-1$, no acoustic ray was generated for the intermittent source. Despite the fact that other active sources exist at iteration t, the ray paths generated by those sources were previously allocated to other filters identifying the convergence regions of those sources. As a result, ray paths in the -1 state are left for the intermittent source. Suppose that the p-th particle filter corresponds to the source. All weights of particles of the p-th particle filter have a uniform probability as the initialization process (Chapter. 2.4.1). At iteration t, newly generated acoustic rays in the -1 state should be propagated to the activated source, and the weight $w_t^{(p,i)}$ is only determined by the distance between the particle and acoustic rays.

2.4.3 Resampling

There may be particles close to or far from the convergence region of ray paths, and their weights indicate how closely they are located to the convergence region. To make particles converge to the convergence region of ray paths in this part, I delete particles located far from the convergence region

and re-generate them inside the convergence region. Intuitively, particles with low weights are removed, and additional particles are generated near existing particles with high weights. Regarding this process, I adopt a basic resampling method [54].

Once resampling is done, I check whether the particles are converged enough to define an estimated sound source; if the particles are thus converged, I can treat the positions of particles as the convergence region of the ray paths. To determine the convergence of the particle positions, I compute the generalized variance (GV), which is a one-dimensional measure for multi-dimensional scatter data and is defined as the determinant of the covariance matrix of the particles [55]. If GV is less than the convergence threshold, $\sigma_c = 0.01$, at the p-th particle filter, I determine that the source emitted the sound and treat the mean position of particles as the estimated position of the source. GV is also used as a confidence measure in the estimation in this Chapter. 2; I use its covariance matrix to draw a 95% confidence ellipsis disk for visualizing the estimated sound region (Figure. 2.1).

2.4.4 Allocating ray paths

Suppose that the sound source is estimated in the resampling step. In such a case, it becomes necessary to check whether or not ray paths are caused by the estimated source; if there is no estimated source, I skip the allocating ray paths phase. Assuming that a ray path is caused by the estimated source, it should propagate to the position of the estimated source. In this step, I only consider ray paths in the -1 state, indicating that I do not know from which sound sources the ray path originated. I now verify whether the ray paths in the -1 state propagate close to the estimated source position.

A simple way to do this is to compute and verify the distances between the estimated source position, i.e., the mean of the particle positions, and the ray paths. However, this simple approach does not consider the shape of the estimated sound region in Figure. 2.1, which represents the 95 % confidence area. To deal with the shape of the estimated sound region, I examine the relationships between the ray paths and particle positions.

I define the probability, $P(S(R_{n'}) \to p)$, of allocating the ray path to the source estimated by the p-th filter as follows:

$$P(S(R_{n'}) \to p) = \sum_{i=1}^{I} P(R_{n'}|x_t^{(p,i)}) w_t^{(p,i)}, \tag{2.12}$$

where $R_{n'}$ is the ray path in the -1 state, $P(R_{n'}|x_t^{(p,i)})$ is the probability density in Eq. 2.9, and $w_t^{(p,i)}$ is the weight of a particle as defined by Eq. 2.11. I allocate the ray path $R_{n'}$ to the sound source estimated by the p-th filter if the probability $P(S(R_{n'}) \to p)$ exceeds a threshold probability, i.e., $P_{th} = 0.2$.

The probability density $P(R_{n'}|x_t^{(p,i)})$ represents how closely much the particle $x_t^{(p,i)}$ is to the ray path $R_{n'}$, and the weight $w_t^{(p,i)}$ indicates the importance of the particle. If the particle is located close to the estimated source, its weight becomes high, and it must be an important particle. Therefore, if many particles with high weights are located close to the ray path $R_{n'}$, the allocating probability $P(S(R_{n'}) \to p)$ becomes higher than the threshold probability P_{th}. Figure. 2.9 shows an example of allocating the ray path.

I continue these iterations of the Monte-Carlo localization (MCL) algorithm, consisting of four parts given a fixed duration, e.g., 200 ms, until the proposed multiple-source localization algorithm, containing acoustic ray tracing and MCL algorithms, is terminated. I decide to make the period of iterations short enough to find the source position quickly, taking into account the calculation time of the proposed approach. If the proposed MCL algorithm is finished within the computation budget, 200 ms, it enters

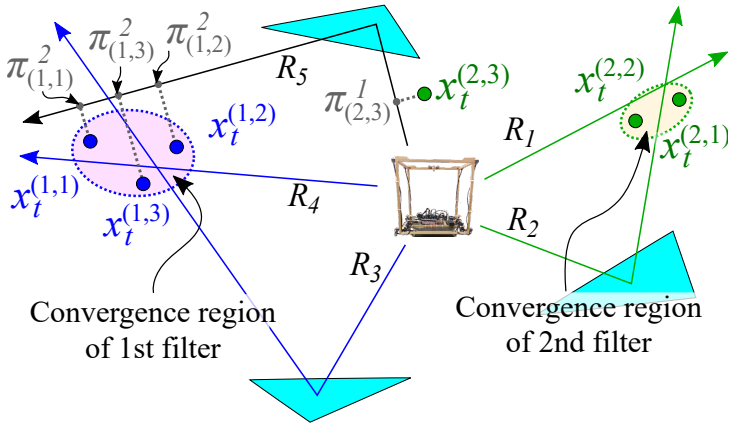


Figure 2.9: An example of allocating the ray path to the convergence region of the particle filter. The ray paths, indicated here by the blue and green lines, are allocated to the convergence regions of the first and second particle filter, respectively; both convergence regions represent the estimated source positions. Ray path R_5, indicated by the black lines, is now considered to be assigned to its proper estimated source. Gray dotted lines denote the distance between the particles and ray path R_5, used to compute the probability $P(R^p_{n'}|x_t^{(p,i)})$ in Eq. 2.11. In this example, ray path R_5 originates from the source estimated by the first filter, and it is allocated to the estimated source of the first particle filter. the allocating probability $P(S(R_5) \rightarrow 1)$ exceeds the threshold probability P_{th}.

an idle state until the next iteration.

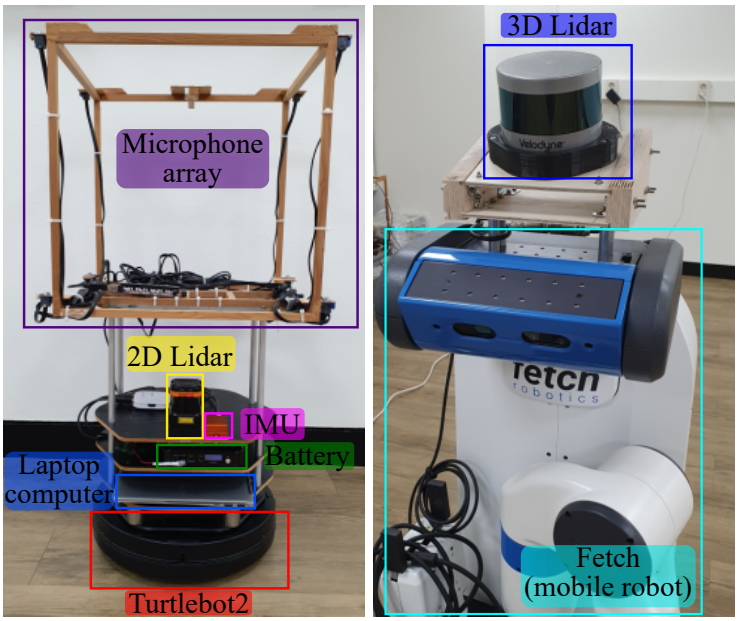
2.5 Results and discussion

In this section, I provide various results and discussions of the proposed approach. The hardware platform is based on Turtlebot2 with a 2D Lidar (UTM-30LX of Hokuyo), an IMU sensor (MTi-30 of Xsens), an 8 channel microphone array [56], and a laptop computer with an Intel i7 process shown in Figure. 2.10(a).

To estimate DoAs, I utilize a delay-and-sum (DAS) beamforming module of ManyEars [57], which is a real-time open software for the DoA estimation, tracking, and separation. Although ManyEars tracks DoA information using the particle filter, the processes between particle filters of ManyEars and the proposed approach are different; I just refer to the process of ManyEars and only utilize the DoA estimator, i.e., the DAS beamformer. ManyEars performs the particle filter given DoAs and energies of DoAs to track the DoA sequentially, but the proposed approach performs the particle filter given acoustic rays to identify convergence regions of those rays.

The DoA estimator and acoustic ray tracing algorithm are performed every 10.67 ms, since the sampling frequency of the audio stream of the microphone array is 48000 Hz, and the number of sound pressure samples is 512, i.e., $L = 512$; 10.67 ms = 512 samples/48000 Hz. For all computations, I use a single core and perform the estimation within every 200 ms, supporting five different estimations in one second.

The experiments contain dynamic sound sources. To make a sound source move, I utilize the mobile robot platform, i.e., Turtlebot2, and the sound sources, an omnidirectional speaker, are placed on the mobile robot. I also measured the odometry of the mobile robot, which contains the sound source, and then utilized measured odometry as the ground truth of moving sources.



(a) (b)

Figure 2.10: Hardware platforms of the proposed approach. (a): to utilize the proposed SSL algorithm in the runtime computation, I add an eight-channel microphone array onto Turtlebot2, a mobile robot, with 2D Lidar, an IMU sensor, and a laptop computer. (b): in the precomputation phase, I extracted the point cloud of the environments using 3D Lidar placed on the top of the Fetch mobile robot.

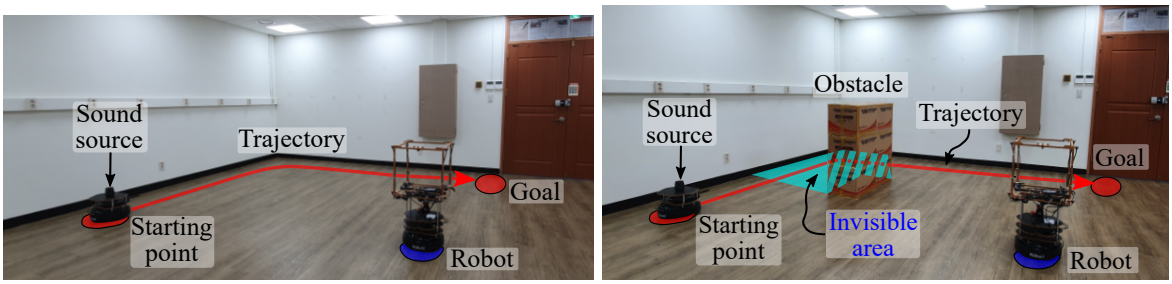
To reconstruct the 3D environments, I perform a SLAM algorithm, i.e., Cartographer of Google [2], using sensor data collected by a 3D Lidar (VLP-16 of Velodyne) and an IMU sensor equipped on Fetch [58] (Figure. 2.10(b)). I also utilize the open source, MeshLab [59], to generate mesh maps from point clouds and improve the quality of meshes.

Wedges needed for supporting diffraction effects are extracted by using primitive fitting techniques [60], where the primitive model is defined by the box shape since the experiments in this Chapter. 2 contain only box-shape obstacles. I expect that different shapes of obstacles can be identified using various primitive models [48, 61].

Benchmarks. I tested the proposed approach in various scenarios and compared the result in this Chapter. 2 to the prior work [1], i.e. ManyEars3D. This method is another version of ManyEars [57], i.e., the open software containing the DAS beamforming module that I utilize. While ManyEars contains a module for estimating DoAs, this method, i.e., ManyEars3D, provides a module for estimating 3D locations of sound sources. While ManyEars3D can identify 3D location of the source, it considers only the direct sound; it estimates the source position by considering direct sound based on the DAS beamformer, and then tracks estimated source positions using a particle filter.

I first conducted a room experiment having 7 m×7 m area and 3 m height with a moving source (Chapter. 2.5.1). In this environment, I verify how well the proposed approach identify a source position given a direct and reflection acoustic rays. I also place an obstacle, blocking direct sound propagation paths, to show the effect of diffraction acoustic rays where the sound source moves around the obstacle. I then analyze diffraction acoustic rays to confirm the benefits of them (Chapter. 2.5.2).

In the above environments, the majority of objects, e.g., wall, floor, and ceiling, consists of specular materials like bricks, thick woods, gypsum boards, and steals; specular materials reflect the most of the



(a) Environment w/o an obstacle.

(b) Environment w/ an obstacle.

Figure 2.11: Testing environments in a 7 m×7 m room with a 3 m height given one moving source w/ and w/o an obstacle: (a) environment without an obstacle and where the sound source moves along the trajectory, highlighted by the red line, (b) environment with an obstacle, i.e., the box shape, where the moving source becomes a non-line-of-sight source when it is located in the invisible area due to the box.

sound energy from incident sound waves. Those specular materials cause sufficient specular reflections helping to generate reflection acoustic rays, whereas diffuse materials absorb most of the energy; therefore, specular reflection does not occur well at those diffuse materials. By replacing specular materials by diffuse materials, I also test the robustness of the proposed approach where the area of specular materials decreases (Chapter. 2.5.3); decreasing the number of specular materials means decreasing the number of reflection propagation paths. I also tested the proposed approach with a different DoA estimator using a different microphone array (Chapter.2.5.4). The quantitative results of scenarios of a single moving source are summarized in Table 2.1.

Because I extend the single sound source localization to the multiple source localization algorithm in this Chapter. 2, I conducted experiments with multiple sources. I tested the proposed approach in two scenes containing multiple static sound sources, which do not move, or moving sound sources (Chapter. 2.5.5); especially, in the three static sources scene, I show localizing intermittent sound sources by controlling the source activation period. The quantitative results of scenarios of multiple sources are summarized in Table 2.2.

To show the robustness for different sizes of environments, I conducted the experiments in a smaller size of the room: 7 m×3 m area with 3 m height (Chapter. 2.5.6).

I apply the proposed sound source localization algorithm to the navigation task. When the source is located behind the obstacle, i.e., the NLOS source, the robot equipped with a microphone array estimates the source position using the proposed approach and navigates to the estimated source position corresponding to the goal position of the navigation (Chapter. 2.5.7).

2.5.1 A moving source w/ or w/o an obstacle

I first show results of the environments with a moving source without or with an obstacle. The sound source moves along trajectories, red lines shown in Figure. 2.11, and emits sound signals. I utilize two kinds of sound signals that are a clapping sound and a human speech; the dominant frequencies of a clapping sound and a human speech are 15k Hz and 275 Hz, respectively. The clapping sound consists of five claps, and the human speech is reading the sentence "Hey, robot, come here" by a woman.

The environment without an obstacle. The results of the environment without an obstacle (Figure. 2.11(a)) are shown in Figure. 2.12. I measure distance errors between the ground truth and

estimated source positions, and the smaller distance error means that the accuracy is higher. The average distance errors of the clapping sound and the human speech are 0.5967 m and 0.7416 m, respectively; I call the average of distance errors as the average distance errors for convenience. Note that both values are smaller than the average distance errors, i.e., 1.6 m and 1.7769 m of the clapping sound and the human speech, of a prior work, and thus the proposed SSL algorithm can identify the source position reasonably well in this case. I observe a 168% and 139% improvement for the clapping sound and the human speech. The prior work only considers the direct sound, and thus the better accuracies of the proposed approach show that it is useful to consider the reflection sound. Also, the reason why the average distance error of the human voice is worse than the clapping sound is that the dominant frequency of the human voice (275 Hz) is lower than that of the clapping sound (15 kHz); the lower frequency sound more frequently causes the diffuse reflection, i.e., scattering by obstacles, rather than the specular reflection [62].

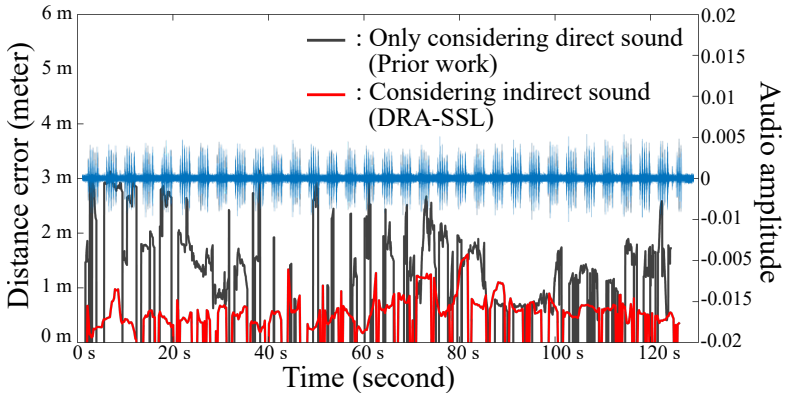
To verify how much the acoustic rays contribute in terms of helping source localization, I check how many ray paths propagate near to the source position. For example, given a ray path consisting of various acoustic rays, I find the smallest distance of acoustic rays contained by a ray path; the distance between the source position and the acoustic ray corresponds to the distance between a point and a line segment. If the smallest distance is less than 1 m, I treat this ray path as helping the source localization. I then check the type of the acoustic ray having the smallest distance and count this ray; I call those rays as the significant ray. The average numbers of significant primary and reflection rays per frame are 6.49 and 10.43 of the clapping sound and 8.84 and 7.23 of the human speech. The diffraction acoustic rays are not generated because the DoA estimator can only detect prominent propagation paths; a diffraction propagation path becomes a prominent propagation path when the source is the non-line-of-sight state in other tested environments.

These results tell us that sufficient primary and reflection rays propagate near the source position. Moreover, the number of significant reflection rays of the clapping sound, i.e., 10.43, is larger than the human speech, i.e., 7.23, because the specular reflection frequently occurs on the higher frequency sound. The larger number of significant reflection rays helps to increase the localization accuracy since reflection rays increase the convergence of rays; the average distance error of the clapping sound is less than the human speech. The propagation directions of primary rays should be similar since they are generated at the robot to the source position, while the propagation directions of reflection rays are determined by a normal vector of a hit obstacle.

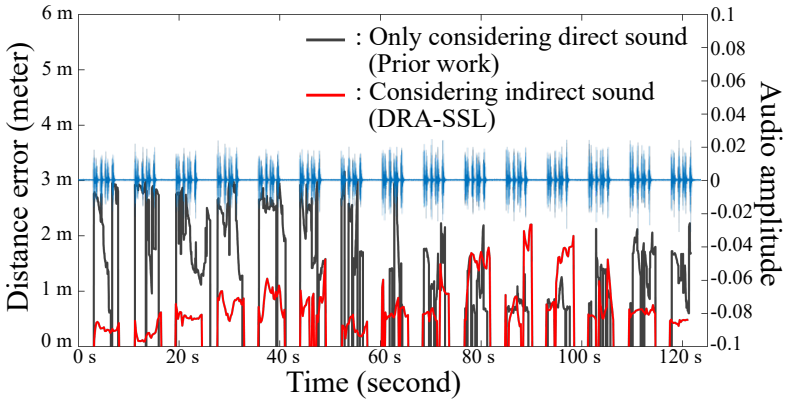
The environment with an obstacle. In the prior experiment without an obstacle, there is a sufficient number of significant primary and reflection rays, and I can localize the moving source by utilizing the primary and reflection acoustic rays where the diffraction propagation path was not a prominent path thus not detected by the DoA estimator. However, I need to consider diffraction on the wedges of the obstacle, especially when an obstacle is located and blocks the direct propagation path of sound, as shown in Figure. 2.11(b); the size of the obstacle is 0.39 m×0.96 m area with 1.05 m height. The diffraction propagation paths become prominent, when the moving source is located in the invisible area: the source in this case becomes the non-line-of-sight (NLOS) source.

I present the results of the environment with an obstacle in Figure. 2.13; I tested with the prior work and two versions of the proposed approach: the first version is only utilizing primary and reflection acoustic rays, and the second version is adding diffraction rays to them. I call the first version as RA-SSL (reflection-aware SSL), and the second version as DRA-SSL (diffraction and reflection-aware SSL) for convenience.

When I use the clapping sound, the average distance errors are 0.6351 m (DRA-SSL), 0.8112 m (RA-



(a) Clapping sound without an obstacle.



(b) Human speech without an obstacle.

Figure 2.12: The results in the environment without an obstacle (Figure. 2.11(a)), where the clapping sound is used in (a) and human (female) speech is used in (b). Both show the distance error of the proposed approach and prior work [1] in the red and gray curves, respectively, between the ground truth and the estimated source positions, and the measured signals in blue curves of the clapping sound in (a) and the human speech in (b).

SSL), and 1.582 (prior work) in Figure. 2.13(a). When I utilize the human speech, the average distance errors are 0.7313 m (DRA-SSL), 0.8803 m (RA-SSL), and 1.7571 m (prior work) in Figure. 2.13(b). These results show that the diffraction acoustic rays help to localize the source better. I observe a 149 % (clapping sound) and 140 % (human speech) improvement over the prior work considering only direct sound, and a 15 % (clapping sound) and 18 % (human speech) improvement when I additionally consider the diffraction rays compared to only utilizing the primary and reflection rays.

Especially, when the sound source becomes a NLOS source, located in the invisible area in Figure. 2.11(b), from 20 s to 80 s, the average distance errors when adding diffraction rays, i.e., DRA-SSL, are 0.7336 m for the clapping sound and 0.7618 m for the human speech, while the average distance errors of the prior work and RA-SSL are 2.1515 m (prior work, clapping sound), 2.3 m (prior work, human speech), 0.7336 m (RA-SSL, clapping sound), and 0.7618 m (RA-SSL, human speech), respectively. I observe a 193 % (clapping sound) and 201 % (human speech) improvement compared to the prior work, and a 31 % (clapping sound) and 26 % (human speech) improvement when adding the diffraction rays compared to RA-SSL.

When the sound source is on the line-of-sight (LOS) from 0 to 20 s and 80 s to 125 s, the averages of significant acoustic rays of the proposed approach per frame are 7.36 (primary), 9.52 (reflection), and 2.32 (diffraction) of the clapping sound, respectively, and 6.9 (primary), 9.79 (reflection), and 1.15 (diffraction) of the human speech, respectively. When the sound source is occluded by the obstacle, i.e., NLOS source, the averages of significant rays per frame are 0.61 (primary), 9.3 (reflection), and 3.87 (diffraction) of the clapping sound, respectively, and 1.55 (primary), 3.83 (reflection), and 6.39 (diffraction) of the human speech, respectively. Ideally, there should be neither diffraction rays during LOS sources nor primary rays during NLOS sources, respectively. However, in practice, primary rays generated immediately after being occluded by the obstacle and diffraction rays generated just before being occluded by the obstacle were counted in significant primary acoustic rays in the NLOS source cases and affect significant diffraction acoustic rays in the LOS source cases.

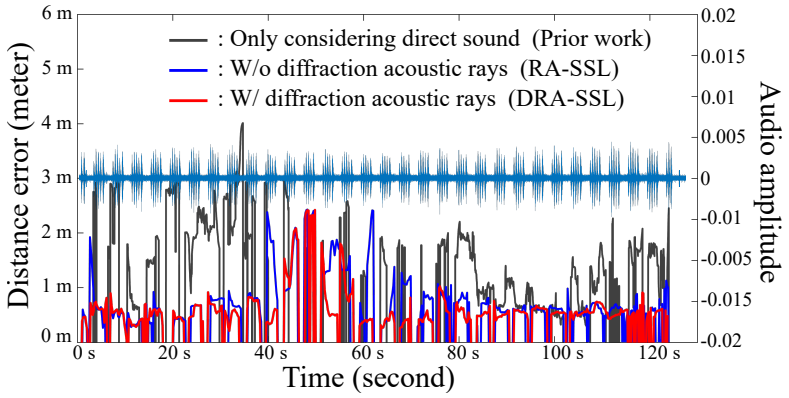
The remarkable aspect is that the most primary acoustic rays are blocked by the obstacle, and the effect of the diffraction rays increases when the source is the NLOS state; the averages of significant diffraction rays become larger compared to the LOS source. Also, the average of significant diffraction rays of the human speech is larger than the clapping since diffraction is a low-frequency phenomenon.

2.5.2 Analysis of the diffraction acoustic rays

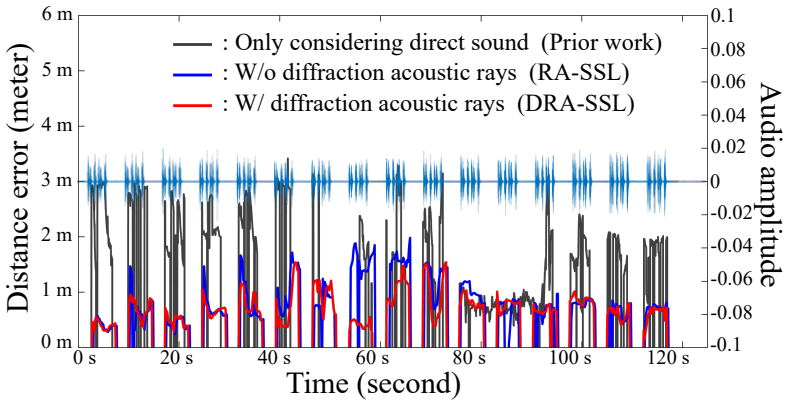
To see effects of considering diffraction acoustic rays in addition to the primary and reflection acoustic rays, I measure the accuracy as a function of the number of diffraction acoustic rays, N_d. As N_d, increases from 0 to 9, I measure the average distance errors and the average of calculation times in the environment containing the obstacle using the clapping sound (Figure. 2.14); the experimental setting with $N_d = 0$, i.e., no diffraction rays, is same to one tested in Figure. 2.13(a).

The average distance errors are gradually reduced until $N_d = 5$, and the accuracy is almost converged after $N_d = 5$. The averages of calculation times increase linearly, as a function of N_d. Since the accuracy changes after $N_d = 5$ are small enough, I use $N_d = 5$ across all the other experiments. Overall, I observe 29 % improvement by using $N_d = 5$ over using no diffraction rays; the average distance errors of $N_d = 0$ and $N_d = 5$ are 0.8112 m and 0.6351 m, respectively.

The average running times for acoustic ray tracing and particle filter are 6 ms and 11 ms; the total average running time is 17 ms corresponding to the average calculation time at $N_d = 5$ in Figure. 2.14.



(a) Clapping sound with an obstacle



(b) Human speech with an obstacle.

Figure 2.13: The results in the environment with an obstacle (Figure. 2.11(b)) and two sound signals: the clapping sound and human speech. In both (a) and (b), the black curves are the distance errors of the prior work [1], the blue curves are the distance errors where I use only the primary and reflection acoustic rays (RA-SSL), and the red curves correspond to the distance errors when handling all types of acoustic rays containing diffraction acoustic rays (DRA-SSL). Measured audio signals are shown in the middle of the graphs.

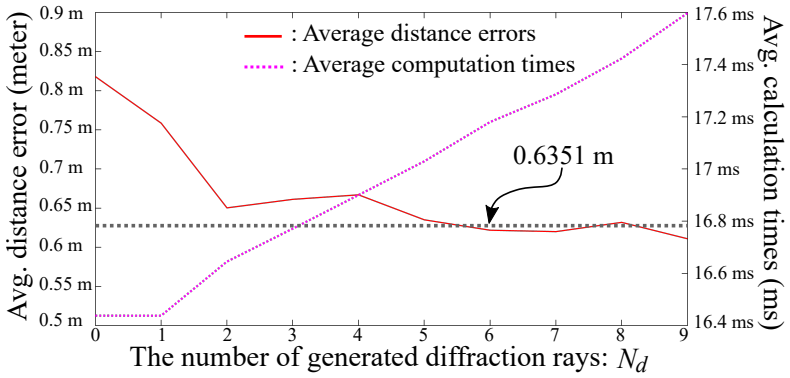


Figure 2.14: The average distance errors and computation times for the proposed method on an Intel i7 6700 processor, as a function of the number of diffraction rays generated for simulating the edge diffraction.

2.5.3 Analysis of specular and diffuse materials

In the previous environments (Chapter. 2.5.1), most materials such a solid cement wall, a thick wooden floor, and a gypsum board ceiling, have low absorption coefficients and tend to generate specular reflections. I verify that most materials have a coefficient of 0.1 or lower for all frequency bands using a hand-held measurement device [63, 64].

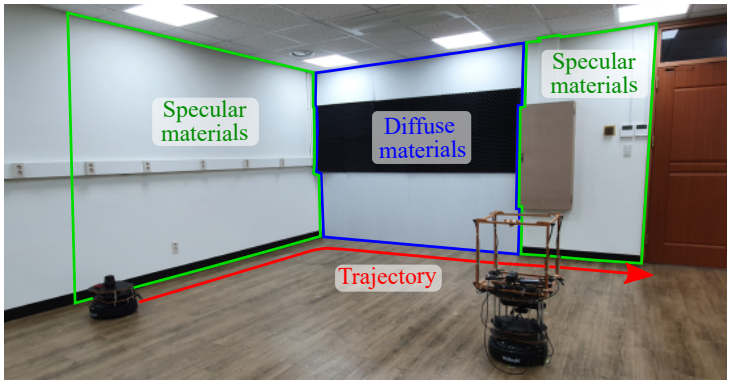
In real environments, however, there can be diffuse materials, e.g., carpets on the floor and curtains for windows, with high absorption coefficients. These scenes can have fewer specular materials and the number of reflection propagation paths, especially those caused by diffuse materials, can therefore decrease.

To see how diffuse materials affect the proposed algorithm, I set the environment with a diffuse material, as shown in Figure. 2.15. I attached diffuse materials, having almost absorption coefficient of 1 for all frequency bands, to a part of the wall (the blue rectangle). Parts of the wall shown by the green and blue rectangles in Figure. 2.15 are the candidates for causing the dominant reflection propagation paths. I cover those walls by the diffuse material, i.e., acoustic foam.

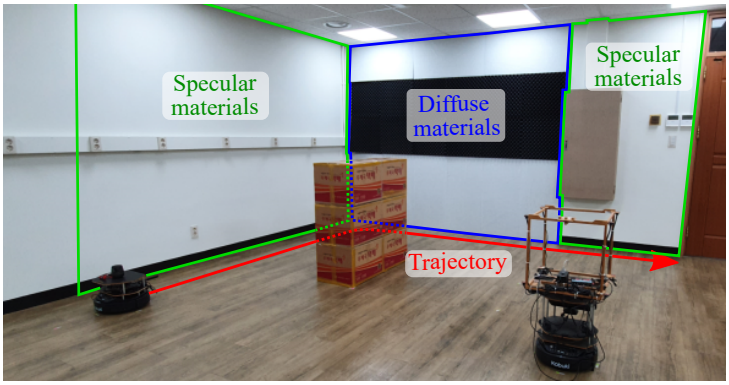
I tested those situations containing diffuse materials in environments without and with an obstacle (Figure. 2.15(a) and Figure. 2.15(b)) using the clapping sound. The corresponding distance errors are shown in Figure. 2.16, where the average distance errors w/o and w/ an obstacle are 0.6176 m and 0.6998 m, respectively. Because there exist direct propagation paths and a sufficient amount of specular materials in the environment without an obstacle in Figure. 2.15(a), the average distance error, 0.6176 m, is similar to the average distance error, 0.5967 m, of the environment consisting mainly of specular materials in Figure. 2.12(a); there is only a 3 % decrease due to the added absorption materials.

In the scene containing the obstacle in Figure. 2.15(b), the average distance error, 0.6998 m, deteriorates compared to the average distance error, 0.6351 m, of the environment consisting of the majority of specular materials in Figure. 2.13(a); about an 10% decrease due to the added absorption materials. When the sound source becomes an NLOS state and direct propagation paths are blocked, the wall w/ the diffuse materials (blue rectangle) becomes the main material to generate prominent propagation paths. However, those prominent propagation paths cannot detected by the DoA estimator utilized in this Chapter. 2 since most energy has been absorbed by the diffuse materials, and this situation is the reason of the deterioration in Figure. 2.16(b).

Even though the portion of specular materials decreases, the proposed approach shows reasonable

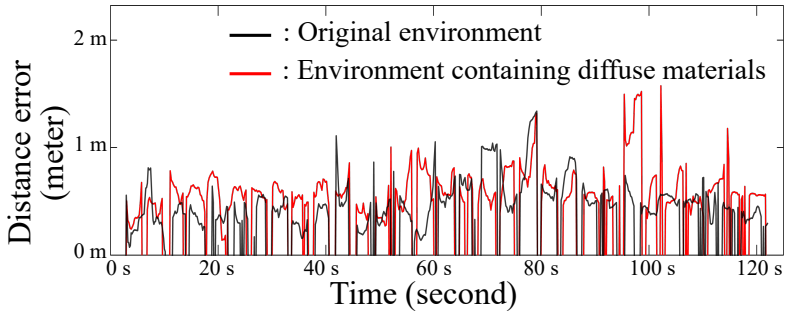


(a) Environment w/ absorption materials w/o an obstacle.

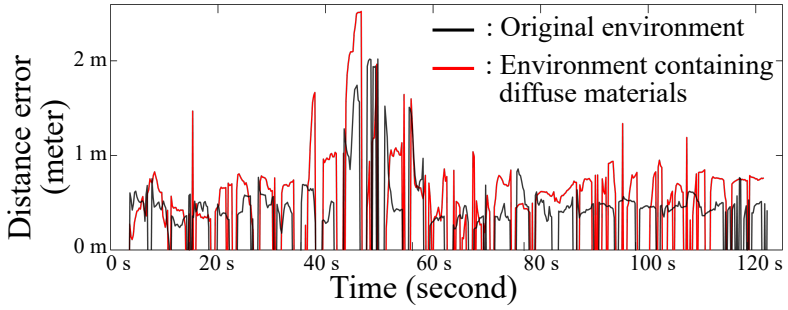


(b) Environment w/ diffuse materials and an obstacle.

Figure 2.15: Environments with one moving source containing high absorption materials, i.e., acoustic soundproofing foam consisting of a sponge, without and with an obstacle. In (a) and (b), I replace part of the specular materials with the diffuse materials from the environments in Figure. 2.11; the specular materials are indicated by the green rectangles and the diffuse materials are indicated by the blue rectangle. These walls strongly affect the proposed approach, as the source moves from the left end to the right end of the walls consisting of the specular (green rectangles) and diffuse (blue rectangle) materials; many propagation paths coming from the moving source to the microphone array interact with those highlighted materials.



(a) Distance errors w/ absorption materials w/o an obstacle.



(b) Distance errors w/ absorption materials and an obstacle.

Figure 2.16: Distance errors, i.e., red graphs, in the environments in Figure. 2.15 containing diffuse materials without and with an obstacle.

localization accuracy compared to the previous environments whose most materials are specular materials; 3 % and 10 % decrease in both scenes. This graceful degradation is achieved since the proposed method still generates and processes a similar number of acoustic rays. The averages of total significant rays of environments containing absorption materials are 18.46 (w/o obstacle), 19.78 (LOS source w/ obstacle), and 13.8 (NLOS source w/ obstacle), respectively; the detailed results for primary, reflection, and diffraction rays are shown in Table. 2.1. These values are similar to the previous environments containing many specular materials, i.e., 16.83 (w/o obstacle), 19.2 (LOS source w/ obstacle), and 13.78 (NLOS source w/ obstacle), respectively. The sound propagation paths that are absorbed by absorption materials and thus are not detected by microphones are compensated by other propagation paths caused by other specular materials, i.e., the green rectangles in Figure. 2.15.

2.5.4 The compatibility w/ different microphone arrays

So far, I basically utilized the 8 channel cube shape microphone array with the delay-and-sum (DAS) beamformer. To show that the proposed approach can be combined with different types of microphone arrays and DoA estimators, I tested the proposed method using the 32 channel microphone arrays [49] with EB-MVDR beamformer [50] that is one of the state-of-the-art DoA estimator.

I tested the different microphone array and DoA estimator in the environment without and with an obstacle (Figure. 2.11), and distance errors are shown in Figure. 2.17. The distance errors w/ and w/o an obstacle are 0.5946 m and 0.6176 m, respectively. These results tell us that the proposed approach can work well based on different types of the microphone array and DoA estimators; both average distance errors are similar or slightly smaller compared to results of the 8 channel microphone array and DAS beamformer, i.e., 0.5967 m and 0.6351 m. The reason why those results are slightly better

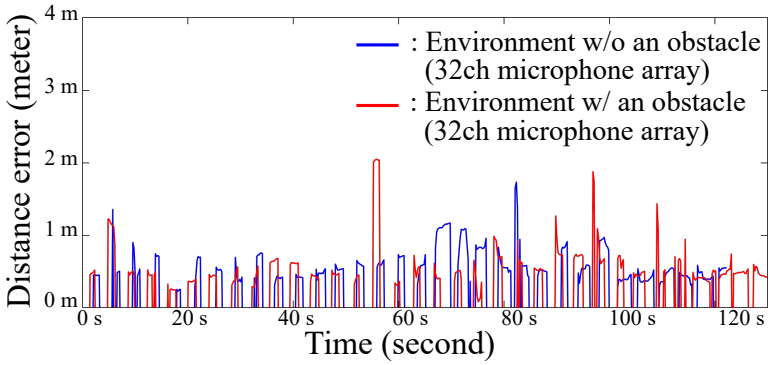


Figure 2.17: Distance errors in the environment shown in Figure. 2.11 without and with an obstacle when using a different microphone array and the DoA estimator: the 32 channel microphone array, i.e., Eigenmike, and EB-MVDR beamformer.

Table 2.1: Quantitative results of single source scenarios

Environments (Figure. 2.11 and 2.15)	w/o obstacle	w/o obstacle	w/ obstacle	w/ obstacle	absorp. mat.		EB-MVDR w/ 32-ch array	
					w/o obstacle	w/ obstacle	w/o obstacle	w/ obstacle
Source signal	clapping	speech	clapping	speech	clapping	clapping	clapping	clapping
Avg. distance error / std. of prior work	1.6 m / 0.6776	1.7769 m / 0.7615	1.582 m / 0.758	1.7571 m / 0.8112	1.7724 m / 0.7767	1.6324 m / 0.7353	- / -	- / -
Avg. distance error / std. of this work **(improvement over the prior work)**	0.5967 m / 0.2617 **(168 %)**	0.7416 m / 0.4448 **(139 %)**	0.6351 m / 0.3698 **(149 %)**	0.7313 m / 0.2679 **(140 %)**	0.6176 m / 0.2106 **(186 %)**	0.6998 m / 0.3858 **(133 %)**	0.5946 m / 0.2392 (-)	0.6176 m / 0.3165 (-)
Avg. significant primary / reflection / diffraction rays of LOS source	6.49 / 10.43 / 0	8.84 / 7.23 / 0	7.36 / 9.52 / 2.32	6.9 / 9.79 / 1.15	7.14 / 11.32 / 0	7.53 / 10.67 / 1.58	6.93 / 11.2 / 0	8.8 / 8.58 / 0.26
Avg. significant primary / reflection / diffraction rays of NLOS source	- / - / -	- / - / -	0.61 / 9.3 / 3.87	1.55 / 3.83 / 6.39	- / - / -	0.83 / 8.3 / 4.67	- / - / -	0.97 / 5.43 / 3.18

than the 8 channel microphone array is that the 32 channel microphone array has a higher number of channels, i.e., 32 ch, with the state-of-the-art DoA estimators. Even if the 32 channel microphone array has a better performance compared to the 8 channel microphone array, the proposed approach has acceptable accuracies w/ the 8 channel microphone array, which is much cheaper than the tested 32 channel microphone array.

2.5.5 Multiple sound sources

In general, localizing multiple sources is more difficult than handling a single source, as reverberant sounds tend to accumulate as the number of sources increases. Moreover, the proposed approach can detect up to N different DoAs at a single frame (Eq. 2.1). As a result, the number of allocated rays for each source decreases as there are more sources, and this can deteriorate the localization accuracy. First, I show results in an environment with multiple stationary sources (Figure. 2.18), remaining at fixed positions, and then present results for multiple moving sources (Figure. 2.20(a)).

Multiple stationary sources. In a multiple stationary source environment (Figure. 2.18), I conducted experiments on two scenes, one with two stationary sources and another with three stationary sources. For the former, I place two sources at the positions of source 1 and 2, highlighted by red circles in Figure. 2.18, where source 1 and 2 emit clapping sounds and human speech, respectively. For the scene

Figure 2.18: An environment with multiple sources. I place up to three sound sources in a room environment. Each red circle indicates a sound location, with each source numbered as source 1, source 2, and source 3.

with three stationary sources, I place three sources at the positions of source 1, 2, and 3 in Figure. 2.18, where source 1 emits human speech, and source 2 and 3 emit clapping sounds. Sources 2 and 3 are active from 0 s to 25 s and from 30 s to 70 s, respectively, and they are intermittent sound sources.

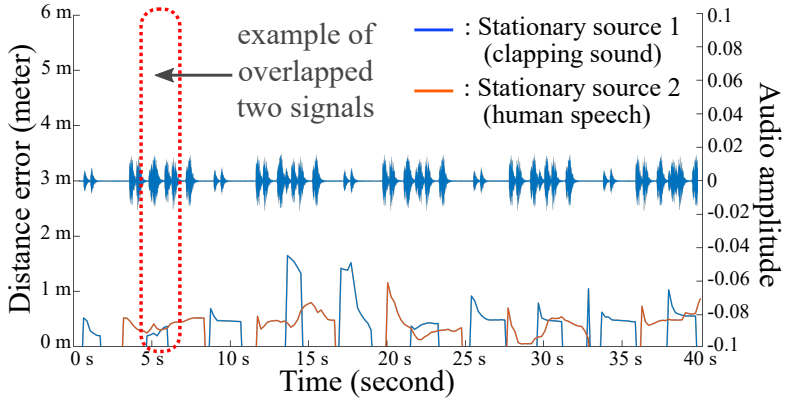
The localization errors of two and three stationary sources scenes are shown in Figure. 2.19. In the scene with two stationary sources, the average distance errors of the proposed approach are 0.5947 m (source 1) and 0.4306 m (source 2), and the average distance errors of the prior work are 1.6712 m (source 1) and 1.6662 m (source 2). In the scene with three stationary sources, the average distance errors of the proposed approach are 0.4263 m (source 1), 0.4856 m (source 2), and 0.5185 m (source 3), and the average distance errors of the prior work are 1.3286 m (source 1), 1.717 m (source 2), and 1.0551 m (source 3); sources 2 and 3 are intermittent sound sources. These results demonstrate that the proposed approach can localize multiple sources reasonably well.

Especially, even if two audio signals coincide, the proposed approach can localize both overlapped signals separately, e.g., a case at 5 s in Figure. 2.19(a) that is highlighted by a dotted box. Furthermore, even if there are intermittent sound sources, i.e., the source 2 and 3 in the three sources scene, the proposed approach can distinguish activation and inactivation of intermittent sources. In Figure. 2.19, when the source 2 is active from 0 s to 25 s, and the source 3 is activate from 30 s to 70 s, the proposed approach only localizes source positions when sources are active and does not react properly to inactivated sources.

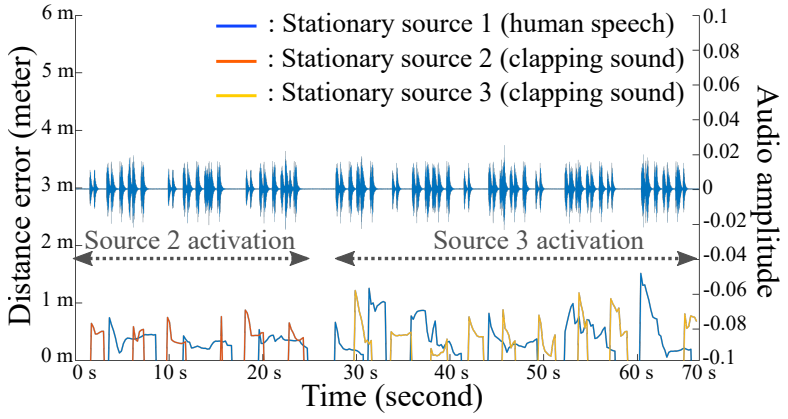
Multiple moving sources. For testing the proposed approach to the scene with multiple moving sources, I place two sources in Figure. 2.20(a), and they follow their trajectories, where sources 1 and 2 emit clapping sound and human speech, respectively. I also put two obstacles between moving sources and the robot to build a more challenging environment; both obstacles block the direct propagation paths of each source.

Distance errors of multiple moving sources are show in Figure. 2.20(b). The average distance errors of the proposed approach are 0.7689 m (source 1) and 0.7246 m (source 2). Since this scenario containing multiple moving sources is challenging, these errors are higher than single source scenarios in Figure. 2.13(a). The accuracy of moving source 1 (clapping sounds) is 17 % decreased compared to one moving source scene containing the obstacle with clapping sounds (Figure. 2.11(b)); the environment setups of both experiments are almost the same, i.e., the same obstacle size and the similar trajectory of the source. Since multiple sources generate more reverberant sounds and the total number of generated rays for each source decreases, the accuracy of moving source 1 becomes worse.

The average distance errors of the prior work is presented in Table 2.2: 1.36 m (source 1) and

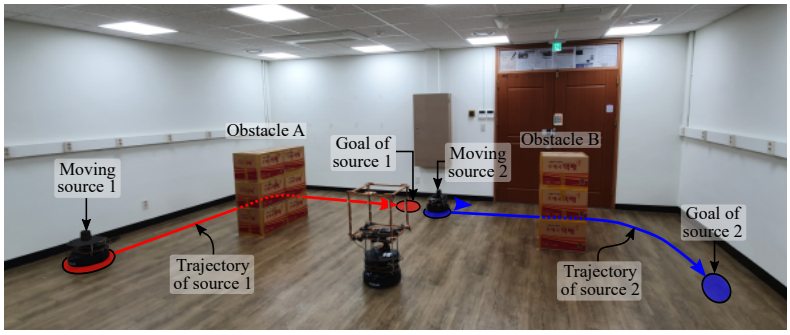


(a) Two stationary sources.

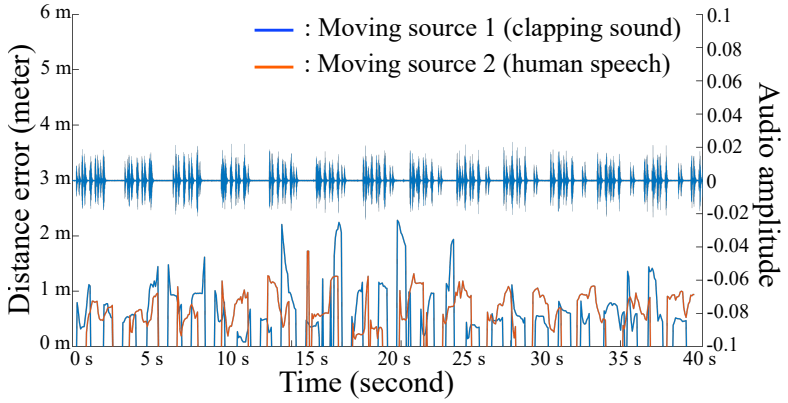


(b) Three stationary sources

Figure 2.19: Distance errors and amplitudes of the measured audio signals of scenes with two (a) and three (b) stationary sources. Sound sources numbering from 1 to 3 correspond to the sources, denoted by the red circles, in Figure. 2.18. The distance errors of the sources are plotted using lines with different colors, and the amplitudes of the measured audio signals are also presented.



(a) Environment with two moving sources.



(b) Accuracy of two moving sources

Figure 2.20: The environment of multiple moving sources in (a) and its accuracy in (b). There are two moving sources, i.e., moving source 1 and 2, and they follow trajectories. Both obstacles, i.e., the obstacle A and B, cause the non-line-of-sight states of each moving source.

1.812 m (source 2). Compared to the average distance errors of the prior work, I can observe that the proposed approach shows better result; 76 % and 150 % improvement for source 1 and 2, respectively.

2.5.6 Different environment sizes

Thus far, I have tested the proposed approach in environments of identical dimensions, i.e., 7 m×7 m in area and with a 3 m height. To determine how different sizes of the environment affect the proposed method, I conducted an experiment in a room 7 m×3.5 m in size and with a height of 3 m as shown in Figure. 2.21(a). I also measured localization accuracy by increasing the distance between the robot and the source from 1.25 m to 4 m.

Figure. 2.21(a) shows twelve locations of sound sources from the source 1 to 12; two adjacent sources have the same distance interval of 0.25 m. I place the sound source at one of source locations, and the source emits the clapping sound for 20 seconds. To demonstrate the benefits of the proposed approach, I also tested the prior work and the proposed approach. The accuracy of the two cases are shown in Figure. 2.21(b).

I observe that the proposed method helps to improve the accuracy of SSL by using acoustic rays since all average distance errors of the proposed approach are smaller than the prior work. By increasing the distance between the robot and the source, the accuracy generally deteriorates in both cases.

The prior work utilizes the time differences of arrival sound to each microphone to estimate the

Table 2.2: Quantitative results of multiple source scenarios.

Scene	Two stationary sources	three stationary sources	two moving sources (Figure. 2.20(a))
Source 1 Avg. distance error / std. of prior work	1.6712 m / 0.6436 (clapping)	1.3286 m / 0.6328 (speech)	1.36 m / 0.6663 (clapping)
Source 2 Avg. distance error / std. of prior work	1.6662 m / 0.6393 (speech)	1.717 m / 0.7958 (clapping)	1.812 m / 0.7043 (speech)
Source 3 Avg. distance error / std. of prior work	- / -	1.0551 m / 0.5016 (clapping)	- / -
Source 1 Avg. distance error / std. of this work **(improvement over the prior work)**	0.5947 m / 0.3452 (clapping, **181 %**)	0.4263 m / 0.271 (speech, **211 %**)	0.7689 m / 0.4629 (clapping, **76 %**)
Source 2 Avg. distance error / std. of this work **(improvement over the prior work)**	0.4306 m / 0.1895 (speech, **286 %**)	0.4856 / 0.1488 (clapping, **246 %**)	0.7246 / 0.2843 (speech, **150 %**)
Source 3 Avg. distance error / std. of this work **(improvement over the prior work)**	- / -	0.5185 / 0.2787 (clapping, **103 %**)	- / -

distance between the microphone array and the source. As the source becomes far away from the microphone array, the change in the time differences of arrival sound decreases. Thus, it becomes difficult to accurately estimate the distance between the microphone array and the source from time differences; the localization accuracy decreases in longer distances, reported in Figure. 2.21(b).

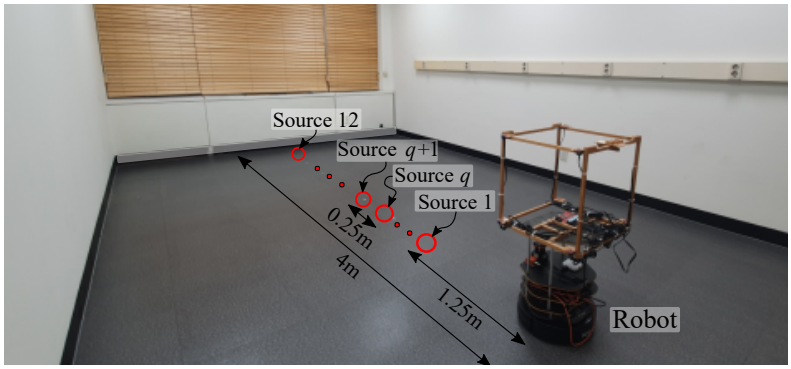
This is attributed by the accumulated propagation errors of acoustic ray paths caused by various noises, e.g, sensor noises of the laser scanner, audio noises of microphones, and odometry noises of the mobile robot. These noises adversely affect the proposed localization algorithm and cause propagation errors. Nonetheless, the proposed approach shows the better and stable accuracy compared to the prior work only considering direct sound.

2.5.7 Navagating to the NLOS source

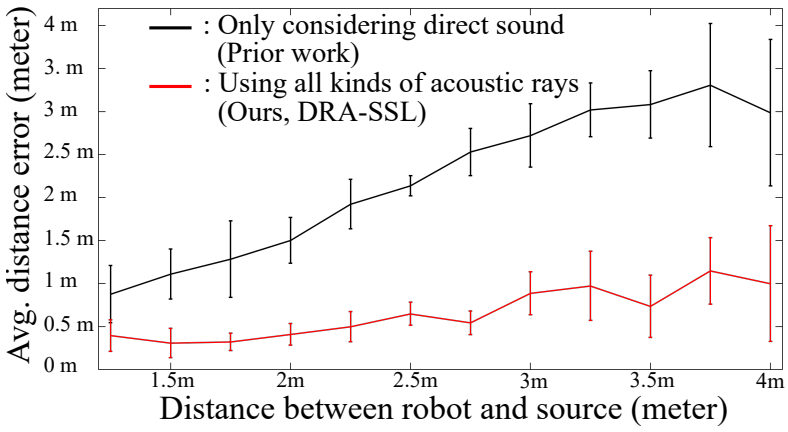
I expect that the proposed approach can be applied to various tasks in robotics. Especially, the proposed approach is useful in cases of containing an NLOS source; the vision based localization approaches do not deal with these cases due to the occlusion by obstacles. Assuming that a user orders the robot to bring something, e.g., a cup of water, the robot has to detect and localize the user and then navigate to the location of the user. If there is an obstacle between the user and the robot, the vision sensor does not see the user, but the sound can be heard through indirect sound propagation; sound becomes very crucial information in the NLOS source cases.

I applied the proposed approach to the navigation task. When the source emits the clapping sound at a specific goal position, which is unknown for the robot, the robot localizes the source and navigates to the estimated goal position by the proposed localization method. To simulate the NLOS case, the sound source is occluded by an obstacle (Figure. 2.22(a)), and I tested the proposed method and the prior work to localize the source.

If localization methods, i.e., the proposed approach and prior work, produce the estimated source position for 2 seconds, the robot sets an estimated goal position as the mean of estimated source positions

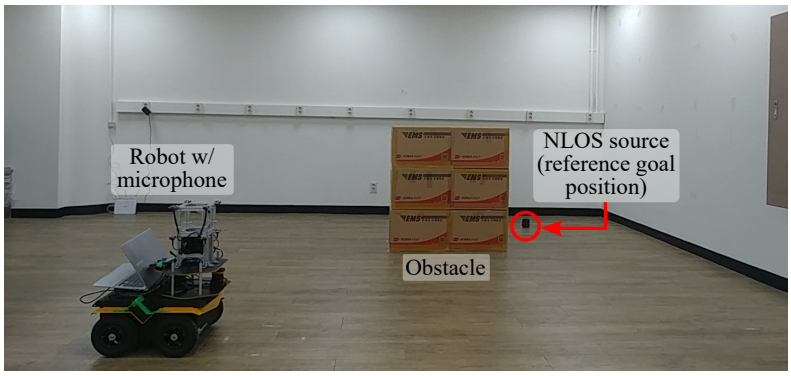


(a) Small testing room

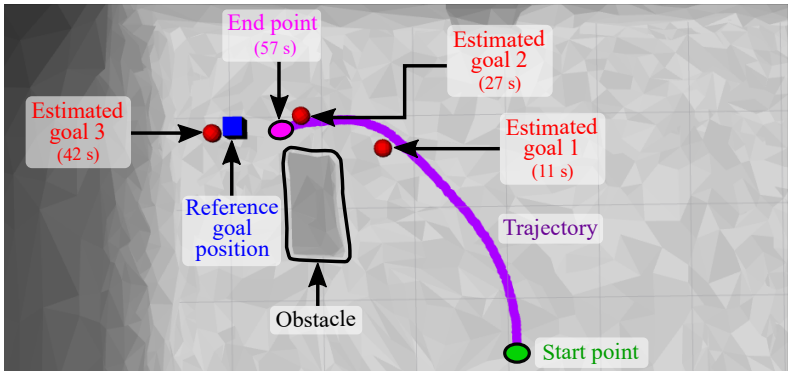


(b) The average distance errors over different source locations.

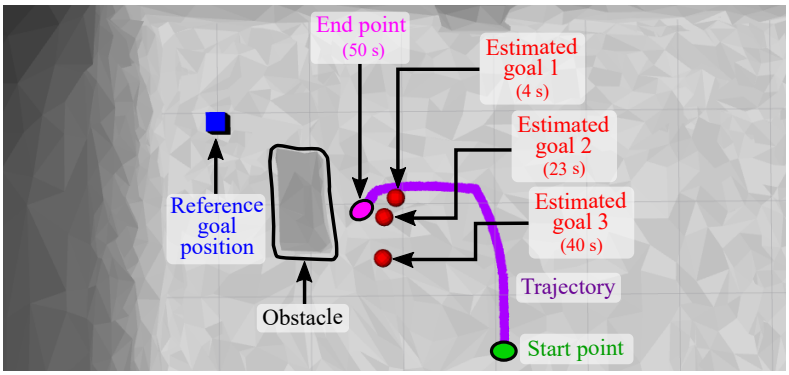
Figure 2.21: (a) shows another testing environment with a small size of 7 m×3.5 m in area and with a 3 m height. Red circles denote tested different source positions whose distance from the robot varies from 1.25 m to 4 m by 0.25 m interval. (b) shows the average distance errors at different source positions. The vertical lines represent the one standard deviation of the average distance errors.



(a) Test environment for navigation to the NLOS case



(b) Estimated goal positions and computed trajectory for navigation (this work)



(c) Estimated goal positions and trajectory for navigation (prior work [1])

Figure 2.22: (a) shows test environment for the navigation task to the NLOS source. (b) and (c) show results of navigation tasks of this work and the prior work, respectively. The blue cubes denote the reference goal position generating a clapping sound, and red spheres represent the estimated goal position at each time. The purple lines are the computed trajectory of the robot given the start point (green circle) and the end point (purple circle).

of localization methods; the duration of the clapping sound is about 2 seconds. During the navigation tasks, the sound source plays the sound clip three times periodically in order to show the localization result over the different robot positions. The robot stops navigation process once the distance to the estimated goal position is less than 1 m.

I utilize Jackal as a mobile robot platform. Jackal provides the open source of the navigation in the ROS system, and I use this open source in this experiment where the linear and angular velocities are 0.1 m/s and 0.314 rad/s, respectively, the linear and angular accelerations are 2.0 m/s^2 and 4.0 rad/s^2, respectively, and other parameters are set to default values. The microphone array is the same 8-ch cube-shaped type as in previous experiments.

The results of navigation tasks of this work and the prior work are shown in Figure. 2.22(b) and 2.22(c). I observe that the robot can reach the reference goal position at 57 s given the proposed localization approach, as shown in Figure. 2.22(b). On the other hand, the robot with the prior work does not consider indirect sound paths, failing in reaching the reference position; the navigation task of the prior work is stopped at 50 s. Especially, the estimated goal positions of the proposed approach are gradually getting close to the reference goal position; the distance errors of estimated goal 1, 2, and 3 are 1.5292 m at 11 s, 0.659 m at 27 s, and 0.3361 m at 42 s, respectively. The estimated goal positions of the prior work get worse while the robot moves closely to the obstacle; the distance errors of estimated goal 1, 2, and 3 are 1.9397 m at 4 s, 2.09 m at 23 s, and 2.1669 m at 40 s, respectively.

The prior work that considers only direct sound was not able to handle the NLOS case caused by the obstacle, but the proposed approach can reach the destination, i.e., the NLOS source, mainly thanks to the consideration of diffraction.

2.6 Conclusion

I have presented a novel reflection and diffraction-aware sound source localization algorithm by utilizing acoustic ray tracing and Monte Carlo localization for multiple sound sources. The proposed approach can also localize non-line-of-sight sound sources and model diffraction using the uniform theory of diffraction. I have evaluated the proposed method in various scenarios with static and moving single or multiple sources using different sound signals. I have also analyzed the properties of the proposed method across a diverse set of configurations with different materials, room sizes, beamforming algorithms, etc. I applied the proposed approach to the navigation task and confirmed the usefulness of the proposed approach.

While I have demonstrated the benefits of the proposed approach, it has some limitations that need to be addressed by future work. The UTD model is an approximate model and mainly designed for infinite wedges. Its accuracy can deteriorate on obstacles that have smooth surfaces. More accurate wave-based diffraction models can be used to deal with this problem, but achieving real-time performance remains as a main technical challenge.

The proposed approach works based on interactions, i.e., reflection and diffraction, with obstacles, and is not suitable for outdoor environments where I do not have obstacles causing interaction. As mentioned in Chapter. 2.5.5, the proposed method may not work property when reverberation becomes prominent. This issue can be mitigated by utilizing semantic information of sound signal that each sound source carries. Overall, I believe that the proposed work takes a meaningful step for sound source localization, and considering the aforementioned issues can open up new research directions.

Chapter 3. Sound Source Localization considering Similarity of Back-Propagation Signals

3.1 Introduction

There has been a significant amount of efforts to localize a sound source by estimating direction of arrival (DoA) of sound waves. Thanks to the advantage of the spherical configuration, many DoA estimation methods focus on using the spherical microphone arrays. Rafaely [65] presented a theoretical framework of spherical harmonic array processing, and the delay-and-sum beamformer is extended to process on the spherical harmonics domain. Many advanced beamforming techniques [50, 66, 67] were proposed by using the minimum variance distortionless response (MVDR) power spectra on the spherical harmonics domain. Li *et al.* [68] presented a MUSIC (Multiple Signal Classification) based DoA estimation algorithm, which uses orthogonality between a noise-only subspace and a signal-plus-noise subspace on the spherical harmonics domain.

Unfortunately, these methods were designed for detecting DoA, not the 3D location of a sound source in an arbitrary environment. Especially, when a sound source is occluded by an obstacle, most prior approaches cannot specify the location of the source generating the sound signal.

To address this issue, I proposed a 3D sound source localization method based on acoustic ray tracing techniques in Chapter. 2. These techniques estimate sound propagation paths from the source to microphones as acoustic rays, generated by the ray tracing technique, and identify the 3D source location by using generated acoustic rays. However, the accuracy of these methods decreases in environments with background noise and imperfect reconstruction of the 3D environments. This low accuracy is caused mainly because several errors, like background noise and imperfect 3D reconstruction, are accumulated along each acoustic ray for estimating the source location.

Main Contributions. To robustly identify the sound source location, I present a novel, sound source localization algorithm using back-propagation signals (Figure. 3.1). Using a beamforming algorithm, I first compute DoA of the sound wave and separation signals corresponding to those specific DoAs (Chapter. 3.2.1). I then estimate sound propagation paths by generating acoustic ray paths in the reverse direction to DoAs of the sound (Chapter. 3.2.2), and compute the back-propagation signals using the impulse response of the acoustic ray path from the separation signal (Chapter. 3.2.3). Intuitively speaking, back-propagation signals are virtually computed signals that could be heard at a particular location on acoustic paths from the measured signals at the microphone array.

Finally, I use the Monte Carlo localization algorithm estimating a location of the sound as a converging region of computed acoustic ray paths. In particular, I utilize the computed back-propagation signals of different acoustic ray paths for robust estimation of the sound location, under the intuitive assumption that acoustic paths coming from the same sound source should have similar back-propagation signals at the estimated location (Chapter. 3.2.4).

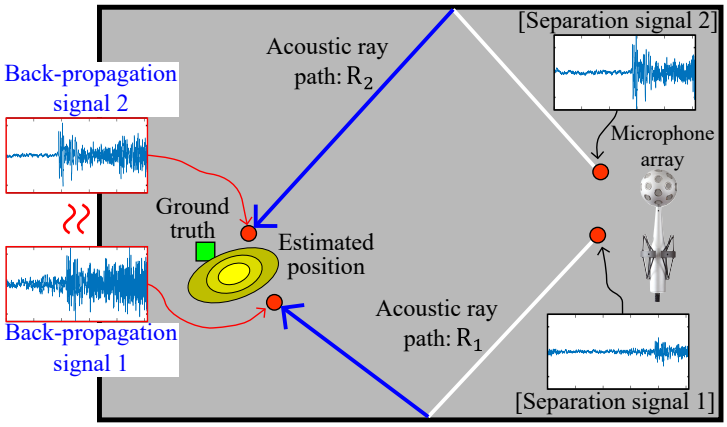


Figure 3.1: The proposed approach generates direct and indirect acoustic ray paths and localizes the sound source while considering back-propagation signals on generated acoustic ray paths. The back-propagation signals are virtually computed signals that could be heard at particular locations and computed by using impulse responses. When two back-propagation signals of acoustic ray paths are highly correlated, I treat them to be originated from the same source.

3.2 Sound source localization using back-propagated signals

The proposed work is built upon ray tracing-based sound source localization (SSL) [69]. In a real environment involving moving sound sources, obstacles, or noise, acoustic rays generated naively by the prior ray tracing based SSL may converge to a position other than the actual location of the sound source.

To solve this problem, I aim to generate and utilize back-propagation signals to a candidate 3D location along each acoustic ray. This back-propagation signals at a location can be computed by simulating the reverse process of sound propagation, i.e., by reversely performing ray tracing.

3.2.1 Beamforming

To generate acoustic rays, I estimate DoAs of the sound waves at the spherical microphone array using a EB-MVDR (Eigenbeam-minimum variance distortionless response) beamformer [50,66,67]. Note that the input signals are measured by almost uniformly sampled microphones on a rigid sphere (32 channel microphone positions), but each microphone signal is, in fact, a mixture of signals from different directions. I therefore aim to extract signals from different DoAs, and for this purpose, the EB-MVDR beamformer is utilized.

The array signal, $\mathbf{x} = [x_1(k), \cdots, x_Q(k)]^T$, measured by Q microphones of the spherical array consists of sound pressure signals $\mathbf{p} = [p_1(k), \cdots, p_Q(k)]^T$ and noise signals, $\mathbf{n} = [n_1(k), \cdots, n_Q(k)]^T$:

$$\mathbf{x} = \mathbf{p} + \mathbf{n}, \tag{3.1}$$

where $k = 2\pi f/c$ is the wavenumber determined by the frequency f and speed of sound c. Note that the measured sound signal $\mathbf{p}$ is the consequence of sound propagation and reflections through a direct or indirect propagation path. I apply the spherical Fourier transform (SFT) to the array signal $\mathbf{x}$ [65], which yields the spherical harmonic (SH) coefficients $\mathbf{x}_{\nu\mu}$ defined over different orders ν and degrees μ of spherical harmonics. For the SH coefficients measured up to the order $\nu = \nu'$, there are $(\nu' + 1)^2$

coefficients in total. Since the SFT is a linear operation, I also have the following relation:

$$\mathbf{x}_{\nu\mu} = \mathbf{p}_{\nu\mu} + \mathbf{n}_{\nu\mu}, \tag{3.2}$$

The objective in this Chapter. 2 is to identify DoAs and extract the sound signal coming from each DoA. The beamformer does this by multiplying a beamformer weight vector $\mathbf{w}_{\nu\mu}(\Omega)$ defined for a specific pair of zenith and azimuth angle $\Omega = (\theta, \phi)$ to the measured signal $\mathbf{x}_{\nu\mu}$. The output of the beamformer S, therefore, can be written as the inner product of $\mathbf{w}_{\nu\mu}(\Omega)$ and $\mathbf{x}_{\nu\mu}$:

$$S(\Omega) = \mathbf{w}_{\nu\mu}(\Omega)^H \mathbf{x}_{\nu\mu}, \tag{3.3}$$

where $(\cdot)^H$ is the Hermitian transpose. Among many beamformers, I adopt the EB-MVDR that is known to provide a good spatial resolution and signal separation performance. With the EB-MVDR beamformer, DoAs are estimated from the beamforming power defined as:

$$\beta_{MV}(\Omega) = \frac{1}{\mathbf{v}_{\nu\mu}(\Omega)^H \mathbf{R}^{-1}_{\mathbf{x}_{\nu\mu}\mathbf{x}_{\nu\mu}} \mathbf{v}_{\nu\mu}(\Omega)}, \tag{3.4}$$

where $\mathbf{R}_{\mathbf{x}_{\nu\mu}\mathbf{x}_{\nu\mu}}$ is the covariance matrix of which elements are cross-spectral densities of measured signals $\mathbf{x}_{\nu\mu}$, and $\mathbf{v}_{\nu\mu}(\Omega)$ denotes a steering vector given by the wave propagation model. In this work, I use the plane wave model to define the steering vector $\mathbf{v}_{\nu\mu}$. Figure. 3.2 shows the beamforming power calculated for every direction Ω; all directions correspond to 10242 grids on the unit sphere that is based on the recursive subdivision of an icosahedron [70].

Local maxima of the beamforming power can represent the direct and indirect DoAs of the propagation paths. That is

$$[\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N] = F_{max}\{\beta_{MV}(\Omega)\}, \tag{3.5}$$

where $\mathbf{d}_n = (\cos\phi_n \sin\theta_n, \sin\phi_n \sin\theta_n, \cos\theta_n)$ denotes a directional vector of the n-th local maximum of the beamforming power among N different local maxima in a frame, and $F_{max}\{\cdot\}$ is a function for finding local maxima of the beam energy function. In practice, I identify top-four local maxima on average in the tested experiments in this Chapter. 3.

I then extract sound signals, called the separated signal S_n, coming from a specific direction Ω_n with the directional vector $\mathbf{d}_n$. The beamformer weight $\mathbf{w}_{\nu\mu}(\Omega_n)$ of the EB-MVDR beamformer is given by:

$$\mathbf{w}_{\nu\mu}(\Omega_n) = \frac{\mathbf{v}_{\nu\mu}(\Omega_n)^H \mathbf{R}^{-1}_{\mathbf{x}_{\nu\mu}\mathbf{x}_{\nu\mu}}}{\mathbf{v}_{\nu\mu}(\Omega_n)^H \mathbf{R}^{-1}_{\mathbf{x}_{\nu\mu}\mathbf{x}_{\nu\mu}} \mathbf{v}_{\nu\mu}(\Omega_n)}, \tag{3.6}$$

which minimizes the total beamforming power while satisfying the distortionless-response constraint to the looking direction Ω_n $\left(\mathbf{w}_{\nu\mu}(\Omega_n)^H \mathbf{v}_{\nu\mu}(\Omega_n) = 1\right)$. This beamformer weight is used in Eq. 3.3 for computing four separated signals S_n.

The separated signals are then back-propagated to the directions $\mathbf{d}_n$ by reconstructing acoustic rays to the true source positions.

3.2.2 Acoustic ray tracing

I explain how to generate acoustic rays from estimated directions $[\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N]$ that are the reverse directions of incoming sounds. I want to estimate propagated paths (e.g., direct and reflection paths) of the sound from its source location to the microphone array location using the acoustic rays. I generate such acoustic rays considering direct and reflection paths based on the RA-SSL algorithm [69].

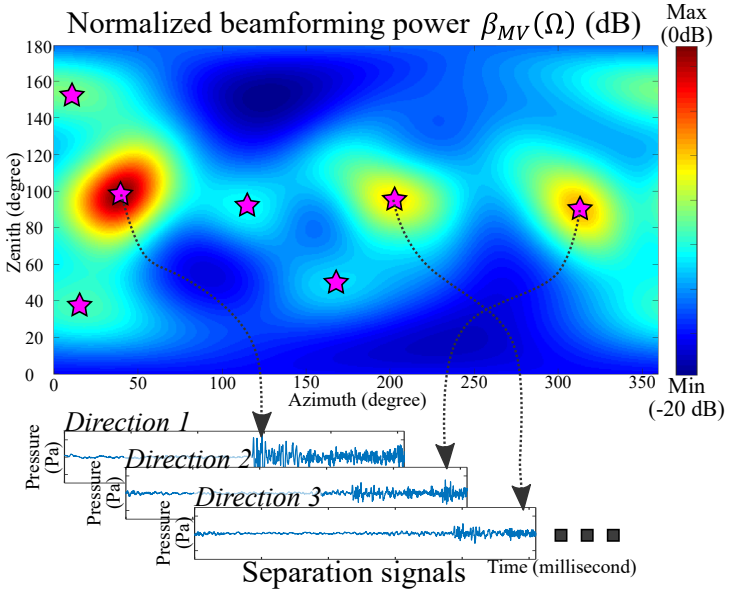


Figure 3.2: A beamforming power is computed by a beamforming algorithm, where the horizontal axis is the azimuth angle and the vertical axis is the zenith angle of the unit sphere. Local maxima of the beamforming power are treated most significant directions of arrival (DoAs) of sound. The sound signal impinging from each DoA is extracted by applying the EB-MVDR beamformer to the signals measured by microphones.

Unlike the prior work of RA-SSL, I use a mesh representation of the surroundings captured from sensors. I construct the mesh that is robust to minor noise, and use it for acoustic interactions between the surroundings and generated acoustic rays. Starting from the point cloud collected by the depth sensor, i.e., Velodyne VLP-16, I apply the voxelization in order to reduce the sensor noise, and then reconstruct the environment in the form of a mesh map from the voxelized point cloud using the Poisson surface reconstruction algorithm [71].

For the n-th acoustic ray path, denoted by R_n, its primary acoustic ray, r_n^0, is created into the n-th direction vector $\mathbf{d}_n$, as shown in Figure. 3.3. If the acoustic ray collides with an obstacle, its secondary, reflection ray is generated by assuming the specular reflection, and is denoted by r_n^1, where the superscript represents the order of the acoustic ray path; refer to [69] for the detailed process on ray generation. When R_n is propagated until a $(D-1)$-th order, it indicates that the acoustic ray path R_n consists of D acoustic rays: i.e., $R_n = [r_n^0, r_n^1, \cdots, r_n^{D-1}]$.

3.2.3 Back-propagation signals

I introduce how to compute back-propagation signals based on the generated acoustic ray paths $[R_1, R_2, \cdots, R_N]$ and separated signals $[S_1, S_2, \cdots, S_N]$; there is a tuple of (R_n, S_n) for the reverse direction vector $\mathbf{d}_n$ of the n-th incoming sound. I want to compute the back-propagation signal P_n from the separated signal S_n by designing and using an impulse response of backward sound propagation based on the acoustic ray path R_n. The impulse response describes the reaction of any linear system as a function of time-independent variables; the input is the separated signal and the output is the back-propagation signal in the proposed approach.

In this work, I utilize the impulse response for the backward propagation to improve the accuracy

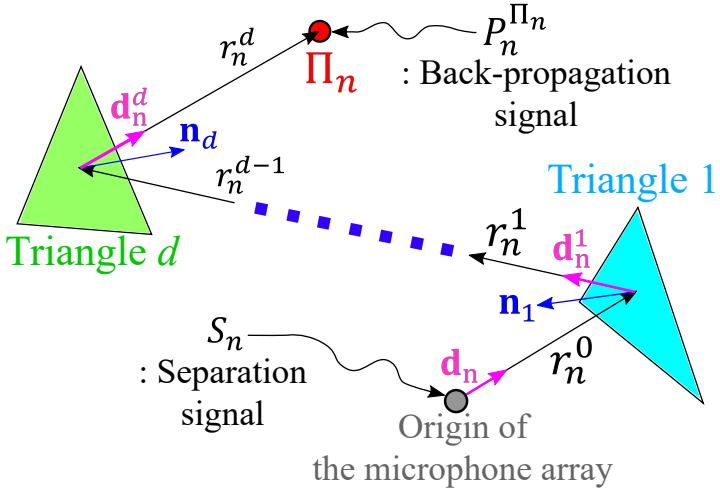


Figure 3.3: An example of generating an acoustic ray path R_n and its back-propagation signal. The primary acoustic ray, r_n^0, of the n-th acoustic ray path R_n is generated to the direction vector $\mathbf{d}_n$ that is the reverse direction of the n-th incoming sound. When the acoustic ray r_n^0 hits an obstacle represented by Triangle 1, its reflection acoustic ray r_n^1 is generated according to the specular reflection based on the normal vector $\mathbf{n}_1$ of Triangle 1. The back-propagation signal P_n is computed by using the impulse response of R_n at a specific point, Π_n, on the path from the separated signal S_n.

of the sound source localization. In forward sound propagations [30, 31, 72, 73], the impulse response of an acoustic ray path is described by sound attenuations according to the travel distance of a ray path and reflection. For example, the travel distance attenuation represents the decrease of sound pressure inversely proportional to the travel distance of the ray path, because the sound is propagated according to the spherical wave in 3D environments; similar for the reflection attenuation.

On the other hand, for the backward propagation problem, the attenuation of travel distance and reflection becomes an amplification of the sound pressure. Suppose that I aim to compute the back-propagation signal from the starting point to a specific point, Π_n (Figure. 3.3), on an acoustic ray path using the backward impulse response, where there is the n-th tuple (R_n, S_n) and the acoustic ray path R_n consists of D acoustic rays $[r_n^0, \cdots, r_n^{D-1}]$; r_n^0 is a primary ray and r_n^d is the d-th reflection ray $(1 \leq d \leq D-1)$. In the frequency domain, the backward impulse response, $H_n^{\Pi_n}$, is described by amplifications because of the travel distance l and the reflection until the d-th order reflection ray r_n^d:

$$H_n^{\Pi_n}[k] = \exp\left(\frac{\mathbf{i}kl}{c}\right) \cdot A^T[l] \cdot A^R[R_n, d, k], \tag{3.7}$$

where the term inside the exponential function is for shifting the back-propagation signal to the time delay of the sound propagation at the specific point Π_n and $\mathbf{i}$ is the imaginary unit. A^T is a coefficient of the travel distance amplification, and is defined by a function of the travel distance l: $A^T[l] = 4\pi(1+l)$. Also, A^R is a coefficient of the reflection amplification, and is defined by considering specular reflections until the d-th order reflection ray:

$$A^R[R_n, d, k] = \prod_{\delta=1}^{d} \left[\frac{1}{\Gamma_\delta[k]}\right], \tag{3.8}$$

where Γ_δ denotes the reflectivity (reflection coefficient) of the triangle hit by the $(\delta-1)$-th order ray; the reflection coefficient is a function of wavenumber k and I refer to coefficient values reported by [74].

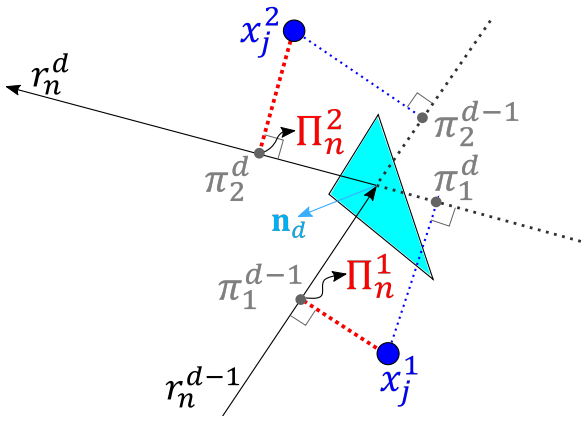


Figure 3.4: Examples of determining the point of the acoustic ray path for computing the back-propagation signal. For the particle of x_j^2, the perpendicular foots π_2^d on all d-th order acoustic rays of the n-th acoustic ray path are computed. I then decide the representative perpendicular foot Π_n^2 satisfying the shortest distance from x_j^2 to R_n.

The back-propagation signal $P_n^{\Pi_n}$ at the specific point Π_n on the acoustic ray path R_n is finally computed by the product of the backward impulse response $H_n^{\Pi_n}$ and the separated signal S_n in the frequency domain:

$$P_n^{\Pi_n}[k] = S_n[k] \cdot H_n^{\Pi_n}[k]. \tag{3.9}$$

3.2.4 Estimating a source position

The proposed estimation process of localizing the sound source is based on the Monte Carlo (MC) localization method. The MC sound source localization identifying the convergence region of acoustic ray paths was suggested in the prior work (RA-SSL) [69]; the convergence region means the area where acoustic ray paths gather.

However, the accuracy of MC localization can decrease in real environments. When there are background noises of sound or complex scene configurations causing uncertainty of the reconstructed environment, they can trigger to generate many arbitrary or incoherent acoustic ray paths. By considering back-propagation signals, I aim to identify those arbitrary and incoherent acoustic ray paths and cull away acoustic ray paths with different back-propagation signals indicating that they are from different sound sources. Intuitively speaking, if there are two acoustic ray paths caused by the same source, their back-propagation signals should be similar near the location of their sound source. In other words, when back-propagation signals of two acoustic ray paths are different at a location, the location is unlikely to be a candidate for a converging region of the sound source.

The MC localization consists of three parts: sampling, computing a weight of particles, and resampling. The main differentiation of the proposed approach over the prior technique is that the proposed method improves the localization accuracy based on a novel module for computing weights of particles based on the proposed back-propagation signals.

Suppose there are i-th particles, x_j^i, representing hypothetical locations of the sound source at a j frame. I compute how close the particle is to acoustic ray paths. For this, I define a specific point Π_n^i, which is decided to be the point satisfying the shortest distance between x_j^i and any point on the n-th acoustic ray path; i.e., $\Pi_n^i = \text{argmin}_{\pi_i^d} ||x_j^i - \pi_i^d||$, where π_i^d is the perpendicular foot on the d-th order

acoustic ray from the x_j^i position (Figure. 3.4). I then compute the proposed back-propagation signal according to Eq. 3.7 at the shortest point Π_n^i on the n-th acoustic ray path from the particle x_j^i.

From the back-propagation signal $P_n^{\Pi_n^i}[k]$ in the frequency domain, I compute the back-propagation signal $p_n^{\Pi_n^i}[t]$ in the time domain signal. I then calculate a particle weight, w_j^i, representing the probability of being a convergence region of the sound source, based on two factors: a distance weight, w_d, representing how away the particle is from the n-th acoustic ray path and a similarity weight, w_s, indicating how similar between $p_n^{\Pi_n^i}[t]$ and other signals given acoustic ray paths:

$$w_j^i = P(O_j|x_j^i) = \frac{1}{n_c}\sum_{n=1}^{N_j}[w_d(x_j^i, R_n)\cdot w_s(x_j^i, R_n)], \tag{3.10}$$

where N_j is the number of acoustic ray paths at the j frame, O_j is the observation containing $[P_1^{\Pi_1^i}, \cdots, P_{N_j}^{\Pi_{N_j}^i}]$ and $[R_1, \cdots, R_{N_j}]$, and n_c is a normalizing constant.

The distance weight w_d is calculated by using the Euclidean distance between the particle location x_j^i and the point Π_n^i:

$$w_d(x_j^i, R_n) = G(||x_j^i - \Pi_n^i||\,|0, \sigma_w), \tag{3.11}$$

where G is the Gaussian distribution function with the zero mean and a standard deviation σ_w. w_d is maximized when the particle x_j^i is on the perpendicular foot Π_n^i. The similarity weight $w_s(x_j^i, R_n)$ measures the similarity between the back-propagation signal $p_n^{\Pi_n^i}$ from the n-th acoustic ray path and ones of other acoustic ray paths:

$$\frac{1}{n_s}\sum_{m=1, m\neq n}^{N_j}\begin{cases}\frac{L-(1-\alpha)\cdot l_{cc}(n,m)}{L}, & \text{if } a_{cc}(n,m) > a_{th}\\ 0, & \text{otherwise,}\end{cases} \tag{3.12}$$

where n_s is the normalizing constant, L is the length of the back-propagation signal, $a_{cc}(\cdot)$ is the peak coefficient in a normalized range of -1 to 1, $l_{cc}(\cdot)$ is the peak coefficient delay, α denotes a parameter for adjusting the similarity weight, and a_{th} denotes the threshold value of $a_{cc}(\cdot)$. Both variables of $a_{cc}(\cdot)$ and $l_{cc}(\cdot)$ are computed by applying the cross-correlation operation between n-th and m-th signals:

$$\begin{aligned} a_{cc}(n,m) &= \max\{(p_n^{\Pi_n^i} \star p_m^{\Pi_m^i})[\tau]\},\\ l_{cc}(n,m) &= \text{argmax}_\tau\{(p_n^{\Pi_n^i} \star p_m^{\Pi_m^i})[\tau]\},\end{aligned} \tag{3.13}$$

where $\star$ is the cross-correlation operator.

As shown in Figure. 3.5, $a_{cc}(\cdot)$ represents how much both back-propagation signals are correlated, and l_{cc} shows the time difference of occurrence between both back-propagation signals. As both back-propagation singles are from the same sound source, ideally a_{cc} and l_{cc} become one and zero, respectively.

Getting back to Eq. 3.12, I treat that two back-propagation signals are similar, when their peak coefficient is bigger than the threshold, i.e., $a_{cc} > a_{th}$. In this case, I assign a higher weight according to the relative time delay of the length of the signal, $(\frac{L-(1-\alpha)\cdot l_{cc}}{L})$ that becomes a value in a range of α to 1; i.e., I give the highest weight when two signals are matched without any delay, under the assumption that those two signals are originated from the same sound source. If there is no back-propagation signal satisfying the condition, $a_{cc} > a_{th}$, the signal similarity weight w_s has a constant value α that is the smallest value of $(\frac{L-(1-\alpha)\cdot l_{cc}}{L})$.

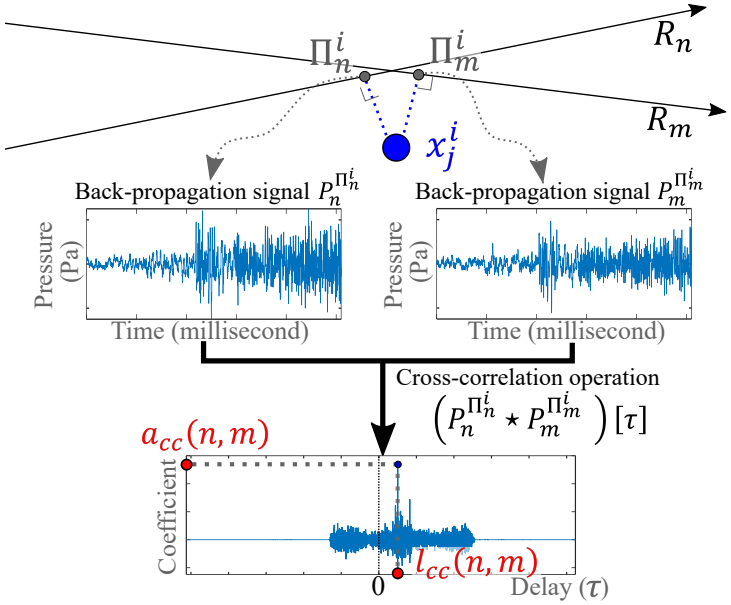


Figure 3.5: An example of computing the peak coefficient a_{cc} and the peak coefficient delay l_{cc} by using the cross-correlation operation. Given two back-propagation signals, $p_n^{\Pi_n^i}$ and $p_m^{\Pi_m^i}$ at Π_n^i and Π_m^i, respectively, I perform the cross-correlation operation between two signals. The maximum coefficient becomes the peak coefficient a_{cc} and the time delay from the time origin, 0, to the time realizing the maximum coefficient becomes the peak coefficient delay l_{cc}.

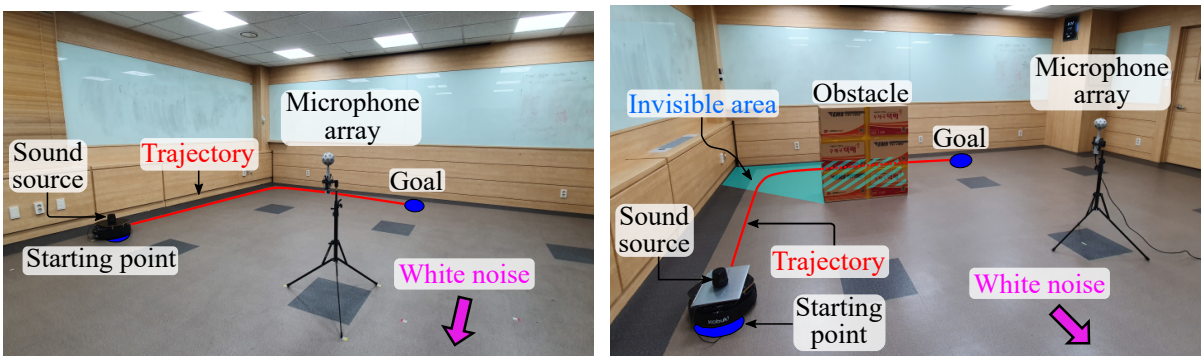
3.3 Result and Discussion

The yellow disk in Figure. 3.1 represents a 95% confidence area for the sound source location estimated by the proposed method. I also compare distance errors of the proposed approach to the prior work (RA-SSL) that does not use the similarity of back-propagation signals, to demonstrate the effectiveness of considering the back-propagation signals.

The hardware platform consists of Eigenmike, the 32-channel microphone array of the mh acoustics, and the i7 CPU computer. As mentioned in Chapter. 3.2.2, I use Velodyne VLP-16 and build a mesh map as the reconstruction of tested indoor environments. The reflection coefficients are appropriately assigned to the triangles by referring to the reported values in [74].

I report values of parameters used for the proposed algorithm: α for controlling the influence of each weight is 0.5, the standard deviation σ_w of the Gaussian distribution function used for computing the distance weight is 0.5 that is determined by the consideration of the size of the indoor environment (about one-tenth of the room width 7m), and the threshold value a_{th} for checking the correlation between back-propagation signals is 0.15.

I use 1024 samples for the separation signal, where the sampling frequency is 12 kHz; 1024 audio samples (85 ms) are a sufficient length for covering direct and first-bounce reflection signals as indicated in [75]. I set the proposed algorithm to estimate the source position every 256 ms in order to respond appropriately to the movement of the source. Specifically, beamforming and generating acoustic rays take 50ms and 0.54ms respectively on average, which are less than the audio length (85ms), and estimating the source position based on the particle filter takes 200 ms on an average that is less than the iteration period 256 ms.



(a) The environment without the obstacle.

(b) The environment with the obstacle.

Figure 3.6: The test environments w/ and w/o an obstacle that can make the sound source non-line-of-sight one. I use the clapping sound in the sound source. I put an additional noise (67 dB and 77 dB white noises) as the distractor in the the back of the test environments.

3.3.1 Benchmarks

Different experiments were conducted in two scenes: the moving sound without and with an obstacle. In both environments (Figure. 3.6(a) and Figure. 3.6(b)), a robot equipped with an omni-directional speaker moved along the red trajectory, and the 32-channel microphone array recorded the audio signals, and these data are used for various tests with the ground truth information on the sound source locations. In Figure. 3.6(b), I put an obstacle made by paper boxes, to cause the robot invisible along the robot's trajectory for the microphone array; at the invisible area, the sound source becomes the non-line-of-sight (NLOS) source.

Handling the NLOS source was reported a quite difficult problem in prior methods [69], because direct sound propagation paths are blocked by the obstacle and I have to rely on indirect sound paths that are incoherent and sensitive to noise. Furthermore, the number of indirect acoustic ray paths passing near the ground truth is usually small, and thus the accuracy of the localization algorithm tends to deteriorate.

Additionally, these scenes are not free from noise (e.g., various noise from outside the room and moving sound of the tested sound source), naturally occurring in a typical environment where the signal-to-noise ratios (SNRs) of both scenes containing the moving sound without and with an obstacle are 20.64 dB and 20.83 dB. To further test the robustness of the proposed method, I expose these scenes additional white noise, whose average sound pressure levels are 67 dB and 77 dB. These noises can cause to trigger many incoherent acoustic ray paths, hindering them to converge in a single location.

3.3.2 A moving sound source

I first show how the proposed approach has the advantage compared to the prior method in a simple scene with a moving sound. In Table 3.1, the accuracy of RA-SSL in the moving source scene gradually deteriorates, as the power of noises increase, where the SNRs containing 67 dB and 77 dB noises are 15.74 dB and 9.35 dB, respectively. On the other hand, the accuracy of the proposed work is rather robust with different power of noise. This shows that the proposed method is robust even in noisy environments, thanks to considering the back-propagation signals on estimated source locations; the similarity weight improves the robustness of the source localization algorithm. To show the positive effect

Table 3.1: The average distance errors w/ different noise levels. Numbers in the parentheses show the improvement.

An moving source w/o an obstacle	w/o white noise	67 dB white noise	77 dB white noise
SNR	20.64 dB	15.74 dB	9.35 dB
The proposed approach	0.57m (7%)	0.58m (18%)	0.56m (38%)
RA-SSL	0.61m	0.69m	0.78m
An moving source w/ an obstacle	w/o white noise	67 dB white noise	77 dB white noise
SNR	20.83 dB	17.33 dB	9.65 dB
The proposed approach	0.51m (64%)	0.54m (75%)	0.53m (100%)
RA-SSL	0.84m	0.95m	1.08m

of back-propagation signals on the 3D sound source localization, I append a description on coherence among back-propagation signals compared to separation signals on the video submission.

Figure. 3.7(a) shows the distance errors of RA-SSL and the proposed approach, where there is 77 dB white noise. The average distance errors are 0.7839 m for RA-SSL and 0.5678 m for the proposed approach; the accuracy of the sound source localization is improved about 38% based on the proposed approach.

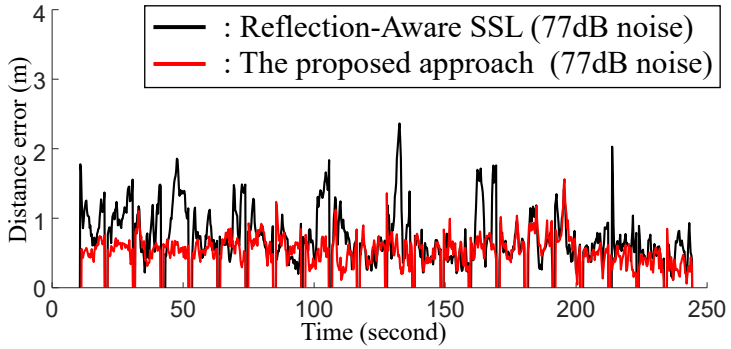
3.3.3 A moving sound around an obstacle

I now show results with the more challenging environment including an obstacle between the source trajectory and the microphone array (Figure. 3.6(b)). Figure. 3.7(b) shows graphs of the distance errors of RA-SSL and the proposed approach with the 77 dB white noise; SNR in this scene is 9.65 dB. The average distance errors of RA-SSL and the proposed approach are 1.083 m and 0.5364 m, respectively. Especially, where the sound source is in the NLOS state from 90 to 180 seconds, the accuracy of RA-SSL decreases drastically, because blocking the direct sound propagation paths makes the convergence of acoustic rays weak near the ground truth. On the other hand, even in this challenging case, I get a stable result, 100% improvement compared to RA-SSL, by considering the similarity between back-propagation signals of indirect acoustic paths.

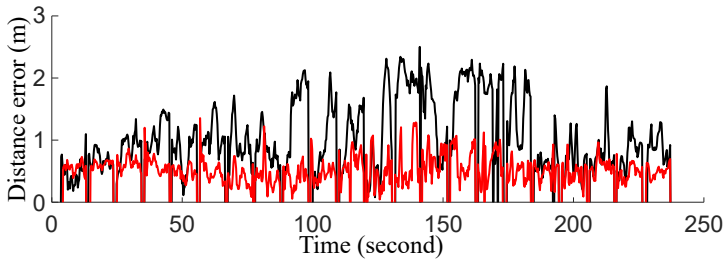
As I have stronger white noise (Table 3.1), SNRs in the moving source scene w/ the obstacle decrease, which are 20.83 dB, 17.33 dB, and 9.65 dB, and the accuracy of RA-SSL then dramatically deteriorates. However, the accuracy of the proposed approach is stable even with different noise energy, demonstrating the robustness and usefulness of the proposed approach.

3.4 Limitations and Future Directions

While I have demonstrated benefits of the proposed approach, it has several limitations and opens up many interesting future directions. When the white noise is larger than 87 dB (sound level like a truck noise [51]), I found that the proposed approach did not work properly because of the relatively weak energy of the sound source (77.34 dB). The acoustic material properties such as reflection coefficients of triangles of objects are not automatically assigned, and recent deep learning approaches showing



(a) Accuracy of moving sound w/o the obstacle containing a 77 dB white noise (Figure. 3.6(a)).



(b) Accuracy of moving sound w/ the obstacle containing a 77 dB white noise (Figure. 3.6(b)).

Figure 3.7: The distance errors between the ground truth and the estimated source positions. In this scene, there is the additional 77 dB white noise, on top of natural occurring noise.

promising results can be employed to solve this problem [74].

Chapter 4. Scalable Microphone Pair Training for Robust Sound source localization with Diverse Array Configuration

4.1 INTRODUCTION

Numerous studies have been dedicated to solving this SSL problem by leveraging various signal processing techniques, including sub-space-based methods such as MUSIC [11, 76], and beamforming methods [9].

In real-world scenarios, robots face complex situations including noise and simultaneous sound events, which traditional signal processing methods struggle with. Considering these challenges, numerous deep learning (DL)-based methods have been presented. He *et al.* [18] introduced a method for the localization of simultaneous speech sources using a multi-layer perceptron. Further, Wang *et al.* [19] enhanced speech localization performance among noisy conditions by creating a trainable mask to reduce noise impact.

Many studies have also been addressed the task of sound event localization and detection (SELD). Adavanne *et al.* [20] suggested a convolutional recurrent neural (CRNN) network to perform the SELD task with simultaneous sources. Schymura *et al.* [77] proposed a self-attention-based network to improve the performance of the SELD task.

However, existing DL-based methods are designed to operate with a specific microphone array type. This design choice results in scalability issues when different microphone array types are involved. Consequently, datasets used for training and testing must be acquired using a consistent microphone array type. Given the diversity of microphone arrays employed in robots, these scalability issues pose a problem: after training with a specific microphone array, the models cannot be utilized effectively with other types of arrays. Additionally, these scalability issues complicate the process of obtaining sufficient datasets for every array type.

Several audio datasets have been released for SSL, such as TUT-CA [20], DCASE2021 [24], and SSLR [18]. However, their combined usage is constrained by the scalability issues associated with the various microphone array types used during their recording. These datasets encompass a wide range of situational audio data. For instance, the TUT-CA dataset contains various sound events recorded in an anechoic chamber, DCASE2021 captures dynamic sources in real, reverberating indoor environments, and SSLR collects human conversations in real environments amid noise interference, such as robot fan noises. Therefore, the potential for improving SSL performance through the combined utilization of these datasets is significant.

Main contribution. To address the scalability issues associated with different types of microphone arrays, I propose a two-stage training methodology. In the first stage, I conduct scalable microphone pair training (Chapter. 4.2). This stage allows the proposed model to access multiple datasets collected by a range of microphone array types for TDoA estimation. As a result, the proposed model gains exposure to the variety of situations present in these datasets.

Additionally, I develop a robust TDoA model, encompassing a Mel scale learnable filter bank (MLFB) (Chapter. 4.2.1) and a hierarchical frequency-to-time attention network (HiFTA-net) (Chapter. 4.2.2). This model is explicitly designed to learn from the variety of situations presented in multiple datasets. Following the scalable microphone pair training stage, the proposed method can be extended to

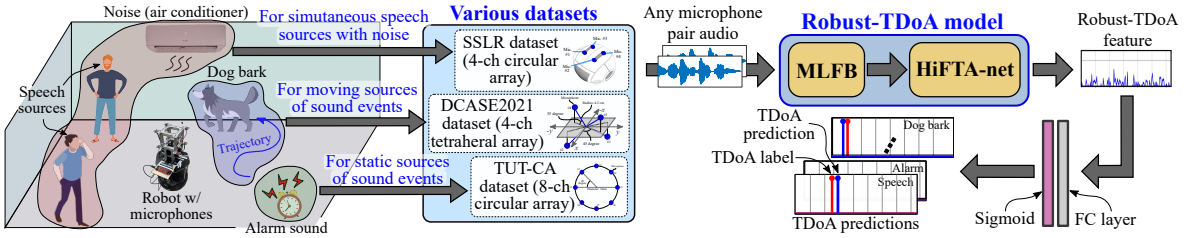


Figure 4.1: The overview of the first scalable microphone pair training stage. The proposed robust-TDoA model is trained by any microphone pair audio from multiple datasets to predict the time difference of arrival (TDoA) of various sound events. Multiple datasets cover different situations like simultaneous speech sources with noise, simultaneous static sources of sound events, or simultaneous moving sources of sound events, e.g., dog bark and alarm. The proposed robust-TDoA model consisting of a Mel scale learnable filter bank (MLFB) and a hierarchical frequency-to-time attention network (HiFTA-net) is designed to effectively learn these different situations. After scalable microphone pair training, the proposed robust-TDoA model can handle these situations in real environments and be applied to the target microphone array in the DoA estimation training stage (Chapter. 4.3).

a variety of microphone arrays for the DoA estimations, i.e., SSL, during the second array geometry-aware training stage (Chapter. 4.3).

Thanks to scalable microphone pair training, the proposed approach demonstrates robust performance across various SSL (i.e., DoA estimation) tasks, such as the localization of simultaneous speech sources (Chapter. 4.4.1, 4.4.2, and 4.4.4) and sound event localization and detection (SELD, Chapter. 4.4.5). Furthermore, through the proposed array geometry-aware training process, the proposed approach is applicable to diverse types of microphone arrays, including the 4-ch circular array in the Pepper robot (Chapter. 4.4.1 and 4.4.2), the 8-ch planar array (Chapter. 4.4.4), and the Respeaker Mic Array v2 from Seeed (Chapter. 4.4.5). I lastly verified that the proposed approach could work in real-time (Chapter. 4.4.6).

4.2 Scalable microphone pair training

I suggest a novel approach of scalable microphone pair training, aiming to enhance robustness under challenging conditions within real-world environments, e.g., containing simultaneous sources of various sound events even with noise. I tackle the scalability issue often encountered in prior deep learning (DL)-based methods and have structured the training process for direction-of-arrival (DoA) estimation into two phases: the scalable microphone pair training phase and the array geometry-aware training phase.

During the first scalable microphone pair training stage, the proposed method focuses on estimating the time difference of arrival (TDoA) of sound events. TDoA is a valuable feature for estimating DoA, given that for two microphone signals; the sound source aligns with a specific TDoA located on a distinct hyperbola, where the locations of both microphones act as foci. In the second phase of array-geometry-aware training, the proposed method, utilizing the previously estimated TDoA information, aims to predict the DoAs of sound events by learning the geometry of the target array.

This two-stage division provides several advantages. It frees the proposed approach from the scal-

ability issue associated with the types of microphone arrays when performing TDoA estimation. TDoA estimation can be seen as a task of computing a time difference between two coherent signals. Consequently, the type of microphone array becomes irrelevant when the proposed method estimates TDoA in the first stage. During the scalable microphone pair training stage aiming to estimate TDoA, the proposed method can access multiple datasets collected by different types of microphone arrays. These datasets, detailed in Chapter. 4.1, represent diverse situations within real-world environments. The method thereby gains exposure to various scenarios. Additionally, having already learned diverse environments in the first stage, the method showcases robust performance with various types of microphone arrays in the second, array-geometry-aware training stage.

The overview of the scalable microphone pair training process is shown in Figure. 4.1. Any microphone pair audio from various datasets is collected. I then train the proposed TDoA estimation model, termed a robust-TDoA model, with this audio, enabling it to predict TDoAs of sound events.

Some TDoA features exist based on generalized cross correlation-phase transform (GCC-PHAT) [5, 19, 78]. Some DL-based methods [18, 19], referred to in Chapter. 4.4.1 for the comparison, utilize these TDoA features as an input to their models. However, existing TDoA features ability to handle challenges like simultaneous sources, noises, and diverse types of sound like sound events needs enhancement. Given the frequency with which robots encounter these challenges in their operational environments, a more robust TDoA feature is required.

I propose a robust-TDoA model, designed to address challenges such as handling simultaneous sound sources, various sound events, and noisy environments by learning from multiple datasets recorded by diverse array types. The model comprises two components: a Mel scale learnable filter bank (MLFB) for generating advantageous audio features and a hierarchical frequency-to-time attention network (HiFTA-net) for estimating TDoAs from these MLFB-generated features.

4.2.1 Mel scale learnable filter bank (MLFB)

In real environments, there exist diverse types of sound like speech, alarm, and dog bark, i.e., different sound events. Moreover, these sound events can occur simultaneously, even with noise like the sound of the air conditioner. The robust-TDoA model needs to be capable of distinguishing sound events while maintaining phase information for estimating TDoAs, even in cases of simultaneous source accompanied by noise. I first generate a useful audio feature for managing sound events in this section, then in Chapter. 4.2.2, I estimate TDoAs while distinguishing between these events.

The Mel-spectrogram [79] serves as a useful hand-crafted feature for capturing different characteristics of various sound events. By converting the frequency scale to the Mel frequency scale, the Mel-spectrogram can show greater discriminative ability for low frequency and be advantageous in differentiating sound events, as many such events like speech, alarms, and dog barks dominate the low-frequency range. Nonetheless, the Mel-spectrogram is not suitable for computing TDoAs because it neglects a phase spectrum of short time Fourier transform (STFT) signals. A phase difference of two microphone signals plays an key role in the TDoA estimation [5, 78], but the Mel-spectrogram is computed only from a magnitude spectrum of STFT signals.

I propose a Mel scale learnable filter bank (MLFB) that generates effective audio features capable of both computing Time-Differences of Arrival (TDoAs) and distinguishing between sound events. Unlike the Mel-spectrogram, the proposed MLFB is designed to take into account the phase difference of two microphone signals, thereby accepting real and imaginary parts of the STFT signals as input. The MLFB

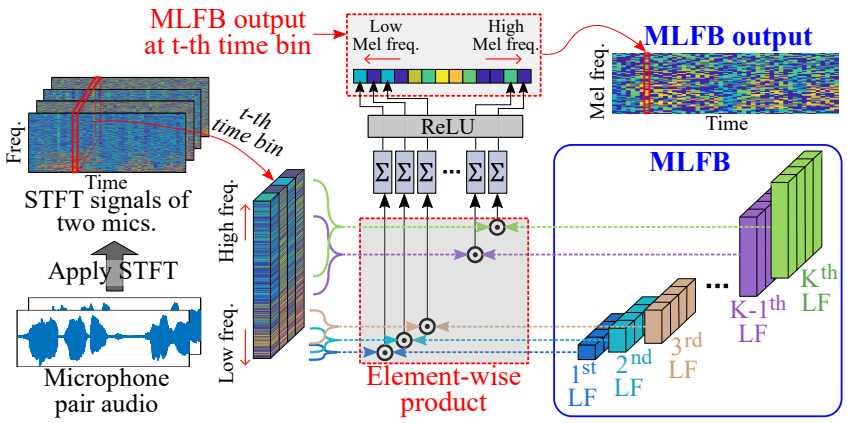


Figure 4.2: An example of applying the Mel scale learnable filter bank (MLFB), consisting of K learnable filters (LFs), to the STFT siganls of two microphones. Each LF consists of 4-ch learnable parameters and has a unique frequency bandwidth. Notably, the frequency bandwidth of each LF becomes more narrow as the frequency decreases. Each LF is tulized on the selectively cropped frequency signal present at the t-th time bin within the STFT signals. As a result, the processed output from the k-th LF subsequently becomes the k-th value of the MLFB output at the respective t-th time bin.

is composed of K learnable filters (LFs), which facilitate the conversion of the frequency scale into the Mel frequency scale.

The process of applying MLFB to a pair of microphone audio inputs is illustrated in Figure. 4.2. The proposed method begins by applying the short-time Fourier transform (STFT) to the microphone pair audio, resulting in STFT signals which include both the real and imaginary parts from the two microphones. The K LFs are then applied to a frequency signal at each time bin of the STFT signals. The process of filtering through MLFB for each time bin of the STFT signals can be expressed as:

$$s_t[k] = \text{ReLU}(\Sigma(x_t \odot LF_k)), \tag{4.1}$$

where $s_t[k]$ represents the k-th value of the MLFB output at the t-th time bin, ReLU is the rectified linear unit, Σ denotes the function summing all elements, $\odot$ is the element-wise product, x_t is the cropped frequency signal at the t-th time bin of the STFT signals, and LF_k is the k-th LF. The proposed method applies Eq. 4.1 across all time bins to produce the comprehensive MLFB output.

To convert the frequency scale to the Mel frequency scale, each LF is designed to possess different frequency bandwidths [80]. LFs for low frequencies have narrower bandwidths compared to those for high frequencies. As a result, when each LF is applied to the cropped frequency signal x_t, which has the same frequency bandwidth, the MLFB output can be more discriminative for low frequencies.

Furthermore, each LF is designed with learnable parameters having 4-channel. Each channel corresponds to real and imaginary components of the STFT signals of two microphones. The learnable parameters of each LF can be trained to consider the phase difference between the two microphones during the scale microphone pair training process. A ReLU function in Eq. 4.1 is employed to facilitate efficient LF training.

Through the implementation of multiple MLFBs, can generate N MLFB outputs, thereby increasing the model's learnable parameters and subsequently its ability to handle diverse situations found in real-world environments.

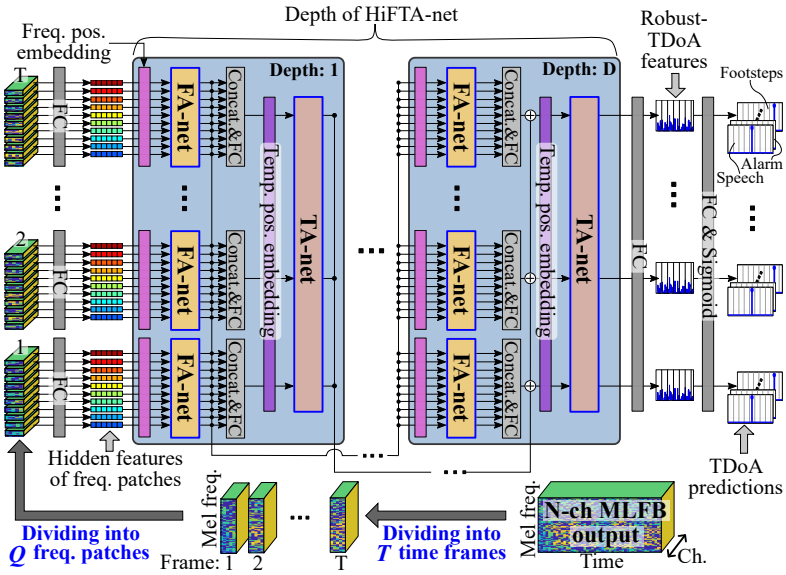


Figure 4.3: An illustration of performing the proposed HiFTA-net. The proposed approach involves the division of the N-channel MLFB output into T time frames. Each time frame is further divided into Q frequency patches. The HiFTA-net is designed to hierarchically comprehend both the freuqnecy and temporal aspects inherent in the input MLFB output, derived from the divided frequency patches. The frequency-attention network (FA-net) initially learns the frequency characteristics within each time frame, followed by the temporal-attention network (TA-net) grasping the temporal properties spanning across all T time frames. Fiannly, from the output of the TA-net, the proposed approach generates robust-TDoA features and calculates the TDoA predictions for the sound events.

4.2.2 Hierarchical frequency-to-time attention network

In this section, I introduce a hierarchical frequency-to-time attention network (HiFTA-net). This model is designed to estimate time differences of arrival (TDoAs) between two microphones while simultaneously distinguishing different sound events from N MLFB outputs, generated as discussed in Chapter. 4.2.1.

Different sound events inherently carry unique frequency and temporal characteristics. In an intuitive sense, each sound event is composed of sequential elements unfolding over time (i.e., temporal characteristics), and these individual components each exhibit distinct frequency characteristics. Consider, for instance, human speech, which comprises a sequence of phonemes, or the periodic pattern of footsteps caused by the alternating contact of two feet with the ground. The temporal aspects, such as the sequential arrangement of phonemes in speech or the rhythm of footfalls, differ greatly. In addition, both phonemes and the sounds generated by individual footsteps exhibit distinct frequency attributes.

In an attempt to estimate TDoAs from a pair of microphones while distinguishing between sound events, the proposed approach strives to comprehend and leverage these temporal and frequency characteristics inherent to sound events. This is particularly relevant in scenarios where multiple sound sources exist simultaneously. The hypothesis is that by considering the unique frequency and temporal properties of each sound source, the proposed method can effectively estimate the multiple TDoAs that originate from these simultaneous sound sources. Furthermore, given this aspect, I anticipate that the proposed approach will display robust performance even in noisy environments.

HiFTA-net operations are demonstrated in Figure. 4.3; I am inspired by the image-based neural architecture [81] when designing the proposed HiFTA-net. The proposed method initially divides the N MLFB outputs into T uniform time frames and seeks to estimate the TDoA of each frame. The HiFTA-net hierarchically learns both frequency and temporal characteristics of sound events. The frequency-attention network (FA-net) initially learns the frequency characteristics of each time frame, followed by the temporal-attention network (TA-network) learning temporal characteristics across the T time frames.

To learn the frequency characteristics of each time frame, each frame of the N-channel MLFB output is divided into Q frequency patches. These patches are then transformed into hidden features using a fully connected layer FC: $FC([p_{(t,1)}, \cdots, p_{(t,Q)}]) \rightarrow [h_{(t,1)}, \cdots, h_{(t,Q)}]$, where $p_{(t,q)}$ is the q-th frequency patch on the t-th time frame and $h_{(t,q)}$ is the q-th hidden feature on the t-th time frame. These hidden features are input into the FA-net, which learns the relationships between hidden features of frequency patches, representing frequency characteristics of various sound events:

$$[h'_{(t,1)}, \cdots, h'_{(t,Q)}] = \text{FA-net}([h_{(t,1)}, \cdots, h_{(t,Q)}] + E_f), \tag{4.2}$$

where $h'_{(t,q)}$ is the q-th FA-net output on the t-th time frame and E_f is the frequency positional embedding (e.g., the sinusoidal signal [82]). FA-net is designed based on the self attention mechanism [81].

The outputs of the FA-net include frequency characteristics of each time frame. The temporal-attention network (TA-net) uses these outputs to learn the temporal characteristics across all time frames. The outputs of the FA-net at the t-th time frames are converted to another hidden feature:

$$H_t = FC(Concat([h'_{(t,1)}, \cdots, h'_{(t,Q)}])), \tag{4.3}$$

where FC is the fully connected layer and $Concat$ is the function concatenating all elements. The proposed approach performs Eq. 4.3 for every T time frame and obtains the T hidden features, $[H_1, \cdots, H_T]$, of all time frames, which is used as the TA-net input:

$$[H'_1, \cdots, H'_T] = \text{TA-net}([H_1, \cdots, H_T] + E_t), \tag{4.4}$$

where H'_t is the output of TA-net for the t-th time frame and E_t is the temporal positional embedding similar to E_f in Eq. 4.2. The TA-net is also based on the self attention mechanism, similar to the FA-net.

In order to accommodate more diverse scenarios in real environments, such as an increase in the number of distinct sound events to be discerned, I expand the trainable parameters of the proposed HiFTA-net. This is achieved by stacking multiple HiFTA-nets, providing the depth D to the network. In this configuration, the output of the FA-net at the t-th time frame ($[h'_{(t,1)}, \cdots, h'_{(t,Q)}]$, in Eq. 4.2) serves as the input for the subsequent FA-net at the same time frame. Furthermore, the outputs of the TA-net ($[H'_1, \cdots, H'_T]$, in Eq. 4.4) are added to the input of the subsequent TA-net, which is the hidden features ($[H_1, \cdots, H_T]$) computed by Eq. 4.3.

The final step involves estimating TDoAs from the outputs of the last TA-net, which has the depth D and ideally should encapsulate the temporal and frequency characteristics of the input microphone pair audio:

$$r\text{-}TDoA_t = \text{FC}(H'_t), \tag{4.5}$$

$$[TDoA^1_t, \cdots, TDoA^S_t] = \sigma(\text{FC}(r\text{-}TDoA_t)), \tag{4.6}$$

where $r\text{-}TDoA_t$ is the proposed robust-TDoA feature at the t-th time frame, $TDoA^s_t$ is the TDoA prediction of the s-th sound event at the t-th time frame, and σ is the sigmoid activation function. I

compute the BCE loss with a TDoA label for training the proposed robust-TDoA model, which consists of the MLFB and HiFTA-net. The TDoA label is computed from the DoA labels provided by datasets, considering the positions of microphone pairs.

4.3 Array geometry-aware training

Thanks to the scalable microphone pair training introduced in Chapter. 4.2, the proposed robust-TDoA model is capable of estimating the time difference of arrival (TDoA) under demanding situations, such as real-world environments featuring simultaneous, noise-infused sources of various sound events. In this section, the goal is to utilize the estimated TDoA information to localize sound sources from various types of microphone arrays.

TDoA offer valuable hints for sound source localization (SSL); a sound source that satisfies a specific TDoA will be located along a hyperbola, with the microphone pair serving as the foci. By utilizing TDoAs from all pairs within a target microphone array and considering the positions of the microphones (i.e., geometry information) within that array, I can localize the sound source, i.e., estimate the directions-of-arrival (DoA). The target microphone array can be various types of microphone array satisfying a condition that every DoA should correspond to the one specific combination of TDoAs [83]. For the 2-D DoA estimation, for instance, there must be at least three microphones on a plane .

I introduce an array geometry-aware training procedure, as depicted in Figure. 4.4. During that procedure, the proposed approach is trained to consider geometry information of the target microphone array to estimate DoA from TDoA information. The initial step involves extracting audio from each pair of microphones within the target microphone array. Subsequently, I employ the robust-TDoA model (trained as described in Chapter. 4.2) to calculate the robust-TDoA feature for each microphone pair, represented as $[r\text{-}TDoA_t^1, \cdots, r\text{-}TDoA_t^P]$, where $r\text{-}TDoA_t^p$ signifies the robust-TDoA feature at the t-th time frame for the p-th microphone pair. Thanks to the exquisite design of the proposed TDoA feature that can learn from various datasets with different goemetric array, this leads us to anticipate that the proposed method can estimate DoAs in diverse situations, including tasks such as localizing simultaneous speech sources (Chapter. 4.4.1 and 4.4.4) and the sound event localization and detection (SELD, Chapter. 4.4.5).

For the t-th time frame, a multi-layer perceptron (MLP) is trained to determine the DoA, denoted as DoA_t, from these robust-TDoA features of all microphone pairs:

$$DoA_t = \Phi(\text{MLP}(Concat([r\text{-}TDoA_t^1, \cdots, r\text{-}TDoA_t^P]))), \tag{4.7}$$

where Φ is the activation function, MLP consists of four fully connected layers, and $Concat$ is a function concatenating all elements. My strategy requires the consideration of microphone positions (i.e., geometric information) within the target microphone array to estimate DoAs from TDoA information. MLP is trained to account for microphone positions during the array geometry-aware training process.

By modifying MLP and the activation function Φ, I can reformat DoA_t to match the desired DoA form; examples include DoA vectors (x, y, z) of various sound events in Chapter. 4.4.5 and DoA angles (360 cells corresponding 360 degrees) of speech in Chapter. 4.4.1 and 4.4.4. For these examples, I respectively use mean squared error (MSE) and binary cross-entropy (BCE) during the array geometry-aware training process.

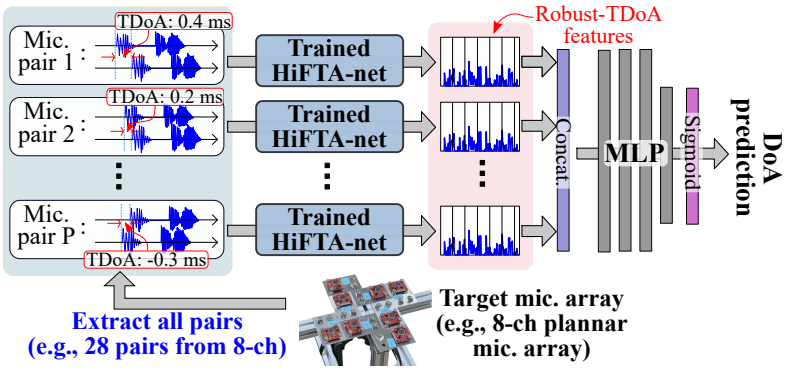


Figure 4.4: An illustration of the process for executing the second array geometry-aware training stage for direction-of-arrival (DoA) estimation. The proposed method initially extracts all microphone pairs from the target microphone array, then utilize the robust-TDoA model consisting of the MLFB and HiFTA-net. The robust-TDoA model is trained through the proposed scalable microphone pair training process in Chapter. 4.2, to compute the robust-TDoA features and utilize the same parameters across all pairs The robust-TDoA feature encompass TDoA information for all microphone pairs. Subsequently, a multi-layer perceptron (MLP) is trained to predict DoAs from the robust-TDoA features, by considering geometry information of the target microphone array.

4.4 Result and discussion

In this section, I evaluate the proposed approach, demonstrating its benefits. Firstly, the proposed robust-TDoA model, which was trained using three datasets, i.e., SSLR, TUT-CA, and DCASE2021, mentioned in Chapter. 4.1, during the scalable microphone pair training stage as detailed in Chapter.4.2, is capable of handling a variety of real-world scenarios. This includes challenging situations such as instances with simultaneous sources amidst noise. Consequently, the proposed approach, which operates based on this robust-TDoA model, delivers substantial performance in various DoA estimation tasks. These tasks include the localization of simultaneous speech sources (Chapter. 4.4.1, 4.4.2, and 4.4.4), and sound event localization and detection (Chapter.4.4.5), which are referred to as speech-SSL and SELD in this section, respectively. I compare the proposed approach with previous methods for speech-SSL [18,19] and SELD [20].

Secondly, following the scalable microphone pair training stage, the proposed approach can adapt to various types of microphone arrays through array geometry-aware training. This is a crucial capability considering the diversity of arrays found in different robots. I have demonstrated the proposed approach by applying three distinct target microphone arrays: the 4-ch circular array in the Pepper robot (Chapter. 4.4.1 and 4.4.2), the 8-ch planar array (Chapter. 4.4.4), and the Respeaker Mic Array v2 from Seeed (Chapter. 4.4.5).

Lastly, I verify that the proposed approach can works in real-time in Chapter. 4.4.6; a demonstration video of the proposed approach are shown in the multimedia attachment.

4.4.1 Speech-SSL with existing dataset

Suppose the target microphone array for the speech-SSL task is the 4-ch circular microphone array within the Pepper robot. The proposed approach then performs array geometry-aware training over 10 epochs using the SSLR dataset [18], collected using the same 4-ch circular microphone array. I compare

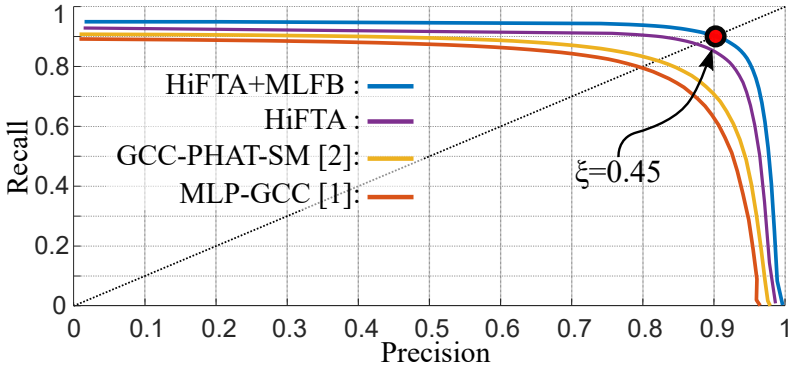


Figure 4.5: The graph of *precision* vs. *recall* curves of different methods by varying the prediction threshold ξ, in the case of an unknown number of sources.

Table 4.1: The accuracy of speech-SSL with existing dataset.

	Overall		*Source=1*		*Source=2*	
	MAE (↓)	ACC (↑)	MAE (↓)	ACC (↑)	MAE (↓)	ACC (↑)
MLP-GCC [18]	4.61°	91.91%	3.80°	95.03%	9.54°	72.74%
GCC-PHAT -SM [19]	3.90°	92.42%	3.13°	95.50%	8.39°	74.02%
HiFTA (ours)	3.20°	93.95%	3.11°	95.19%	3.42°	90.64%
HiFTA+ MLFB (ours)	**2.84°**	**94.94%**	**2.81°**	**95.91%**	**2.92°**	**92.34%**

the proposed approach with previous speech-SSL methods, such as *MLP-GCC* [18] and *GCC-PHAT-SM* [19]. They use the existing TDoA feature based on the generalized cross correlation-phase transform (GCC-PHAT) as input and are also trained with the SSLR datasets. Other datasets like DCASE2021 and TUT-CA cannot be used in conjunction because they were recorded using different microphone array types.

Additionally, I conduct an ablation study to highlight the advantages of the proposed components. *HiFTA+MLFB* incorporates all components. *HiFTA* applies solely the HiFTA-net without the MLFB.

I apply evaluation metrics previously suggested in the work [18] for the speech-SSL, covering two scenarios where the number of sources are known or unknown. The results of the first condition, i.e., the results with the known number of sources, are shown in Table. 4.1. The mean absolute error (MAE) is the average azimuth estimation error, and the accuracy (ACC) denotes the percentage of correct azimuth estimates. I measure the results of both metrics under three conditions based on the number of overlapping sources: *Source=1* (a single source), *Source=2* (two simultaneous sources), and *Overall* (averaging *Source=1* and *Source=2*)

Both versions of the proposed method, namely *HiFTA* and *HiFTA+MLFB*, demonstrate improved accuracy in all situations across both metrics compared to prior works like *MLP-GCC* and *GCC-PHAT-SM*. It means that the proposed robust-TDoA model effectively learns to manage challenging scenarios, such as handling simultaneous sources with noise, using multiple datasets during the scalable microphone

Table 4.2: The accuracy by increasing the size of the training dataset.

Training datasets for the robust -TDoA model	*Overall* MAE (↓)	ACC (↑)	$\xi = 0.45$ Precision (↑)	Recall (↑)
SSLR	3.53°	91.27%	85.51%	86.45%
SSLR, DCASE2021	2.97°	93.63%	87.84%	88.94%
SSLR, DCASE2021 TUT-CA	**2.84°**	**94.94%**	**90.38%**	**89.95%**

pair training process. Furthermore, the proposed components, MLFB and HiFTA-net, appear to be well-suited to learning in these complex scenarios as evidenced by the superior performance of the *HiFTA+MLFB* version, which utilizes all components.

This observation is evident in Table. 4.1. Particularly, in the complex scenario *Source=2* containing simultaneous sources, the performance of the prior works regarding MAE and ACC significantly deteriorated compared to the simpler scenario *Source=1*. For instance, the ACCs of both prior works fell by 22.29 % and 21.48 %, respectively. However, the proposed approach, *HiFTA+MLFB*, showed a marginal decrease in accuracy, with the ACC fall of only 3.57 %.

The results of the second type of metrics, i.e., the results with an unknown number of sources, are shown in Figure. 4.5. The *precision* vs. *recall* curve is proposed by varying the prediction threshold ξ [18] to verify the ability of detection as well as localization; the curve showing better results is closer to 1 value in both axes corresponding to *precision* and *recall*. The proposed approach, *HiFTA+MLFB*, shows the best performance among the reported results.

4.4.2 The ablation study with varying sizes of datasets

The proposed approach is evaluated by expanding the training datasets during the scalable microphone pair training process. This experiment aims to ascertain the impact of learning from an extensive training dataset. I assess the proposed method using various dataset sizes, incrementally adding the DCASE2021 and TUT-CA datasets to the SSLR dataset. This results in three versions: SSLR, (SSLR + DCASE2021), and (SSLR + DCASE2021 + TUT-CA).

I utilize the MAE and ACC metrics of the *Overall* case, as well as the *precision* and *recall* values discussed in Chapter. 4.4.1. The *precision* and *recall* values have a trade-off relation according to the prediction threshold ξ. For the proposed approach to work on the fly in a real-time, ξ needs to be chosen with a fixed value. I choose the proper value of ξ, which makes both precision and recall have large values. In Figure. 4.5, I assume that both *precision* and *recall* can be large at the intersection point between the *precision* vs. *recall* curve and the symmetric line, i.e., the dotted black line; thus, I utilize $\xi = 0.45$, i.e., the red point.

The results are shown in Table. 4.2. The performance of the proposed method improves across all metrics when the size of the training dataset is increased. These findings suggest that a large dataset is advantageous in the scalable microphone pair training stage, implying that the proposed robust-TDoA model, comprised of MLFB and the HiFTA-net, has sufficient capacity to learn from multiple datasets. Moreover, even in the case of just utilizing the SSLR dataset during the scalable microphone pair training procedure, the speech-SSL performance is better than the prior works, i.e., *MLP-GCC* and *GCC-PHAT-SM*, in Table. 4.1.

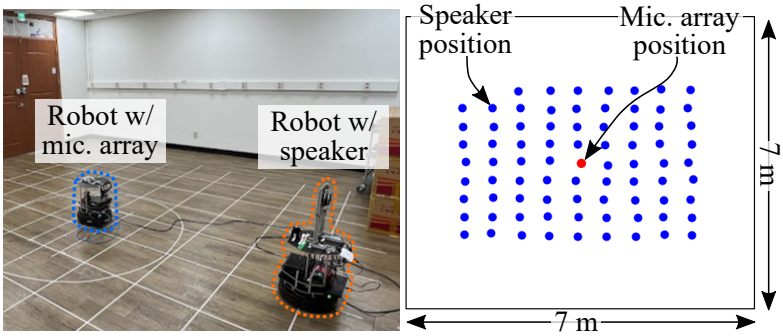


Figure 4.6: The experiment for recording real RIRs in the left figure and the positions of robots equipped with the speaker (blue dots) and the microphone array (a red dot), respectively, in the right figure. The positions of robots are obtained using a SLAM technique, i.e., Cartographer [2], with 2D LiDAR and IMU sensors.

4.4.3 Synthetic datasets for training and evaluating processes.

The proposed method is designed to operate with diverse types of microphone arrays mounted on various robots, made possible through the proposed array geometry-aware training stage. Notably, when the dataset of the target microphone array for array geometry-aware training is not available, the proposed method can effectively leverage a synthetic dataset generated by sound simulators. This involves initially generating room impulse responses (RIRs) using a sound simulator, then creating a synthetic dataset by convoluting RIRs with dry signals. I exclusively utilize dry speech signals for the speech-SSL task and all dry signals for the SELD task.

To generate synthetic datasets with two target microphone arrays, i.e., the ReSpeaker Mic Array v2 and 8-ch planar microphone array, I employ the Habitat 2.0 sound simulator [84] and a NIGENS general sound events database [85], which contains dry signals of fourteen sound events. The sound simulator can operate in a 3-D reconstructed environment using an iPhone equipped with a camera and LiDAR sensors, replicating a real environment of 7m width, 7m depth, and 3m height. The synthetic dataset includes 5 hours of multi-channel audio with up to two simultaneous sources.

In addition, I record real RIRs with robots in an actual environment to verify the proposed approach following the array geometry-aware training process using synthetic datasets from the sound simulator. I equip two mobile robots with each microphone array and a speaker, then record RIRs at 78 locations using sine sweeps [86], as illustrated in Figure. 4.6. I generate a 2-hour evaluation dataset using real RIRs, thereby making the evaluation dataset more realistic than the synthetic dataset from the sound simulator.

To mimic the noisy real-world conditions, I incorporate additional white noise; both the synthetic dataset, created using a sound simulator, and the evaluation dataset, utilizing real RIRs, have average signal-to-noise ratios (SNR) of 10 dB and 18 dB, respectively. Furthermore, the reverberation time (RT60s), which signifies the reverberation factor in the proposed experimental environments, is 0.45 seconds for the synthetic dataset and 0.4 seconds for the evaluation dataset.

4.4.4 Speech-SSL with synthetic dataset

When the target microphone array for the speech-SSL task is the 8-ch planar microphone array, no published datasets exist that were recorded using this array. For the array geometry-aware training

Table 4.3: The accuracy of speech-SSL with synthetic dataset of a 8-ch planar array using the sound simulator.

	Overall		*Source=1*		*Source=2*	
	MAE (↓)	ACC (↑)	MAE (↓)	ACC (↑)	MAE (↓)	ACC (↑)
MLP-GCC [18]	18.49°	63.08%	17.87°	64.95%	18.80°	62.14%
GCC-PHAT-SM [19]	15.19°	62.72%	14.57°	65.70%	15.49°	61.22%
MUSIC [76]	11.68°	78.02%	6.74°	79.61%	13.99°	77.27%
Ours	**6.23°**	**83.43%**	**5.23°**	**86.44%**	**6.74°**	**81.91%**

Table 4.4: The accuracy of SELD with synthetic dataset of ReSpeaker v2 using the sound simulator.

	SED metrics		*DoA metrics*		*Overall*
	ER (↓)	F-score (↑)	DOA error (↓)	Frame recall (↑)	SELD score (↓)
SELDnet [20]	0.67	47.92	28.77	66.21	0.42
Ours	**0.45**	**69.79**	**11.90**	**73.43**	**0.27**

process, I utilize the synthetic and evaluation datasets, generated by that 8-ch planar microphone array (Chapter. 4.4.3). Prior DL-based methods, *MLP-GCC* [18] and *GCC-PHAT-SM* [19], are modified to access the 8-ch audio and re-trained using that synthetic dataset, given the disparity in the microphone array type utilized in the earlier studies in Chapter. 4.4.1 and the target microphone array.

Additionally, I contract the proposed approach with MUSIC [76], the sub-space-based method using signal processing techniques. As MUSIC can function without a training process, it can easily be used when a training dataset is unavailable. To apply MUSIC, I must be aware of the number of active sources at each frame; I run MUSIC with the actual number of active sources, gleaned from the DoA labels. Therefore, the results of MUSIC in this paper can be seen as the optimal performance achievable when utilizing MUSIC. I utilize the evaluation metrics, i.e., MAE and ACC in the three instances of *Source=1*, *Source=2*, and *Overall*, as utilized in Chapter. 4.4.1.

The evaluation results are depicted in Table. 4.3. Even though the sound simulator can generate a realistic synthetic dataset, it does not fully mimic data recorded in a real environment. Because of the scalability issues associated to microphone array types, prior DL-based methods, *MLP-GCC* and *GCC-PHAT-SM*, can only use the synthetic dataset for model training, leading to suboptimal performances. Furthermore, the results of these methods are inferior to the signal processing-based method, *MUSIC*. However, the proposed method, being capable of learning realistic scenarios from multiple real datasets during the scalable microphone pair training, exhibits superior and more reasonable performance compared to other methods.

4.4.5 SELD with synthetic dataset

When the target microphone array is the 4-ch Respeaker Mic Array v2, for which no published dataset exists, I use synthetic and evaluation datasets (Chapter.4.4.3) for array geometry-aware training and performance evaluation. I also extend the proposed approach to the Sound Event Localization and

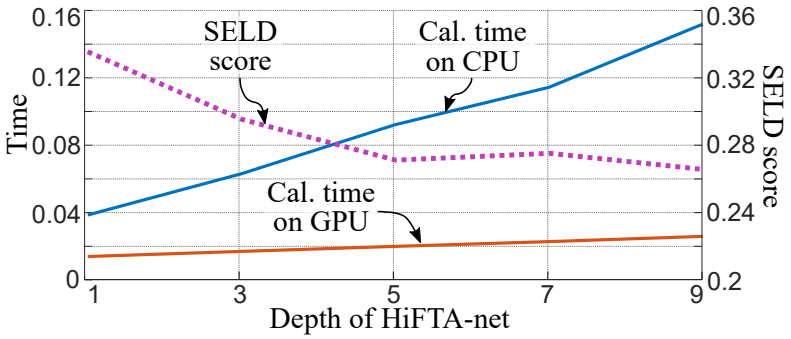


Figure 4.7: The calculation times on CPU and GPU and the SELD scores of the SELD task by increasing the depth of the proposed HiFTA-net.

Detection (SELD) task, which requires differentiating sound events and is a different DoA estimation task than the experiments in Chapter.4.4.1, 4.4.2, and 4.4.4.

This task demands the differentiation of sound events, making traditional signal processing-based methods like MUSIC [76] unsuitable. I compare the proposed approach with a prior SELD method [20], which uses the GCC-PHAT feature and the mel spectrogram within the CRNN model. This method is retrained using the synthetic dataset from the 4-ch Respeaker Mic Array v2 due to the discrepancy in the microphone array types used in its study [20] and the target microphone array. Scalability issues associated with the microphone array types prevent this method from utilizing other datasets, such as SSLR, DCASE2021, and TUT-CA, simultaneously.

For evaluation, I use the metrics proposed in previous SELD work [20], which comprise of four metrics: *ER*, *F-score*, *DoA error*, *Frame recall*, and *SELD* score. The *ER* and *F-score* are used to evaluate sound event detection (SED), whereas *DoA error* and *Frame recall* measure the performance of DoA estimation. The *SELD* score represents the overall SELD performance by averaging the other four metrics. Detailed information about these metrics is available in [20].

The proposed approach outperforms the prior work in all metrics, suggesting that it can be successfully applied to different SSL tasks with high performance, attributed to the proposed scalable microphone pair training process from multiple datasets.

4.4.6 Real-time computation verification

So far, the results and analysis confirm the applicability of the proposed approach to various microphone array types and diverse tasks. In this section, I aim to validate the feasibility of the proposed approach for real-time application with actual robots. The key factor influencing computation time is the depth D of the proposed HiFTA-net, as detailed in Chapter. 4.2.2. I tested the proposed work in the SELD task with Respeaker Mic Array v2 in Chapter. 4.4.5 by incrementally increasing the depth D from 1 to 9.

The proposed approach is computed in a laptop computer with Intel i7-11800H CPU and NVIDIA GeForce RTX 3070 Laptop GPU. However, even at maximum depth of 9, the computation time is approximately 0.15 seconds, validating that the proposed approach can operate in real-time. Furthermore, I observe that the SELD score reaches saturation beyond a depth of 5. Therefore, in all experiments, I set the proposed model to have a depth of 5. I demonstrate the real-world applicability of the proposed method with real robots by implementing it in an SSL application, for example, directing the robot to

face the speech sources.

4.5 Conclusion

I have demonstrated that the proposed approach is capable of robust DoA estimation across two SSL tasks and can be applied to various microphone array types, due to the proposed scalable microphone pair training. I have also confirmed that the proposed method can operate effectively with real robots in real-time settings. I anticipate that the proposed method holds potential for dealing with indirect sounds, such as reflections and diffractions, by integrating it with ray tracing-based SSL methods [87]. Furthermore, I believe that the proposed method could be adapted for open-world problems by incorporating self-supervised or online learning techniques [88].

Bibliography

[1] J-M Valin, François Michaud, and Jean Rouat, "Robust 3d localization and tracking of sound sources using beamforming and particle filtering", in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. IEEE, 2006, vol. 4, pp. IV–IV.

[2] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor, "Real-time loop closure in 2d lidar slam", in *ICRA*, 2016.

[3] Craig C Douglas and Robert A Lodder, "Human identification and localization by robots in collaborative environments", *Procedia Comput. Sci.*, vol. 108, pp. 1602–1611, 2017.

[4] Muhammad Imran, Akhtar Hussain, Nasir M Qazi, and Muhammad Sadiq, "A methodology for sound source localization and tracking: Development of 3d microphone array for near-field and far-field applications", in *IBCAST*, 2016, pp. 586–591.

[5] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay", *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 4, pp. 320–327.

[6] François Grondin and François Michaud, "Time difference of arrival estimation based on binary frequency mask for sound source localization on mobile robots", in *IROS*, 2015.

[7] François Grondin and James Glass, "Fast and robust 3-d sound source localization with dsvd-phat", in *IROS*, 2019.

[8] J.-M. Valin, F. Michaud, and J. Rouat, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering", *Robot. Auton. Syst.*, vol. 55, no. 3.

[9] Cha Zhang, Dinei Florêncio, Demba E Ba, and Zhengyou Zhang, "Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings", *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 538–548, 2008.

[10] S. Argentieri and P. Danes, "Broadband variations of the music high-resolution method for sound source localization in robotics", in *IROS*, 2007.

[11] K. Nakamura, K. Nakadai, F. Asano, Y. Hasegawa, and H. Tsujino, "Intelligent sound source localization for dynamic environments", in *IROS*, 2009.

[12] Keisuke Nakamura, Kazuhiro Nakadai, and Gökhan Ince, "Real-time super-resolution sound source localization for robots", in *IROS*, 2012.

[13] Y. Sasaki, R. Tanabe, and H. Takemura, "Probabilistic 3d sound source mapping using moving microphone array", in *IROS*, 2016.

[14] D. Su, T. Vidal-Calleja, and J. V. Miro, "Towards real-time 3d sound sources mapping with linear microphone arrays", in *ICRA*, 2017.

[15] Pragyan Mohapatra Prasant Misra, A. Anil Kumar and Balamuralidhar P., "Droneears: Robust acoustic sound localization with aerial drones", in *ICRA*, 2018.

[16] Quan V Nguyen, Francis Colas, Emmanuel Vincent, and François Charpillet, "Localizing an intermittent and moving sound source using a mobile robot", in *IROS*, 2016.

[17] Alban Portello, Gabriel Bustamante, Patrick Danès, Jonathan Piat, and Jérôme Manhes, "Active localization of an intermittent sound source from a moving binaural sensor", in *European Acoustics Association Forum Acusticum*, 2014, p. 12p.

[18] Weipeng He, Petr Motlicek, and Jean-Marc Odobez, "Deep neural networks for multiple speaker detection and localization", in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 74–79.

[19] Jiadong Wang, Xinyuan Qian, Zihan Pan, Malu Zhang, and Haizhou Li, "Gcc-phat with speech-oriented attention for robotic sound source localization", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5876–5883.

[20] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2018.

[21] Yin Cao, Qiuqiang Kong, Turab Iqbal, Fengyan An, Wenwu Wang, and Mark D Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy", *arXiv preprint arXiv:1905.00268*, 2019.

[22] Huy Phan, Lam Pham, Philipp Koch, Ngoc QK Duong, Ian McLoughlin, and Alfred Mertins, "On multitask loss function for audio event detection and localization", *arXiv preprint arXiv:2009.05527*, 2020.

[23] Qing Wang, Huaxin Wu, Zijun Jing, Feng Ma, Yi Fang, Yuxuan Wang, Tairan Chen, Jia Pan, Jun Du, and Chin-Hui Lee, "The ustc-iflytek system for sound event localization and detection of dcase2020 challenge", *IEEE AASP Chall. Detect. Classif. Acoust. Scenes Events*, 2020.

[24] Archontis Politis, Sharath Adavanne, Daniel Krause, Antoine Deleforge, Prerak Srivastava, and Tuomas Virtanen, "A dataset of dynamic reverberant sound scenes with directional interferers for sound event localization and detection", *arXiv preprint arXiv:2106.06999*, 2021.

[25] Jani Even, Jonas Furrer, Yoichi Morales, Carlos Toshinori Ishi, and Norihiro Hagita, "Probabilistic 3-d mapping of sound-emitting structures based on acoustic ray casting", *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 333–345, 2016.

[26] Nagasrikanth Kallakuri, Jani Even, Yoichi Morales, Carlos Ishi, and Norihiro Hagita, "Using sound reflections to detect moving entities out of the field of view", in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5201–5206.

[27] Jani Even, Yoichi Morales, Nagasrikanth Kallakuri, Carlos Ishi, and Norihiro Hagita, "Audio ray tracing for position estimation of entities in blind regions", in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1920–1925.

[28] H. Kuttruff, *Acoustics: an introduction*, CRC Press, 2007.

[29] Michael Vorländer, "Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm", *J. Acoust. Soc. Am.*, vol. 86, no. 1, pp. 172–178, 1989.

[30] Carl Schissler, Ravish Mehra, and Dinesh Manocha, "High-order diffraction and diffuse reflections for interactive sound propagation in large environments", *ACM Trans. Graph.*, vol. 33, no. 4, pp. 39, 2014.

[31] Hengchin Yeh, Ravish Mehra, Zhimin Ren, Lakulish Antani, Dinesh Manocha, and Ming Lin, "Wave-ray coupling for interactive sound propagation in large complex scenes", *ACM Trans. Graph.*, vol. 32, no. 6, pp. 165, 2013.

[32] Joseph B Keller, "Geometrical theory of diffraction", *JOSA*, vol. 52, no. 2, pp. 116–130, 1962.

[33] Diego Di Carlo, Antoine Deleforge, and Nancy Bertin, "Mirage: 2d source localization using microphone pair augmentation with echoes", in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 775–779.

[34] J-M Valin, François Michaud, Jean Rouat, and Dominic Létourneau, "Robust sound source localization using a microphone array on a mobile robot", in *IROS*, 2003.

[35] B Teng and R Eatock Taylor, "New higher-order boundary element methods for wave diffraction/radiation", *Appl. Ocean Res.*, vol. 17, no. 2, pp. 71–77, 1995.

[36] Sara R Martin, U Peter Svensson, Jan Slechta, and Julius O Smith, "A hybrid method combining the edge source integral equation and the boundary element method for scattering problems", in *Proc. of Meet. on Acoust.* ASA, 2016, vol. 26.

[37] Atul Rungta, Carl Schissler, Nicholas Rewkowski, Ravish Mehra, and Dinesh Manocha, "Diffraction kernels for interactive sound propagation in dynamic environments", *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 4, pp. 1613–1622, 2018.

[38] Nicolas Tsingos and Jean-Dominique Gascuel, "Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments", in *AES Convention*. AES, 1998.

[39] U Peter Svensson, Roger I Fred, and John Vanderkooy, "An analytic secondary source model of edge diffraction impulse responses", *J. Acoust. Soc. Am.*, vol. 106, no. 5, pp. 2331–2344, 1999.

[40] Andreas Asheim and U Peter Svensson, "An integral equation formulation for the diffraction from convex plates and polyhedra", *J. Acoust. Soc. Am.*, vol. 133, no. 6, pp. 3681–3691, 2013.

[41] Robert G Kouyoumjian and Prabhakar H Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface", *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 62, no. 11, pp. 1448–1461, 1974.

[42] Lakulish Antani, Anish Chandak, Micah Taylor, and Dinesh Manocha, "Efficient finite-edge diffraction using conservative from-region visibility", *Appl. Acoust.*, vol. 73, no. 3, pp. 218–233, 2012.

[43] Nicolas Tsingos, Thomas Funkhouser, Addy Ngan, and Ingrid Carlbom, "Modeling acoustics in virtual environments using the uniform theory of diffraction", in *SIGGRAPH*. ACM, 2001.

[44] Micah Taylor, Anish Chandak, Zhimin Ren, Christian Lauterbach, and Dinesh Manocha, "Fast edge-diffraction for sound propagation in complex virtual environments", in *EAA auralization symposium*, 2009, pp. 15–17.

[45] Micah Taylor, Anish Chandak, Qi Mo, Christian Lauterbach, Carl Schissler, and Dinesh Manocha, "Guided multiview ray tracing for fast auralization", *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 11, pp. 1797–1810, 2012.

[46] Michael Kazhdan and Hugues Hoppe, "Screened poisson surface reconstruction", *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29, 2013.

[47] Marco Tarini, Nico Pietroni, Paolo Cignoni, Daniele Panozzo, and Enrico Puppo, "Practical quad mesh simplification", in *Comput. Graph. Forum*. Wiley Online Library, 2010, vol. 29, pp. 407–418.

[48] Ruwen Schnabel, Roland Wahl, and Reinhard Klein, "Efficient ransac for point-cloud shape detection", in *Comput. Graph. Forum*. Wiley Online Library, 2007, vol. 26, pp. 214–226.

[49] Marco Binelli, Andrea Venturi, Alberto Amendola, and Angelo Farina, "Experimental analysis of spatial properties of the sound field inside a car employing a spherical microphone array", in *AES Convention*. AES, 2011.

[50] Haohai Sun, Edwin Mabande, Konrad Kowalczyk, and Walter Kellermann, "Joint doa and tdoa estimation for 3d localization of reflective surfaces using eigenbeam mvdr and spherical microphone arrays", in *ICASSP*. IEEE, 2011.

[51] Yang-Hann Kim and Jung-Woo Choi, *Sound visualization and manipulation*, John Wiley & Sons, 2013.

[52] Francis X Giraldo, "Lagrange–galerkin methods on spherical geodesic grids", *J. Comput. Phys.*, vol. 136, no. 1, pp. 197–213, 1997.

[53] Finn Jacobsen, Torben Poulsen, Jens Holger Rindel, Anders Christian Gade, and Mogens Ohlrich, "Fundamentals of acoustics and noise control", *Dep. of Electr. Eng., Tech. Univ. of Denmark*, 2011.

[54] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, MIT press, 2005.

[55] T. W. Anderson, Ed., *An Introduction to Multivariate Statistical Analysis*, Wiley, 1984.

[56] S. Briere, J.-M. Valin, F. Michaud, and D. Létourneau, "Embedded auditory system for small mobile robots", in *ICRA*, 2008.

[57] F. Grondin, D. Létourneau, F. Ferland, V. Rousseau, and F. Michaud, "The manyears open framework", *Auton. Robots*, vol. 34, no. 3.

[58] Melonee Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich, "Fetch and freight: Standard platforms for service robot applications", in *Workshop on Auton. Mob. Serv. Robot.*, 2016.

[59] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool", in *Eurographics Italian Chap. Conf.* 2008, The Eurographics Association.

[60] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz, "Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments", in *IROS*, 2009.

[61] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz, "Towards 3d point cloud based object maps for household environments", *Rob. Auton. Syst.*, vol. 56, no. 11, pp. 927–941, 2008.

[62] Heinrich Kuttruff, *Room acoustics*, Crc Press, 2016.

[63] Hans-Elias de Bree, Martin Nosko, and Emiel Tijs, "A handheld device to measure the acoustic absorption in situ", *SNVH, GRAZ*, 2008.

[64] Reinhilde Lanoye, Hans-Elias de Bree, Walter Lauriks, and Gerrit Vermeir, "a practical device to determine the reflection coefficient of acoustic materials in-situ based on a microflown and microphone sensor", in *ISMA*. Citeseer, 2004, vol. 1, p. 3.

[65] Boaz Rafaely, *Fundamentals of spherical array processing*, vol. 8, Springer, 2015.

[66] Daniel P Jarrett, Emanuël AP Habets, and Patrick A Naylor, "Spherical harmonic domain noise reduction using an mvdr beamformer and doa-based second-order statistics estimation", in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 654–658.

[67] Shefeng Yan, Haohai Sun, U Peter Svensson, Xiaochuan Ma, and Jens M Hovem, "Optimal modal beamforming for spherical microphone arrays", *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 2, pp. 361–371, 2011.

[68] Xuan Li, Shefeng Yan, Xiaochuan Ma, and Chaohuan Hou, "Spherical harmonics music versus conventional music", *Appl. Acoust.*, vol. 72, no. 9, pp. 646–652, 2011.

[69] I. An, M. Son, D. Manocha, and S. Yoon, "Reflection-aware sound source localization", in *ICRA*, 2018.

[70] F. Gyorgy, "Rendering and managing spherical data with sphere quadtrees", in *Proceedings of the First IEEE Conference on Visualization: Visualization '90*, Oct 1990, pp. 176–186.

[71] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe, "Poisson surface reconstruction", in *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006, vol. 7.

[72] Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou, "Interactive sound propagation with bidirectional path tracing", *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 180, 2016.

[73] Dingzeyu Li, Timothy R Langlois, and Changxi Zheng, "Scene-aware audio for 360 videos", *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 111, 2018.

[74] Carl Schissler, Christian Loftin, and Dinesh Manocha, "Acoustic classification and optimization for multimodal rendering of real-world scenes", *IEEE transactions on visualization and computer graphics*, vol. 24, no. 3, pp. 1246–1259, 2018.

[75] Jingdong Chen and Jacob Benesty, "A time-domain widely linear mvdr filter for binaural noise reduction", in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*. IEEE, 2011, pp. 105–108.

[76] R. Schmidt, "Multiple emitter location and signal parameter estimation", *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280.

[77] Christopher Schymura, Benedikt Bönninghoff, Tsubasa Ochiai, Marc Delcroix, Keisuke Kinoshita, Tomohiro Nakatani, Shoko Araki, and Dorothea Kolossa, "Pilot: Introducing transformers for probabilistic sound event localization", *arXiv preprint arXiv:2106.03903*, 2021.

[78] Charles Blandin, Alexey Ozerov, and Emmanuel Vincent, "Multi-source tdoa estimation in reverberant audio using angular spectra and clustering", *Signal Processing*, vol. 92, no. 8, pp. 1950–1960, 2012.

[79] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman, "A scale for the measurement of the psychological magnitude pitch", *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.

[80] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy, *Spoken language processing: A guide to theory, algorithm, and system development*, Prentice hall PTR, 2001.

[81] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang, "Transformer in transformer", *arXiv preprint arXiv:2103.00112*, 2021.

[82] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, 2017.

[83] Ashok Kumar Tellakula, "Acoustic source localization using time delay estimation", *Degree Thesis. Bangalore, India: Supercomputer Education and Research Centre Indian Institute of Science*, 2007.

[84] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra, "Habitat 2.0: Training home assistants to rearrange their habitat", in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[85] Ivo Trowitzsch, Jalil Taghia, Youssef Kashef, and Klaus Obermayer, "The nigens general sound events database", *arXiv preprint arXiv:1902.08314*, 2019.

[86] Angelo Farina, "Advancements in impulse response measurements by sine sweeps", in *Audio engineering society convention 122*. Audio Engineering Society, 2007.

[87] Inkyu An, Youngsun Kwon, and Sung-eui Yoon, "Diffraction-and reflection-aware multiple sound source localization", *IEEE Transactions on Robotics*, 2021.

[88] Yoshiki Masuyama, Yoshiaki Bando, Kohei Yatabe, Yoko Sasaki, Masaki Onishi, and Yasuhiro Oikawa, "Self-supervised neural audio-visual sound source localization via probabilistic spatial modeling", in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4848–4854.

Acknowledgments in Korean

2015년 11월 가을 인턴으로 대학원 생활을 시작하여, 8년 가까운 시간이 흘렀습니다. 이제 대학원 과정을 마무리하고, 새로운 시작을 하려고 합니다. 많은 분들의 도움이 있었기 때문에 길었던 대학원 과정을 잘 지나온 것 같습니다. 이 글을 빌려 감사의 인사를 드리고자 합니다.

저에게 언제나 많은 가르침을 주신 윤성의 교수님께 감사드립니다. 연구자로서, 또는 인생의 선배로서 항상 바른 길로 이끌어 주셨기에, 지금의 저로 성장할 수 있었던 것 같습니다. 항상 교수님의 가르침을 잊지 않고 살아가겠습니다. 그리고, 최정우 교수님과 조병호 박사님, 함께 연구할 수 있어서 영광이었습니다. 많은 가르침 주셔서 감사합니다.

SGVR lab에 들어온 것은 저에게 큰 행운이었던 것 같습니다. 좋은 환경에서 훌륭한 동료들과 마음껏 연구할 수 있었습니다. 저와 함께해준 우리 우리 SGVR lab 멤버들 감사합니다.

로봇팀의 정신적 지주이자 룸메이트인 김동혁 박사님, 함께했던 보드 게임, 부산 역사 탐방 등은 너무 좋은 추억입니다. 이웃사촌이었던 김수민 박사님, 같은 기혼자로서 고민 상담을 해주셔서 감사했습니다. 조언이 큰 도움이 되었습니다. 로봇팀의 실권자 권용선 박사님, 항상 궂은 일도 도맡아 솔선해주셨다는 걸 이제야 깨달았습니다. 연구적으로도 항상 많은 도움 주셔서 감사합니다. 제 연구의 Co-author 김태영 박사님, 함께 연구해서 큰 힘이 되었습니다. 비슷한 연구를 함께하여 큰 도움이 되었습니다. 옆 자리에 앉아 가장 길게 대학원 생활을 함께한 강민철 박사님, 자리도 옆자리에 앉아서 함께한 시간이 길었던 것 같습니다. 함께 많은 수업을 들었는데, 강민철 박사님 도움 덕분에 지금의 학점을 유지할 수 있었던 것 같습니다. 박사 동기 우빈군, 딥-러닝 공부를 시작하며 관련해서 물어봐도 항상 친절하게 알려줘서 감사했습니다. 로봇팀 직속 후배 희찬군, 항상 믿고 일을 부탁하고, 함께할 수 있는 든든한 동료였습니다. 이미지 재윤군, 스타랩 과제 제안서를 작성하며 책임감 있는 일처리를 보고 많이 배웠습니다. 연구실 기둥 현 랩장 준식군, 마지막까지 최고년차라고 많이 배려해줘서 너무 고마웠습니다. I'm grateful to Xu Yin for the entertaining conversations and to Guoyuan An for his invaluable research contributions. 육군 소령 김민철 형님, 아기 태어나 바쁘신데도 항상 군인 자세로 열심히 연구에 임하는 모습이 멋지십니다. 마린보이 윤기군, MT 준비할 때 윤기군이 있어서 항상 든든했습니다. 렌더링 재윤군, ITRC 과제를 함께하며 큰 도움이 되었습니다. Fetch의 새로운 주인 민성군, 항상 연구에 대한 열정적인 자세를 보며 본받을 점이 많았습니다. 제주 KCC2022의 추억 세빈군, 같이 논문 떨어졌다고 필름이 끊길 정도로 술을 마셨던 기억이 이제는 좋은 추억이 되었네요. Brilliant 우재군, 영어 교정 부탁할 때마다 귀찮았을 텐데, 친절하기 도와줘서 고마웠습니다. 먼저 졸업하신 선후배, 현철형, 윤석형, 웅직형, 정수형, 재형군, 병윤군, 농구를 좋아했던 Pio Claudio, 나의 동기 영기군, 첫 논문의 co-author 명배군, 재원군, 홍선군, 훈민군, 창호군, 진혁군, 인영군, 형열군, 규연군, 진원군 함께 연구실 생활하며 좋은 추억 만들어주셔서 감사합니다. 앞으로 우리 SGVR를 이끌어갈, 규범군, 주민양, 주형군, 우정군, Yaxin, 영주, 태근, 제일, 충수형, 문성주 연구원님, 하인우 연구원님의 빛나는 미래도 응원합니다. 그리고 연구에 매진할 수 있게 행정일을 책임져주신 김슬기나 선생님, 김나영 선생님께도 감사드립니다.

긴 대학원 기간을 묵묵히 옆에서 기다려준 사랑하는 아내, 이성민씨에게도 감사를 전합니다. 힘든 순간도 많았지만, 아내와 함께했기 때문에 잘 이겨낼 수 있었던 것 같습니다. 항상 바쁜 대학원생 남편을 이해해주고, 지원해줘서 감사합니다. 너무나도 사랑하는 아들, 안병현에게도 건강하게 잘 자라줘서 고맙다고 전하고 싶습니다. 퇴근 후 반겨주는 아들 덕분에, 더욱 힘내서 잘 마무리할 수 있었습니다. 항상 저를 믿어주고 응원해주신 부모님께도 감사의 인사드립니다. 부모님이 그 동안 저에게 주신 사랑과 가르침을 기반으로 이렇게 성장할 수 있었습니다. 그리고 장인어른과 장모님께도 항상 따뜻하게 반겨주셔서 감사합니다. 처갓집에서 맛있는 음식 많이 먹고 힘내서 더 열심히 할 수 있었습니다. 동생들 유정이와 상규, 주화 처형에게도 항상 응원해줘서 고맙다고 전하고 싶습니다.

Curriculum Vitae in Korean

이 름: 안 인 규

생 년 월 일: 1989년 08월 24일

학 력

2009. 3. – 2016. 2. 동국대학교 전자공학과 (학사)

2016. 3. – 2018. 2. 한국과학기술원 로봇공학학제전공 (석사)

2018. 3. – 2023. 8. 한국과학기술원 전산학부 (박사)

연 구 업 적

1. **Inkyu An**, Guoyuan An, Taeyoung Kim, and Sung-Eui Yoon, *Scalable Microphone Pair Training for Robust Sound Source Localization with Diverse Array Configurations*, (Under review)
2. Taeyoung Kim, **Inkyu An**, and Sung-Eui Yoon, *Inexpensive indoor acoustic material estimation for realistic sound propagation*, Computer Animation and Virtual Worlds (CAVW), 2022
3. **Inkyu An**, Youngsun Kwon, and Sung-Eui Yoon, *Diffraction-and Reflection-Aware Multiple Sound Source Localization*, IEEE Transactions on Robotics (T-RO), 2021
4. **Inkyu An**, Byeongho Jo, Youngsun Kwon, Jung-woo Choi, and Sung-Eui Yoon, *Robust Sound Source Localization considering Similarity of Back-Propagation Signals*, IEEE International Conference on Robotics and Automation (ICRA), 2020
5. **Inkyu An**, Doheon Lee, Jung-woo Choi, Dinesh Manocha, and Sung-Eui Yoon, *Diffraction-aware sound localization for a non-line-of-sight source*, IEEE International Conference on Robotics and Automation (ICRA), 2019
6. Youngsun Kwon, Donghyuk Kim, **Inkyu An**, and Sung-Eui Yoon, *Super rays and culling region for real-time updates on grid-based occupancy maps*, IEEE Transactions on Robotics (T-RO), 2019
7. **Inkyu An**, Myungbae Son, Dinesh Manocha, and Sung-Eui Yoon, *Reflection-aware sound source localization*, IEEE International Conference on Robotics and Automation (ICRA), 2018
8. Pio Claudio, **Inkyu An**, and Sung-Eui Yoon, *A content-aware non-uniform grid for fast map deformation*, the conference on computer animation and social agents (CASA), 2017