sensors

MDPI

Article

An Audio-Based SLAM for Indoor Environments: A Robotic Mixed Reality Presentation

Elfituri S. F. Lahemer * and Ahmad Rad

Autonomous and Intelligent Systems Laboratory, School of Mechatronic Systems Engineering, Simon Fraser University, Surrey, BC V3T 0A3, Canada; arad@sfu.ca
* Correspondence: elahemer@sfu.ca

Abstract: In this paper, we present a novel approach referred to as the audio-based virtual landmark-based HoloSLAM. This innovative method leverages a single sound source and microphone arrays to estimate the voice-printed speaker's direction. The system allows an autonomous robot equipped with a single microphone array to navigate within indoor environments, interact with specific sound sources, and simultaneously determine its own location while mapping the environment. The proposed method does not require multiple audio sources in the environment nor sensor fusion to extract pertinent information and make accurate sound source estimations. Furthermore, the approach incorporates Robotic Mixed Reality using Microsoft HoloLens to superimpose landmarks, effectively mitigating the audio landmark-related issues of conventional audio-based landmark SLAM, particularly in situations where audio landmarks cannot be discerned, are limited in number, or are completely missing. The paper also evaluates an active speaker detection method, demonstrating its ability to achieve high accuracy in scenarios where audio data are the sole input. Real-time experiments validate the effectiveness of this method, emphasizing its precision and comprehensive mapping capabilities. The results of these experiments showcase the accuracy and efficiency of the proposed system, surpassing the constraints associated with traditional audio-based SLAM techniques, ultimately leading to a more detailed and precise mapping of the robot's surroundings.

Keywords: audio-based SLAM; landmarks; EKF/ellipsoidal landmark-based SLAM; robotic mixed reality; Microsoft HoloLens; landmarks; HoloSLAM; Nao humanoid robot

Citation: Lahemer, E.S.F.; Rad, A. An Audio-Based SLAM for Indoor Environments: A Robotic Mixed Reality Presentation. *Sensors* **2024**, *24*, 2796. https://doi.org/10.3390/s24092796

Academic Editor: Hui Kong

Received: 7 March 2024
Revised: 21 April 2024
Accepted: 25 April 2024
Published: 27 April 2024

1. Introduction

Advancements in artificial intelligence (AI) have facilitated the transition to a new era of versatile, efficient, and affordable autonomous robots [1]. They are employed in various indoor and outdoor tasks such as mapping, localization, pathfinding, obstacle avoidance, guiding, guarding, and providing care for the elderly, and most notably, autonomous navigation [2]. Autonomous navigation is crucial in many applications as it enables a robot to safely and effectively traverse complex, unstructured environments. The robot must be able to simultaneously build a map of its surroundings and determine its location within that map [3,4]. This process is now well established and is referred to as Simultaneous Localization and Mapping (SLAM) [5]. The SLAM problem can be applied to a wide variety of environments, including both static and dynamic, indoor and outdoor, with different robotic platforms such as ground robots, underwater robots, and aerial drones [5–7]. Considerable effort has been dedicated to crafting efficient solutions for the SLAM problem [8,9]. In its early stages, SLAM primarily relied on range sensors, such as sonar, lasers, and cameras, as the principal information sources for constructing maps and ascertaining the robot's position and orientation [10]. Within indoor environments, the majority of SLAM and navigation systems depend on visual data. Vision imparts a wide range of capabilities to robots, rendering cameras a universally integrative component [11]. However, vision-based SLAM faces limitations like a restricted field of view and occlusion,

hampering target exploration. Ground conditions heavily affect its accuracy [12]. Sensor fusion and advanced algorithms, like machine and deep learning, enhance mapping accuracy and resilience against limitations and changes [13]. An alternative approach is to integrate audio input into the robot's navigation system, thereby broadening its sensory capabilities and facilitating navigation in scenarios where visual data may be inadequate [14–17]. Indeed, an auditory system enables the robot to comprehend spoken instructions, identify particular sounds or speakers, determine the source of sounds accurately, and react to important environmental auditory cues. This includes sound source localization, speaker tracking, speech separation, recognition, and audio-based SLAM [18–20]. Computational auditory scene analysis (CASA) has progressed in understanding environmental sounds, focusing on source localization and separation [16,21,22]. In contrast, audio-based SLAM algorithms are relatively less mature and face certain challenges that may hinder their widespread adoption in robotics [23,24]. Audio-based SLAM typically involves several important steps. Firstly, data are acquired using one or more microphones, capturing sound waves from the environment. Preprocessing enhances data quality by filtering noise. Features are then extracted for localization and mapping. Sound source position is estimated using techniques like beamforming, triangulation, or time-delay estimation, along with combinations of signal processing methods, sensor fusion, and potentially machine learning approaches. Data association maintains the correspondence between observed sound sources and landmarks. State estimation integrates information from various sensors, and optimization refines trajectory and map estimates, minimizing errors [25].

When equipped with a microphone array, a robot can estimate sound source directions, but accurately gauging distance presents challenges, particularly when the distance surpasses the array's dimensions. As a result, deducing Cartesian source positions from Direction of Arrival (DoA) estimates presents an issue with multiple unknowns. Additionally, the presence of reverberation and noise introduces errors in estimation that may result in incorrect source position estimations. Moreover, instances of silence, such as during human speech pauses, can result in the absence of audio source estimations. Consequently, research in sound source localization and SLAM for mobile robots has mainly focused on detecting the sound sources' directions. Numerous theories and methods exist for microphone-array-based sound source localization, including Received Signal Strength (RSS), Angle of Arrival (AOA), Time of Arrival (TOA), Time Difference of Arrival (TDOA), Frequency Difference of Arrival (FDOA), Multiple Signal Classification (MUSIC), beamforming, and other advanced techniques [26]. In the TDOA algorithm, accurately estimating the sound source location hinges on effectively gauging the time difference of signals received by microphones [27]. This is achieved through two main approaches: cross-correlation methods like Generalized Cross-Correlation (GCC) and cross-power spectrum phase, and obtaining TDOA estimation via path impulse response calculations. The GCC with Phase Transform (GCC-PHAT) stands out as a specific approach frequently utilized for various sound localization tasks [19,28,29].

Current audio-based SLAM methods [30–33] typically assume open spaces and clear paths to multiple sound sources. Real-world scenarios, however, often feature reflective surfaces like narrow hallways, causing localization challenges. In addition, these audio-based SLAM solutions incorporate either artificial or natural audio sources as landmarks and these landmarks are progressively integrated into the robot's map over time. Existing audio-based SLAM solutions often overlook the presence of landmarks due to the unavailability of direct paths to sound sources caused by reflections or detection issues. This limitation hampers the comprehensive mapping and localization of environments, urging the need for advancements in techniques to address such complexities effectively [34]. In an audio-based SLAM, landmarks can be extracted from audio signals or use audio sources themselves. In environments with multiple audio sources, localization ambiguity increases due to overlapping signals, reflections, and reverberations. Mapping becomes complex as each source contributes to the acoustic map. Identifying and tracking multiple sources requires advanced signal-processing techniques like source separation and

clustering. Techniques like triangulation improve location estimation. However, managing multiple sources increases computational demands and system costs. Conversely, single-source audio SLAM offers simplicity and reduced complexity, albeit with potential localization and mapping inaccuracies. Identifying and tracking the main audio signal in single-source SLAM is crucial, yet poses challenges in complex environments with signal distortion and multiple sources. Our method utilizes single-source audio-based SLAM. For localization, we exclusively estimate the direction of the primary signal (active speaker) using the GCC-PHAT technique with a microphone array.

Mixed Reality (MR) offers a promising avenue to overcome the limitations of traditional landmark-based SLAM systems. By overlaying digital information onto physical surroundings, MR expands the perceptual capabilities of robots beyond tangible landmarks, allowing for more versatile mapping and localization. In MR-enhanced SLAM systems, virtual landmarks can be dynamically generated and manipulated, offering flexibility in adapting to diverse environments. These virtual landmarks may include not only visual cues but also auditory or spatial markers, aligning with the capabilities of audio-based SLAM methods. Additionally, MR and holographic displays facilitate the creation of interactive and immersive experiences, enabling robots to interact with both physical and virtual elements for enhanced localization and mapping accuracy [35]. Integrating Microsoft HoloLens or any other mixed-reality device [36,37] with a robot's real-world environment enables the robot to effectively follow, track, communicate, and interact with specific speakers. Simultaneously, it empowers the robot to conduct a virtual audio-based SLAM with high accuracy and success, revolutionizing its ability to navigate and perceive its surroundings in dynamic and complex environments.

The main contribution of this paper over and above the state of the art is the integration of a microphone array platform, mixed-reality, and holographic displays on Microsoft HoloLens [37] to perform audio landmark-based SLAM in indoor environments. The proposed system is verified on a Nao robot [38] platform. The system begins with identifying a specific speaker in a multi-audio environment and extracting sound source information using the microphone array. Simultaneously, it employs a Short-Time Fourier Transform (STFT) to transform input signals into the complex domain and extract features using a combination of GFCC (Gammatone Frequency Cepstral Coefficients) and MFCC (Mel-frequency Cepstral Coefficients) [39]. These extracted features are then fed into multiple speaker classifiers, including Gaussian Mixture Model (GMM) [40], Support Vector Machine (SVM) [41], Convolutional Neural Network (CNN) [42], Deep Neural Network (DNN), and Time-Delay Neural Network (TDNN) [43], for keyword detection and sound source identification, including the source's angle. This angle information is then used to position virtual landmarks in the robot's environment via Microsoft HoloLens holographic apps. The research also involves the application of a traditional ellipsoidal SLAM algorithm to estimate the robot's path and integrate virtual landmarks in the mapped environment. The robot is then able to navigate toward the specified speaker while avoiding interfering sound sources. This study demonstrates the effective localization of both the robot and sound sources in indoor environments, which has implications for improving robot navigation and interaction in real-world scenarios.

The paper's organization is as follows: Section 2 provides a comprehensive review of the relevant literature, highlighting the present state of audio-based SLAM and its primary challenges. Section 3 elucidates the intricate design of our proposed system. Section 4 integrates simulation studies and an extensive exploration of the benefits inherent in our architecture. The paper culminates with a conclusion in Section 5. This structured approach guides readers through the background, system details, empirical assessments, and final insights of our research.

2. Related Studies

The idea of robot audition was first reported by Nakadai et al. [44]. Subsequently, researchers have explored numerous approaches to enhance sound source localization

(SSL) for various applications in robotics [45–47]. The use of SSL in robotics is relatively new, dating back to 1989 when Squirt, the first robot equipped with an SSL module, was introduced [48]. After the Squirt robot was equipped with the ability to locate surrounding sound sources in 1989, the SSL field has continuously advanced to address challenges. In 1995, MIT's Robert installed a basic robot auditory system. In 2006, the Honda Research Institute pioneered real-time tracking with IRMA and a robot-head microphone array integration [49]. These approaches involve collecting acoustic data from sound sources such as microphone arrays and integrating them with other sensory data such as vision and odometry information [50]. Filtering techniques [51,52] are then applied to leverage sound information alongside robot movement data to accurately estimate their position and orientation, which can be valuable for tasks such as navigation, mapping, and interaction with their environment. Conventional audio-based SLAM approaches primarily integrate SSL with SLAM [53]. These methods typically initiate TDOA estimation using multi-channel audio data from microphone arrays. Subsequently, the relative distances or angles between sound sources are then computed to assist SLAM implementation. In [33], a collection of sound sources served as landmarks, and a microphone array was mounted on a wheeled robot. This setup was designed for the concurrent localization of both the sound sources and the robot. Meng et al. [29] introduced an approach utilizing a microphone array combined with Light Detection and Ranging (LiDAR). The study successfully located the robot and mapped its environment in experiments. Nonetheless, to achieve satisfactory outcomes, precise motion data from odometry or LiDAR were required. The robot's motion and its performance were hindered due to signal sync, noise, and DOA errors from indoor acoustics. Inaccurate motion reports limit effectiveness. Some SSL methods presented in [54–56] combined audio and visual data, focusing solely on visible sound sources, and are unsuitable for robot navigation when targets are hidden. Sasaaki et al. [57] designed a mobile robot with a microphone array for estimating multiple sound source positions by triangulating observations from various robot positions. Echoes [58] and multipath [59] have also previously been employed for SLAM and, more broadly, for estimating room geometry [60]. In [61], a method was proposed for localizing a mobile robot using structured sound sources that emit unique codes, similar to the GPS system, where the exact positions of each sound source are known beforehand. This is different from the SLAM approach where landmark locations are not known a priori. While effective for static sound sources, these methods struggle to adapt when the robot moves amidst dynamic sound changes and to accurately estimate the distance between the sources and the robots in such situations.

In contrast, the system introduced in this paper is initiated by identifying unique sound targets via pre-registered voiceprints. The angle of the target speaker is all that is required; the algorithm tracks the sound of interest, facilitating navigation and SLAM task execution with a virtual map. The estimated direction serves as observation data, and this became a standard bearing-only SLAM problem solely for guiding the robot to track the active speaker while localizing itself and creating a map of its environment. However, since there is a lack of additional information to perform a complete SLAM operation, the Ellipsoidal HoloSLAM algorithm [62] is employed. Ellipsoidal HoloSLAM addresses this problem by incorporating virtual landmarks into the mapping process, allowing for an accurate and realistic SLAM implementation without a need for an active predefined sound source locationprior to location. As the robot moves and the active speaker's location and direction change, the SSL and SLAM algorithms work together to continually update the robot's position and orientation within the built map, allowing it to follow the speaker and build a detailed virtual map of the environment at the same time. This approach has potential in indoor applications in areas such as human—robot interaction, assistive robotics, and indoor navigation.

3. Materials and Method

3.1. Overall System Architecture

The inference part of the proposed system was implemented on a Nao robot for real-time operation [63] as illustrated in Figure 1. The robot is equipped with a microphone circular array module, specifically the ReSpeaker microphone array module (Seeed Technology Co., Ltd., Shenzhen, China), on its head. The ReSpeaker module is connected to a Raspberry Pi 4B (Raspberry Foundation, Cambridge, UK), which is used to collect and process the recorded acoustic data [64].

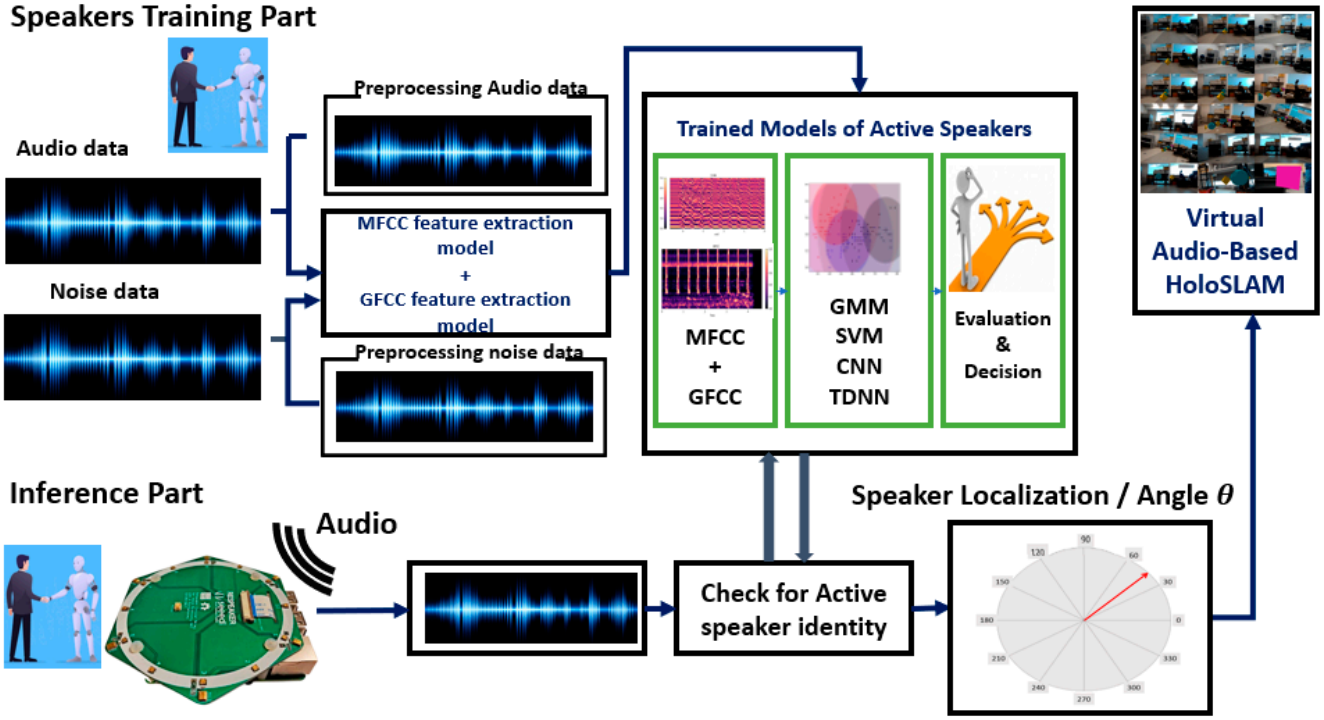

Figure 1. System overview of real-time virtual HoloSLAM process and active speaker identification and localization.

Upon detecting a keyword, the ReSpeaker microphone array records acoustic data, feeding them into various classifier models (GMM, SVM, DNN, CNN, TDNN) to discern the active speaker's angle. This allows the robot to adjust its position, promoting natural interaction and movement tracking. This has applications in human-robot dialogue, social robotics, and guided robot tours.

Subsequently, the robot employs the virtual Ellipsoidal HoloSLAM technique to map its surroundings and establish its position within the map. Unlike traditional acoustic-based SLAM, this approach leverages Microsoft HoloLens and mixed-reality technology to virtually map the environment. Virtual landmarks are incorporated into the robot's surroundings while it moves, tracking the active speaker and efficiently avoiding obstacles.

3.2. Active Speaker Localization Using Microphone Array

This section focuses on Sound Source Localization (SSL), which involves identifying the direction and distance of detected sounds using electronic receivers, like microphones. We use the term "Sound Localization" for sound direction estimation. SSL system design is an active area of research, employing techniques like beamforming, cross-correlation, time delay estimation, and machine learning [65,66]. Figure 2 outlines the key stages of sound source direction estimation.

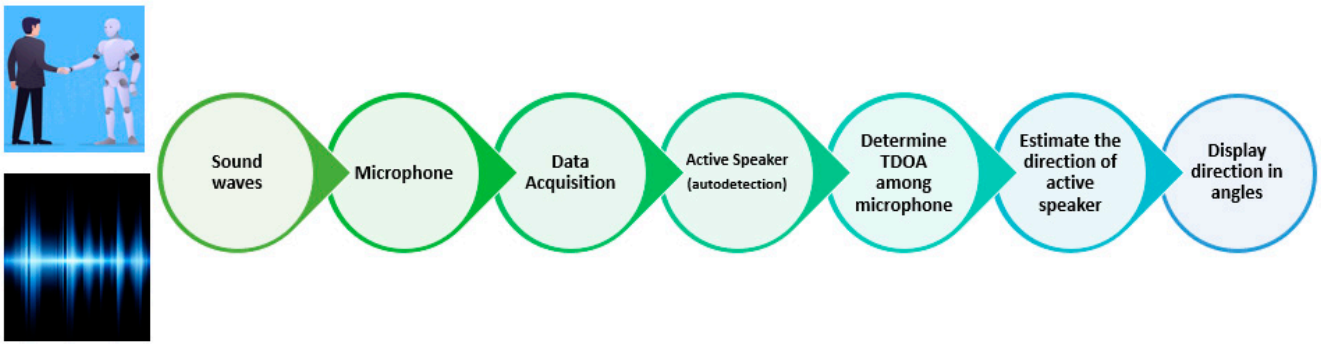

Figure 2. SSL design steps.

Upon confirming a received signal as a signal of interest, data from microphones undergo processing, including band-pass filtering. Time Difference Of Arrival (TDOA) between microphones is measured to determine the sound source location, often using the Generalized Cross-Correlation with Phase Transform weighting function (GCC-PHAT) method [67] (Figure 3).

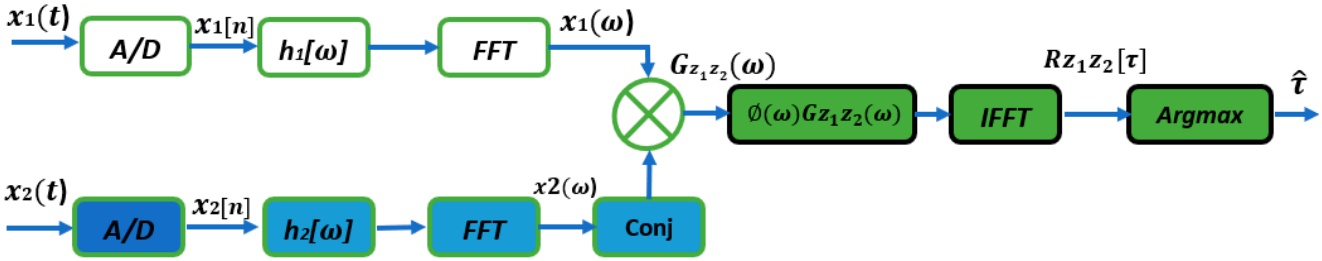

Figure 3. The Generalized Cross-Correlation-PHAT block diagram.

The cross-correlation between two discrete signals $x_1(t)$ *and* $x_2(t)$ received from the left and right channels can be defined by [29]:

$$R_{x_1x_2}(k) = \sum_{n=-\infty}^{\infty} x_1(n) \cdot x_2(n+k) \quad (1)$$

The cross-correlation can also be represented with the help of the convolution operator as follows:

$$R_{x_1x_2}(k) = x_1(-k) * x_2(k) \quad (2)$$

In practice, a limited signal segment is processed, estimating cross-correlation. The equation applies to two signals of length N:

$$R_{x_1x_2}(k) = \frac{1}{N} \sum_{n=0}^{N-1-K} x_1(n) \cdot x_2(n+k) \quad (3)$$

For longer signals, Fourier transformation simplifies calculations, enabling frequency domain multiplication. The spectral cross-power density is defined as the Fourier transform of the cross-correlation function by:

$$S_{X_1xX_2}(f) = X_1(f) \cdot X_2^*(f) \quad (4)$$

Complex conjugation is denoted by $(\cdot)$. Cross-correlation is calculated via inverse Fourier transformation. When one signal is a time-shifted version of another, cross-correlation has a peak at time D. The delay is expressed as:

$$\hat{D} = \underset{k}{arg\max} \hat{R}_{x_1x_2}(k) \quad (5)$$

Real-time factors affect the maximum position. To enhance stability, Generalized Cross-Correlation (GCC) in [68] uses weight functions on cross-power spectral density. The general GCC equation is:

$$\hat{R}_{x_1x_2}^{(g)}(\tau) = \mathcal{F}^{-1}\{X_1(f)\cdot X_2^*(f)\cdot \psi(f)\} \tag{6}$$

where ψ stands for a weighting function.

Various weighting functions are available for GCC to improve time delay sensitivity. PHAT weight, as introduced in [69], can be defined as:

$$\psi_p(f) = \frac{1}{|X_1(f)\cdot X_2^*(f)|} = \frac{1}{|S_{X_1X_2}(f)|} \tag{7}$$

Inserted into Equation (6), the GCC-PHAT results in:

$$\hat{R}_{x_1x_2}^{(p)}(\tau) = \mathcal{F}^{-1}\left\{\frac{X_1(f)\cdot X_2^*(f)}{|X_1(f)\cdot X_2^*(f)|}\right\} = \mathcal{F}^{-1}\left\{\frac{S_{X_1X_2}(f)}{|S_{X_1X_2}(f)|}\right\} \tag{8}$$

The position of the maximum $\hat{R}_{x_1x_2}^{(p)}(\tau)$ of corresponds to the delay between the signals:

$$\hat{D}_p = arg\max_k \hat{R}_{x_1x_2}(\tau) \tag{9}$$

For discrete signals, $\hat{D}_p$ represents time units through signal sampling frequency. Shift replaces delay. In acoustic source localization, a microphone pair, known as an "active" pair, is utilized to estimate sound source direction. The choice of this pair varies based on the array geometry and sound source direction. Typically, it consists of the two closest or the pair with the greatest time delay difference [68]. The calculation of the angle based on the signal propagation time difference between two microphone signals takes place here using the following equation:

$$\theta = sin^{-1}\left(\frac{\tau}{\tau_{max}}\right) \tag{10}$$

The τ is the signal runtime difference and τ_{max} is the maximum runtime between two microphones. τ_{max} can be calculated by the following:

$$\tau_{max} = \frac{d}{c} \tag{11}$$

where d is the distance between the two microphones and c is the speed of sound.

By substitute τ_{max} from Equation (11) into Equation (10), the angle based on the signal propagation time difference can be calculated as:

$$\theta = sin^{-1}\left(\frac{c\cdot\tau}{d}\right) \tag{12}$$

Using the plane wave model, the distance required for the wavefront to pass through both microphones can calculated as shown in Figure 4. The time difference can be calculated as:

$$\tau = \frac{d\,\sin(\theta)}{c} \tag{13}$$

An angle of 0° signifies a wavefront perpendicular to the microphone axis, while $\pm 90^\circ$ indicates a wavefront aligned with the microphone axis. When the signal propagation time difference τ is zero, the angle θ is 0°. At $\tau = \tau_{max}$, θ is 90°, and at $\tau = -\tau_{max}$, θ is -90°. For $\tau \neq \pm\tau_{max}$, two angles are observed on the full circle, mirroring each other along the microphone axis. This means only angles between -90° and 90° are determinable in a linear microphone array, without information about whether the wavefront is above or below the microphone axis.

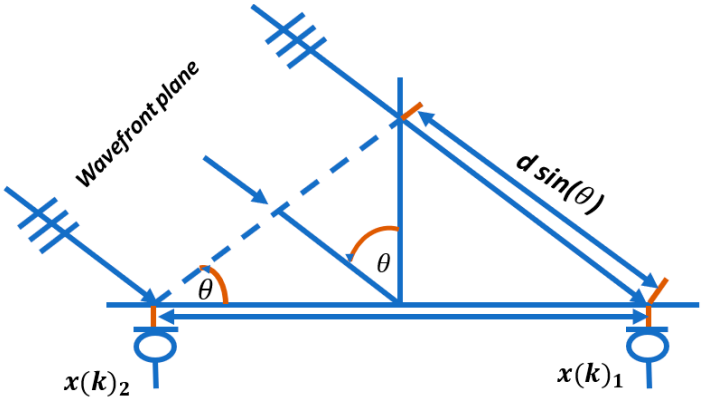

Figure 4. Calculating the angle of arrival between two microphones via τ.

3.3. Audio-Based Ellipsoidal Virtual HoloSLAM Algorithm Implementation

In robotics, auditory systems are vital for human interactions and navigation tasks. Current research addresses multiple aspects, such as speaker localization, speech separation, enhancement, recognition, and speaker identification [65,70,71]. Speaker localization using biological hearing principles or microphone arrays has been a long-standing focus. Sound source localization (SSL) aims to automatically locate sound sources, which is crucial for a robot's self-localization and mapping. The research objective here is to enable a robot to autonomously determine its location and map its surroundings while in motion, even without prior sound source knowledge. **Localization** here solely refers to estimating the robot's position over time in a global frame, without prior knowledge of natural or artificial sound source landmarks.

HoloSLAM revolutionizes the landmark-based SLAM in autonomous robot navigation by merging the real and virtual worlds using Microsoft HoloLens and mixed-reality techniques [62]. It combines established methods to provide real-time environment construction and robot position tracking. Mixed Reality, as demonstrated with HoloLens, seamlessly integrates virtual and physical elements, enabling the robot to interact with both [37,72]. This breakthrough eliminates the need for real multi-sound sources, as virtual landmarks can be generated and placed in scenarios lacking physical landmarks. For detailed HoloSLAM implementation, please consult the reference [62].

In this project, the Microsoft HoloLens-Mixed Reality landmark-based SLAM (HoloSLAM) is utilized along with the ellipsoidal set-membership filter method [11] to address the challenges associated with landmarks in landmark-based acoustic-based SLAM. This approach allows for accurate robot localization and mapping even without multiple sound sources are required. With HoloSLAM, the robot gains the capability to dynamically place virtual landmarks within its environment in real time, using its robot voice as a means of interaction. A virtual landmark represents a digital entity serving as a recognizable point or feature in augmented or mixed-reality environments. It includes items like 3D models, holographic representations, or interactive elements strategically placed in the robot's physical space, seamlessly blending with reality to enrich its environment. These virtual landmarks play a crucial role in the robot's accurate self-localization, eliminating the need for explicit sound source locations.

Devices with powerful CPUs and GPUs are required for processing these virtual landmarks (the virtual digital data) and real-world information, alongside display devices like lenses or screens to showcase generated digital content, facilitating immersive environments. Common extended reality devices include Microsoft HoloLens, Magic Leap One, Epson Moverio, and Google Glass, while popular VR choices encompass HTC Vive, Oculus Quest, Valve Index, and Sony PlayStation VR. Additionally, companies like Microsoft offer HMD display devices for mixed-reality production, alongside various smart glasses [62].

The HoloSLAM with Mixed Reality and Microsoft HoloLens operates within a framework defined by two distinct scenarios. In the first scenario, real landmarks are fully

accessible and detectable, providing the robot with tangible points of reference for navigation. In contrast, the second scenario arises when real landmarks are either unavailable or undetectable, necessitating the reliance on virtual landmarks presented through the device to facilitate navigation. The audio-based HoloSLAM system is specifically crafted to function solely with virtual landmarks.

HoloLens seamlessly integrates virtual elements into the real world utilizing spatial mapping and tracking technology. This innovative process incorporates advanced sensors, cameras, and algorithms to ensure that virtual objects maintain their position and perspective within the environment, adapting to changes in the user's viewpoint or location. Through depth cameras and IMUs, HoloLens constructs a detailed 3D representation of the surroundings, continuously updating it to reflect any alterations. This spatial mapping capability distinguishes HoloLens as a mixed-reality device, setting it apart from standard augmented reality tools.

Spatial mapping involves generating a three-dimensional depiction of the physical space, while scene understanding interprets the elements within it, recognizing objects and their attributes. In Unity, this is achieved by creating a 3D mesh using depth data from the camera, representing surfaces in the environment. Each surface triangle is linked to a world-locked spatial coordinate system, ensuring consistency in virtual object placement [37].

Despite the complexity, HoloLens adjusts virtual objects in real-time based on robot head movement and perspective changes, applying perspective corrections to maintain realism. However, challenges such as variations in perspective and lighting, as well as occlusions, can affect visual integration. Microsoft has not officially disclosed details regarding algorithms and hardware precision, but practical experiments suggest an accuracy within a few centimeters. This margin of error is considered in implementations like audio-based HoloSLAM. HoloLens offers a compelling augmented reality experience through its sophisticated integration techniques.

Upon successfully deploying an audio-based virtual landmark, the robot proceeds to localize itself within the environment and seamlessly incorporates this landmark into the existing map, establishing it as a pivotal reference point for subsequent navigation tasks. These virtual landmarks, displayed through the HoloLens, serve as surrogate points of reference, enabling the robot to continue its navigation task despite the absence of real audio cues.

Figure 5 depicts the typical Unity3D engine interface showcasing the HoloSLAM virtual landmark hologram application.

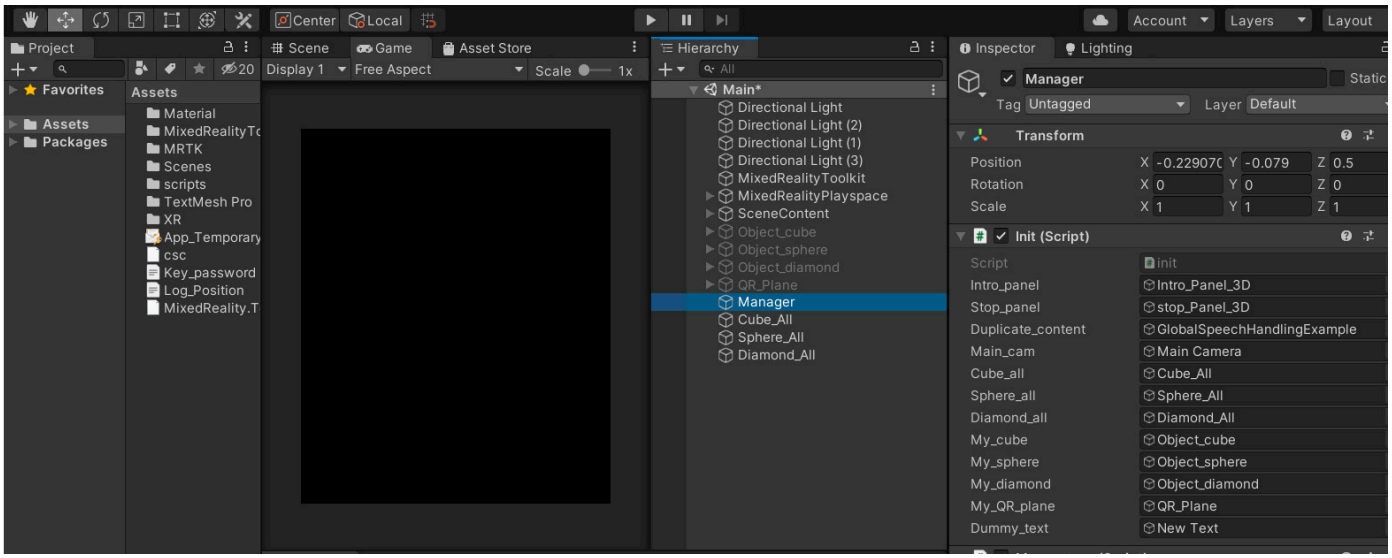

Figure 5. The Unity3D window of the HoloSLAM virtual landmark hologram app [62].

The following is the pseudo-code for the hologram app that employs the HoloLens mixed reality technique to place the virtual landmarks in the robot environment. The execution of this hologram occurs exclusively within the HoloLens device and is triggered solely by a voice command (***Start***) generated through the robot's speakers.

Start-launch *the Holo-landmark hologram app.*
voice function command (Start)
Place Virtual Landmark Observation–
-Voice function command (place virtual landmark).
Take a Picture (if needed) –
-voice function command (takepicture).
Exit- *close the Holo-landmark hologram app.*
voice function (Exit)

Once the Holo-landmark hologram app is deployed in the HoloLens device through the robot's voice command function (***Start***), the robot initiates the HoloLens to ensure the app is operational. Subsequently, when the active speaker is recognized and the robot receives a movement command, it prompts the Holo-landmark hologram to place virtual landmarks in predetermined positions using the voice command (***place virtual landmark***) through the Holo-landmark hologram app. In this paper, the Holo-landmark hologram app is purpose-built to introduce a singular type of virtual landmark, such as a cube, sphere, or diamond shape, with each execution of the voice command "***place virtual landmark***". Different applications require different functions and different virtual landmarks. Nonetheless, the Holo-landmark app can easily incorporate additional objects or landmarks by simply incorporating new functions. Unity3D, Morphi, 3D Slash, Fusion 360, and Blender are valuable resources that provide a vast collection of pre-existing 2D/3D objects that can serve as audio virtual landmarks [73]. These virtual landmarks can be placed in random positions, offering flexibility and freedom in selecting their desired locations. The functionalities embedded within the virtual holographic application on the HoloLens may vary across different applications, offering flexibility in the features available for navigation assistance. However, irrespective of the complexity or simplicity of the audio-based navigation task, the fundamental design principles governing the creation and utilization of virtual landmarks remain consistent. These virtual landmarks play a crucial role in facilitating audio-based virtual HoloSLAM, ensuring robust and reliable navigation capabilities under varying environmental conditions.

Presented below is the pseudo-code for the integration of an audio-based HoloSLAM hologram with Ellipsoidal-SLAM. The core structures of Ellipsoidal-SLAM remain intact, while the HoloLens-Mixed Reality operations come into play when a virtual landmark is placed. These virtual landmarks are then incorporated into the global location mapping using Ellipsoidal-SLAM.

Start
Initialization*—SLAM Initialization, NAO Robot Initialization,* ***Launch Holo-landmark hologram app (voice function command(start)).***
Get Observation (4)–Is the Active Speaker Identified?
Yes*—****Holo-landmark*** *hologram app*
*-Place Virtual Landmark (**voice function command (place virtual landmark)**)*
No*—****No*** *action (wait for speaker identification).*
while not_stop
Prediction Step*—Check for a safe distance to move by* ***sonar****. (**Move command**)*
No*—safe distance. Turn 180 degrees.*
Is the Active Speaker Identified?
Yes*—****Holo-landmark*** *hologram app*
*-Place Virtual Landmark (**voice function command (place virtual landmark)**)*
No*—No action (wait for speaker identification)*
Data Association(5)-
Virtual Landmark *matching and data-association simplification*
Correction_Step*—Run standard Ellipsoidal—SLAM update step.*
Augmented_Map*—Add new* ***Virtual Landmarks*** *to the map*
Check if iteration numbers are achieved.
No—Go to step 4
End*—****Close-Holo-landmark hologram app.*** ***Voice function command (stop)***

During the initialization process of Ellipsoidal-SLAM, the robot simultaneously initiates the Holo-landmark hologram app by issuing a voice command (start). If the robot detects an active speaker through the detected keyword commands, it responds to their command by placing a virtual landmark through the voice function command (place virtual landmark) first and then responds to the movement command.

The primary contribution of integrating HoloLens-Mixed Reality into sound-based Ellipsoidal-SLAM lies in the utilization of virtual landmarks in situations where multiple sound sources or acoustic landmarks are unavailable, and the full SLAM process cannot be accomplished.

3.4. Active Speaker Representation and Modeling

Human speech conveys various information like words, emotions, gender, and identity. Initially, speech comprehension outpaced speaker identification [74]. Speaker recognition research started in the 1960s, notably by Pruzansky and Mathews in 1964, who used digital spectrograms to verify speakers [75]. This formed the foundation for extensive research in speaker recognition, investigating various feature extraction and similarity measurement techniques. Automatic speaker recognition, or voice biometrics, identifies individuals using vocal traits. It has applications in security, forensics, and human–computer interaction. Unlike speech recognition, it focuses on verifying or identifying speakers, involving **speaker verification** to confirm claimed identity and **speaker identification** to determine identity from a group of candidates [76,77].

Speaker recognition systems have core components: **feature extraction** and **feature matching**. Extraction involves capturing data from the voice signal to represent speakers. Matching identifies unknown speakers by comparing their features to known speakers. This field has text-dependent and text-independent recognition. Text-dependent recognition requires accurate utterance of a password, while text-independent recognition verifies identity without content restrictions [42].

This study targets active speaker identification, employing MFCC and GFCC. Multiple recognition methods, including HMM, GMM, RF, SOM, statistical approaches, and deep neural networks, are applicable. GMM, SVM, and deep neural network-based classifiers were tested for active speaker classification.

3.4.1. Feature Extraction Techniques for Active Speaker Identification

Audio feature extraction, part of signal modeling, transforms audio data into a domain that groups similar instances and separates distinct categories. Inspired by human audi-

tory and articulatory systems, these methods yield meaningful representations. Effective feature extraction reduces data dimensionality, offering several benefits such as decreased computational complexity and the removal of redundant or irrelevant information [78]. This study utilizes GFCC and MFCC features for speaker identification, as they capture diverse voice signal characteristics. Figure 5 visually depicts the MFCC and GFCC feature extraction processes. Both share a common sequence of stages, differing only in filter bank types applied to the frequency domain signal obtained through FFT and the subsequent compression step. MFCC uses a Mel filter bank, followed by logarithmic compression and DCT. In contrast, MFCC uses a Mel filter bank, followed by logarithmic compression and Discrete Cosine Transform (DCT). In contrast, GFCC applies a Gammatone filter bank, pre-loudness, and DCT. Figure 6. shows the Block diagram of MFCCs and GFCCs feature extraction modules.

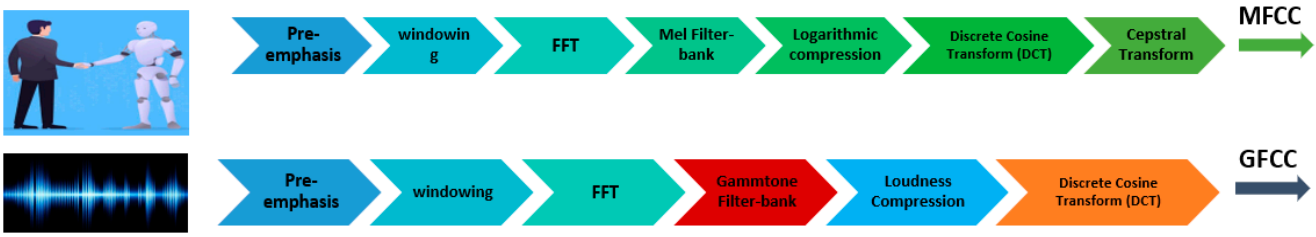

Figure 6. Block diagrams of MFCC and GFCC feature extraction modules.

The MFCC feature vectors target human speech frequencies up to 1000 Hz using linear and logarithmic filters, capturing spoken word spectral characteristics precisely.

For the MFCC feature, let us consider $X(n)$ as the original input speech signal, and $Y(n)$ as the enhanced or amplified speech signal, given by Equation (14):

$$\boldsymbol{Y(n) = X(n) - a{*}X(n-1)} \tag{14}$$

The pre-emphasis factor, typically chosen from the range of 0.95 to 0.98, is applied. Subsequently, a smoothing window function, such as Hamming windows (Equation (15)), is used on the pre-emphasized speech signal $Y(n)$.

$$\boldsymbol{W(n) = 0.54 - 0.46{*}cos\,cos\left(\frac{2\pi n}{N-1}\right),\ 0 \le n < N-1} \tag{15}$$

The time-domain signal is then transformed to the frequency domain using Fast Fourier Transform (FFT). A Mel filter bank, designed for speaker recognition, refines the spectrum. In the final step, the log Mel spectrum is converted back to time, yielding MFCC using logarithmic compression and discrete cosine transform (Equation (16)).

$$\boldsymbol{C_n = \sum_m^M [log\,log\,S(m)]\,cos\,cos\left[\frac{\pi n}{M}\left(m - \frac{1}{2}\right)\right],\ 0 \le n < N-1} \tag{16}$$

M represents the output of an M-channel filter bank, and $\boldsymbol{n}$ is the index of the cepstral coefficient. This cepstral representation effectively captures the local spectral properties of the signal for the given frame analysis.

To analyze transitions, another method involves calculating the first difference of MFCC signal features, referred to as the feature's delta (Δf). This delta signifies the rate of change in a feature over time, providing insights into transitions between speech sounds. It is computed using a simple formula:

$$\boldsymbol{\Delta f_k = \left(f_{k+1} - f_{k-1}\right)/2}$$

where f_{k+1} and f_{k-1} are feature values at adjacent time points. Figure 6 displays the MFCC, Delta, and Delta-Delta features in our dataset. These features provide hierarchical

representations of audio signals, capturing spectral characteristics, temporal dynamics, and higher-order temporal variations, respectively. The output graph illustrates that MFCC-delta-delta contains fewer coefficients compared to both MFCC-delta and MFCC. The figure generates a more comprehensive depiction of the frame's context, resulting in improved accuracy. In this representation, the x-axis denotes time, while the y-axis represents the MFCC coefficient values.

GFCC's feature computation parallels MFCC's with a significant distinction: the use of gammatone filters. These filters, inspired by human cochlear processing, enhance feature extraction from the FFT spectrum. Like MFCCs, the process involves pre-emphasis, windowing, and FFT. Gammatone filters are then applied, extracting distinct features for GFCC representation, and capturing auditory system-inspired insights. The formula representing each filter's impulse response is as follows [39]:

$$g(t) = at^{n-1}e^{-2\pi bt}\, cos\, cos\, (2\pi f_c t + \varphi) \tag{17}$$

As 'a' remains constant, 'n' and 'φ' are consistent throughout the filter bank. The Gammatone filter bank's frequency selectivity relies mainly on two parameters: central frequency 'f' and filter bandwidth 'b'. A common approach for setting these values approximates them based on human cochlear filters using the Equivalent Rectangular Bandwidth (ERB), following Moore's model [79]. This approach effectively simulates the human auditory system.

$$ERB(f_c) = 24.7 + 0.108\, f_c \tag{18}$$

To align the Gammatone filter with human auditory characteristics, follow Moore's recommendation [79] and Patterson et al.'s use of the ERB concept [80]. Set the filter parameters as bandwidth (b) = **1.019 ∗ ERB** and filter order (n) = 4. This ensures better compatibility with the human auditory system. Moore's guidance suggests spacing center frequencies uniformly on the ERB frequency scale, creating the relationship between the number of ERBs and corresponding center frequencies, denoted as f_c, which can be represented by the following expression:

$$numberofERBs = 21.4\, log_{10}(0.00437\, f_c + 1) \tag{19}$$

The ERB scale, logarithmic in nature, relates center frequencies and frequency energy distribution in speech, following a $\frac{1}{f}$ pattern. Gammatone filters adapt bandwidth, narrowing at lower frequencies and broadening at higher frequencies. Figure 7 displays the MFCC, Delta, and Delta-Delta features in our dataset. Figure 8 illustrates the cochleagram response, a Gammatone filter bank, and a typical spectrogram in response to our dataset.

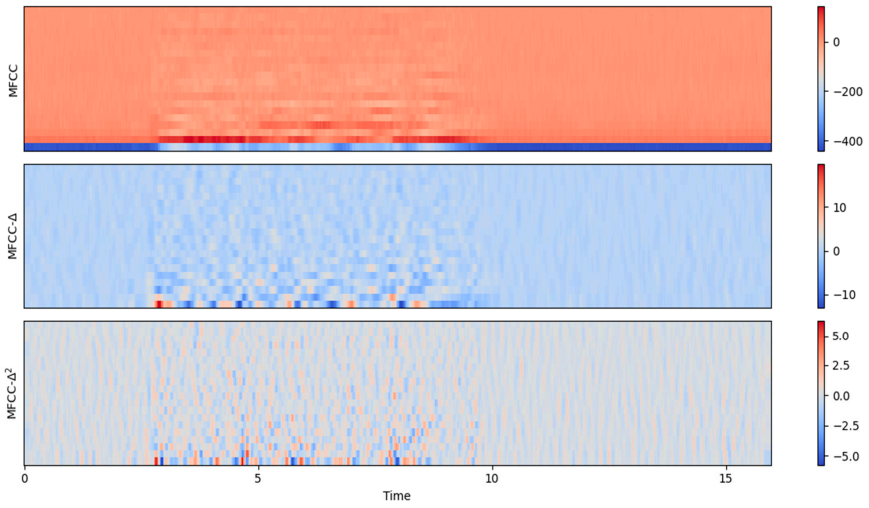

Figure 7. MFCC output of our dataset.

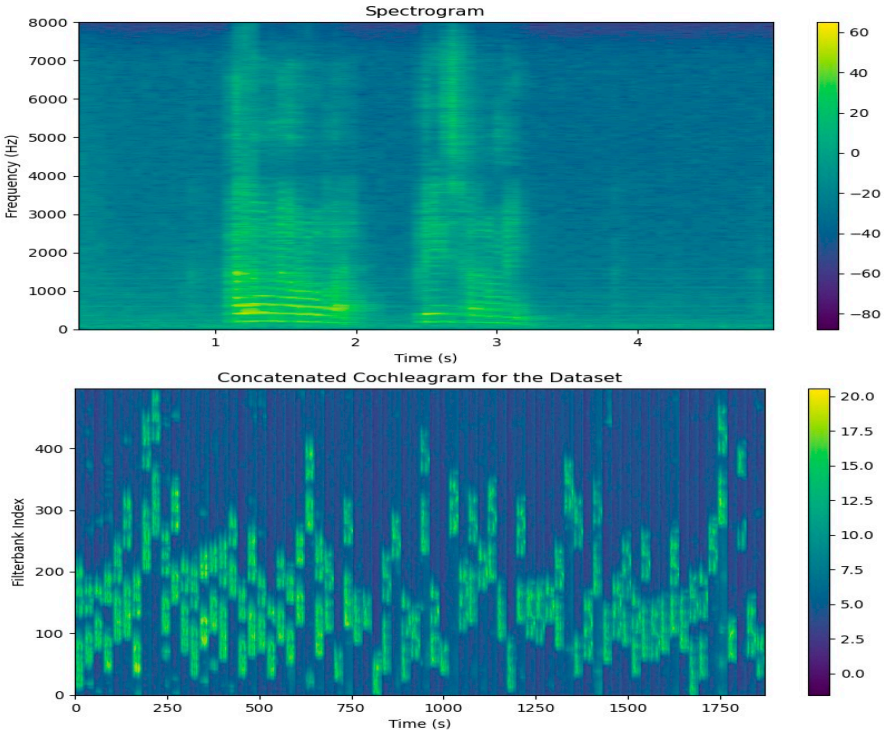

Figure 8. The Spectrogram and Cochleagram of a sample speech signal from our dataset.

3.4.2. Active Speaker Representations (Classification Algorithms)

Following the extraction of distinct features from an audio signal, a classifier is utilized to differentiate these features, creating a model for each speaker. These models are then used to compare new input features with stored reference templates to determine identity. Speaker classification involves **stochastic** (parametric) models like Gaussian mixture models and Hidden Markov Models, which use probabilistic pattern matching, and **template** (non-parametric) models like Dynamic Time Warping and Vector Quantization, which employ deterministic pattern matching [81,82]. The choice of classification method depends on the specific application, with dynamic time warping and hidden Markov models suited for text-dependent recognition and vector quantization and Gaussian mixture models commonly used for text-independent recognition. This section covers established classification algorithms extensively used in speech recognition and active speaker identification in this work.

Gaussian Mixture Model

In this section, we clarify the structure of the GMM and its rationale for representing active speaker identity in text-independent speaker identification. GMM is a robust tool in Speaker Recognition Evaluations (SREs), adept at addressing data analysis and clustering challenges through a mixture of Gaussian densities. Through unsupervised techniques like clustering, GMM offers a valuable probabilistic model for data grouping. In contemporary text-independent GMM systems, the Expectation Maximization (EM) algorithm is commonly used to estimate background model parameters, ensuring GMM-based methods remain at the forefront of speaker recognition advancements. The GMM model is defined as a likelihood function with a mixture of *M* Gaussians, expressed by the following equation [40]:

$$p\left(\vec{x}\middle|\lambda\right) = \sum_{i=1}^{M} w_i p_i(\vec{x}) \quad (20)$$

where $p(\lambda)$ is the frame-based likelihood function, λ is the hypothesis or likelihood function, x is a set of features (MFCCs or GFCCs), and (x) is the individual Gaussian density function.

The model is estimated by a weighted linear combination of D-variate Gaussian density function $p_i\left(\vec{x}\right)$, each parameterized by a mean $D \times 1$ vector, μ_i, mixing weights, which are constrained by $w_i \geq 0$, $\sum_{i=1}^{M} w_i = 1$, and a $D \times D$ covariance matrix, Σ_i as:

$$p_i(\vec{x}) = \frac{1}{2\pi^{D/2}|\Sigma_i|^{1/2}} exp\{\frac{1}{2}(x-\mu_i)'(\Sigma_i{}^{-1})(x-\mu_i)\} \quad (21)$$

Once the model has completed its training, the subsequent step involves evaluating the log-likelihood of this model using a test set comprising feature vectors MFCCs/GFCCs.

$$p(\lambda) = \sum_{i=1}^{M} p(\lambda) \quad (22)$$

Support Vector Machine (SVM)

SVM, a binary classifier, distinguishes speakers from impostors via a separation hyperplane. Exploring SVM techniques assesses novel classification methods for speaker identification, enhances comprehension of the challenge, and determines whether SVMs offer insights beyond traditional GMM approaches. SVM utilizes a kernel function to create a binary classifier, with the sequence kernel based on generalized linear discriminants. Notably, it directly expands into the SVM feature space while maintaining computational efficiency and increased accuracy. SVM complements and competes effectively with other methods, including Gaussian mixture models. It seeks the optimal hyperplane that maximizes the margin between data and the separation boundary, resulting in the best generalization performance [41]. Figure 9 shows the principle of the optimal hyperplane and the optimal margin in SVM modeling.

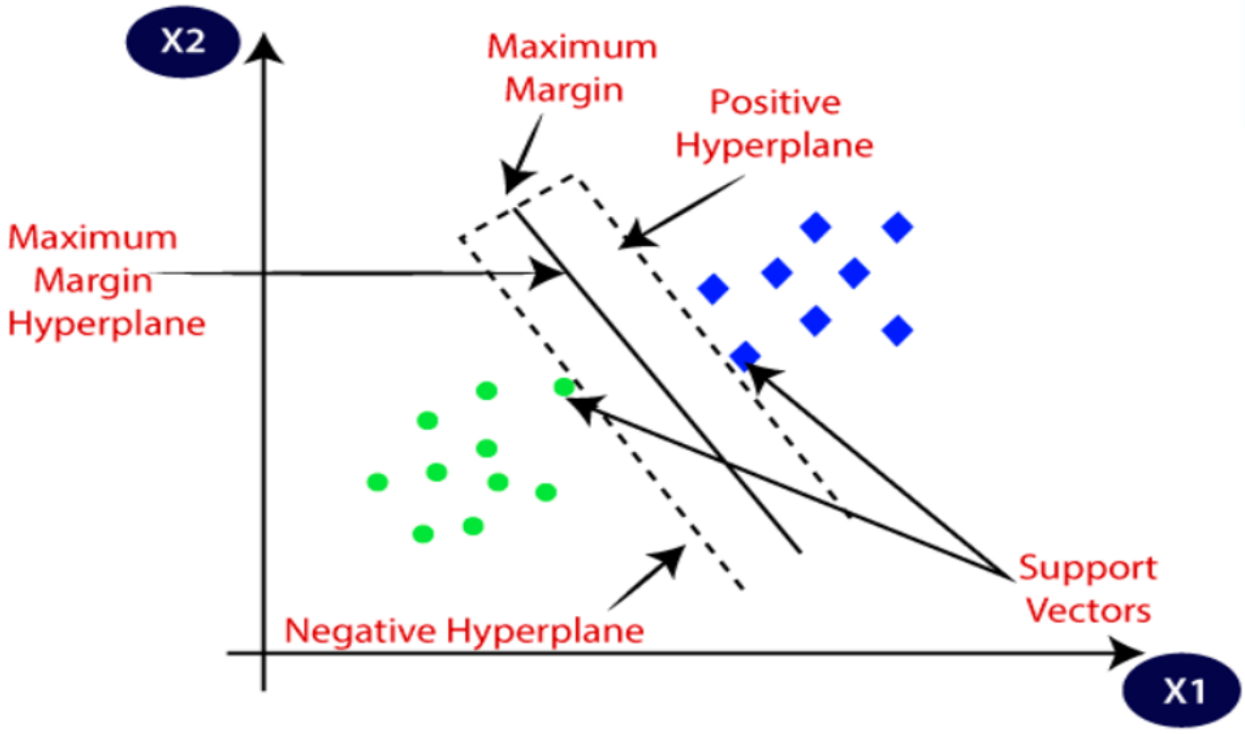

Figure 9. Principle of support vector machines.

The discriminant function of the SVM is given by:

$$f(x) = \sum_{i=1}^{N} \alpha_i t_i K(x, x_i) + d \quad (23)$$

where the t_i are the ideal outputs, $\sum_{i=1}^{N} \alpha_i t_i = 0$, and $\alpha_i > 0$. The vectors x_i are support vectors and are obtained from the training set by an optimization process. The ideal outputs are either 1 or -1, based on the support vector class. The kernel $K(.\,,\,.)$ is constrained to have certain properties (the Mercer condition), so that $K(.\,,\,.)$ can be expressed as:

$$K(.,.) = b(x)^t b(y) \tag{24}$$

$b(x)$ maps input space in SVM, where a two-class model for speaker identification is trained. The known non-targets comprise the second class, with class 0 assigned to the target speaker's utterances.

SVM can be represented as a two-class problem: target and nontarget speaker. If ω is a random variable representing the hypothesis, then $\omega = 1$ represents the target being present and $\omega = 0$ represents the target not being present. A score is calculated from a sequence of observations $y_1, \ldots, y_n$ extracted from the speech input. The scoring function is based on the output of a generalized linear discriminant function of the form $g(y) = \omega^t b(y)$ where ω is the vector of classifier parameters and b is an expansion of the input space into a vector of scalar functions [33]:

$$b(y) = [b_1(y) b_2(y) \ldots . b_n(y)]^t \tag{25}$$

If the classifier is trained with a mean-squared error training criterion and ideal outputs of 1 for $\omega = 1$ and 0 for $\omega = 0$, then $g(y)$ will approximate the posterior probability $\mathbf{p}(\omega = 1|y)$. We can then find the probability of the entire sequence, $\mathbf{p}(y_1 \ldots y_n|\omega = 1)$ as follows:

$$\mathbf{p}(y_1 \ldots y_n|\omega) = \prod_{i=1}^{n} \mathbf{p}(y_i|\omega) = \prod_{i=1}^{n} \frac{\mathbf{p}(\omega|y_i)\mathbf{p}(y_i)}{\mathbf{p}(\omega)} \tag{26}$$

Taking *log* on both sides [33], we obtain the discriminant function:

$$d^{'}(y_1^n|\omega) = \sum_{i=1}^{n} \log\left(\frac{\mathbf{p}(\omega|y_i)}{\mathbf{p}(\omega)}\right) \tag{27}$$

For classification purposes, we discard $p(y_i)$. Using $\log(x) \approx x - 1$:

$$d(y_1^n|\omega) = \frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{\mathbf{p}(\omega|y_i)}{\mathbf{p}(\omega)}\right) \tag{28}$$

Assuming $g(y) \approx p(\omega = 1|y)$:

$$d(y_1^n|\omega = 1) = \frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{w^t \mathbf{b}(y_i)}{\mathbf{p}(\omega = 1)}\right) = \frac{1}{np(\omega = 1)} w^t \left(\sum_{i=1}^{n} \mathbf{b}(y_i)\right) = \frac{1}{p(\omega = 1)} w^t \bar{b}_y \tag{29}$$

where the mapping $y_1^n \rightarrow b_y$ by is:

$$y_1^n = \frac{1}{n}\sum_{i=1}^{n} b(y_i) \tag{30}$$

In the scoring method, for a sequence of input vectors $x_1, x_2, \ldots, x_n$ and a speaker model w, we can construct b using (30). For speaker identification, if the score is above a threshold, then we declare the identity claim valid; otherwise, the claim is rejected as an impostor attempt.

Deep Learning-Based Models Architecture

In recent years, deep learning-based models have become the cornerstone for audio classification tasks, enabling the automatic categorization of audio signals into various classes, such as speech recognition, music genre classification, and environmental sound analysis. These models, characterized by their sophisticated architectural design, have

demonstrated remarkable performance in handling complex audio data, making them an indispensable tool in various domains including multimedia analysis, content recommendation, and surveillance systems. DNN excels here by leveraging multiple filters during training to extract unique features from input spectrograms. These features improve the representation of active speakers in speech data, autonomously learned and then used for identification by a classifier [83]. In this section, we explore CNN-LSTM and TDNN architectures as the two main ones that have been employed in this work.

1. Convolutional Long Short-Term Memory Network

CNN, a deep learning model based on convolution, is primarily used for image analysis in machine learning. However, it has shown broad utility in recognizing audio patterns, improving images, processing natural language, and forecasting time series data. The CNN architecture, introduced by Lecun et al. [84], consists of an input layer, an output layer, and concealed layers, with convolutional layers performing dot product operations between input matrices and convolutional kernels.

A Long Short-Term Memory (LSTM) Network belongs to the category of recurrent neural networks (RNNs), which are essentially neural networks with feedback loops [85]. RNNs perform well in speech recognition, language modeling, and translation, but they face a key challenge: the vanishing gradient problem. This occurs when the error gradient dwindles or grows explosively during backpropagation, especially across multiple time steps, leading to limited memory capacity, often called '**short memory**'. LSTM network architecture offers a solution by using a special memory cell to control information flow. This selectively retains or discards data, preventing gradient problems and enabling the learning of long-term dependencies in sequential data. Figure 10 illustrates the architecture of an LSTM cell. Each cell receives two critical inputs: the output sequence produced by the previous LSTM cell and the hidden state value from the previous cell, denoted as h_{t-1}. Inside the cell, there are three gates: the forget gate f_t, the input gate i_t, and the output gate O_t.

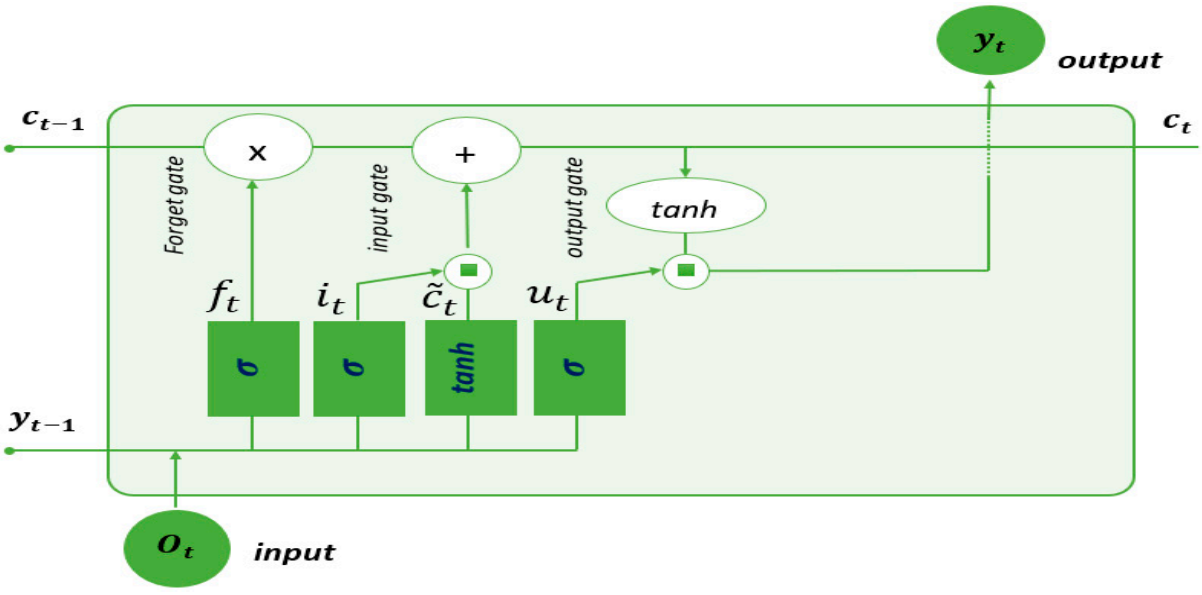

Figure 10. The architecture of an LSTM cell.

Information from the previous hidden state h_{t-1} and information from the current input x_t are passed through the sigmoid function. The forget gate acts as a filter to forget certain information about the state of the cell. To this end, a term-to-term multiplication is carried out between f_t and c_{t-1}, which tends to cancel the components of c_{t-1} close to 0. A filtered cell state is then obtained as follows:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \quad (31)$$

where σ denotes the sigmoid activation function, which is a nonlinear function that maps its input to a value between 0 and 1, W_f is the weight of the forget gate, and b_f is the bias. The weights and bias values are acquired through the training process of the LSTM.

LSTM employs the input gate for data integration into the memory cell, comprising the input activation gate and the candidate memory cell gate. The input activation gate manages data integration, while the candidate memory cell gate governs data storage within the memory cell.

By considering both the previous hidden state h_{t-1} and the current input node x_t, the input gate in an LSTM generates two essential vectors: the input vector i_t and the candidate memory cell vector $\widetilde{c}_t$. Equation (32) describes the operation of the input activation gate, which involves the weight matrix W_i and bias vector b_i. Simultaneously, Equation (33) demonstrates the formation of the candidate memory cell $\widetilde{c}_t$ by applying the hyperbolic tangent activation function (**tanh**) to the same set of inputs, utilizing the weight matrix W_c and bias vector b_c.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{32}$$

$$\widetilde{c}_t = \mathbf{tanh}(W_c[h_{t-1}, x_t] + b_c) \tag{33}$$

The input vector and the candidate memory cell vector are merged to update the previous memory cell c_{t-1}, as shown in Equation (34). In this equation, the symbol $\odot$ represents element-wise multiplication.

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t \tag{34}$$

The output gate controls data transfer from the memory cell to the current hidden state, serving as the LSTM's output. The output gate vector o_t is computed with this equation:

$$o_t = \sigma(W_o[h_{t-1}, x_t, c_t] + b_o) \tag{35}$$

Subsequently, the current hidden state, h_t, is derived using following equation:

$$h_t = o_t \odot tanh(c_t) \tag{36}$$

The new cell state and that of the hidden state are then directed to the next time step. Training sequential neural networks minimizes loss over data sequences using back-propagation through time (BPTT) for temporal gradients. Weight updates are computed mathematically based on loss function L and the learning rate η can be expressed as:

$$\Delta W = -\eta \frac{\partial L}{\partial W} \tag{37}$$

In this paper, the CNN-LSTM architecture utilizes CNN layers to construct a model of an active speaker from input data to enhance the model's ability to make sequence predictions.

2. Time-delay neural networks (TDNNs)

A Time-Delay Neural Network (TDNN) is a dynamic network designed to capture temporal relationships between events and maintain temporal translation invariance. Initially introduced to enhance modeling of extensive temporal context [43], TDNN models have found applications in spoken word and online handwriting recognition. TDNN remains a common choice for acoustic modeling in modern speech recognition software such as Kaldi [84]. Its primary function is to convert acoustic speech signals into sequences of phonetic units, known as 'phones'. The network takes acoustic feature frames as input and produces output depicting probability distributions for each phonetic unit. The network takes acoustic feature frames as input and produces a probability distribution for a defined set of target language phones. The goal is to classify each frame into the phonetic unit with the highest likelihood. In a single TDNN layer, each input frame is represented as a column vector, symbolizing a time step, with rows representing feature values. A compact weight

matrix, often called a kernel or filter, slides over the input signal, performing convolution to generate the output.

Consider an input vector x_t in $\mathbb{R}^m$ as a matrix containing m numerical values, such as amplitudes at a specific frequency or the values of acoustic features within a filter bank bin. At each time step t, we will have a matrix of input features $\boldsymbol{X} \in \mathbb{R}^{m\ x\ t}$ where each vector represents one-time step t of our speech signal with a trainable weight matrix $\boldsymbol{W} \in \mathbb{R}^{m\ x\ l}$, where the kernel maintains a consistent height of m and a width of l, as illustrated in Figure 11.

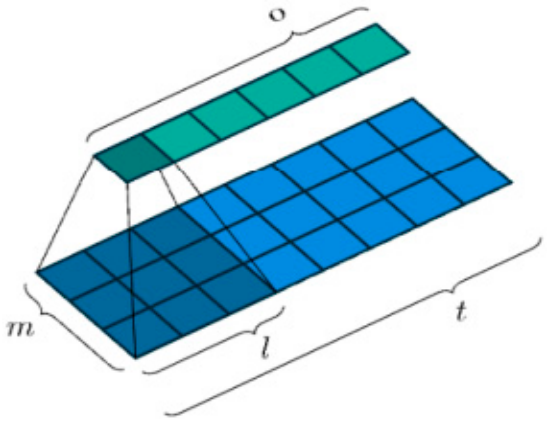

Figure 11. Defining a trainable weight matrix for the TDNN [85].

The kernel W moves across the input signal with a space of s making s steps in each movement. The area on the input feature map that the kernel encompasses is termed the "receptive field". Depending on the specific implementation, it is possible that the input may be filled with null values at both ends of height of m and a length of p. The output width o, resulting from the number of times the kernel can fit over the length of the input sequence, can be calculated as follows:

$$\boldsymbol{o} = \left\lfloor \frac{\boldsymbol{t} - \boldsymbol{l} + \boldsymbol{2p}}{\boldsymbol{s}} \right\rfloor + \boldsymbol{1} \tag{38}$$

where $\lfloor . \rfloor$ represents the floor function.

During each time step t, the TDNN conducts a convolution operation, which involves performing an element-wise multiplication (commonly known as the Hadamard product) between the kernel weights and the input located below it, followed by the summation of these resulting products.

Within the neural network, a trainable bias term b is included (which is not shown in the images above). The outcome is then processed through a non-linear activation function denoted as ϕ (examples of which include sigmoid, rectified linear, or p-norm functions). This process results in the formation of an output $z_q\ \epsilon\ z$ where z represents the entire output vector, achieved by performing this operation across all time steps (depicted as the light green vector in the images). Hence, the concise representation of the scalar output for a single element, denoted as $z_q\ \epsilon\ z$, at the q-th output step within the set {1,2. . ., o}, can be expressed as:

$$z_q = \phi\left(\boldsymbol{W} * \boldsymbol{X_q} + \boldsymbol{b}\right) \tag{39}$$

where $*$ denotes the convolution operation and $\boldsymbol{X_q}$ are the inputs in the receptive field. It can also be equivalently given by:

$$z_q = \phi\left(\sum_{i=1}^{m} \sum_{k=1}^{l} \boldsymbol{w_{i,k}} \boldsymbol{x_{i,k}} + \boldsymbol{b}\right) \tag{40}$$

In this equation, the initial summation extends across the height of the acoustic features, while the subsequent summation covers the width of the receptive field or the width of the kernel. It is important to note that the kernel weights $w_{i,k}$ are shared across all output steps q. Because the weights of the kernel are shared across the convolutions, the TDNN acquires a representation of the input that remains insensitive to the precise location of a phone within the broader sequence. Additionally, this sharing of weights reduces the quantity of parameters that need to be trained.

Considering that we need to repeat the same convolution operation as before, denoted as $z_q = \phi(W*X_q + b)$, it is important to note that the input vectors X_q have also grown due to the expanded receptive field. In simpler terms, this process involves extracting the receptive field from its input, combining it, and applying the identical convolution operation.

Finally, in the context of employing multiple kernels, represented as H kernels, where each kernel is can be represented as $W^{(h)}\ \epsilon\ W^{(1)}, \ldots, W^{(H)}$, where each kernel similarly moves across the input. This process results in the generation of a sequence of output vectors, which can be structured into an output matrix $\mathbf{Z}\ \epsilon\ R^{hxo}$.

Within a deep neural network architecture, this output can subsequently serve as the initial hidden layer of the network and be employed as input for the subsequent layer of the TDNN.

4. Experimental Results

4.1. Active Speaker Identification

In this section, we present the results of our implemented speaker identification system experiments. We conducted a performance comparison of five different classification methods: GMM, SVM, CNN, DNN, and TDNN. The evaluation is based on a dataset featuring two distinct active speaker classes, Speaker-1 and Speaker-2, recorded in a noise-free environment at the AISL. We used two feature extraction methods: one with MFCC and additional features, and the other with GFCC and similar features. Our primary aim is to assess the accuracy of each method in precisely distinguishing between these two speakers. The experimental analysis provides valuable insights into the strengths and weaknesses of each approach in the context of active speaker identification tasks.

The GMM was fitted using EM and clusters were determined using K-Means clusters. EM showed stable convergence. We explored different covariance types in GMM. Figures 12 and 13 visualize clustering. GMM-MFCC features effectively grouped data. GMM-GFCC captured some speaker patterns but with lower accuracy. Future work can enhance feature extraction, use advanced clustering, or increase training data for better accuracy.

Figure 12. Scatter plot illustrating the clustering outcome of GMM-MFCC.

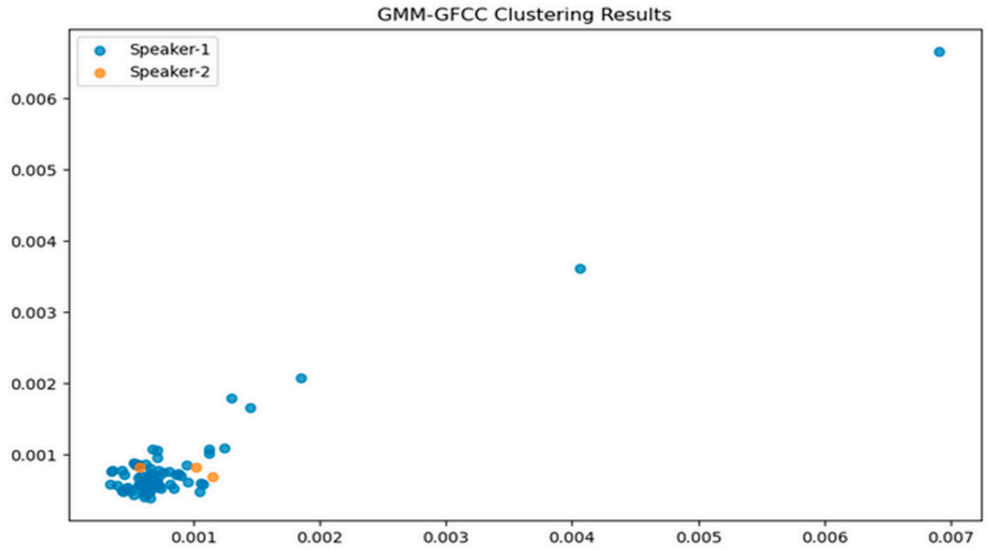

Figure 13. Scatter plot illustrating the clustering outcome of GMM-GFCC.

Table 1 compares how different covariance types in GMMs affect K-fold split data, showing their influence on GMM's clustering ability. GMM clustering with GFCC features yielded 31% accuracy, while MFCC features improved accuracy to 80.9%. Although not ideal, the scatter plot suggests the model identified speaker similarities, offering the potential for further clustering enhancements.

Table 1. Comparative Analysis of Gaussian Mixture Model Covariance Types on K-Fold Split MFCC and GFCC Data.

	Covariance Type							
k-th Split	**MFCC**				**GFCC**			
k	**Full**	**Diag**	**Tied**	**Spherical**	**Full**	**Diag**	**Tied**	**Spherical**
1	79.687500	43.750000	35.937500	32.031250	39.062500	50.000000	44.791667	43.359375
2	21.875000	14.843750	17.187500	18.359375	39.062500	39.062500	39.062500	40.234375
3	78.461538	42.307692	55.384615	46.153846	38.461538	36.153846	36.923077	43.846154
4	21.538462	13.846154	35.384615	31.923077	38.461538	50.000000	46.153846	43.076923
5	21.538462	56.923077	45.641026	39.615385	44.615385	41.538462	40.512821	40.384615
6	20.000000	56.923077	44.615385	53.076923	32.307692	35.384615	35.897436	36.153846
7	84.615385	87.692308	86.666667	68.846154	32.307692	33.846154	32.820513	33.461538
8	84.615385	46.923077	57.435897	46.923077	40.000000	37.692308	47.179487	45.000000
9	83.076923	44.615385	36.410256	47.307692	40.000000	50.000000	54.871795	50.000000
10	23.076923	14.615385	36.410256	47.307692	69.230769	53.846154	48.717949	45.769231

Next, we reveal the outcomes of two SVM models employing MFCC and GFCC feature extraction. Our aim is to assess the effect of feature extraction on SVM performance. Both models used a linear kernel with C = 1.0. The SVM-MFCC model achieved 100% accuracy, demonstrating the power of these features for classification. In contrast, the SVM-GFCC model, while slightly less accurate, displayed robust classification abilities with some overlap between classes, indicating potential for refinement. These results highlight how feature extraction impacts SVM performance: MFCCs excel, while GFCCs offer solid performance with opportunities for improvement in separating classes. Figure 14 shows the decision boundary for the SVM model using MFCC features, clearly separating the two classes, confirming its 100% accuracy. In Figure 15 the decision boundary for the SVM

model with GFCC features is more intricate, with some class overlap, explaining the slightly lower accuracy. However, it effectively separates most data points while acknowledging some overlap.

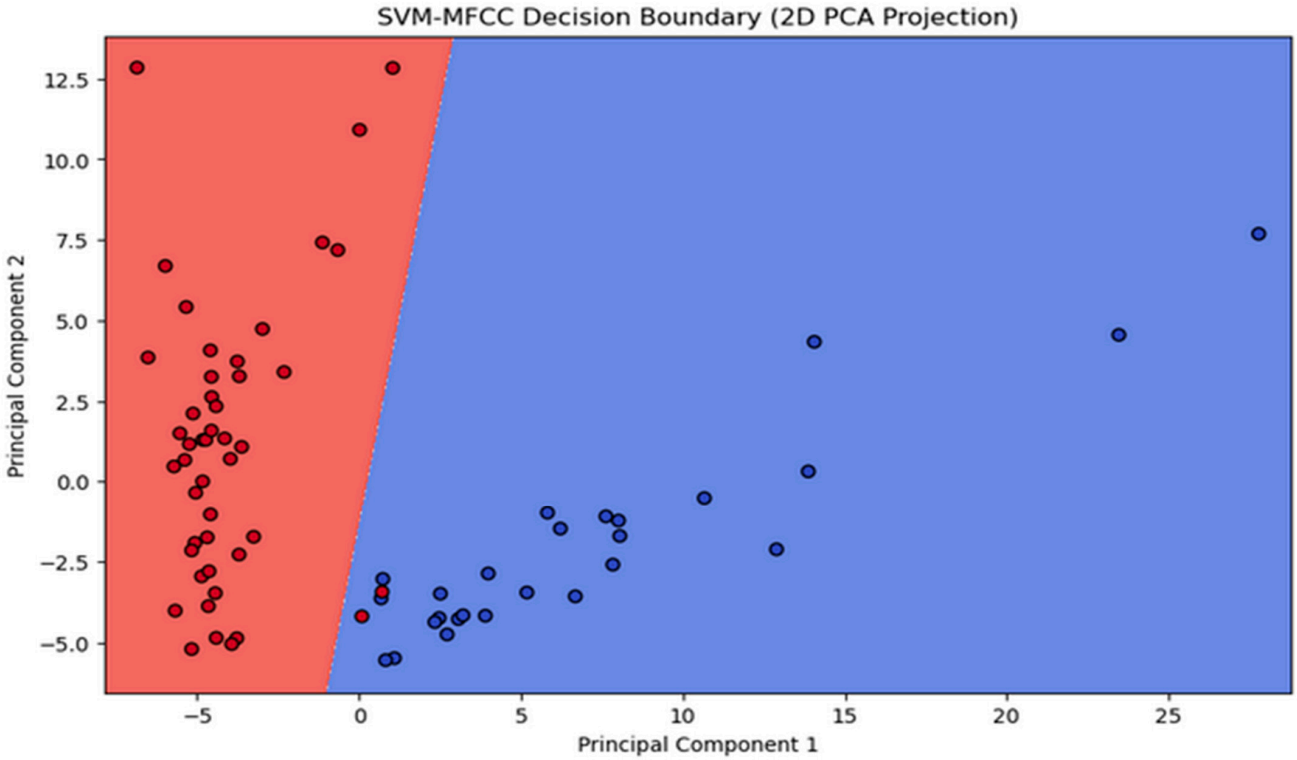

Figure 14. Scatter plot illustrating the clustering outcome of SVM-MFCC.

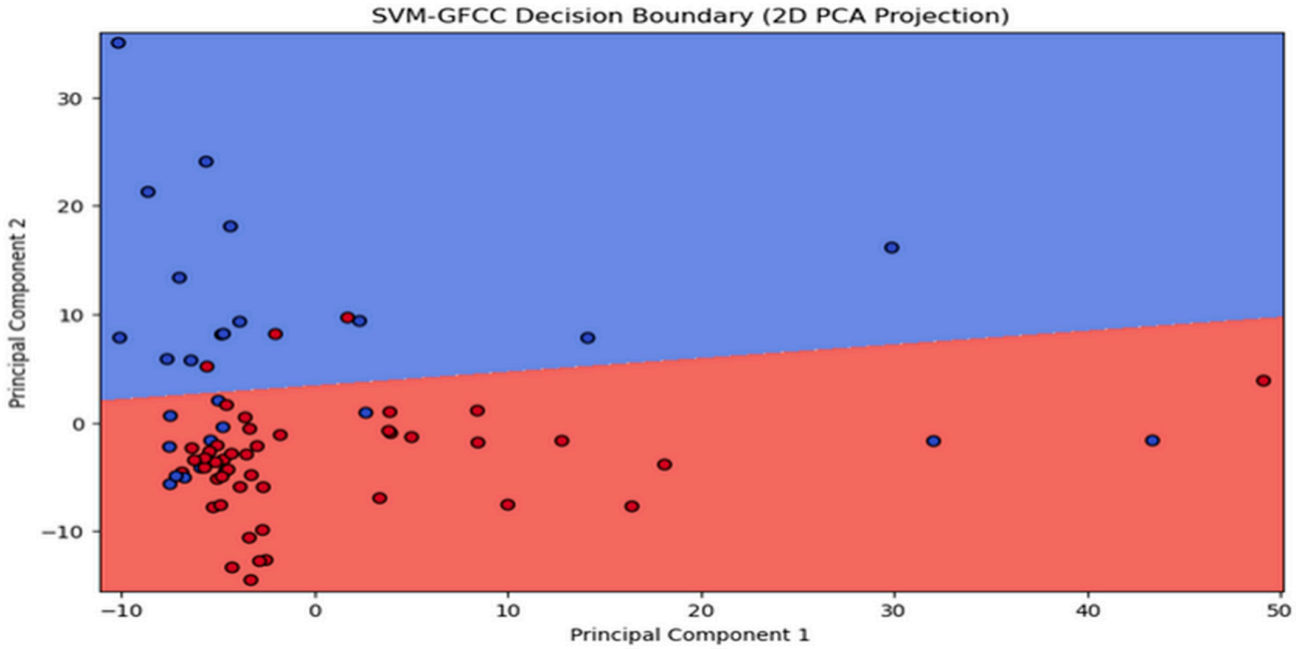

Figure 15. Scatter plot illustrating the clustering outcome of SVM-GFCC.

Visualizing decision boundaries provides insights into model behavior. The distinct boundary in the MFCC model highlights their suitability for this dataset. The GFCC model's boundary, while effective, suggests potential benefits from additional feature

engineering or model refinement for improved performance. Both SVM models with MFCC and GFCC show promise in dataset classification, with the choice depending on data characteristics and the balance between accuracy and interpretability.

We will now present the results of our speaker identification system using deep learning architectures, including Conv-LSTM, DNN, and TDNN models. These models were trained using combined features based on MFCC and GFCC, along with additional features such as chroma, Mel frequency, zero-crossing rate (ZCR), root mean square energy (RMSE), delta, and delta-delta features. Our goal is to evaluate how well these models accurately identify and distinguish between two active speakers. The recognition rate graph for the CNN, DNN, and TDNN models using MFCC features visually represents their performance in speaker identification. The graph displays changing recognition rates along the x-axis, reflecting their accuracy in identifying speakers. Early stopping was used to prevent overfitting.

Figure 16 displays the connection between training epochs and accuracy/recognition rate. It shows an initial steady increase in accuracy, signifying effective learning. However, there is a plateau, indicating diminishing returns with more training. Early stopping prevented overtraining by restoring the best weights, as evident in the stabilized accuracy curve.

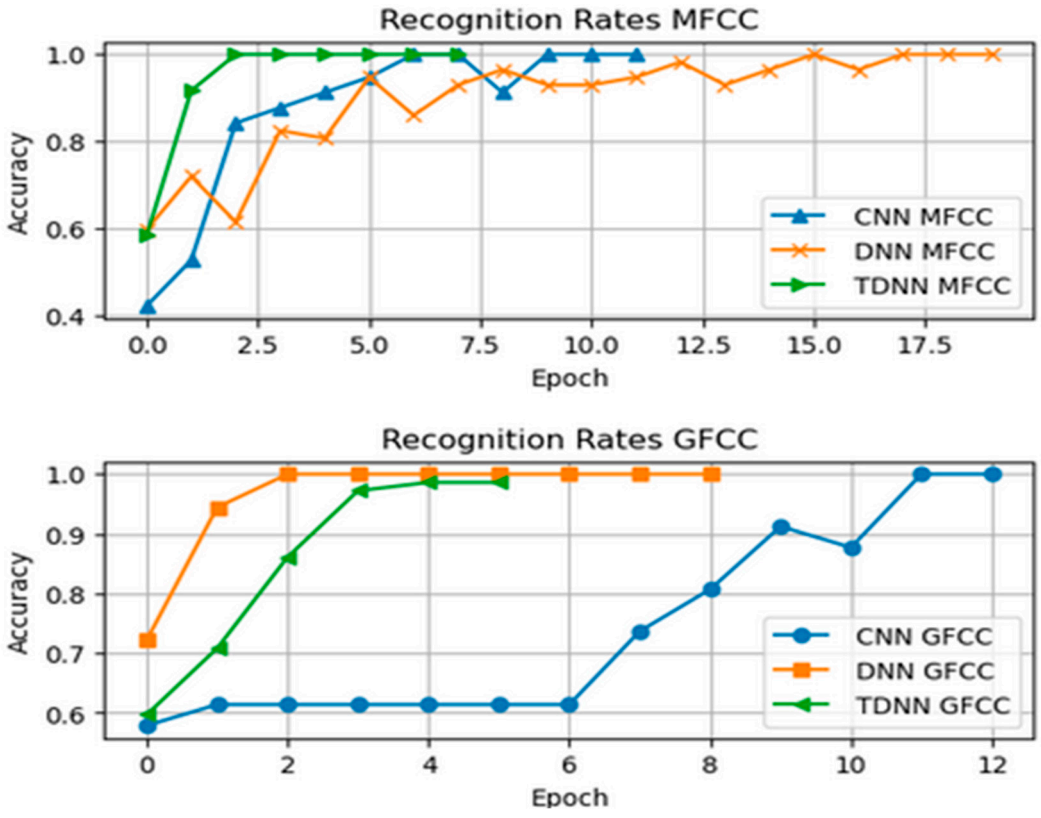

Figure 16. Recognition Rates of the Neural Networks using MFCC and GFCC combined features.

Figure 17 reinforces accuracy findings by showing how the loss function changes over epochs. The loss aligns with accuracy, decreasing rapidly initially. However, like accuracy, it gradually levels off after a certain number of epochs, highlighting the ideal training duration. Early stopping effectively curbed loss deviation caused by overfitting.

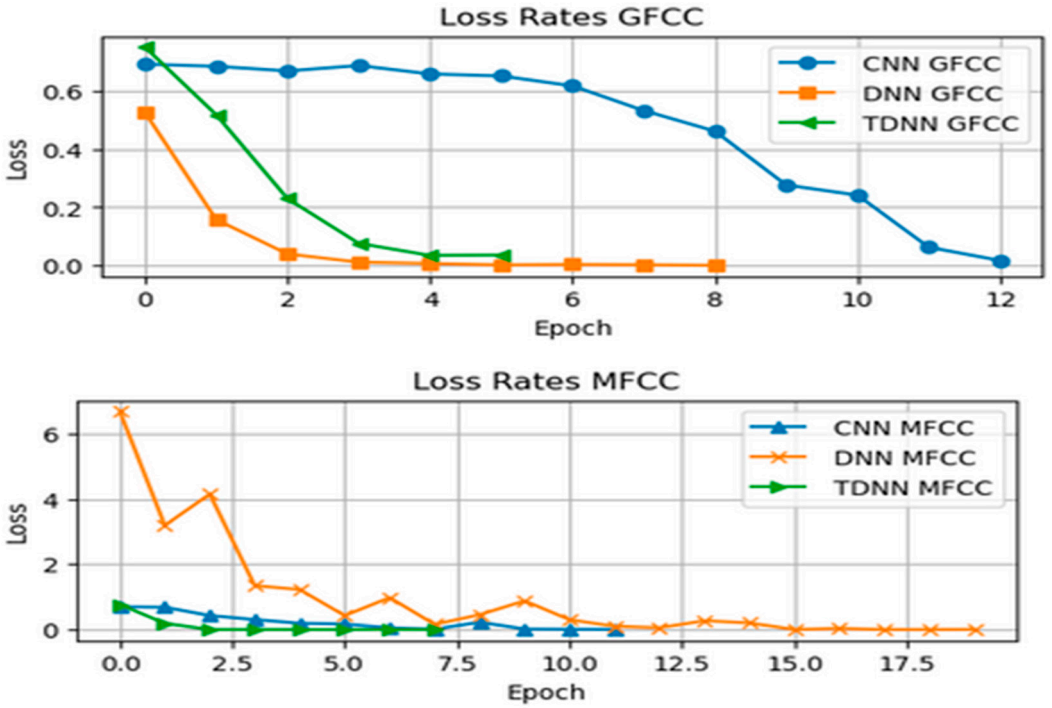

Figure 17. Loss Rates of the Neural Networks using MFCC and GFCC combined features.

The performance of the six models was graphically presented for easy comparison. Remarkably, all six models consistently reached a final training accuracy of 100%, as seen in Figure 18 (Recognition Rate) and Figure 19 (Loss Function Value).

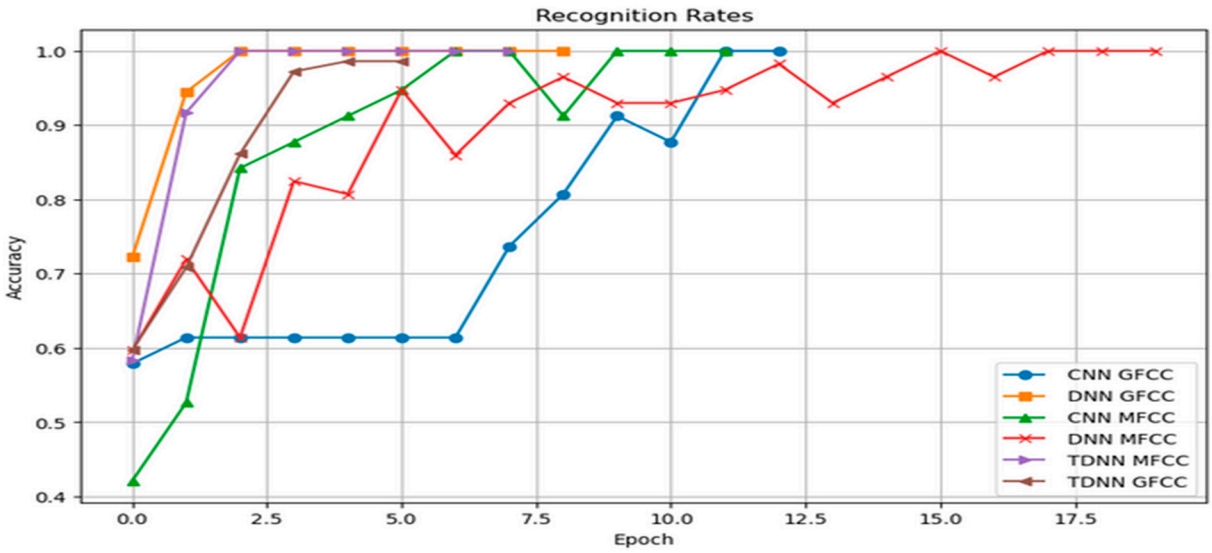

Figure 18. Recognition Rates of Neural Network Models for Active Speaker Classification.

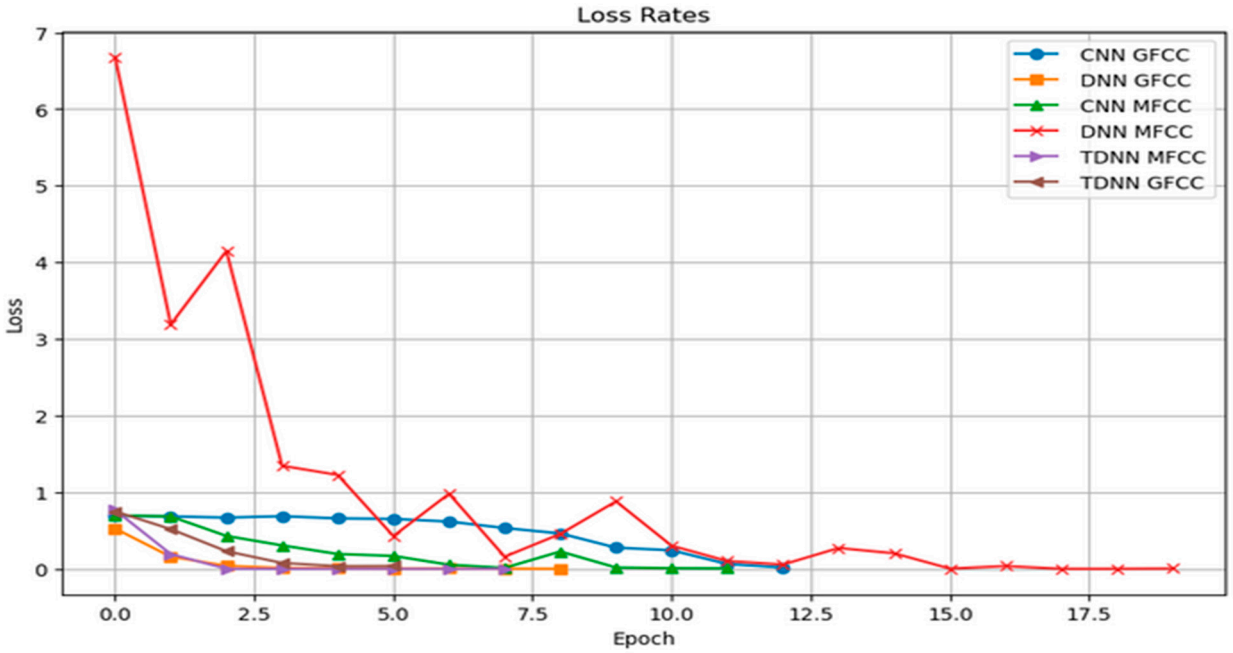

Figure 19. Loss Rates of Neural Network Models for Active Speaker Classification.

The models' high accuracy demonstrates their effectiveness in recognizing unique speaker traits from input features. Table 2 summarizes our thesis results. TDNN excelled in speaker classification compared to Conv-LSTM CNNs and traditional DNNs. Its specialized temporal sequence modeling effectively captured complex speech patterns, enhancing speaker discrimination accuracy.

Table 2. Summary of results of all six models trained across their testing Recognition Rates and Loss Function Values along with the features used during preprocessing.

	Training Parameters	Recognition Rate (in %)	Loss Function Value	Combined Features Used (MFCC/GFCC)
Conv-LSTM	824,322	93.75	0.1989	MFCC
Dense Neural Network	429,936	87.5	0.6421	MFCC
Time Delay Neural Network	**88,322**	**100**	**0.0003**	**MFCC**
Conv-LSTM	824,322	90.625	0.2838	GFCC
Dense Neural Network	429,936	93.75	0.1724	GFCC
Time Delay Neural Network	**88,322**	**93.75**	**0.1666**	**GFCC**

Optimizer = Adam; Learning rate = 0.0001; Loss = Categorical cross entropy; Metrics = Accuracy; Training iterations = 20.

The use of early stopping played a key role in achieving these results by ending training at 20 epochs, preventing overfitting by monitoring performance on a validation set. The models' ability to stop early highlights their fast learning and efficient parameter optimization. The thorough evaluation of the six models (Conv-LSTM, TDNN, and DNN) using both MFCC and GFCC features yielded impressive outcomes. Perfect accuracies within a limited number of epochs showcase their ability to capture intricate speaker traits. Different convergence speeds emphasize each architecture's efficiency with specific features. Early stopping safeguards against overfitting, enhancing model reliability. This highlights deep learning models' potential for speaker classification while stressing the

importance of model selection based on feature characteristics. Future research could explore interpretability and robustness in real-world scenarios.

4.2. Audio Ellipsoidal-HoloSLAM Algorithm

This study utilizes the Seeed ReSpeaker Core v2.0 microphone, tailored for voice interface tasks, with a 16 kHz sampling rate. Operating on the GNU/Linux system via the Arduino device, the ReSpeaker microphone array connects to the Nao robot's USB port [86]. Integrated into the Nao robot head alongside Microsoft's HoloLens, the microphone array is depicted in Figure 20.

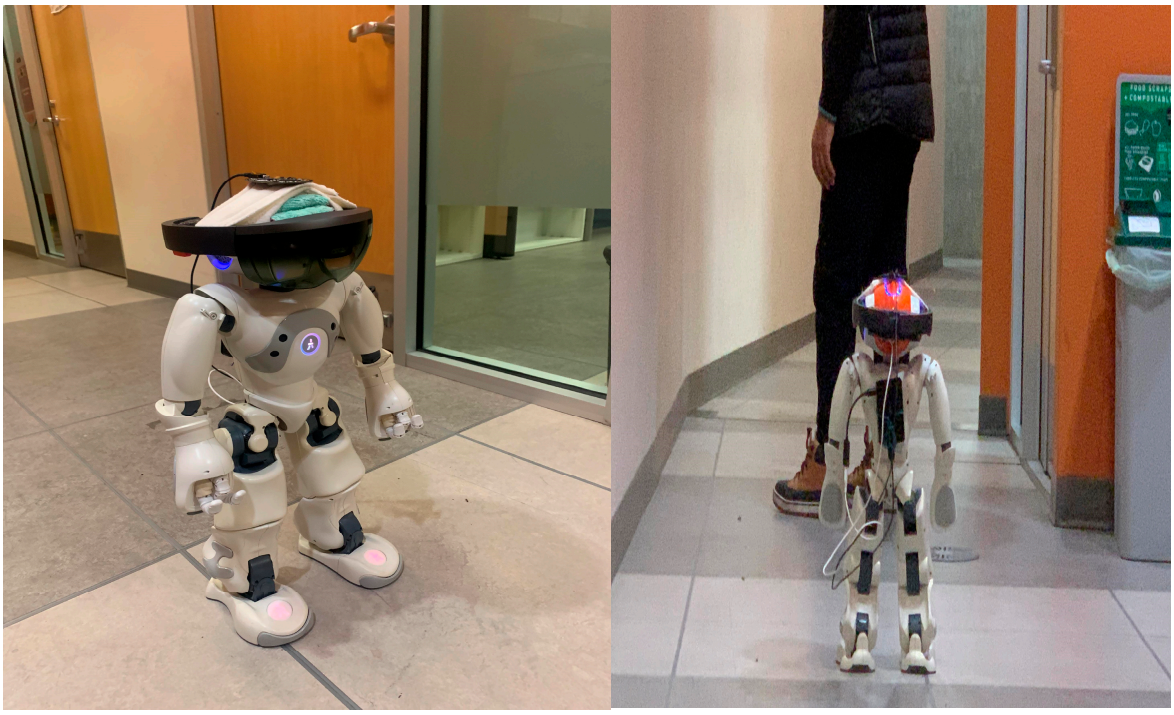

Figure 20. The integrated system into the Nao robot head alongside Microsoft's HoloLens and the microphone array.

Table 3 displays some parameters of the Nao robot utilized in these experiments. For further information regarding this robot, please refer to [38].

When an individual speaks within the vicinity of the microphone array, the microphones capture the sound, which is subsequently transmitted to the onboard ADC on the ReSpeaker. The resulting data are then processed on the Raspberry Pi board. Numerous experiments have been undertaken with various speakers positioned at different distances to calibrate the ReSpeaker microphone array and determine the ideal angle for effective communication with the robot. This initial investigation provided early observations on humans' ability to gauge the direction of a voice, even in situations with clear speech angles, such as 0°. Evidently, someone positioned directly facing the ReSpeaker (i.e., 0° off-axis) would be regarded as the optimal orientation. The ReSpeaker is capable of capturing sound within a 5-m range. The variations in DOA estimation of the microphone pairs are consolidated to a median value, yielding a unified live DOA output. This illustrates that our microphone array system is capable of precisely determining the sound source location. The ReSpeaker microphone array system can pinpoint the sound source with an average deviation of 5 degrees. Such precise localization enables robots to effectively perceive their surroundings and make informed navigational decisions.

Table 3. Nao robot parameters.

Specification	Details
Height	58 cm (22.8 inches)
Weight	4.3 kg (9.5 lbs.)
Degrees of Freedom	25
Sensors	-two HD cameras
	-four microphones
	-Touch sensors (head, hands, feet)
	-Inertial measurement unit (IMU)
	-Ultrasonic sensors
Processing Unit	Intel Atom Z530 processor
Memory	1 GB RAM
Operating System	Linux-based NAOqi OS
Connectivity	-Ethernet
	-Wi-Fi
	-Bluetooth
Power Source	Rechargeable lithium-ion battery
Battery Life	Up to 90 min of continuous operation
Development Framework	Choregraphe (graphical programming software)
	Python SDK
	C++ SDK

In audio-based navigation systems, the utilization of multiple microphones strategically placed throughout the indoor environment is paramount for accurate localization and mapping by the robot. With prior knowledge of the microphones' locations, the robot can triangulate sound sources more effectively, thereby improving the accuracy of its localization capabilities.

However, the task of localizing the robot and mapping its environment becomes significantly more challenging and impossible when fewer microphones are employed, especially in the absence of prior knowledge regarding their locations. This can lead to inaccuracies in localization and mapping, potentially resulting in navigation errors and reduced overall performance.

The Microsoft HoloLens, with its Mixed Reality-based Ellipsoidal SLAM (**HoloSLAM**), facilitates the placement of virtual landmarks if needed within the robot's environment via the Holo-landmark hologram app when necessary. The Virtual Landmark App offers a variety of three distinct virtual landmark types to choose from, including diamonds, spheres, and cubes.

This innovative technology enables the robot to interact with these landmarks dynamically, offering functionalities such as moving up, moving down, and moving right. Moreover, the robot possesses the capability to modify these virtual landmarks in real-time through voice commands. This feature empowers the robot to adapt and customize its audio environment according to changing requirements or unforeseen circumstances swiftly. This ultimately enhances its navigational capabilities and overall functionality.

The experiments detailed in this paper were conducted utilizing the Nao robot inside an SFU campus environment. The main goal of the experiment is to assess the effectiveness of audio-based virtual HoloSLAM in estimating the robot's position and mapping its environment. This evaluation specifically focuses on the robot's performance when equipped with a single microphone array designed to track the active speaker. The target or

active individual traverses a realistic path scenario, with the robot tracking and following their voices.

In this experiment, the robot first identifies the active speaker and their direction, then turns toward them. Subsequently, it places random virtual landmarks in space, capturing images and removing them from their surroundings, as depicted in Figure 21. These virtual objects are consistently positioned 2 m toward the speaker. The virtual landmarks are then utilized to complete the SLAM process.

Figure 21. The audio-based virtual HoloSLAM robot environments.

The robot is assigned the task of identifying the active speaker through a combination of pre-trained models and subsequently mapping its environment and localizing itself within it autonomously, without prior knowledge or human intervention. The robot is programmed not to follow the speaker unless a recognition rate of 93.75% is achieved by six models (GMM, SVM, Conv-LSTM, DNN, and TDNN) utilizing both MFCC and GFCC features, ensuring accurate identification of the correct active speaker.

After identifying the active speaker using a hybrid pre-trained model, the robot utilizes ReSpeaker's sound source localization to estimate the speaker's direction. This angle guides the robot in orienting itself towards the speaker. Through a virtual hologram app, the robot instructs the HoloLens to place random virtual landmarks within its environment, depicted in Figure 21. These landmarks are consistently positioned 2 m along the x-axis of the Nao robot. This capability enables the robot to precisely position and remove virtual landmarks as needed, facilitating real-time communication with them. Moreover, it grants the robot greater autonomy and control over its mapping process. Consequently, even if the robot's sensors fail to detect any landmarks during observation, the audio-based SLAM algorithm remains reliable.

Figures 22 and 23 provide visual representations of the estimated robot position and virtual landmarks obtained using the audio-based virtual Ellipsoidal-HoloSLAM system.

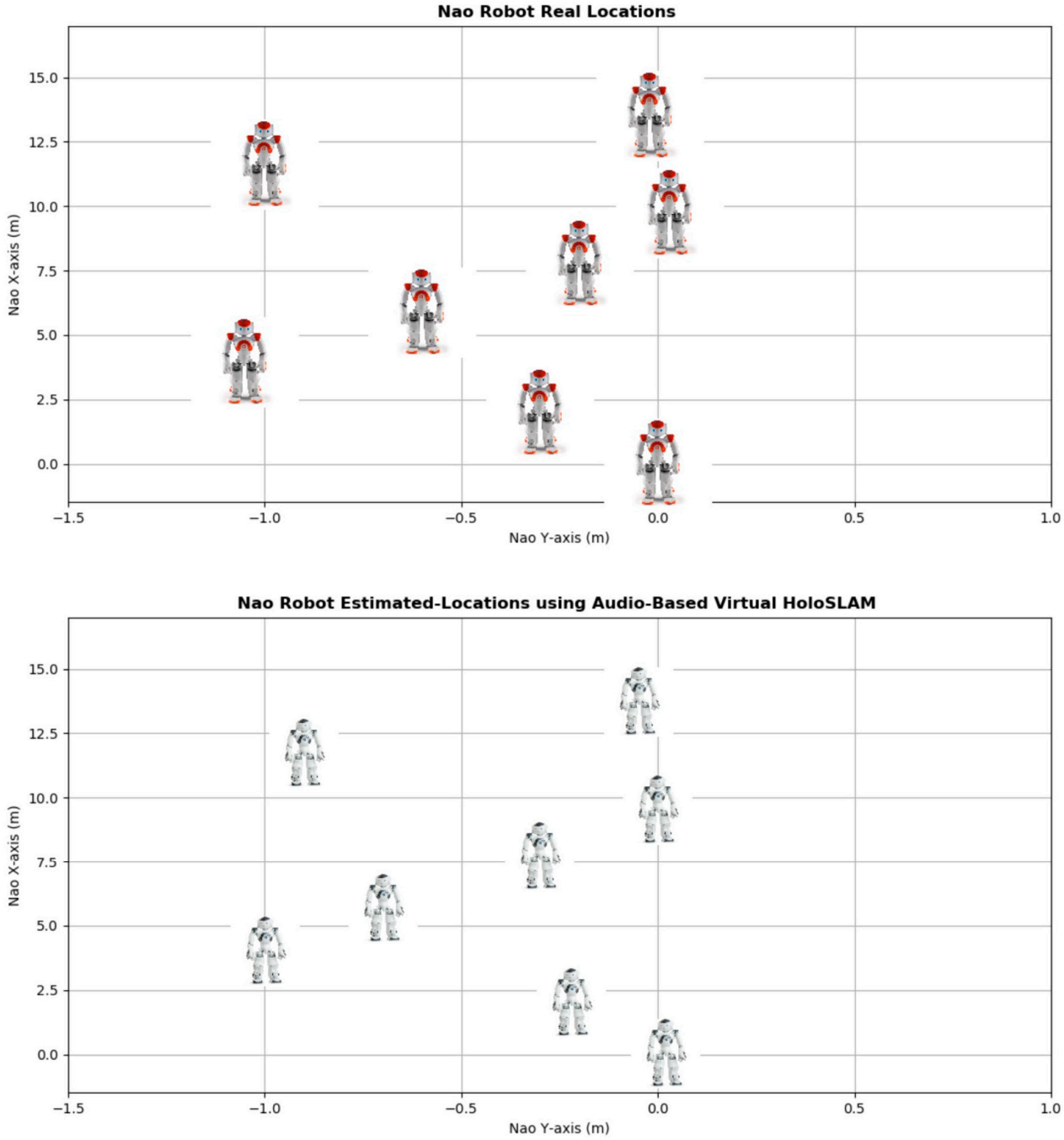

Figure 22. Estimated Nao robot locations using audio-based HoloSLAM.

Upon comparing the estimated positions with the actual robot position, it becomes evident that the newly implemented SLAM technology has adeptly tracked the robot's movement and accurately constructed virtual landmarks with minimal margin for error.

Table 4 offers an intricate breakdown of outcomes garnered from real-time experiments aimed at assessing the efficiency of the implemented system. These findings offer valuable insights into the overarching performance of the audio-driven virtual Ellipsoidal HoloSLAM algorithm, particularly regarding its accuracy and reliability. Additionally, the evaluation extends to scrutinizing the estimated positions of the virtual landmarks.

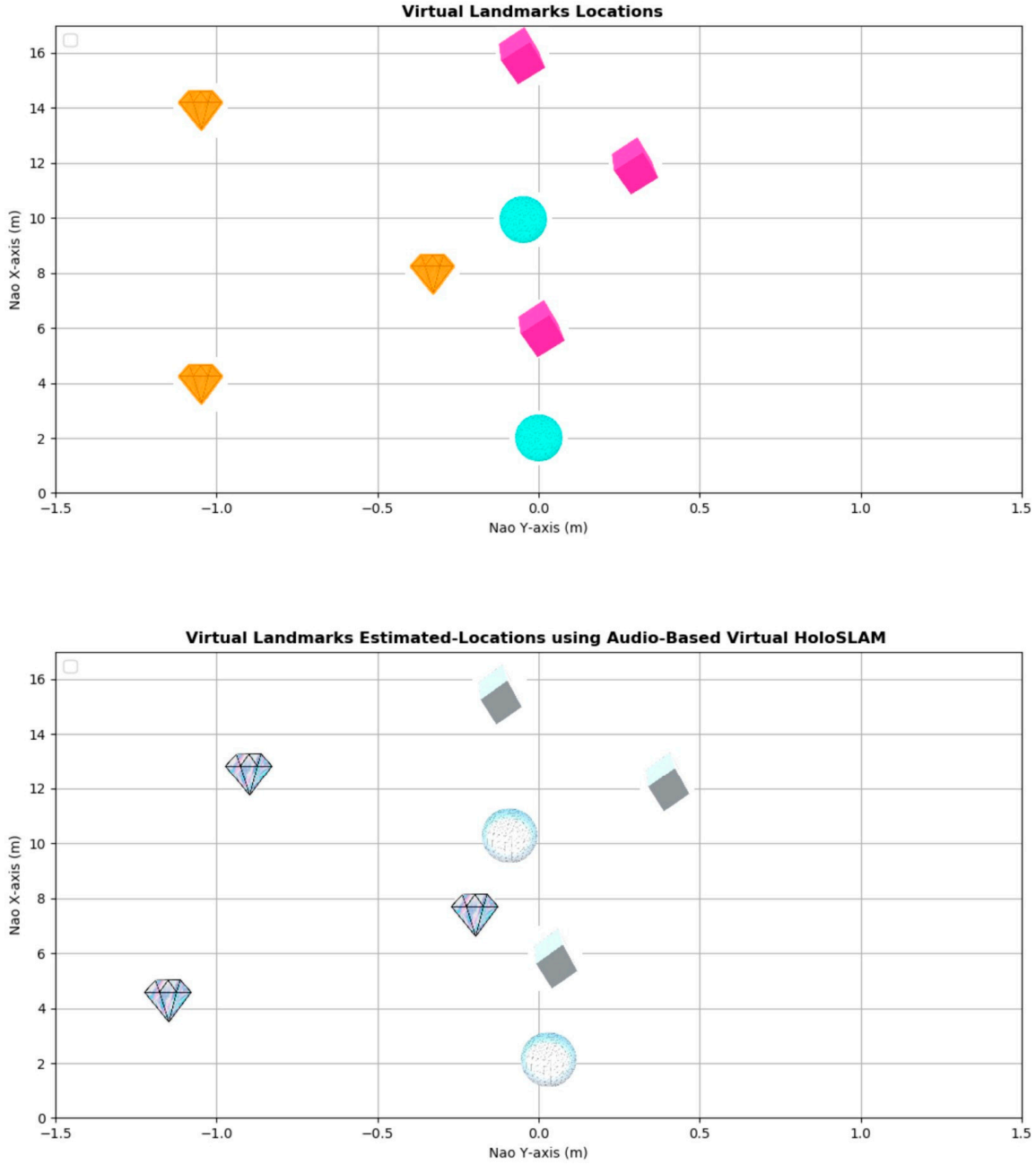

Figure 23. Estimated virtual landmark location using audio-based HoloSLAM.

Table 4. Analyzing the performance of implemented audio-based virtual HoloSLAM.

Algorithm	Nao Position Error/m	Nao Orientation Error/rad	Virtual Landmarks Error/m
Nao IMU	33.01	0.675	
Audio-based Ellipsoidal Virtual HoloSLAM	0.0184	0.119	0.010
Total Times of identification speaker called		23	

The data within the table highlight a notable observation: the IMU of the Nao exhibits the most significant error concerning both the robot's position and orientation. However, through the utilization of Ellipsoidal HoloSLAM, these errors are mitigated, showcasing

a substantial improvement in localization accuracy. Notably, the algorithm consistently succeeds in accurately localizing and tracking the active speaker and building a map of the unknown environment at each iteration, indicative of its robustness and efficacy in real-time scenarios.

The new audio-based SLAM enables precise estimation of the robot's position and meticulous mapping of its virtual environment at each stage of operation. However, a notable bottleneck emerged during active speaker identification, prolonging the process and highlighting the imperative for optimization to augment the model's operational speed and efficiency.

5. Discussion

This paper advances the current state-of-the-art in audio-based SLAM by introducing a novel integration of a microphone array platform with Microsoft HoloLens, a robotic mixed reality device. The approach eliminates the need to prepare the robot environment with multiple audio sources and audio landmarks to perform complete and successful audio-based SLAM in indoor environments. This approach can operate with a single audio source and a solitary microphone array, ensuring precise localization of both the audio sources and the robot. The study utilizes a pre-trained or voice-printed speaker as the target audio source for the robot to follow and interact with. It also facilitates the mapping of audio landmarks to the robot's environment, addressing the challenges associated with multiple audio sources and landmarks in indoor settings. Additionally, this approach successfully maps the environment with virtual landmarks, providing a comprehensive solution to the complexities associated with audio-based SLAM in indoor settings.

6. Conclusions

In this study, we proposed an audio-based SLAM system integrated with the Microsoft HoloLens mixed reality device to enhance the capabilities of intelligent robots. The main objective of this system is to conduct audio-based SLAM with minimal auditory requirements, presenting a novel perspective through the HoloLens and robotic mixed reality concept. The proposed system operates in several stages. Firstly, it leverages the audio features to identify a unique speaker by employing pre-registered voiceprints through deep learning in a multi-audio environment, utilizing a microphone array. The extracted audio is then utilized to estimate the direction of the speaker. Subsequently, the robot utilizes this estimated direction to track the active speaker while simultaneously localizing itself and generating a map of its surroundings. Due to the limited availability of audio landmarks, the Ellipsoidal HoloSLAM incorporates virtual landmarks into the mapping process. This inclusion allows for an accurate and realistic SLAM implementation without the need for prior knowledge of sound source locations. As the robot moves and the location and direction of the active speaker change, the implemented audio HoloSLAM algorithm continuously updates the robot's position and orientation within the built map. This enables the robot to dynamically follow the speaker and simultaneously construct a detailed virtual map of the environment.

A comparative analysis with state-of-the-art audio-based SLAM systems revealed that the audio HoloSLAM achieved more accurate trajectories for the robot without the addition of extra sensors or reliance on additional audio landmarks or pre-known locations of audio sources. Real-world experiments were conducted to validate the implemented audio HoloSLAM system. The results demonstrated that the audio-based virtual HoloSLAM algorithm successfully mapped the environment and exhibited a more robust robot trajectory. The system accurately estimated the robot's position at each movement with minimal errors. This approach exhibits significant potential in various indoor applications, including human–robot interaction, assistive robotics, and indoor navigation. The successful integration of the Microsoft HoloLens mixed reality device with audio-based SLAM opens up new possibilities for enhancing the spatial awareness and interaction capabilities of intelligent robots in various environments.

Author Contributions: Conceptualization: E.S.F.L. and A.R.; methodology: E.S.F.L. and A.R.; software: E.S.F.L.; validation, E.S.F.L. and A.R.; formal analysis: E.S.F.L. and A.R.; investigation: E.S.F.L. and A.R.; resources: A.R.; data curation, E.S.F.L. and A.R.; writing—original draft preparation, E.S.F.L.; writing—review and editing, A.R.; visualization, E.S.F.L. and A.R.; supervision, A.R.; project administration, A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be available upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Javaid, M.; Haleem, A.; Singh, R.P.; Suman, R. Substantial capabilities of robotics in enhancing Industry 4.0 implementation. *Cogn. Robot.* **2021**, *1*, 58–75. [CrossRef]
2. Nilsson, N.J.; Park, M. A Mobile Automaton: An Application of Artificial Intelligence Techniques. In Proceedings of the International Joint Conference on Artificial, Washington, DC, USA, 7–9 May 1969.
3. Yasuda, Y.D.V.; Martins, L.E.G.; Cappabianco, F.A.M. Autonomous Visual Navigation for Mobile Robots: A systematic literature review. *ACM Comput. Surv.* **2020**, *53*, 1–34. [CrossRef]
4. Campos-Macías, L.; Aldana-López, R.; de la Guardia, R.; Parra-Vilchis, J.I.; Gómez-Gutiérrez, D. Autonomous navigation of MAVs in unknown cluttered environments. *J. Field Robot.* **2021**, *38*, 307–326. [CrossRef]
5. Taheri, H.; Xia, Z.C. SLAM; definition and evolution. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104032. [CrossRef]
6. Alsadik, B.; Karam, S. The Simultaneous Localization and Mapping (SLAM)—An Overview. *J. Appl. Sci. Technol. Trends* **2021**, *2*, 147–158. [CrossRef]
7. Takleh, T.T.O.; Abu Bakar, N.; Rahman, S.A.; Hamzah, R.; Aziz, Z.A. A Brief Survey on SLAM Methods in Autonomous Vehicle. *Int. J. Eng. Technol.* **2018**, *7*, 38. [CrossRef]
8. Basilico, N. Recent Trends in Robotic Patrolling. *Curr. Robot. Rep.* **2022**, *3*, 65–76. [CrossRef]
9. Panigrahi, P.K.; Bisoy, S.K. Localization strategies for autonomous mobile robots: A review. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 6019–6039. [CrossRef]
10. Munguía, R.; Grau, A. Concurrent Initialization for Bearing-Only SLAM. *Sensors* **2010**, *10*, 1511–1534. [CrossRef]
11. Lahemer, E.S.; Rad, A. An Adaptive Augmented Vision-Based Ellipsoidal SLAM for Indoor Environments. *Sensors* **2019**, *19*, 2795. [CrossRef]
12. Tourani, A.; Bavle, H.; Sanchez-Lopez, J.L.; Voos, H. Visual SLAM: What Are the Current Trends and What to Expect? *Sensors* **2022**, *22*, 9297. [CrossRef] [PubMed]
13. Barros, A.M.; Michel, M.; Moline, Y.; Corre, G.; Carrel, F. A Comprehensive Survey of Visual SLAM Algorithms. *Robotics* **2022**, *11*, 24. [CrossRef]
14. Keyrouz, F. Advanced Binaural Sound Localization in 3-D for Humanoid Robots. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 2098–2107. [CrossRef]
15. Wang, C.-C.; Lin, C.-H.; Hu, J.-S. Probabilistic Structure from Sound. *Adv. Robot.* **2009**, *23*, 1687–1702. [CrossRef]
16. Risoud, M.; Hanson, J.-N.; Gauvrit, F.; Renard, C.; Lemesre, P.-E.; Bonne, N.-X.; Vincent, C. Sound source localization. *Eur. Ann. Otorhinolaryngol. Head Neck Dis.* **2018**, *135*, 259–264. [CrossRef] [PubMed]
17. Trowitzsch, I.; Schymura, C.; Kolossa, D.; Obermayer, K. Joining Sound Event Detection and Localization Through Spatial Segregation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 487–502. [CrossRef]
18. Sun, Y.; Chen, J.; Yuen, C.; Rahardja, S. Indoor Sound Source Localization with Probabilistic Neural Network. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6403–6413. [CrossRef]
19. Lee, R.; Kang, M.-S.; Kim, B.-H.; Park, K.-H.; Lee, S.Q.; Park, H.-M. Sound Source Localization Based on GCC-PHAT With Diffuseness Mask in Noisy and Reverberant Environments. *IEEE Access* **2020**, *8*, 7373–7382. [CrossRef]
20. Nadiri, O.; Rafaely, B. Localization of Multiple Speakers under High Reverberation using a Spherical Microphone Array and the Direct-Path Dominance Test. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 1494–1505. [CrossRef]
21. Wang, D.; Brown, G.J. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*; Wiley-IEEE Press: Hoboken, NJ, USA, 2006. [CrossRef]
22. Liaquat, M.U.; Munawar, H.S.; Rahman, A.; Qadir, Z.; Kouzani, A.Z.; Mahmud, M.A.P. Localization of Sound Sources: A Systematic Review. *Energies* **2021**, *14*, 3910. [CrossRef]
23. Su, D.; Vidal-Calleja, T.; Miro, J.V. Simultaneous asynchronous microphone array calibration and sound source localisation. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015. [CrossRef]

24. Chen, X.; Sun, H.; Zhang, H. A New Method of Simultaneous Localization and Mapping for Mobile Robots Using Acoustic Landmarks. *Appl. Sci.* **2019**, *9*, 1352. [CrossRef]
25. Qiu, W.; Wang, G.; Zhang, W. Acoustic SLAM Based on the Direction-of-Arrival and the Direct-to-Reverberant Energy Ratio. *Drones* **2023**, *7*, 120. [CrossRef]
26. Zhao, J.; Zhang, G.; Qu, J.; Chen, J.; Liang, S.; Wei, K.; Wang, G. A Sound Source Localization Method Based on Frequency Divider and Time Difference of Arrival. *Appl. Sci.* **2023**, *13*, 6183. [CrossRef]
27. Thai, D.Z.; Hashemi-sakhtsari, A.; Pattison, T. *Speaker Localisation Using Time Difference of Arrival*; Technical Report (Defence Science and Technology Organisation (Australia)); DSTO: Edinburgh, Australia, 2008; pp. 1–6.
28. Knapp, C.; Carter, G. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust. Speech Signal Process.* **1976**, *24*, 320–327. [CrossRef]
29. Meng, L.; Li, X.H.; Zhang, W.G.; Liu, D.Z. The Generalized Cross-Correlation Method for Time Delay Estimation of Infrasound Signal. In Proceedings of the 2015 Fifth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, China, 18–20 September 2015; pp. 1320–1323. [CrossRef]
30. Evers, C.; Naylor, P.A. Acoustic SLAM. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1484–1498. [CrossRef]
31. O'Reilly, J.; Cirstea, S.; Cirstea, M.; Zhang, J. A Novel Development of Acoustic SLAM. In Proceedings of the 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Istanbul, Turkey, 27–29 August 2019; pp. 525–531. [CrossRef]
32. Hu, J.S.; Chan, C.Y.; Wang, C.K.; Lee, M.T.; Kuo, C.Y. Simultaneous Localization of a Mobile Robot and Multiple Sound Sources Using a Microphone Array. *Adv. Robot.* **2011**, *25*, 135–152. [CrossRef]
33. Valin, J.-M.; Michaud, F.; Rouat, J. Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Robot. Auton. Syst.* **2007**, *55*, 216–228. [CrossRef]
34. Narang, G.; Nakamura, K.; Nakadai, K. Auditory-aware navigation for mobile robots based on reflection-robust sound source localization and visual SLAM. In Proceedings of the2014 IEEE International Conference on Systems, Man and Cybernetics, San Diego, CA, USA, 5–8 October 2014. [CrossRef]
35. Milgram, P.; Fumio, K. A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Inf. Syst.* **2003**, *E77-D*, 1321–1329.
36. Flavián, C.; Ibáñez-Sánchez, S.; Orús, C. The impact of virtual, augmented and mixed reality technologies on the customer experience. *J. Bus. Res.* **2019**, *100*, 547–560. [CrossRef]
37. Vroegop, D. *Microsoft HoloLens Developer's Guide*; Packt Publishing: Birmingham, UK, 2017. Available online: https://learning.oreilly.com/library/view/microsoft-hololens-developers/9781786460851/ (accessed on 1 January 2020).
38. Gelin, R. NAO. In *Humanoid Robotics: A Reference*; Goswami, A., Vadakkepat, P., Eds.; Springer: Dordrecht, The Netherlands, 2019; pp. 147–168. ISBN 978-94-007-6046-2.
39. Al-Qaderi, M.; Lahamer, E.; Rad, A. A Two-Level Speaker Identification System via Fusion of Heterogeneous Classifiers and Complementary Feature Cooperation. *Sensors* **2021**, *21*, 5097. [CrossRef]
40. Reynolds, D.A.; Quatieri, T.F.; Dunn, R.B. Speaker Verification Using Adapted Gaussian Mixture Models. *Digit. Signal Process.* **2000**, *10*, 19–41. [CrossRef]
41. Jakkula, V. Tutorial on Support Vector Machine (SVM). School of EECS, Washington State University. 2011, pp. 1–13. Available online: http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf (accessed on 1 January 2020).
42. El-Moneim, S.A.; Sedik, A.; Nassar, M.A.; El-Fishawy, A.S.; Sharshar, A.M.; Hassan, S.E.A.; Mahmoud, A.Z.; Dessouky, M.I.; El-Banby, G.M.; El-Samie, F.E.A.; et al. Text-dependent and text-independent speaker recognition of reverberant speech based on CNN. *Int. J. Speech Technol.* **2021**, *24*, 993–1006. [CrossRef]
43. Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K. Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 328–339. [CrossRef]
44. Nakadai, K.; Lourens, T.; Okuno, H.G.; Kitano, H. Active Audition for Humanoid. In Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, AAAI 2000, Austin, TX, USA, 30 July–3 August 2000.
45. Rascon, C.; Meza, I. Localization of sound sources in robotics: A review. *Robot. Auton. Syst.* **2017**, *96*, 184–210. [CrossRef]
46. Desai, D.; Mehendale, N. A Review on Sound Source Localization Systems. *SSRN Electron. J.* **2021**, *29*, 4631–4642. [CrossRef]
47. Argentieri, S.; Danès, P.; Souères, P. A survey on sound source localization in robotics: From binaural to array processing methods. *Comput. Speech Lang.* **2015**, *34*, 87–112. [CrossRef]
48. Flynn, A.M.; Brooks, R.A.; Wells, W.M., III; Barrett, D.S. *Squirt: The Prototypical Mobile Robot for Autonomous Graduate Students*; DTIC: Fort Belvoir, VA, USA, 1989.
49. Irie, R.E.; Brooks, R.A.; Morgenthaler, F.R. Robust Sound Localization: An Application of an Auditory Perception System for a Humanoid Robot. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.
50. Liu, M.; Zeng, Q.; Jian, Z.; Peng, Y.; Nie, L. A sound source localization method based on improved second correlation time delay estimation. *Meas. Sci. Technol.* **2023**, *34*, 045102. [CrossRef]
51. Klee, U.; Gehrig, T.; McDonough, J. Kalman Filters for Time Delay of Arrival-Based Source Localization. *EURASIP J. Adv. Signal Process.* **2006**, *2006*, 012378. [CrossRef]

52. Kallakuri, N.; Even, J.; Morales, Y.; Ishi, C.; Hagita, N. Probabilistic approach for building auditory maps with a mobile microphone array. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013. [CrossRef]
53. Zhong, X.; Hopgood, J.R. Particle filtering for TDOA based acoustic source tracking: Nonconcurrent Multiple Talkers. *Signal Process.* **2014**, *96*, 382–394. [CrossRef]
54. Ogiso, S.; Kawagishi, T.; Mizutani, K.; Wakatsuki, N.; Zempo, K. Self-localization method for mobile robot using acoustic beacons. *ROBOMECH J.* **2015**, *2*, 1364. [CrossRef]
55. Lee, B.-G.; Choi, J.; Kim, D.; Kim, M. Sound source localization in reverberant environment using visual information. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010), Taipei, Taiwan, 18–22 October 2010. [CrossRef]
56. Ham, I.; Asano, F.; Kawai, Y.; Kanchiro, F.; Yamamoto, K.; Asoh, H.; Ogata, J.; Ichintura, N.; Hirukawa, H. Robust speech interface based on audio and video information fusion for humanoid HRP-2. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004. [CrossRef]
57. Sasaki, Y.; Kagami, S.; Mizoguchi, H. Multiple Sound Source Mapping for a Mobile Robot by Self-motion Triangulation. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006. [CrossRef]
58. Dokmanić, I.; Parhizkar, R.; Walther, A.; Lu, Y.M.; Vetterli, M. Acoustic echoes reveal room shape. *Proc. Natl. Acad. Sci. USA* **2013**, *110*, 12186–12191. [CrossRef]
59. Gentner, C.; Jost, T. Indoor positioning using time difference of arrival between multipath components. In Proceedings of the 2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Montbeliard, France, 28–31 October 2013. [CrossRef]
60. Antonacci, F.; Filos, J.; Thomas, M.R.P.; Habets, E.A.P.; Sarti, A.; Naylor, P.A.; Tubaro, S. Inference of Room Geometry From Acoustic Impulse Responses. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 2683–2695. [CrossRef]
61. Ureña, J.; Hernández, A.; Jiménez, A.; Villadangos, J.; Mazo, M.; García, J.; Álvarez, F.; De Marziani, C.; Pérez, M.; Seco, F. Advanced sensorial system for an acoustic LPS. *Microprocess. Microsyst.* **2007**, *31*, 393–401. [CrossRef]
62. Lahemer, E.S.; Rad, A. HoloSLAM: A novel approach to virtual landmark-based SLAM for indoor environments. *Complex Intell. Syst.* **2024**, 1–26. [CrossRef]
63. SoftBank Robotics, Nao Humanoid Robot. Available online: https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/ (accessed on 1 January 2020).
64. ReSpeaker Mic Array v2.0. Available online: https://wiki.seeedstudio.com/ReSpeaker_Mic_Array/ (accessed on 25 April 2024).
65. Valin, J.-M.; Michaud, F.; Rouat, J.; Letourneau, D. Robust sound source localization using a microphone array on a mobile robot. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27–31 October 2003. [CrossRef]
66. Valencia-Palma, A.; Córdova-Esparza, D.M. Sound Source Localization Using Beamforming and Its Representation in a Mixed Reality Embedded Device. In *Pattern Recognition*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2019. [CrossRef]
67. Jang, Y.; Kim, J.; Kim, J. The development of the vehicle sound source localization system. In Proceedings of the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Hong Kong, China, 16–19 December 2016. [CrossRef]
68. Brandstein, M.; Silverman, H. A robust method for speech signal time-delay estimation in reverberant rooms. In Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, 21–24 April 1997. [CrossRef]
69. Li, X.; Liu, H.; Yang, X. Sound source localization for mobile robot based on time difference feature and space grid matching. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011), San Francisco, CA, USA, 25–30 September 2011. [CrossRef]
70. Hornstein, J.; Lopes, M.; Santos-Victor, J.; Lacerda, F. Sound Localization for Humanoid Robots—Building Audio-Motor Maps based on the HRTF. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006. [CrossRef]
71. Nakadai, K.; Hidai, K.-I.; Okuno, H.G.; Kitano, H. Real-time multiple speaker tracking by multi-modal integration for mobile robots. In Proceedings of the 7th European Conference on Speech Communication and Technology (EuroSpeech 2001), Aalborg, Denmark, 3–7 September 2001. [CrossRef]
72. Bray, B.; Zeller, M.; Schonning, N. What Is Mixed Reality? Microsoft. 2018. Available online: https://docs.microsoft.com/en-us/windows/mixed-reality/mixed-reality (accessed on 12 February 2024).
73. Alexandrea, P. Top 12 Best 3D Software for Beginners. 2019. Available online: https://www.3dnatives.com/en/3d-software-beginners100420174/ (accessed on 1 April 2019).
74. Mariani, J. *Spoken Language Processing*; ISTE Ltd.: London, UK, 2010; ISBN 9781848210318. [CrossRef]
75. Bai, Z.; Zhang, X.-L. Speaker recognition based on deep learning: An overview. *Neural Netw.* **2021**, *140*, 65–99. [CrossRef]
76. Naik, J. Speaker verification: A tutorial. *IEEE Commun. Mag.* **1990**, *28*, 42–48. [CrossRef]

77. Jahangir, R.; Teh, Y.W.; Nweke, H.F.; Mujtaba, G.; Al-Garadi, M.A.; Ali, I. Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges. *Expert Syst. Appl.* **2021**, *171*, 114591. [CrossRef]
78. Sharma, G.; Umapathy, K.; Krishnan, S. Trends in audio signal feature extraction methods. *Appl. Acoust.* **2019**, *158*, 107020. [CrossRef]
79. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
80. Altalbe, A. RETRACTED ARTICLE: Audio fingerprint analysis for speech processing using deep learning method. *Int. J. Speech Technol.* **2022**, *25*, 575–581. [CrossRef]
81. Staroniewicz, P.; Majewski, W. SVM Based Text-Dependent Speaker Identification for Large Set of Voices. In Proceedings of the European Signal Processing Conference, Nice, France, 31 August–4 September 2015.
82. Jawarkar, N.P. Speaker Identification in Noisy Environment. *Int. J. Curr. Eng. Sci. Res.* **2017**, *4*, 37–43.
83. Abeßer, J. A Review of Deep Learning Based Methods for Acoustic Scene Classification. *Appl. Sci.* **2020**, *10*, 2020. [CrossRef]
84. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. *The Kaldi Speech Recognition Toolkit*; IEEE Signal Processing Society: Piscataway, NJ, USA, 2011.
85. Tchistiakova. Time Delay Neural Network. Available online: https://kaleidoescape.github.io/tdnn (accessed on 20 October 2023).
86. Nao Documentation. Available online: http://doc.aldebaran.com/2-8/home_nao.html (accessed on 12 February 2024).