Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ens.ewi.tudelft.nl/

SONY

M.Sc. Thesis

Room geometry estimation from stereo recordings using neural networks

Giovanni Bologni B.Sc.

Abstract

Acoustic room geometry estimation is often performed in ad hoc settings, i.e., using multiple microphones and sources distributed around the room, or assuming control over the excitation signals. To facilitate practical applications, we propose a fully convolutional network (FCN) that localizes reflective surfaces under the relaxed assumptions that (*i*) a compact array of only two microphones is available, (*ii*) emitter and receivers are not synchronized, and (*iii*), both the excitation signals and the impulse responses of the enclosures are unknown. Our FCN is designed to extract spectral and temporal patterns from stereo recordings, aggregate the temporal information over time-frames, and predict the likelihood of virtual sources corresponding to reflective surfaces at specific locations. Whereas most source localization algorithms are limited to direction-of-arrival (DOA) estimation, the proposed method jointly estimates distances and DOAs. Numerical experiments confirm that the network is able to generalize to mismatched microphone array sizes, sensor directivity patterns, or audio signal types, while highlighting front-back ambiguity as a prominent source of uncertainty. When a single reflective surface is present, up to 80% of the sources are detected, while this figure approaches 50% in rectangular rooms. Further tests on real-world recordings report similar accuracy as with artificially reverberated speech signals, validating the generalization capabilities of the framework.

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Room geometry estimation from stereo recordings using neural networks

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Giovanni Bologni B.Sc.
born in Montepulciano, Italy

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Applied Mathematics and Computer Science
Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Applied Mathematics and Computer Science for acceptance a thesis entitled **"Room geometry estimation from stereo recordings using neural networks"** by **Giovanni Bologni B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: March 6, 2020

Chairman: ______________________
Prof.dr.ir. R. Heusdens

Advisor: ______________________
Dr.-Ing. F. Giron

Committee Members: ______________________
Dr.ir. R.C. Hendriks

____
Dr. M. Loog

Contents

Introduction 1

Many animals rely on sound to communicate, hunt preys and more generally characterize their surroundings. Notable examples include bats, dolphins and humans. Similar to sonars, some species of bats emit a probe sound and locate targets based on the echoes that return from objects in their proximity. With a similar strategy, it is possible to guess the dimensions of a closed environment through sound. Imagine someone is walking in a closed, quiet space. The echoes produced by their steps will be different in a cathedral or in a meeting room. Through reflected sounds, one can predict the shape and the acoustic characteristics of the enclosure. The intuition behind this phenomenon is that when a sound is emitted inside a room, the walls will reflect and distort the signals, until such reflections eventually hit our ears (Fig 1.1). Our brain is able to link spectral and temporal features of the echoes with the shape of the enclosure.

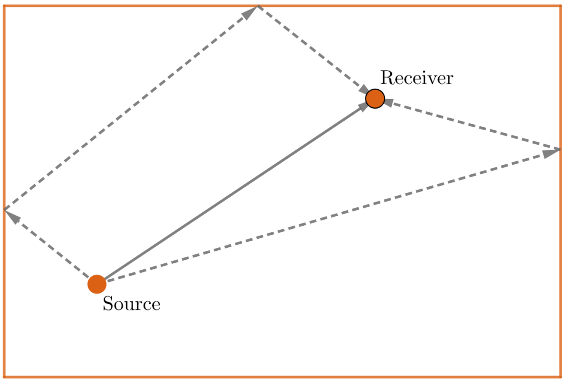


Figure 1.1: Schematic of a room with reflections

Following the same reasoning, algorithms that estimate the geometry of a closed space can be designed. Discerning the geometry of an enclosure from acoustic echoes has found applications in speech enhancement and separation [1, 2], robotics, auralization [3], and spatial upmixing, among others. The method described in this thesis is especially suited for estimating the size of an enclosure from musical stereo recordings, with the ultimate goal of performing reverberation-informed, stereo to surround upmix. By extracting spatial information from a stereo recording, it is possible to derive a surround upmix that promotes a more realistic perception of spaciousness. In binaural hearing, the shadowing effect of the head and the pinna generates spectral and spatial cues which are useful for localization. In contrast, because music stereo tracks result from summing recordings of separated instruments, each processed by artificial reverberation and other effects, it is not possible to assume specific sensor directivity patterns. Accordingly, the proposed approach aims at inferring the dimensions of an enclosure from generic, two-microphones recordings.

In fact, it is possible to establish a link between acoustic and geometric properties of a room. Assuming that an enclosure can be modeled as a linear, time-invariant (LTI) system, its acoustic behaviour is completely characterized by the impulse response between a source and a microphone [4]. Therefore, a room impulse response (RIR) has a one-to-one correspondence with the shape of a convex enclosure (up to rigid transformations) [5]. Although theoretically possible, estimating the geometry of a room from a single measurement is a challenging task, and the majority of previous research has employed recordings from multiple microphones, with few exceptions [6]. If multichannel recordings are available, the estimation pipeline is the following. First, in most practical cases, the impulse response of the enclosure is not measured directly. Instead, it has to be recovered from recordings of different excitation signals [7, 8, 9]. Once the impulse response has been estimated, the peaks corresponding to the reflective surfaces have to be identified. By applying multi-microphone source localization techniques, one can detect and localize the reflections [7, 10, 11, 12]. If the distance between any two microphones is larger than the distance between a microphone and a reflective surface, the reflected signals will arrive in a different order at the different microphones. Such ambiguity is known in literature as the *echo-labeling* problem [5, 13]. In this scenario, additional steps are required to correctly assign each peak in the impulse response to each reflection [14, 15]. Finally, the reflections can be mapped to the geometry of an enclosure through Procrustes analysis or other shape matching techniques.

In general, the geometry of an enclosure can be estimated from acoustic measurements as described above. Most methods, however, present two major limitations. First, they require a large number of microphone channels (typically $N > 5$). Larger sensor arrays might not suit space or hardware constraints, and lead to higher computational costs. Using many receivers allows to localize the source at the intersection of hyperboloids defined by each microphone pair, thanks to trilateration. In contrast, if a single microphone pair is available, range and angle localization is only possible by exploiting level differences between sensors, i.e. assuming *near-field* settings. The second drawback of parametric methods is that they rely on simplifying hypotheses about the audio propagation model. In particular, they are built upon geometric acoustic models such as the image-source model (ISM) [16], which assumes rigid and perfectly reflective surfaces. According to the ISM, the walls will only reflect the incident sound waves, regardless of their arriving angles and spectral characteristics.

To avoid the explicit modeling of the a priori assumptions about a system, supervised machine learning methods use an annotated training set to learn a mapping between an input and a target signal. Such data can either be collected from real-world measurements or generated through software simulators. Advancements in the machine learning community have shown deep neural networks (DNNs) achieving human or super-human performance in classifying images, playing strategy games, captioning videos and generating speech [17]. Recently, similar data-driven algorithms have also been applied to acoustic source localization (ASL), which consists of locating and identifying sound sources from multi-channel audio measurements [18]. In other words, ASL is aimed at finding objects who are actively emitting sound. In contrast, room geometry estimation consists of locating passive sources like walls, which bounce back scattered and diffracted versions of incident sound waves. Locating such reflecting surfaces is necessary to reconstruct the geometry of the enclosure. Although DNNs have successfully been applied in acoustics for ASL, estimation of reverberation time [19], early decay time [20], and room volume [21], direct localization of room boundaries or mirror image sources using DNNs is an open research path. Thus, this report proposes a DNN approach to answer the following question:

- Is it possible to fully localize multiple acoustic sources with two microphones only, when 1. the absolute time-delays of the sources are unknown, i.e. sources and receivers are not synchronized, and 2. the emitted signals are unknown and highly correlated?

The core of our method is a convolutional neural network designed to extract spectral and temporal patterns from the multichannel audio, aggregate the temporal information over multiple time-frames, and predict the likelihood of the reflections being at specific locations. Similar to prior work, the locations of reflective surfaces are estimated from multichannel audio recordings. Unlike previous research, the reflections are localized from generic sounds recorded by only two microphones. In addition, whereas most localization strategies are limited to the estimation of the source angles, our network is also able to predict the source distances. When a single reflective surface is present, the proposed algorithm is able to localize the real and corresponding image source in 80% of the cases, while in rectangular rooms this figure reaches 49%. We also conduct two experiments to check the quality of the learned features. In the first it is shown that the network, trained on white Gaussian noise signals only, is able to reasonably generalize to speech signals. The second experiment reveals that using microphone arrays of different sizes for train and test, biases the estimates in a predictable way, confirming that the proposed neural network relies on meaningful cues for localization.

The reminder of this report is organized as follows. Chapter 2 introduces the reader to the image-source model and outlines the advances in the fields of room geometry estimation and acoustic source localization. Next, the proposed DNN for room geometry estimation from stereo recordings is detailed in Chapter 3, while Chapter 4 presents some numerical results. The last chapter summarizes the report and highlights possible future directions.

Background 2

Localizing the reflective surfaces of an enclosure is equivalent to knowing its geometric shape. To begin with, this chapter outlines the principle of the most common framework for room geometry estimation: the image-source model. In addition, relevant literature in the field of room geometry estimation is reviewed, with a special focus on parametric and data-driven methods for acoustic source localization. Finally, an outline of the proposed approach is presented.

2.1 The image-source model

Sound propagates through air or water as a pressure wave. The propagation of a sound wave in a lossless fluid is governed by a second order partial differential equation that describes how the pressure varies over time and space [4, Chapter 1]. To ease the modeling of a sound field in a closed space, several approaches known as *geometrical acoustics* assume that sound propagates along straight rays at a constant speed c. According to these models, a sound ray emitted from a point source s located at position $\mathbf{s} \in \mathbf{R}^3$ will arrive at a receiver x, placed in $\mathbf{x} \in \mathbf{R}^3$, after a delay τ given by

$$\tau = \frac{\|\mathbf{s} - \mathbf{x}\|}{c}. \tag{2.1}$$

Moreover, like for waves with a spherical wavefront, the energy of the emitted sound ray will decrease proportionally to $1/\|\mathbf{s} - \mathbf{x}\|^2$, the squared inverse of the distance travelled by the ray.

Similar to reflection of light in a mirror, some of the energy content of sound waves is reflected specularly when they encounter an obstacle (e.g. a wall). Generally, the rest of the energy associated with the sound wave is either diffracted in non-specular directions, when the wavelength is comparable to the size of the obstacle, or dissipated into heat. In geometrical acoustics, however, every surface is assumed to be purely reflective. This implies, for example, that when source and receiver are located in an open space, and there is an obstacle between them, the receiver will not hear any sound. One of the most popular models for geometrical acoustics is the image-source model (ISM) [16]. The ISM neglects typical wave effects such as diffraction and interference. As such, it is more accurate at higher frequencies where the wavelength of the incident waves are small compared to the dimensions of the enclosure under exam [22]. When there is a single, planar surface, an ideal specular reflection can be modelled via a mirror image source, that is found by reflecting the source against the reflecting surface (see

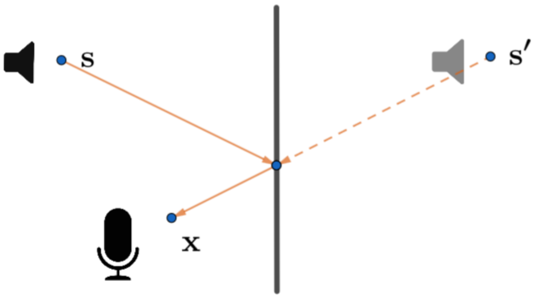


Figure 2.1: A sensor $\mathbf{x}$ and a speaker $\mathbf{s}$ are located next to a reflective wall. The image-source $\mathbf{s}'$ emulates the effects of the reflection of sound on the wall.

Figure 2.1). The real and the mirror (or virtual, or image) sources emit the same sound simultaneously. As a result, the receiver will record delayed copies of the same signal. If the sound ray is reflected in a single boundary, it is called a first-order reflection. In an enclosure, the sound rays can be reflected several times before impinging on the receiver, and each reflection can be modelled by an additional image source. The mirror sources who model successive reflections are represented by higher order image sources (see Figure 2.2).

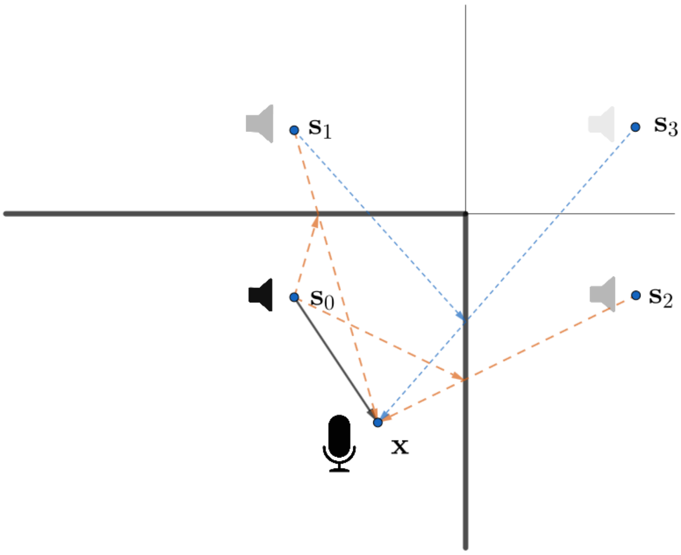


Figure 2.2: The image-source model. In this example, there is one real source $\mathbf{s}_0$ in an room with two reflective surfaces (black in the picture). The receiver is $\mathbf{x}$. The first-order virtual sources are $\mathbf{s}_1$ and $\mathbf{s}_2$ (dark grey), and the only second-order image source is $\mathbf{s}_3$ (light grey).

Consider again a system with a source s and a receiver x, located in an enclosure at locations $\mathbf{s}$ and $\mathbf{x}$, respectively. Let the omnidirectional point source s emit an impulse signal $\delta(t)$. The impulse response $h(t)$, recorded from an ideal noiseless receiver, is the sum of the impulses emitted from the real and the image sources,

$$h(t) = \sum_{i=0}^{I} h_i(t) \tag{2.2}$$

where I is the number of mirror sources considered in the model and $h_i(t)$ the impulse response between the i-th source and the receiver. The real source corresponds to the index $i = 0$, so that $h_0(t)$ models the *direct path* between source and receiver.

The i-th source can be associated with several reflective surfaces at a time, where each boundary has a frequency-dependent impulse response. In general, reflective surfaces attenuate high frequency content more. Let us denote with $0 < \gamma_i(t) \leq 1$ the combined effect of all the surfaces impacting with the ray from the i-th source. Then,

$$h_i(t) = \gamma_i(t) * \frac{\delta(t - \Delta_i)}{\|\mathbf{x} - \mathbf{s}_i\|} = \frac{\gamma_i(t - \Delta_i)}{\|\mathbf{x} - \mathbf{s}_i\|}, \tag{2.3}$$

where Δ_i is the absolute time-of-arrival (TOA) of the i-th image source (Equation 2.1), $\|\mathbf{x} - \mathbf{s}_i\|$ is the distance between the receiver and the i-th source and '$*$' denotes convolution. Because the index $i = 0$ is associated with the real source, $\mathbf{s}_0 = \mathbf{s}$ and $\gamma_0(t) = 1$.

Equations 2.2 and 2.3 define the acoustic impulse response (AIR) of the room according to the ISM. By applying the AIR $h(t)$ to a target audio $s(t)$ through linear convolution, one can simulate the acoustic characteristics of a specific source-receiver pair $(\mathbf{s}, \mathbf{x})$ inside an enclosure,

$$x(t) = s(t) * h(t) = \sum_{k} s(k)\ h(t - k). \tag{2.4}$$

By using the ISM, one can efficiently model the soundfield in an enclosure with the stratagem of virtual sources located outside the enclosure itself. Thanks to geometric duality of the ISM, there is an association between the set of image sources and the locations of the reflective surfaces of a room. In particular, the locations of first and second order virtual sources have a one-to-one mapping with the geometry of a convex enclosure [5]. Thus, by estimating the locations $\mathbf{s}_i$ of the sources, it is possible to reconstruct the geometry of any convex room.

2.2 Prior work

2.2.1 Room geometry estimation

In the previous section, it was shown how an acoustic impulse response can be linked to the geometry of an environment thanks to the image-source model.

Let us consider a typical room geometry estimation pipeline, as depicted in Figure 2.3 (top row). Most approaches begin by estimating the AIR of the enclosure from

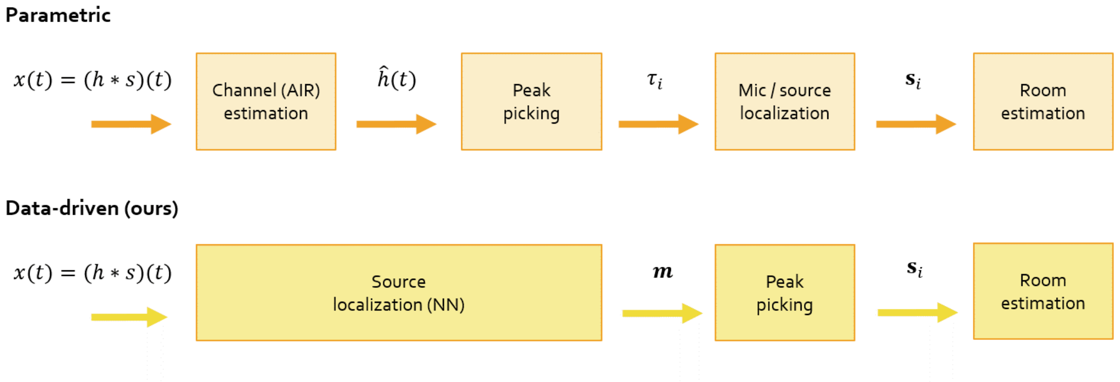


Figure 2.3: A typical pipeline for room geometry estimation.

measured data $x(t) = (h * s)(t)$. When no prior information about $s(t)$ is available, estimation of the AIR $h(t)$ is an instance of the blind-channel identification problem [23]. On the other hand, assuming that the excitation signal $s(t)$ is known a priori, one can estimate $h(t)$ through (sparse) deconvolution [24]. The second step consist of identifying the peaks of $\hat{h}(t)$, corresponding to the time-of-arrivals (TOAs) associated with the sources. In general, when source and receiver are not synchronized, the emission time of the source is unknown. If a sufficient number of receivers is available, synchronization can be achieved from the measured path delays through self-calibration methods [25, 26]. However, when the prerequisites for the self-calibration algorithms are not fulfilled, the absolute TOAs Δ_i are not recoverable. In such cases, the relative time-difference-of-arrival (TDOA) of the sound at each sensor pair can rather be exploited. The absolute or relative delays define hyperbolic shapes (see section A.1) from which it is possible to localize the reflections $\mathbf{s}_i$ and successively the room boundaries [27, 28].

In most cases, localization of the reflective surfaces of an enclosure from audio measurements is performed in ad-hoc settings, i.e. with multiple microphones and sources *distributed* around the room, with oracle knowledge about their locations, or assuming control over the excitation signal $s(t)$. On the other hand, if the information about room geometry has to be recovered, for example, from microphones placed on robots, smartphone or smart speakers, the distance between sensors will be smaller and the recorded signals, often speech or music, will be unknown.

2.2.2 Acoustic source localization

Under the conditions (1) the excitation signal $s(t)$ as well as the emission time are unknown, and only noisy or reverberant recordings $x(t)$ can be measured, (2) receivers are placed in a *compact* array with known geometry, one could attempt to localize the image sources $\mathbf{s}_i$ from the recording $x(t)$ using algorithms for acoustic source localization (ASL). In this report, one such algorithm for full localization of multiple mirror-image sources based on deep neural networks (DNN) is presented.

Let us now delve deeper into the topic of acoustic source localization. Parametric methods have been the preferred choice for ASL for years, while investigation of data-driven methods for localization has begun only recently.

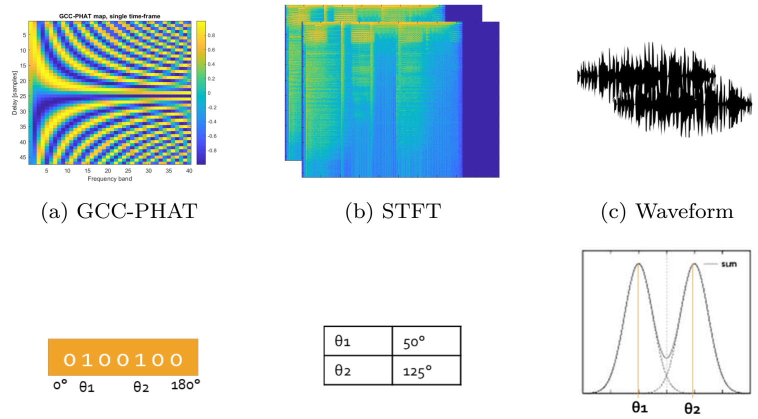
2.2.2.1 Parametric methods

Model-based approaches for ASL can loosely be divided in three categories. The first is that of the high-resolution spectral estimation based locators, comprising all algorithms that perform eigendecomposition of the sensor spatial covariance matrix [29]. Despite being very accurate at localizing independent sources in far-field, the performance of these algorithms degrades when trying to locate correlated sources, or in reverberant environments. In addition, given an array of M sensors, they can only find up to $M-1$ distinct speakers. Approaches belonging to the second group localize sources based on the time-difference of arrival (TDOA). The time-delay at which a pair of sensors shows the highest correlation is often obtained from the generalized cross-correlation phase-transform function (GCC-PHAT) [30, 31]. A shortcoming of such methods is that they are mostly designed for single source scenarios and fail when there is a mismatch between the actual scenario and the model. The third type of parametric algorithms are based on the maximization the steered response power of a beamformer [32, 33]. By filtering, weighting and summing the data at the sensors, these methods virtually steer the receivers towards all possible directions in a grid. The source is found at the direction corresponding to the highest collected power. The most popular of such methods, SRP-PHAT, performs well even in adverse condition, but it is computationally expensive due to the grid search over possible directions.

2.2.2.2 Data-driven methods

To overcome some of the limitations of parametric methods, several approaches that rely on DNNs to perform direction-of-arrival estimation have been proposed in the last years. These methods adopt supervised learning strategies, where DNNs are trained on large data sets to learn a mapping between multi-channel recordings and speaker locations. DNN based approaches can be categorized according to the way they represent input and target quantities, the nature of the training set and the structure of the computational graphs.

Input encoding As for the input representation, most methods for data-driven ASL utilize features extracted from the multi-channel audio waveforms. One such hand-crafted features is the GCC-PHAT map (Figure 2.4a), whose peaks can readily be linked to the TDOA [34, 35, 36]. However, GCC-PHAT maps implicitly assume joint wide-sense stationarity across signals received at different sensors [37], and do not contain sufficient information to retrieve the full location of sources (see section A.2). Al-



θ1	50°
θ2	125°

(a) GCC-PHAT (b) STFT (c) Waveform

(d) Classification (e) Regression (f) Likelihood

Figure 2.4: Several representations for input (top row) and target data points (bottom row) to be used in the DNN.

ternative input representations include interaural phase differences and interaural level differences [38, 39, 40], especially for binaural localization, or the intensity vector for Ambisonics recordings [41]. Another possibility is that of applying an STFT transform to the audio waveform before feeding it to the DNN (Figure 2.4b), a strategy which has revealed successful for tasks like music genre classification [42] or sound source separation [43]. Adavanne et al. employ both phase and magnitude of the STFT coefficients to localize multiple overlapping speakers [44, 45, 46], whereas Chakrabarty and Habets use only the phase of the STFT coefficients to estimate the DOA of non-superimposed speakers [47, 48, 49, 50]. Under the *far-field* assumption, they claim that the STFT magnitude can be discarded, being the same at all microphones. More recently, ASL has been performed directly from the audio waveform (Figure 2.4c), avoiding the feature extraction step completely. The sample domain input is typically fed to a cascade of convolutional layers, which should automatically extract relevant features [51]. Vecchiotti et al. train a set of convolutional finite-impulse-response (FIR) filters and let the network learn a frequency decomposition [52]. Although their trainable filterbank does not generalize to reverberant recordings if trained on anechoic ones, it is shown to outperform a fixed Gammatone decomposition when the training set is representative of the test set.

Target encoding DNN approaches for acoustic localization also differ in the way they represent the target source location. In most cases, estimation of the DOA is either cast as a multi-label classification problem or as a direct regression of Cartesian coordinates [53]. In the multi-label classification problem on the discretized unit circle (Figure 2.4d), the speaker at each discretized angle is either present or absent [47, 48, 49, 50]. Using a standard cross-entropy loss between target and estimated distributions, the major drawback of classification is that it does not allow to take distance between angles into account. For instance, assume that a source is found at the 50° azimuth, and that angles are discretized in 5° steps. The cross-entropy loss occurring for an estimated DOA of 45° is the same as for an estimate of 30°, although the former guess is intuitively better than the latter. A second way to cast the localization problem is by using direct regression of Cartesian coordinates corresponding to the target DOA (Figure 2.4e), i.e. by minimizing a distance between target and estimated coordinates [44, 45, 46]. Regression might seem more appropriate both because the output space is structured and continuous, and because it yields a direct correspondence between the cost function to be minimized and the localization error. On the other hand, it is not clear how to extend regression to a variable number of speakers, because the number of locations to be estimated needs to be fixed in advance. A third strategy is that of performing regression on a map representing the likelihood of a source being at a specific spatial coordinate (Figure 2.4f) [35]. Similar to classification, it allows to encode an arbitrary number of sources. In addition, since it performs a soft assignment of the output values, such encoding enables the network to take the correlation between close angles into account, akin to regression. Related forms of soft encodings have been recently employed in different classification problems with promising results [54]. The advantages of likelihood regression come at the price of increased dimensionality of the output.

2.3 Our method

Somewhat surprisingly, all of the methods for ASL using DNN presented above only estimate the 2- or 3D angle-of-arrival of the speakers, neglecting the distance information. When acoustic localization is an intermediate step in estimating the geometry of an environment, the distance of the mirror image sources is strictly necessary. Therefore, This thesis aims to investigate the feasibility of angle and range localization of multiple overlapping sources with DNNs. The sources to localize are the mirror image sources, which in turn reveal the geometry of the environment under analysis. The idea of He et al. [35], who predict the likelihood of finding a source at a specific angle (1D), is extended to enable the estimation of both angle and range (2D). In our method, the peaks in the likelihood map identify the full 2D positions of real and image sources, which in turn reveal the reflective surfaces (see Figure 2.3, bottom row).

In the remainder of this report, we will focus on shoe-box rooms only, because of their practical relevance. Moreover, the problem of ASL will be restricted to the two dimensional case, i.e. the elevation coordinate will be ignored for the sake of simplicity. Unlike previous work, our algorithm can perform room geometry estimation in restrictive conditions, such as *i)* the excitation signal as well as the emission time are unknown, and only noisy or reverberant recordings can be measured and *ii)* only a compact array of two microphones is available, the theoretical minimum for acoustic source localization. Whereas other solutions solve the problem with high accuracy using ad-hoc instruments and measurements, our DNN based algorithm tackles the geometry estimation task under very mild assumptions.

3 Proposed method

The proposed room geometry estimation neural network takes as input a stereo waveform and returns the location of multiple acoustic sources, corresponding to the reflective surfaces. This chapter begins with describing the real and simulated datasets used for training and testing the DNN. In the second section, design choices and methodology of the proposed strategy is detailed.

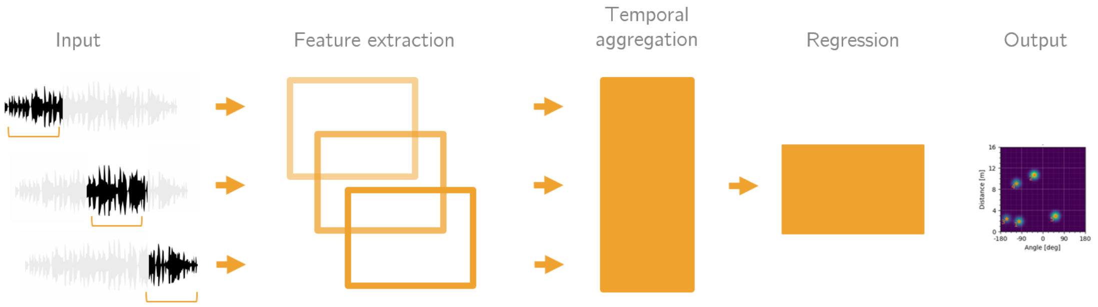
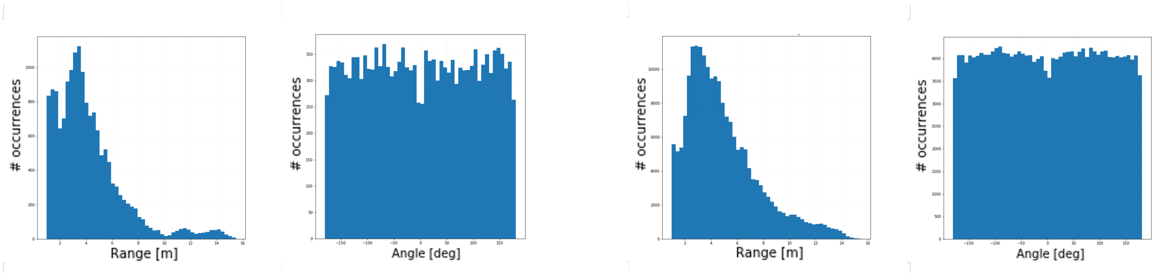


Figure 3.1: A block diagram depicting the high-level architecture of the proposed network for source localization from multichannel audio.

3.1 Data generation

Multiple datasets of acoustic impulse responses (AIRs) are generated to simulate the scenarios of anechoic environments, or enclosures with a single or multiple reflective surfaces. The training sets used in most experiments are then created by convoluting realizations of 1s of white Gaussian noise (WGN) with AIRs of rooms of different sizes and acoustic characteristics. In contrast to speech or music signals, WGN contains the complete frequency range and will excite all the different modes of the enclosures, thus easing the localization of the virtual sources.

In all datasets, the AIRs are generated through the software MCRoomSim [55] using the image-source model at $f_s = 48$kHz, and room dimension is varied between 2×2 m^2 and 7×7 m^2. Higher-order reflections up to the third order are modeled. When multiple reflective surfaces are present, the lateral walls of each room share the same frequency-dependent acoustic responses, whereas floor and ceiling are perfectly absorbing to simulate the 2D case. Source and array positions are randomly sampled from a grid with 0.075 m spacing, and constrained to satisfy three requirements: the



(a) `WGN1`, range.

(b) `WGN1`, angle.

(c) `WGN4`, range.

(d) `WGN4`, angle.

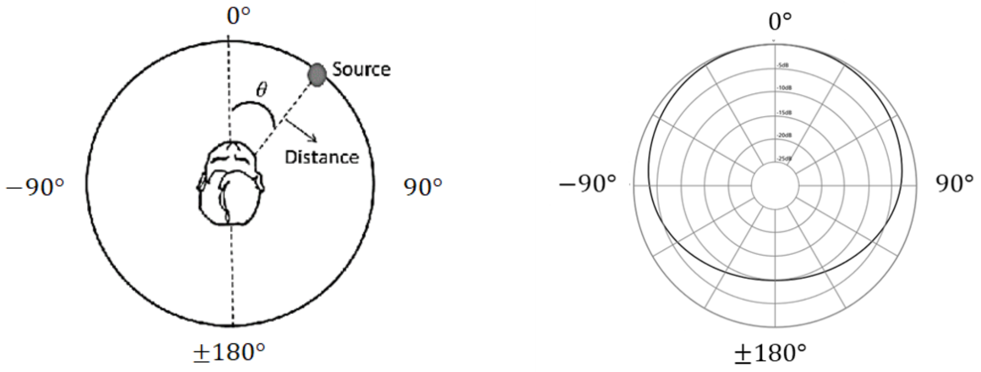
Figure 3.2: Distribution of real and virtual sources locations in `WGN1` and `WGN4` datasets in terms of distance or direction of arrival to the microphone array.

distance between source and microphone must be be greater than 0.5 m, the distance between source or microphone and any wall must be greater than 0.5 m, and the source must not lie on the perpendicular bisector of the array, i.e. the source must not be exactly in front of or behind the microphone array, with a tolerance of ±3°. The latter condition is required to avoid under-determined settings where both the amplitude and the TOA at the sensors are the same [56]. The anechoic and single reflective surface WGN datasets (denominated as `WGN0` and `WGN1`, respectively) contain about 3 hours recordings each, while the four reflective surfaces dataset `WGN4` consists of about 15 hours of reverberant WGN. Extra effort was put on ensuring that the locations of real and image sources were distributed approximately uniformly around the sensor array to avoid biasing the estimates of the DNN. However, the locations of the virtual sources are determined by the location of the real source and the shape of the enclosure, and achieving perfectly uniform angle and distance distributions is a non trivial task (see Figure 3.2). The source is spatially stationary, i.e. it does not move within a recording.

All the simulations use a 2-channel microphone array, with inter-microphone spacing of 10 cm. The microphone array is randomly rotated for each AIR. To reduce the front-back ambiguity, the microphones have a directional, subcardioid response pattern that attenuates sounds coming from behind and from the sides. Following the binaural convention that assigns an azimuth of 0° in front of the array, positive angles on the right emisphere, and negative angles on the left one (see Figure 3.3), the subcardioid attenuation pattern is described by:

$$a(\phi) = \frac{2}{3}\left(1 + \frac{1}{2}\cos\phi\right), \quad -\pi \leq \phi \leq \pi. \tag{3.1}$$

In addition to *synthetic noise* evaluation sets generated similarly to the training sets, and follow the same name convention, we obtain *synthetic speech* sets given by the convolution of the synthetic AIRs and clean utterances in Spanish, German and English from the King-ASR corpora 178, 182 and 249, respectively. The AIRs generated for noise and speech evaluation sets are created by simulating similar reverberation settings



(a) Angle convention

(b) Subcardioid directivity.

Figure 3.3: Angle convention (a) and subcardioid directivity pattern (b).

to the training set. Three squared rooms of sizes 3 × 3 m^2, 4.5 × 4.5 m^2 and 6 × 6 m^2 were selected. The acoustic characteristics of the reflective surfaces in the evaluation set differ from those in the training set, whereas the microphone array geometry is identical.

Lastly, to check the generalization capabilities of the proposed DNN, an additional *real speech* dataset is obtained from recordings of actual meetings in two different rooms using a microphone array of mismatched size and directional response. The meeting rooms are located in the Sony European R&D office in Stuttgart (DE), and their sizes are 5.3 × 5.7 m^2 and 6.5 × 5.6 m^2. Fragments of the recordings where the speakers did not overlap were handpicked and manually annotated, for a total of about 1 minute per each room. The 6-channels recordings were performed using a ReSpeaker circular sensor array for Raspberry Pi[1]. The audio from two opposed microphones, located at about 0.092 m distance, was selected.

3.1.1 Generation of the likelihood maps

Our network predicts a nonlinear mapping $f_\theta(\cdot)$ from a multichannel audio sample x to a map m that indicates the likelihood of a source being at each position. The parameters θ of the network are adjusted to minimize an empirical risk over a training set,

$$\mathcal{L} = \frac{1}{n}\sum_{j=1}^{n} d\left(f_\theta(x^{(j)}),\ m^{(j)}\right)^2, \tag{3.2}$$

where n is the number of samples in the training set, $d(\cdot)$ is the Euclidean distance, $x^{(j)}$ is the jth training sample and $m^{(j)}$ the corresponding target heat-map.

The output of the network is encoded as a 2D image $m \in \mathbb{R}^{L\times L}$, where the first dimension is associated with the source distance $\rho \in [0, d_{\max}]$, and the second with the angular direction $\phi \in [-\pi, \pi]$. Each source corresponds to a Gaussian-like functions

[1] https://respeaker.io/6_mic_array/

centred on the ground truth location $p_i^{(s)} = [\rho_i^{(s)}, \phi_i^{(s)}]$. A naive implementation that neglects the periodicity of the angular coordinate would result in the following basis function:

$$\hat{g}_i = \begin{cases} e^{-d\left(p_i,\ p_i^{(s)}\right)^2/\sigma^2} & \text{if } \rho_i \in [0, d_{\max}],\ \ \phi_i \in [-\pi, \pi] \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

where σ controls the spread of the Gaussian-like curves and $d(\cdot)$ is the Euclidean distance. To avoid truncating the Gaussian-like functions it is possible to define a wrapped basis function g_i that takes into account the circular nature of the angular coordinate:

$$g_i = \sum_{k=-\infty}^{\infty} \mathbb{1}_{[0,\ d_{\max}]\times[k\pi,\ (k+1)\pi]}\ e^{-d\left(p_i,\ p_i^{(s)}\right)^2/\sigma^2}, \tag{3.4}$$

where $\mathbb{1}_{\mathcal{A}\times\mathcal{B}}$ has value 1 on the 2D interval $\mathcal{A}\times\mathcal{B}$, and 0 elsewhere, and $p_i^{(s)} = [\rho_i^{(s)}\phi_i^{(s)}]$ is the ground truth location of the ith image source in polar coordinates (see Figure 3.4). Similar to $\hat{g}_i$ from Equation 3.3, the support of the basis function g_i is on the interval $[0, d_{\max}] \times [-\pi, \pi]$.

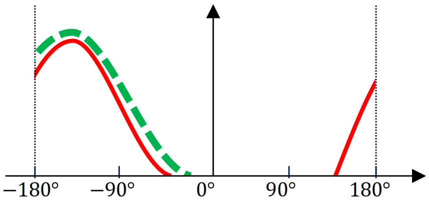


Figure 3.4: 1D representations of $\hat{g}_i$ (dashed green line) and the wrapped version g_i (solid red line) for a source located at approximately −125°.

In addition, we noticed in preliminary experiments how incorporating knowledge about the microphone directivity in the target likelihood maps would benefit the learning process of the DNN. Thus, the height of each Gaussian-like component is rescaled according to the microphone directivity of Equation 3.1. A likelihood map is then computed as:

$$m' = \sum_{i=0}^{I} a(\phi_i^{(s)})\ g_i, \tag{3.5}$$

where $a(\phi_i^{(s)})$ is the attenuation for the ith source located at angular direction $\phi_i^{(s)}$ as defined in Equation 3.1, g_i is the basis function defined in Equation 3.4 and I is the number of sources considered in the model. The real source is located at p_0. As a last step, the sum of the components is normalized to one, to obtain $m = m'/\max(m')$.

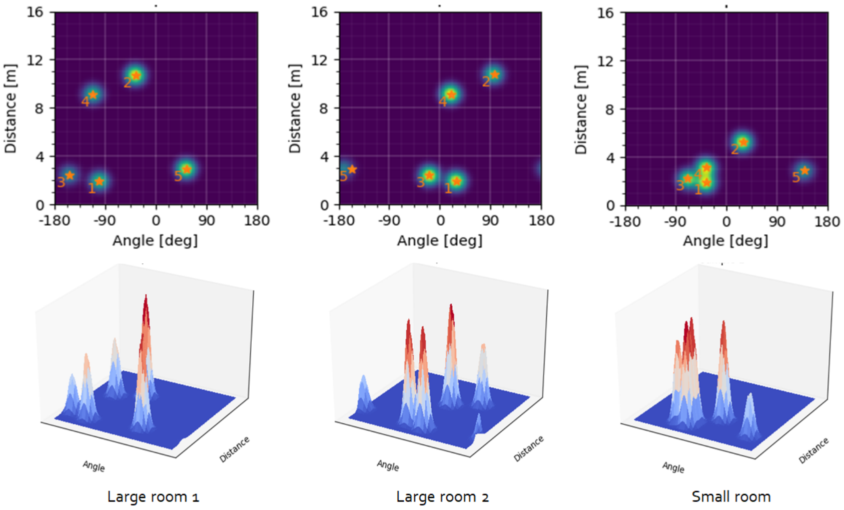


Figure 3.5: Ideal likelihood maps for real and first-order virtual sources, in polar coordinates. The real source always has the smallest distance to the array. Per each column of the figure, top and bottom rows represent the same configuration from different perspectives.

By assuming that any single dimension of the room will not exceed 8m, as it is the case in our training set, the distance between any first-order image source and the microphone array will not exceed $d_{\max} = 16$m. To generate the likelihood maps, it is possible to uniformly quantize both the support of the distance ρ and the angle ϕ in L steps.

3.2 Proposed method: rationale

Our method employs a fully convolutional neural network to localize real and virtual acoustic sources from multi-channel audio recordings. The architecture is designed to perform spectral and spatial feature extraction from the input audio, temporal context aggregation over multiple time frames, and likelihood map regression. The peaks in the two dimensional likelihood map correspond to the estimated sources locations.

Because the input audio signal can be modelled as a stationary random process, and its statistical distribution does not change over a time frame, a network using small convolutional filters should be preferred to an architecture based on matrix multiplication [17]. In fact, convolutional layers can detect local patterns in the data using a fraction of the learnable parameters of affine layers, thus reducing the size of the network and consequently the *over-fitting* issue. In addition, fully convolutional architectures can function with inputs of any size without modifications.

Feature extractor The feature extractor block performs spectral, spatial and temporal analysis of the stereo microphone signals, mapping the waveforms down to multiple space-frequency representations. Due to the lowpass filtering behaviour of the reflective surfaces, the spectra of real and virtual sources differ. Thus, it should be possible to disambiguate between real and virtual sources from their spectral characteristics. The first convolutional stage, `fConv`, contains long FIR filters intended to perform a spectral decomposition similar to a time-domain filterbank [57]. Such spectral filters are identical across microphones. Next, each filtered multi-channel output is passed to a set of spatial filters, which act on all channels simultaneously (`sConv`). These learnable beamformers are capable to steer the microphone array to multiple directions by applying different time-delays to different channels.

With this arrangement, we hypothesize that the first stage should implement a finer frequency decomposition, while the multichannel filters acting on the spectrally decomposed signals should realize spatially selective filters, similar to early implementations of broadband beamformers in time domain [58]. Finally, a sequence of shorter, one dimensional FIR filters is deployed to detect edges and relevant patterns in the spectrally and spatially filtered audio signals (`tConv`). The local time pattern identification concludes the feature extractor block.

Temporal aggregation The feature extraction is performed over multiple overlapping segments, as depicted in Figure 3.1. The filtered outputs are combined together in the following temporal convolutional block [59]. Whereas most strategies for sequence modeling with DNNs employ long-short time memory and recurrent networks, temporal convolutional networks (TCNs) have recently demonstrated competitive performances in a variety of sequence modeling tasks [60]. In addition, TCNs parameters are easier to optimize than their recurrent counterpart [59]. For these reasons, and to keep the overall architecture fully convolutional, we decided to adopt them in our network. By combining the space-frequency representations of successive frames through temporal convolution, we wish to capture the long term dependencies in the data. For instance, acoustic reflections cause the support of the cross-correlation function across sensor channels to extend up to delays of $\tau_{\max} = f_s \cdot d_{\max}/c$ samples, where $d_{\max}$ is the distance between the farthest virtual source and the microphone array. The temporal aggregation block combines the contributions of multiple windowed frames into a single space-frequency representation.

Regression Next, the low dimensional output of the temporal aggregation block should be upsampled and reshaped as to produce the desired position likelihood map.

In DNN-based image processing, this process is typically carried out using successive transposed convolutions, or convolutions followed by deterministic upsampling [61]. Recently, the *DenseNet* architecture for image classification was introduced [62, Chapter 7.7], and shortly after extended to the image segmentation task [63]. The main idea behind DenseNet is that each convolutional layer should receive as inputs the feature maps from all preceding layers, and its own output should be used as input to all following layers (see Figure 3.6). This way, the authors claim that ”(DenseNets) alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters” [64]. Inspired by this approach, Zhang et al. deviced an architecture where upsampling is achieved by stacking many small feature-maps, then reshaping them into a few, larger maps, and denominated this method as *dense upsampling* [65]. In our project, we adapt their method to upsample the output of the temporal aggregation block into the two dimensional likelihood map of the sources locations. The prediction of the likelihood map is the last operation executed inside the computational graph.

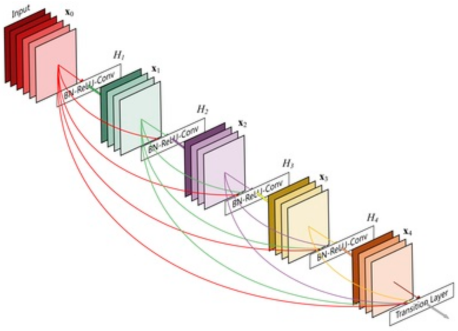


Figure 3.6: A single dense block with $D = 5$ layers having $k = 4$ filters each [64].

Matched filtering and peak picking The network described above produces likelihood maps where the mean positions of 2D Gaussian-like curves correspond to the locations of real source and first-order image sources in polar coordinates. In general, the maps generated by the network will present inaccuracies, as if an unknown, possibly correlated noise was added to the oracle targets. Because the maps are built as a weighted sum of Gaussian-like curves with different mean values (see section 3.1 for details), it is possible to employ a *matched filter* to refine the estimates. The matched filter, or replica correlator, maximizes the signal-to-noise ratio (SNR) at the output of a linear FIR filter, and can be designed in its simplest form without further assumptions about the noise statistics [66]. The output SNR of the matched filter could be further improved if an estimate of the noise covariance matrix was available. After refining the likelihood maps through matched filtering, it possible to finally extract the modes (peaks) of each curve, which correspond to the locations of the separated sources, as depicted in Figure 3.5.

3.3 Network architecture

Having already introduced the main intuitions behind the proposed strategy for multiple source localization, this section will explain more in detail the operation carried out within the computational graph.

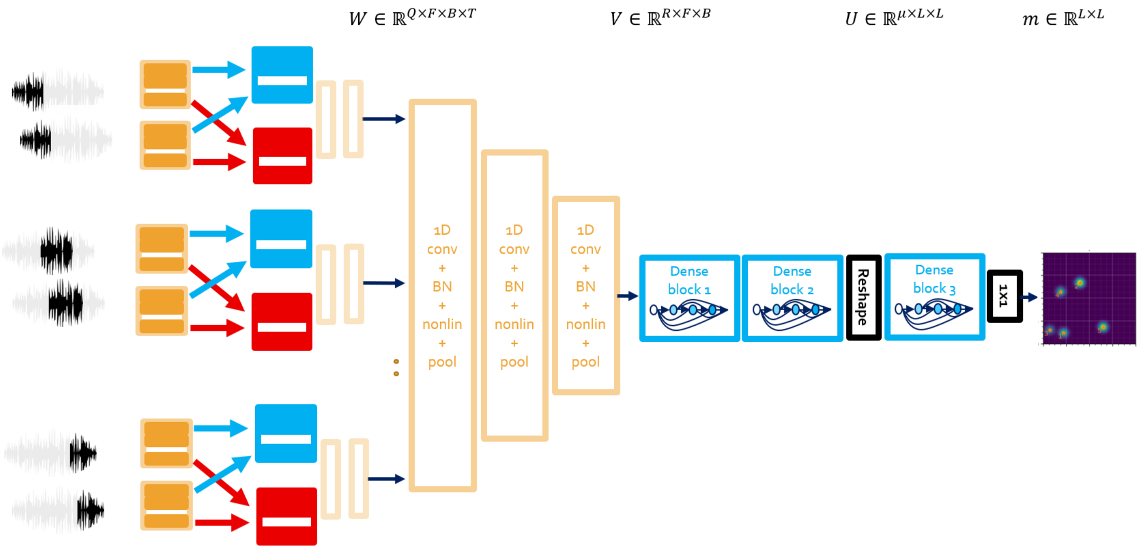

$W \in \mathbb{R}^{Q\times F\times B\times T}$
$V \in \mathbb{R}^{R\times F\times B}$
$U \in \mathbb{R}^{\mu\times L\times L}$
$m \in \mathbb{R}^{L\times L}$
1D conv + BN + nonlin + pool
1D conv + BN + nonlin + pool
1D conv + BN + nonlin + pool
Dense block 1
Dense block 2
Reshape
Dense block 3
1x1

Figure 3.7: A schematic of the proposed architecture. On the left, multiple feature extractor blocks with shared weights act on different frames of the multichannel audio. On the center, a the temporal aggregation block with 3 layers is depicted in light orange. The dense blocks are shown in blue. In this example, $T = 8$ and consequently there are 3 layers in the aggregation block.

3.3.1 Feature extraction

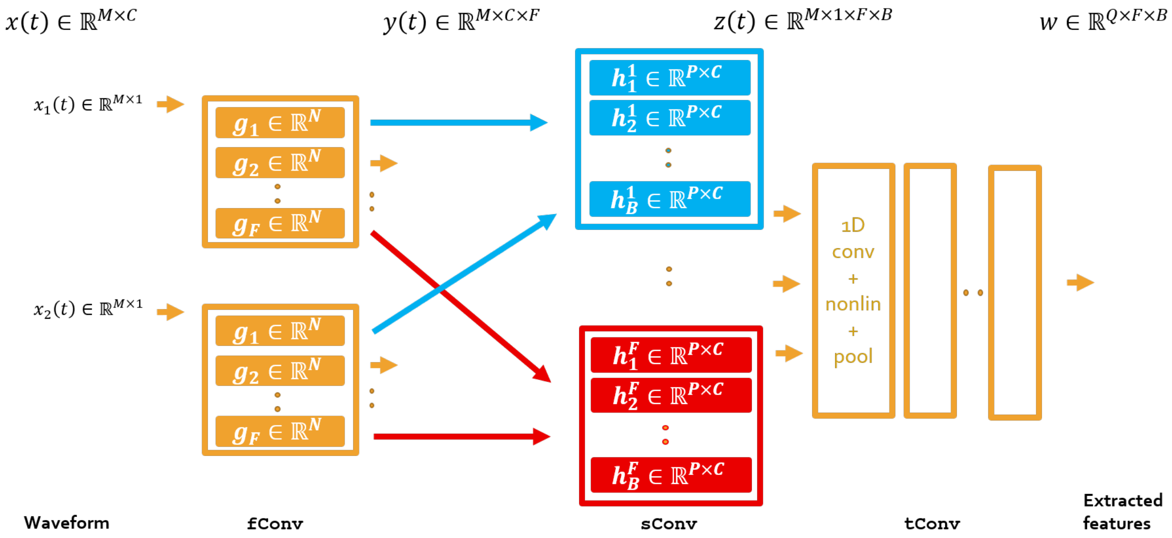
Let us first analyze more in detail the feature extractor block shown in Figure 3.8. As a first step, a multichannel window of length M samples is taken from the input waveform. All the three stages `fConv`, `sConv` and `tConv` perform convolution strided by 1 in time across M samples. Unless otherwise stated, the signals are padded with zeroes before each convolution, so that the outputs have the same lengths as the inputs. To reduce the fluctuations in the values of the intermediate variables during training, batch normalization is applied after the convolution operations [67]. In these cases, the bias parameter of the convolutional filters is ignored to avoid redundancies.

3.3.1.1 Spectral analysis (`fConv`)

The spectral analysis is performed similarly as in [52], where a spectral finite impulse response (FIR) filterbank $g^f = \{g^1, g^2, \ldots, g^F\}$ where $g^f \in \mathbb{R}^N$, is applied to all the input channels. F is the number of FIR filters and N is the number of FIR filter taps. The output of this spectral analysis layer `fConv` will be a set of filtered, multi-channel signals,

$$y_c^f(t) = \sum_{n=0}^{N-1} g^f(n) \cdot x_c(t-n) \tag{3.6}$$

$$= g^f * x_c(t), \qquad c = 0, 1, \ldots, C-1 \tag{3.7}$$



$x(t) \in \mathbb{R}^{M\times C}$
$y(t) \in \mathbb{R}^{M\times C\times F}$
$z(t) \in \mathbb{R}^{M\times 1\times F\times B}$
$w \in \mathbb{R}^{Q\times F\times B}$
$x_1(t) \in \mathbb{R}^{M\times 1}$
$x_2(t) \in \mathbb{R}^{M\times 1}$
$g_1 \in \mathbb{R}^N$
$g_2 \in \mathbb{R}^N$
$g_F \in \mathbb{R}^N$
$h_1^1 \in \mathbb{R}^{P\times C}$
$h_2^1 \in \mathbb{R}^{P\times C}$
$h_B^1 \in \mathbb{R}^{P\times C}$
$h_1^F \in \mathbb{R}^{P\times C}$
$h_2^F \in \mathbb{R}^{P\times C}$
$h_B^F \in \mathbb{R}^{P\times C}$
1D conv + nonlin + pool
Waveform
fConv
sConv
tConv
Extracted features

Figure 3.8: The feature extractor block for a single time frame. The actual implementation with two sensors, $C = 2$, is shown.

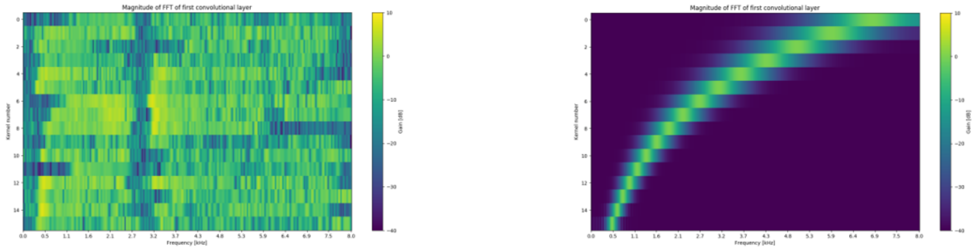
where $x_c(t)$ is the recording from microphone c at time t, $y_c^f(t)$ is the output of the filter g^f for channel c and N is the number of taps for the FIR filters. Notice that an identical filter g^f is applied to all channels, i.e. the filter coefficients are shared among microphones.

In our experiments, we either optimize the filter coefficients jointly with the network, or we fix them to accommodate a fixed Gammatone filterbank, similar to [52]. Whereas the centre frequencies of bandpass filters implemented by a FFT transform are distributed uniformly, the centre frequencies of Gammatone filters are logarithmically spaced. In acoustic source localization, frequencies above $f_{\max} = 2d/c$, d being the inter-microphone distance, lead to an ambiguity known as *spatial aliasing*, for which the beamformers amplify equally signals coming from multiple directions [68]. As a consequence, the logarithmic spacing of the Gammatone filterbank was preferred to a uniform one, because it allows to take into account spatial aliasing by allocating more filters at lower frequencies.

3.3.1.2 Spatial analysis (sConv)

Next, batch normalization is applied to the output of the fConv stage before proceeding with the spatial analysis. The sConv stage is related to filter-and-sum beamforming, where a FIR filter weighs and delays the signals at each microphone before summing them together. Generally, a filter-and-sum beamformer implements the following equation:

$$z(t) = \sum_{c=0}^{C-1} \sum_{p=0}^{P-1} y_c(p) \cdot h_c(t - p - \tau_c), \tag{3.8}$$



(a) Learned filterbank

(b) Gammatone filterbank

Figure 3.9: FFT log magnitude of some of the g_f filters of the first block, `fConv`, as a function of frequency. The response for filters learned by the network is shown on the left (figure (a)), while the fixed Gammatone filters are shown on the right (figure (b)).

where $h_c(t)$ is the FIR filter associated with microphone c, $y_c(t)$ is the cth channel of the signal provided as input, P is the number of taps of the FIR beamformer and τ_c is the time-delay applied to the cth channel to virtually steer the beamformer to the desired direction. For instance, in the DOA estimation technique SRP-PHAT, the weights $h_c(t)$ are calculated as to minimize the contribution of the magnitude of the signal, whereas the time-delays τ_c are adjusted to steer the sensor array to every angle in a grid.

In contrast, our aim is to estimate filter coefficients and steering delays jointly with the network weights, by optimizing a localization likelihood target. A bank of $F \times B$ multi-channel filters $h^{f,b} = \{h^{1,1}, h^{1,2}, \ldots, h^{1,B}, h^{2,1}, h^{2,2}, \ldots, h^{F,B}\}$ where $h^{f,b} \in \mathbb{R}^{P \times C}$ is deployed to simultaneously steer the microphones to multiple directions. P is the length of each beamformer and C the number of microphone channels. A single, spectrally filtered signal $y^f(t)$ is passed to several beamformers, as to encourage each spatial filter $h^{f,b}$ to specialize to a specific frequency content and spatial look. The beamforming operation is described by:

$$z_b^f(t) = \sum_{c=0}^{C-1} \sum_{p=0}^{P-1} y_c^f(t) \cdot h_c^{f,b}(t-p) \tag{3.9}$$

$$= \sum_{c=0}^{C-1} y_c^f(t) * h_c^{f,b}, \tag{3.10}$$

where $z_b^f(t) \in \mathbb{R}^{M \times 1}$ follows from convoluting the filtered multi-microphone signal $y^f(t) \in \mathbb{R}^{M \times C}$ from the `fConv` stage with the beamformer $h^{f,b} \in \mathbb{R}^{P \times C}$ and summing over channels. It is worth noting that the steering delay τ_c of traditional filter-and-sum beamformers of Equation 3.8 is implicitly absorbed inside $h_c^{f,b}$. Therefore, our method does not require explicit estimation of the steering delays τ_c, and the coefficients $h^{f,b}$ of the spatial filters are optimized jointly with the network.

For comparison, our experiment include fixing $h^{f,b}$ to accommodate delay-and-sum beamformers pointing to uniformly spaced directions in the frontal hemisphere. In the latter case, identical beamformers are used across frequencies, i.e. $h^{f,b} = h^{1,b}$, $\forall f \in F$, and differences in damping across microphones due to propagation effects are neglected (*far-field* assumption). Delay-and-sum is preferred to minimum variance distortionless or generalized eigendecomposition beamformers for its simplicity [69]: being completely model based, delay-and-sum does not require estimating the spatial correlation matrices of the signals.

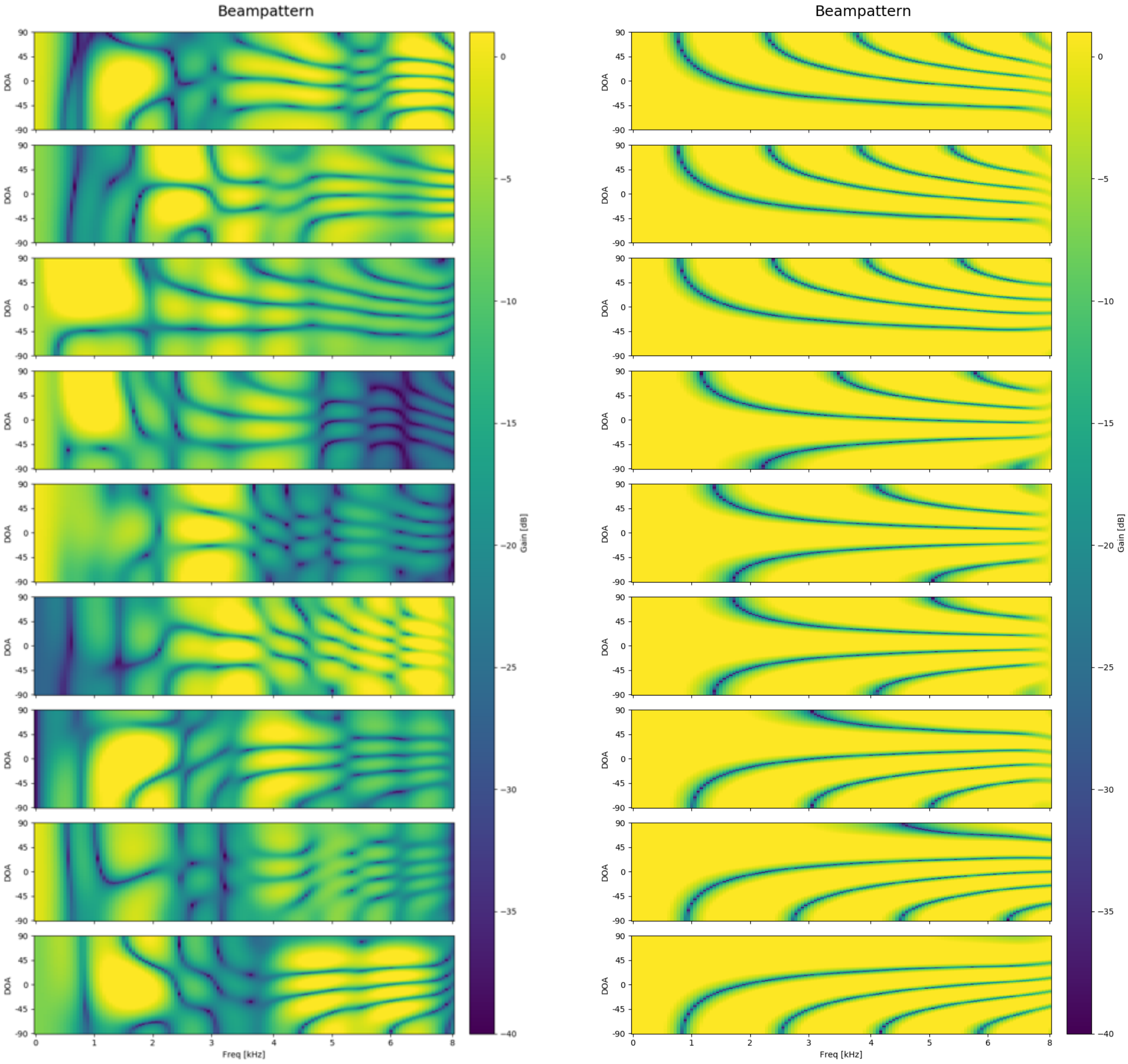


Figure 3.10: Beampatterns. Log magnitude of $h^{f,b}$ filters from the second block, `sConv`, as a function of frequency (horizontal axis) and DOA (vertical axis) assuming incoming planar waves. The response for filters learned by the network is shown on the left, while the fixed delay&sum beamformers are shown on the right.

3.3.1.3 Extraction of local time patterns (`tConv`)

The output of `sConv` layer is a set of $B \times F$ single-channel signals $z_b^f(t) \in \mathbb{R}^{M \times 1}$ having the same sampling rate as the input signal.

Differently from the architecture presented in [57] for speech recognition, we do not perform pooling directly after the beamformers, to preserve the fine-grained shifts which allow DOA estimation. Instead, to detect local time patterns in the band-passed, spatially directional signals, these are processed with batch normalization followed by a non-linearity, and fed to a sequence of four identical 1D convolutional layers [52], each having R taps. Such filters are designed to be much shorter than the ones from the `fConv` and `sConv` layers, such that $R \ll P \ll N$.

Each of the four layers of the `tConv` stage involves convolution over the time axis, batch normalization, non-linearity, and max pooling every 4 samples. The max pooling deterministic block gradually reduces the dimensionality, allowing the network to extract information at different time scales. Lastly, a global max pooling ensures that the time dimension is consumed completely, yielding feature maps w of size $w \in \mathbb{R}^{Q \times F \times B}$, where Q is the number of convolutional filters of the 1D convolutional layers.

3.3.2 Temporal context aggregation

The feature extraction described above is executed over a small window of M samples, and yields a tensor $w \in \mathbb{R}^{Q \times F \times B}$ per each frame. The window is then shifted by $M(1-o)$ samples, where $0 < o \leq 1$ is an overlap factor, and the feature extraction is repeated T times. By concatenating the outputs w_i, $i = 1, 2, \ldots, T$ given by the feature extraction block, one obtains a sequence $W = [w_1 w_2 \ldots w_T] \in \mathbb{R}^{Q \times F \times B \times T}$. The temporal aggregation block implements a series of 1D convolutional layers which act over the time-frame dimension only [59]. Each layer is composed convolutional filters of length S acting over the time-frame axis, batch normalization, non-linearity and max pooling with a kernel size 2. Before max pooling, each side of the input is zero-padded by one sample. Each convolutional layer reduces the time-frame dimension from T' to $T'' = \text{floor}(T'/2) + 1$, where T' is the size of the time-frame dimension at its input. Lastly, a global max pooling layer ensures that the time-frame dimension is completely consumed. In this way, the output of the temporal context aggregation block is a tensor V of size $V \in \mathbb{R}^{A \times F \times B}$, where A is the number of filters of the last convolutional layer of the block. Choosing the length of each conv filter to be $S < T$, the number of convolutional layers in this block is set to $\log_2(T)$, the maximum possible in relation to the number of time frames T.

In practice, this block is similar to the `tConv` stage described above, with three key differences, 1. convolution is performed over different *time-frames*, not *samples*, 2. convolution is applied without zero padding and 3. the number of layers in this block depends on the total number of time-frames T selected from each audio sample.

3.3.3 Likelihood map regression

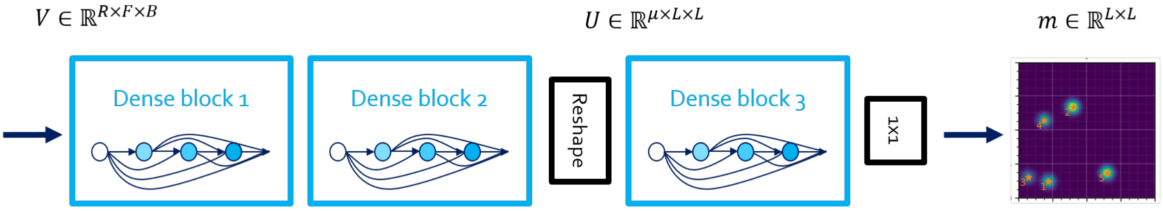

$V \in \mathbb{R}^{R\times F\times B}$
$U \in \mathbb{R}^{\mu\times L\times L}$
$m \in \mathbb{R}^{L\times L}$
Dense block 1
Dense block 2
Reshape
Dense block 3
1x1
2
4
3
1
5

Figure 3.11: A schematic of the regressor.

To upsample the low dimensional output $V \in \mathbb{R}^{A\times F\times B}$ from the temporal aggregation block into the desired output map $\hat{m} \in \mathbb{R}^{L\times L}$, we employ two dense blocks, followed by a reshape operation and a third dense block. A 1×1 convolution followed by a global max pooling yields the final 2D likelihood map (see Figure 3.11). The ith dense block contains D_i convolutional layers, each having k_i convolutional filters [64]. Every layer of a block contains batch normalization, nonlinearity and convolution with padding to preserve the size, and receives as input the concatenation of all feature-maps of the preceding layers (see Figure 3.6 for an example). To account for the fact that angles of ±180° coincide conceptually, but are in practice at the opposite sides of the likelihood maps (see subsection 3.1.1), *reflection padding* is chosen in place of traditional zero padding along the angular dimension. In this way, the network receives a hint about the cylindrical nature of the likelihood maps.

For the first two dense blocks, the number of layers D_i and corresponding filters k_i are chosen as to produce a tensor of size $V' \in \mathbb{R}^{A'\times F\times B}$ which can then reshaped to $U \in \mathbb{R}^{\gamma\times L\times L}$. The reshape operation is only possible if the product of the dimensions of the tensors are equal: $A' \cdot F \cdot B = \gamma \cdot L^2$. In the original proposal of the *dense upsampling* strategy [70], L is the width (and height) of the final output maps, and γ is the number of desired output maps. Although our network always produces a single map $\hat{m} \in \mathbb{R}^{L\times L}$, leaving γ as a configurable hyperparameter allows us to easily control the number of learnable elements in the overall architecture, and consequently the learning capacity.

Once the number of layers $D_1 = D_2$ inside the first two blocks and the number of latent maps γ are fixed, the number of filters $k_1 = k_2$ must be computed as to allow for the reshaping from V' to U:

$$k_i = \frac{1}{2D_i}\left(\frac{\gamma \cdot L^2}{B \cdot F} - A\right), \qquad i = 1, 2$$

where L_i is the number of layers inside the ith dense block, γ is the number of latent maps, L is the width of the final likelihood map, B and F are respectively the numbers of beamformers and frequency channels, and A is the number of feature-maps from the temporal aggregation block.

After the reshaping, a third dense block with $D_3 = 4$ layers and $k_3 = 20$ filters per layer, followed by a 1×1 convolution and a global max pooling along the channel dimension yield the desired 2D likelihood map $\hat{m}$. Lastly, element-wise rectified linear unit $\max(0, x)$ and clipping operator $\min(1, x)$ restrict the values of the map to the target range $[0, 1]$.

3.3.4 Post-processing

3.3.4.1 Matched filtering

The noise in the likelihood maps estimated by the networks can be reduced by applying a matched filter, i.e. a filter that has the same shape as the basis functions that form the map. As described in subsection 3.1.1, the ground truth maps are generated as a superimposition of Gaussian-like functions. Therefore, the impulse response of the desired matched filter h_{mf} can be calculated from Equation 3.5 by substituting $p_0^{(s)} = [d_{\max}/2, 0]$, $I = 0$ and the width parameter σ used for the generation of the maps. The resulting likelihood map h_{mf} corresponds to the desired filter, and can be convolved with the output $\hat{m}$ of the DNN to refine the estimate.

3.3.4.2 Peak picking

The I sources correspond to I peaks in the likelihood map m. To retrieve the peaks, we rely on a publicly available algorithm that mimics the `findpeaks` function of Matlab[2]. All parameters are left to default, except for `threshold_rel=1/20` to discard possible peaks if they are more than 20 times smaller than the highest peak in the map.

[2] https://nbviewer.jupyter.org/github/demotu/BMC/blob/master/notebooks/DetectPeaks.ipynb

4 Numerical results

The previous chapter described the proposed strategy for the identification of reflective surfaces based on real and virtual sources localization. In the following, the performance of the proposed method is tested on both simulated and real data. A random search procedure used for selecting the network hyperparameters (number of convolutional filters, length of each filter etc.) is also presented.

4.1 Implementation details

The DNN is trained in mini-batches using Adam, a first-order extension of stochastic gradient descent that share properties of second-order methods [71]. The code is implemented using the open-source deep learning library NNabla[1]. To stabilize the estimates of the individual moments before commencing updating the parameters, the global learning rate is set to $\mu = 0$ during the first epoch. At epoch 2, the learning rate is changed to $\mu = 1\text{e}{-3}$, and successively reduced by a factor 10 after every 150 epochs without decreases on the validation loss, until it reaches $\mu_{\text{min}} = 1\text{e}{-6}$. To limit numerical errors in the evaluations of the gradients, the stereo input waveforms are normalized such that the variance per each time sample across training points is approximately one [72]. The 1s long audio waveforms are then downsampled from $f_s = 48$kHz to 16kHz to reduce the computational burden. Next, the waveforms are rescaled by a random factor $\beta \in [0.5,\ 1.5]$. This procedure is fundamental to force the network to elicit source distance based on amplitude ratio across microphones, rather than on absolute volume. As a last preprocessing step, we perform *sampling without replacement* by randomly selecting T consecutive, overlapping frames of length $M = 320$ from each audio file per each iteration. Selecting the frames can either be achieved using a rectangular or a Hanning window without consequences to the final accuracy, as shown in section 4.3. The effect of varying the number of time frames T is investigated in section 4.6.1.

With the sole exception of the last layer, leaky rectified linear unit nonlinearity is used throughout the network, with negative slope $\alpha = 0.2$. To limit overfitting, we introduce $\ell 2$ regularization with weight $\lambda = 1.6\text{e}{-6}$, early stopping after 350 epochs without improvements on the validation set, and dropout. Zeroing of entire channels - known as *spatial dropout* [61] - is also applied with probability $p = 0.3$ on the output of the `fConv` block, to discourage the network from relying on a particular frequency range only.

[1] https://github.com/sony/nnabla

In most cases, the values of the hyperparameters of the DNN have little influence on the final accuracy, as shown in the next section. However, because of the dense upsampling strategy, the number of learnable parameters depends greatly on the output shape. The size of the output likelihood map is then set to 128 × 128 for experiments involving none or a single reflective surface, whereas larger 256×256 maps are used when training on the four reflective surfaces dataset `WGN4`. The idea behind this choice is to use a faster network ($L = 128$) with $\approx$ 257k learnable parameters on the smaller `WGN0` and `WGN1` datasets, and a more complex DNN with $L = 256$ and $\approx$ 1.2M learnable parameters for the larger dataset `WGN4`. In the following, the two networks will be referred to as `NN128` and `NN256`, respectively.

4.2 Evaluation metrics

The accuracy of the network is evaluated in terms of distance between the ground truth locations of real and virtual sources $\mathbf{p} = [p_1, p_2, \ldots, p_I] \in \mathbb{R}^{I\times 2}$ and the peaks extracted from the likelihood map $\hat{\mathbf{p}} = [\hat{p}_1,\ \hat{p}_2, \ldots, \hat{p}_I] \in \mathbb{R}^{I\times 2}$, where $p_i = [\rho_i, \phi_i]$. For brevity, we omit the superscript $^{(s)}$ for the ground truth locations. To evaluate the accuracy of the proposed framework, I peaks corresponding to I sources are detected from the estimated map $\hat{m}$. Among all the possible $I!$ permutations of the unsorted array of detected peaks $\hat{\mathbf{p}}$, the ordering that yields the lowest mean absolute error (MAE) is selected, where the error is calculated as:

$$\begin{aligned} \mathrm{MAE}(\mathbf{p}, \hat{\mathbf{p}}) &= \frac{1}{I}\sum_{i=1}^{I} |p_i - \hat{p}_i| && (4.1)\\ &= \frac{1}{I}\sum_{i=1}^{I} |\rho_i - \hat{\rho}_i| + |\phi_i - \hat{\phi}_i|. && (4.2) \end{aligned}$$

In addition, we will make use of an accuracy indicator, defined as the percentage of (real or virtual) sources in the evaluation set which are localized within 2m and 30° from their ground truth positions.

4.3 Random search of hyperparameters

A drawback of neural networks is the large number of configurable variables, known as hyperparameters, that determine the structure of the computational graph. Empirically finding the best combination for a given task involves a search over a often high-dimensional hyperparameter space. Performing grid-search with sufficient granularity is computationally expensive, in that a complete training is needed to evaluate the performance of each configuration of hyperparameters [17]. As an alternative, it is possible to conduct trials where the values of all hyperparameters are chosen at random from given supports, and select the best configuration according to a given metric. Such procedure is known as *random search* [73]. In our approach, 37 DNNs with different

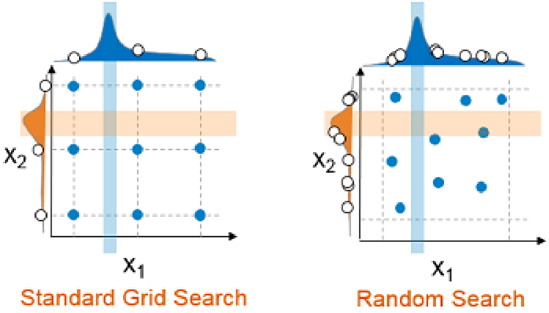


Figure 4.1: Grid and random hyperpameter search. Adapted from [2].

hyperparameters are trained on the single surface dataset `WGN1` to identify the best performing configuration and investigate possible correlations between hyperparameters and the final accuracy. All models use 128×128 output maps, a span of $T = 4$ time frames and batch size 8. Learning rate is reduced by a factor 10 with every 75 epochs without decrease on validation loss, while training is stopped after a max of 300 epochs or 100 epochs without improvements.

Description	Name	Admitted values	Correlation	p-value
# taps `fConv`	N	$[8, 1024]$	0.05	0.75
# taps `sConv`	P	$[4, 64]$	-0.21	0.22
# taps `tConv`	R	$[3, 7]$	0.26	0.12
# filters `tConv`	Q	$[8, 128]$	0.46	< 0.01
# filters 1×1		$[8, 128]$	-0.15	0.39
Blob width	σ	$[0.1, 0.5]$	0.19	0.26
Window type		$\{\text{none}, \text{hann}\}$	0.01	0.96
Chunk lenght	M	$[N, 1024]$	0.10	0.55
Overlap	o	$[0, 0.75]$	0.05	0.77
$\ell 2$ regular.	λ	$[1\text{e}{-9}, 1\text{e}{-5}]$	0.19	0.27
DO		$\{0, 1\}$	-0.32	0.05
DO, p		$[0.1, 0.5]$	-0.11	0.53
Spatial DO		$\{0, 1\}$	-0.71	< 0.01
Spatial DO, p		$[0.1, 0.5]$	-0.41	0.01

Table 4.1: Details of random hyperparameter search. DO stands for dropout. Square brackets denote uniform distribution over ranges of values. Curly brackets denote discrete distribution over values in the sets.

Table 4.1 presents the details of the random search experiment. Pearson's correlation coefficient between hyperparameters under test and network accuracy is calculated, together with the corresponding p-value, to highlight possible linear relationships. The

[2] https://blogs.sas.com/content/subconsciousmusings/2016/09/20/local-search-optimization-for-hyperparameter-tuning/

p-value roughly correspond to the probability that two uncorrelated Gaussian distributions would yield the resulting correlation coefficient (null hypothesis). In line with previous research [18, 42], the low correlation between most variables under test and network accuracy suggests that a large number of widely different DNNs can produce similar results. As an exception, there is a significant positive correlation between accuracy and number of convolutional filters in the `tConv` block of the feature extractor, indicating that a higher number of filters is beneficial to extract meaningful patterns in the spectrally and spatially filtered signal. It is also seen how dropout and spatial dropout, applied here to all layers of temporal aggregation and regressor blocks, tend to worsen the final accuracy.

The hyperparameter set yielding the best performing network (see Appendix C for details) is selected for the experiments of the next sections.

4.4 Anechoic room

The first two experiments deal with the scenario of a single source emitting white Gaussian noise in an anechoic environment (dataset `WGNO`). First, it is shown how the use of directional microphones improves localization by reducing front-back ambiguity (subsection 4.4.1). Second, sensitivity studies are conducted, where the DNN is trained on `WGNO`, but tested on datasets simulating microphone arrays of either smaller or larger sizes (subsection 4.4.2).

4.4.1 Subcardioid vs omnidirectional microphones

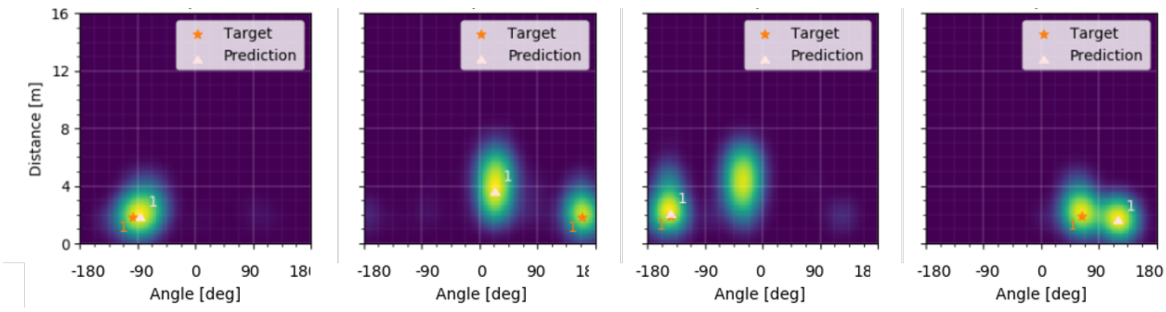


Figure 4.2: Typical DNN output for single noise source in anechoic environment, in polar coordinates. The red stars mark the target locations, while the white triangles are the predicted locations. The first prediction on the left corresponds to a source lying on the same line as the microphones; its location is uniquely determined. In all other predictions, the DNN assigns some likelihood to both the actual source location and its reflected counterpart. This leads to front-back errors in the second picture from the left, and the last to the right.

As a first experiment, `NN128` is trained on two sets of anechoic recordings that only differ for their microphone array directivity patterns. Namely, `WGNO_omni` employs omnidirectional microphones, with $a_{\text{omni}}(\phi) = 1 \ \forall \phi \in [-\pi, \ \pi]$, whereas `WGNO` employs subcardioid microphones with DOA-dependent attenuation, as in Equation 3.1.

Dataset	Mean absolute error		Detections (%)		FB (%)
	dist (m)	ang (deg)	< 2m	< 30°	
WGNO_omni	0.8 ± 0.8	31 ± 48	88	73	38
WGNO	$\mathbf{0.4 \pm 0.6}$	$\mathbf{7 \pm 19}$	**95**	**95**	**11**

Table 4.2: Results for localization of real source in anechoic conditions.

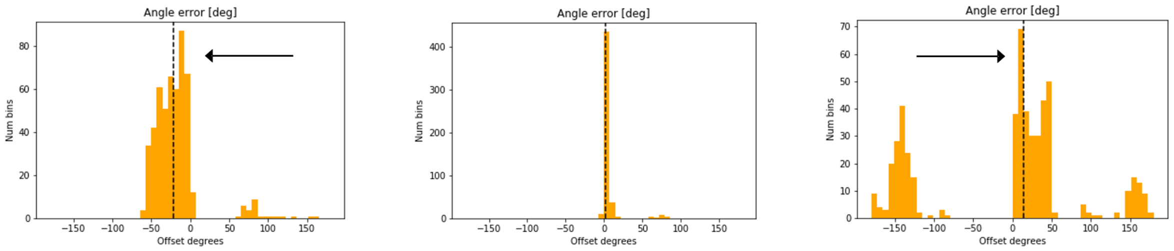
It can be seen from Table 4.2 how the angle-dependent volume attenuation of subcardioid microphones leads to higher accuracy in all localization metrics compared the omnidirectional case. When omnidirectional sensors are used, there is an ambiguous solution located specularly to the real one with respect to the microphones axis (section A.1). If subcardioid microphones are used instead, the ambiguous solution will not always exist. As a consequence, subcardioid microphones drastically reduce front-back confusions, and they will be preferred to omnidirectional ones in all the following experiments.

4.4.2 Sensitivity to array size mismatch

The second experiment is designed to inspect the behavior of the DNN when the microphone arrays used for train and test phases have different sizes. In localization based on time-difference of arrival (TDOA) and amplitude ratio across microphones (section A.2), assuming a wrong microphone spacing would bias the localization estimates in a predictable way. For instance, assume that a stereo array with spacing d measures a TDOA τ for a source located at $\phi = 90°$. A smaller array of size $d/2$ will measure, for the same source, a TDOA $\tau/2$, corresponding to smaller angles. A similar phenomenon occurs with amplitude ratio, which is generally closer to 1 for smaller sensor arrays, biasing the predictions towards higher distances.

As a consequence, it should be possible to predict the biases occurring in the localization estimates in presence of a mismatched microphone array. In this experiment, NN128 is trained on a subset of WGNO that comprises sources lying in the first quadrant only, i.e. sources that satisfy $0 < \phi < 90°$. The stereo array used for training is $d = 10$cm long. After the weights have been fixed, the DNN is tested on simulated recordings from smaller ($d = 5$cm), matched ($d = 10$cm) or bigger ($d = 15$cm) microphone arrays.

The distribution of the localization error for stereo arrays of different sizes are shown in Figure 4.3, and summarized in Table 4.3. As expected, testing the DNN on recordings provided by a smaller array results in predictions which tend to overestimate the source distance and underestimate its DOA, compared to the matched case. Similarly, using larger arrays causes the network to bias the estimates towards closer distances and larger angles. Additionally, it is noted from Table 4.3 how the distance predictions have an undesired bias of 0.12m in the matched case, possibly due to the uneven distribution



(a) Smaller ($d = 0.05$m)

(b) Matched ($d = 0.10$m)

(c) Larger ($d = 0.15$m)

Figure 4.3: Angular error distribution for mismatched test array size. The dashed black line represents the median, and the arrows indicate the direction of the bias.

of sources locations in the training set.

Some insight can also be gained from the error distribution of Figure 4.3, particularly for the case of larger array (Figure 4.3c). Due to the greater distance between microphones occurring during the test phase, some sources will induce TDOAs which are strictly longer than any delay occurring in the training set. It can be argued that the numerous error outliers in Figure 4.3c are to be attributed to the presence of such unexpectedly long delays in the test data.

Microphone array	Error median	
	distance (m)	angle (deg)
smaller, $d = 0.05$m	0.28	-21
matched, $d = 0.10$m	0.12	2
larger, $d = 0.15$m	0.00	14

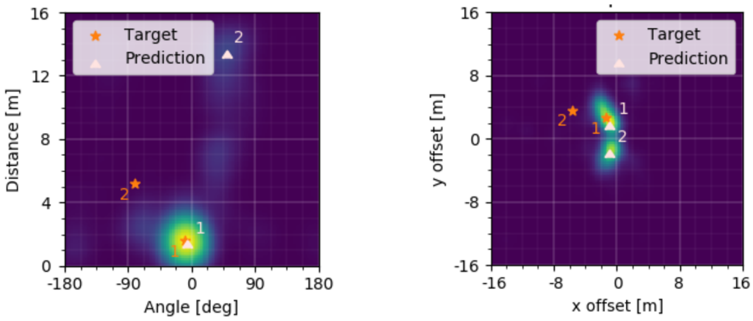
Table 4.3: Localization with matched or mismatched microphone arrays.

4.5 Single reflective surface

Having shown that the proposed DNN can successfully localize a noise source in anechoic environments, the next set of experiments deals with the more complex task of localizing a reflective surface by estimating the positions of real speaker and corresponding virtual source ($I = 2$). The reflective surface stands perpendicularly in the middle of the line connecting the points. First, we compare several variants of the proposed DNN architecture, and observe that separated, learnable filterbanks perform better than handcrafted ones, confirming previous findings [52, 57]. Second, we identify front-back ambiguity as one of the main sources of uncertainty in the proposed framework, and show how constraining the sources to be in front of the array improves localization considerably.

4.5.1 Architecture variations

As a first experiment, several variants of the proposed architecture are trained on the WGN1 dataset and compared in terms of their localization accuracy. The first variant under comparison is denoted as 'Cartesian' in Table 4.4. As opposed to the likelihood maps in subsection 3.1.1, where the emitters are identified with their polar coordinates, 'Cartesian' maps report vertical and horizontal displacement of sources with respect to the microphones (Figure 4.4). The ground truth targets $m \in \mathbb{R}^{L\times L}$ are calculated by virtually translating and rotating the reference system such that the microphones lie in horizontal symmetry with respect to the center of the map $O = (L/2, L/2)$.



(a) Polar

(b) Cartesian

Figure 4.4: The same geometrical settings, where real and virtual sources are in front of the microphone array, is represented with polar (on the left) or Cartesian coordinates (right).

Additional variants of the NN128 network are obtained by fixing certain stages of the feature extractor block to accommodate deterministic filterbanks ('Gammatone', 'Delay & sum'), or merging the fConv and sConv stages into a single learnable layer ('Unfactored'). Namely, the 'Gammatone' variant consists in replacing the fConv stage of the feature extractor block with a fixed Gammatone filterbank having $F = 16$ filters distributed between 512Hz and 8kHz. In the 'Delay & sum' DNN, the sConv stage is substituted by $B = 16$ fixed delay & sum beamformers pointing to uniformly spaced directions in a grid, per each output of the fConv stage, as described in subsubsection 3.3.1.2. The last variant under analysis consists of merging fConv and sConv stages into a single, learnable convolutional layer with a total of 256 filters of length 65. The filter length is chosen to keep the number of learnable parameters similar to the other variants, and the name 'unfactored' follows from [57]. Based on the fact that filter-and-sum beamformers can be viewed as performing band-pass filtering followed by narrow-band beamforming [58], joining the filterbank and beamforming layers into one could yield similar results to the 'factored' case, denoted here as NN128.

The proposed architectures are trained on the WGN1 dataset, and tested on unseen WGN and speech recordings. All variants are equally accurate in localizing noise sources, with 'full' detection percentages ranging between 47% and 51%, meaning that these fractions of sources lie closer than 2m and less than 30° from the corresponding

DNN	Test data	Mean absolute error		Detected (%)		
		dist (m)	ang (deg)	< 2m	< 30°	full
NN128	WGN1	2.3 ± 2.9	38 ± 48	67	62	51
Cartesian		2.2 ± 2.6	40 ± 48	68	56	47
Gammatone		1.8 ± 2.2	42 ± 52	72	59	50
Delay & sum		1.6 ± 1.8	47 ± 52	75	55	47
Unfactored		1.7 ± 2.1	40 ± 49	72	60	51
NN128	SPEECH1	2.8 ± 2.7	75 ± 53	52	27	14
Cartesian		3.3 ± 2.3	73 ± 51	34	26	9
Gammatone		5.4 ± 3.3	60 ± 52	18	41	9
Delay & sum		2.5 ± 1.7	74 ± 54	40	31	11
Unfactored		3.8 ± 3.1	66 ± 55	32	37	11

Table 4.4: Performance of several variants of the original architecture on WGN or speech datasets with one reflective surface.

targets. When the DNNs, trained with WGN samples only, are tested on speech signals, the general performance degrades significantly, with the proposed NN128 network still scoring best in terms of detection rate. Mapping TDOAs and amplitude ratios across microphones to angles of arrival and distances, respectively, is easier than mapping the same stereo cues to Cartesian coordinates in the likelihood maps. This might lead to the superior generalization performance of polar coordinates compared to Cartesian.

4.5.2 The front-back ambiguity

Angle-dependent microphone directivity patterns are beneficial for localization, but do not solve the front-back ambiguity, as discussed in subsection 4.4.1. In completely stationary settings, discriminating sounds coming from the front or from the back is also problematic for humans, despite the additional spectral cues provided by binaural hearing [38]. Constraining the sources to lie in front of the array should ease localization significantly.

Dataset		Mean absolute error		Detected (%)			FB (%)
Train	Test	dist (m)	ang (deg)	< 2m	< 30°	full	
All	Front	2.1 ± 2.3	41 ± 50	62	59	48	42
Front	**Front**	$\mathbf{1.2 \pm 1.4}$	$\mathbf{7 \pm 15}$	**80**	**95**	**80**	**0**

Table 4.5: All stands for WGN1, whereas Front is a subset of WGN1 featuring sources in the positive hemisphere only, i.e. $-90° < \phi < 90°$.

In this experiment, it is shown how NN128 trained on a *subset* of WGN1 featuring sources in the positive hemisphere (denoted as 'Front' in Table 4.5) significantly outperforms its counterpart trained on the complete dataset ('All'). The two identical networks, trained on different dataset, are tested on unseen data where sources and reflective surfaces face the microphones. As seen from Table 4.5, eliminating the ambiguous solution by infusing prior knowledge about the source angle in the training set drastically improves localization: distance error is reduced to half, while angular error is 5 times smaller for the DNN trained on the positive hemisphere only.

It is worth noting that humans, despite the additional spatial and spectral cues given by binaural hearing, can localize a single, static speaker with an average distance error of about 1.3m, and angular error close to 17° [74]. We are not aware of any experiment that measures human performance in localizing reflective surfaces.

4.6 Multiple reflective surfaces

The last set of experiments investigates the performance of the proposed algorithm in localizing real and first-order image sources of rectangular rooms. Only the lateral surfaces are considered, yielding a total of $I = 5$ sources. The effects of varying the number of time frames T, and the number of spectral and spatial filters of the fConv and sConv stages are analyzed. To verify the generalization capabilities of the framework, each configuration is tested not only with simulated WGN and speech recordings, but also on real-world data. The experiments are conducted by training the larger NN256 architecture on the WGN4 dataset, as mentioned in section 4.1.

4.6.1 Influence of time context

The temporal aggregation block of the proposed DNN receives as input the features extracted from multiple time frames, and summarizes them into a unified representation. For multichannel, reverberant white Gaussian noise, a larger time context could either help the DNN capturing long term cross-correlations due to the reflections, or improve the estimation of statistical properties of the signals via averaging. A higher number of windows implies an increased computational cost. In this experiment, the NN256 architecture is trained with varying number of time frames $T \in \{1, 4, 8, 16\}$, fixed Gammatone filterbank and batch size 4. Given the length of a time window $M = 320$, the overlap factor $o = 0.4$ and the sampling frequency $f_s = 16\text{kHz}$, a single frame corresponds to 20ms of audio, while $T = 16$ frames amount to 200ms.

Figure 4.6 displays the number of detected sources per each configuration, where each network is tested on unseen recordings of reverberant noise, synthetic speech, and real speech signals. Localization of real and virtual noise sources appears to highly benefit from longer time contexts: the detection rate increases steadily from 32% to 45% when T increases from 1 to 8. Further extending T to 16 frames leads to minor improvements. Localization with synthetic speech signals follows a similar trend. The relationship between time context and accuracy is less linear for real speech, but the best results are again recorded for the largest T. Interestingly, the performance of the

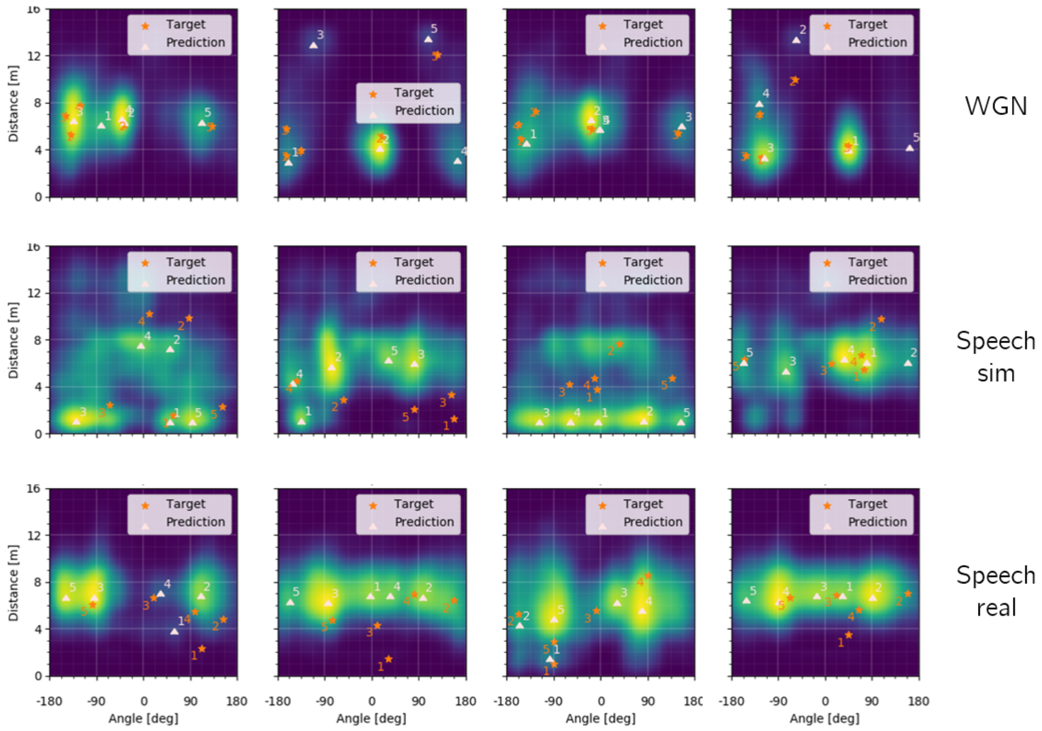


Figure 4.5: Typical DNN output for noise (top row), artificial speech (middle) or real speech (bottom row) sources in rectangular rooms. The predictions are visibly more accurate for the WGN source.

proposed architecture is similar for synthetic and real speech recordings, suggesting that domain adaptation from simulated RIRs to real-world data is indeed possible, in spite of the mismatches in the array size and microphone directivity.

4.6.2 Varying the number of spectral and spatial filters

In all previous experiments, the number of spectral filters F and spatial filters B has been fixed to 16 to limit training time and computational burden. This experiment analyzes the changes in localization accuracy when the spectral filterbank has $F \in \{16, 32, 64, 128\}$ learnable convolutional filters and $B = 8$ beamformers. Alternatively, F is set to $F = 16$ and the number of learnable beamformers per frequency band is varied, $B \in \{16, 32, 64\}$. The batch size is 3 and the time context $T = 4$. For each (F, B) network configuration, the number of learnable parameters is kept approximately equal by adjusting the number μ of latent maps of the dense upsampling layer between $\mu = 1$ and $\mu = 8$ (see subsection 3.3.3). Similar to the previous experiment, the DNNs are tested on noise, simulated speech, and real speech signals.

Table 4.6 reports the number of fully detected sources as a function of the number of filters in `fConv` and `sConv`, and the data type. As a general tendency, higher number

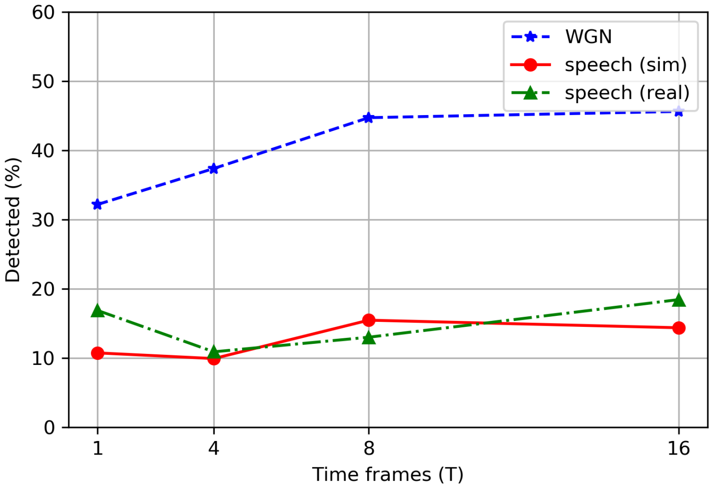


Figure 4.6: The effect of varying the time context T for different test data: white Gaussian noise (dashed blue line), artificially reverberated speech (solid red line) and real speech (pointed green line). The vertical axis represents the number of detected sources.

Dataset	Detected sources (%)						
	Spectral F				Spatial B		
	16	32	64	128	16	32	64
WGN	44	47	49	48	49	49	48
speech (sim)	15	18	19	18	10	11	16
speech (real)	12	13	16	24	22	15	12

Table 4.6: Percentage of correctly localized sources as a function of the number of learnable spectral and spatial filters, for three different datasets.

of spectral filters F consistently improve localization for all types of audio data. Such trend is less apparent for simulated noise and speech sources, where the detection rate improves by 3 to 4 percent points, and more marked for real speech, where the detection improves from 12% ($F = 16$) to 24% ($F = 128$). On the other hand, changing the number of beamformers B has minor consequences for localization of noise sources, and leads to opposing effects for real and simulated speech sources. A possible cause is in the mismatched array size, which is 10cm for the training set `WGN4` and the simulated speech, but only 9.2cm for real recordings. Higher number of beamformers might point to more closely spaced angles, leading to more severe over-fitting to a given array size. The small size of the real-speech dataset, less then three minutes in total, might also contribute to some of the fluctuations in the results.

4.7 Discussion

This chapter analyzed the performance of the proposed neural network for localization of real and image sources using stereo microphones. To begin with, random hyperparameter search allowed us to identify a DNN architecture that performed reasonably well in terms of localization accuracy. The low correlation between parameter values and network accuracy revealed that widely different network configurations can produce similar results. Next, the algorithm was tested on the increasingly complex tasks of localizing a single source in an anechoic environment, a source and a single reflecting surface, or a source and four reflective surfaces. Numerical evidence suggested that the use of directional microphones improves localization and reduces front-back ambiguity, a major cause of uncertainty. To verify the assumption that the proposed DNN estimates distance and DOA of sources by means of time and amplitude differences across microphones, we conducted experiments with sensor arrays of mismatched sizes. The estimation biases occurred when testing the DNN with different microphone spacings were in accordance with our hypothesis. Next, the role of the feature extractor block was examined by substituting the learnable weights in the `fConv` stage with a Gammatone filterbank, or the spatial filters of the `sConv` stage with delay & sum beamformers. Another variation considered merging spectral and spatial filters into a single block. None of the proposed variants performed significantly better than the original data-driven architecture. In addition, it was found that extending the length of the temporal context and raising the number of spectral filters can lead to a better localization of both noise and speech sources, whereas increasing the number of learnable beamformers does not improve the results.

In conclusion, the numerical experiments confirmed the feasibility of the proposed strategy for localization of reflective surfaces in simulated and real settings, despite the challenges presented by the low number of microphones and the absence of synchronization between emitter and receivers.

Conclusion and future work 5

Conclusion

Localization of reflective surfaces from stereo recordings enables spatially-aware surround upmixing. By knowing the locations of first-order image sources in an enclosure, it is possible to generate surround musical content that maintains the reverberation characteristics of the original recordings. In contrast to the traditional room geometry estimation scenario, in typical audio recordings *i*) only a compact array of $C = 2$ microphones is available *ii*) emitter and receivers are not synchronized, so that the absolute time-of-arrivals (TOAs) at the sensors are unknown, and only time-difference-of-arrivals (TDOAs) among microphones can be exploited, and *iii*) both the dry excitation sound $s(t)$ and the RIR $h(t)$ are unknown, and have to be blindly estimated from the reverberant output $x(t) = s(t) * h(t)$. With this challenging set of constraints, the room estimation problem is reshaped as a multiple source localization task, where the sources to be found are the real source and the image sources corresponding to each reflective surface.

This report proposed a supervised learning approach for source localization, where the parameters of a convolutional DNN modelled a mapping between reverberant recordings of white Gaussian noise and position likelihood of sources. In section A.2, it was shown how knowledge of the cross-correlation function between sensors is not sufficient to perform distance and angle localization. As a consequence, the raw audio waveforms were used as an input for the DNN. As for the DNN target, we introduced a 2D position likelihood encoding that admits a variable number of sources, and allows the loss term to take distance between estimate and target locations into account. The proposed architecture adopts data-adaptive spectral and spatial filterbanks, which are applied to multiple overlapping time frames. The features extracted from different frames are further processed by a temporal convolutional block, and reshaped to produce the desired position likelihood maps.

A random hyperparameter search procedure allowed us to identify a suitable network configuration. Additionally, it showed that large variations in the network architecture lead to minor changes in final accuracy. Numerical experiments validated the capabilities of the proposed algorithm in localizing multiple overlapping sources, and investigated the inner functioning of the method by testing it with different microphone array sizes, sensor directivity patterns and audio signal types. Empirical evidence indicated that the DNN localizes sources based on time- and amplitude-differences across microphones as expected, and highlighted front-back ambiguity as a prominent source of localization uncertainty. Additionally, it was found that larger spectral filterbanks and longer time contexts consistently improve localization. Lastly, the network was shown to perform reasonably on real speech recordings, although localization is significantly more accurate if white Gaussian noise excitation signals are used instead. In conclusion, despite the challenges presented by the low number of microphones and the absence of synchronization between emitter and receivers, the numerical experiments confirmed the feasibility of the proposed strategy for localization of reflective surfaces in both simulated and real settings.

Future work

This section lists some possible directions of further development for the proposed framework.

- Learnable width of Gaussian-like blobs in target maps. Instead of fixing the width σ of the Gaussian-like blobs when creating the target maps, it would be possible to assign individual widths to each blob in the map, then optimize their values jointly with the other parameters of the network, as in [75]. This would allow the DNN to expand or shrink the size of the target curves during the training process. Potentially, sharper and smaller Gaussian-like blobs could correspond to sources whose locations are known with lower uncertainty.

- Network target closer to actual objective. The likelihood maps produced by the network need to be followed by a peak picking procedure to assess the sources locations. It would be desirable to optimize a loss function that allows for end-to-end training. A starting point might be the procedure described in [76] for single source, where the peak picking is integrated into the computational graph by means of simple matrix operations.

- Another limitation of the proposed target maps, briefly mentioned in chapter 3, is that the use of rectangular convolutional filters does not allow to capture the cylindrical nature of polar coordinates. Circular or spherical convolution, which has been proposed for analysis of 3D objects, might be able to fully exploit the intrinsic periodicity of polar representation.

- It has been shown recently that small, dilated convolutional filters often outperform longer ones in audio classification and source separation [77, 42]. Similar approaches could be evaluated and compared with the proposed network for source localization.

- Better understanding of the proposed network could be achieved by investigating the variation of the filterbank response as a function of microphone spacing. Moreover, it would be of interest to show analytically how directional microphone patterns lead to better localization by reducing the number of ambiguous solutions.

Source localization with two receivers A

In the following, it is shown how a speaker can be localized in free-field conditions using stereo time-domain recordings. The speaker location is found, up to a front-back ambiguity, by combining amplitude and time-delay differences information at the sensor array.

A.1 Sound propagation in free-field

Consider the following two-dimensional example, where two sensors at locations $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$ record sound in free-field condition, i.e. without any obstacle. A point source located in $\mathbf{s} = (x, y)$ emits a signal $s(t)$ propagating in spherical waves. The receivers measure attenuated and delayed copies of $s(t)$,

$$x_c(t) = \alpha_c \cdot s(t - \Delta_c), \quad c = 1, 2 \tag{A.1}$$

where α_c is the distance dependent attenuation factor at the c-th channel and Δ_c the time delay. The attenuation α_c is inversely proportional to the distance between source and receiver, $\alpha_c = 1/\|\mathbf{s} - \mathbf{x}_c\|$; the time delay Δ_c is calculated as $\Delta_c = \|\mathbf{s} - \mathbf{x}_c\|/c$, where $c = 340\text{m/s}$ is the speed of sound. Therefore, the time difference of arrival (TDOA) between the received signals depends on the distances between the elements of the array and source, and it is defined as

$$t_{12} \triangleq \Delta_1 - \Delta_2 = \frac{\|\mathbf{s} - \mathbf{x}_1\| - \|\mathbf{s} - \mathbf{x}_2\|}{c}. \tag{A.2}$$

A.2 Source localization

Source localization consists of recovering the unknown source position $\mathbf{s}$ from measured signals $x_c(t)$, $i = 1, \ldots, C$, where C is the number channels of the array. With two sensors ($C = 2$), the source position can be retrieved, up to a front-back ambiguity, by combining amplitude ratio and TDOA information about received signals.

Amplitude The received signal at each sensor is attenuated according to the distance between sensor and source. The set of points that have a specified ratio of distances to two fixed points, $S_1 = \{\ \mathbf{s} \mid \|\mathbf{s} - \mathbf{x}_2\|/\|\mathbf{s} - \mathbf{x}_1\| = \alpha_1/\alpha_2 = k\}$, describes an Apollonian

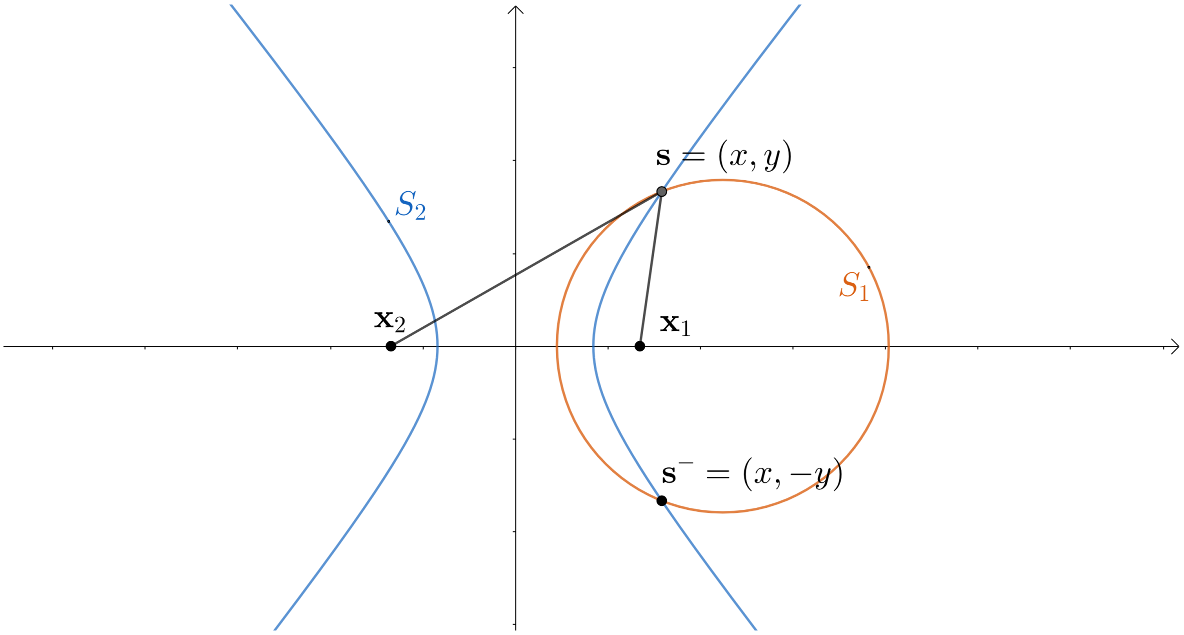


Figure A.1: Source localization with two receivers $\mathbf{x}_1$ and $\mathbf{x}_2$. The circle S_1 of constant amplitude ratio is drawn in orange for $k = 0.5$. The hyperbola S_2 of constant TDOA is depicted in blue. The actual source location $\mathbf{s}$ and the ambiguous solution $\mathbf{s}^-$ are found at the intersections of the two curves.

circle. [1] Thus, if an attenuation ratio $k = \alpha_1/\alpha_2$ has been measured, the source must lie on a circle S_1 (see Figure A.1).

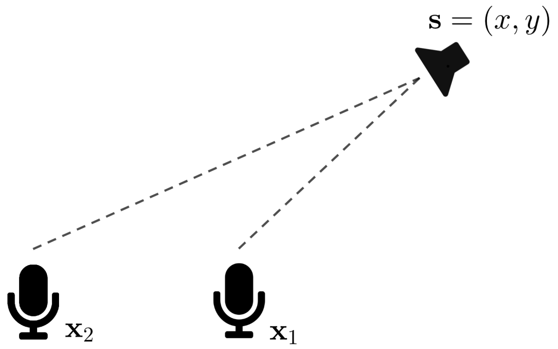
Time delay The set of possible locations of the source, for which the time difference of arrival t_{12} between received signals is constant, is a hyperbola $S_2 = \{\, \mathbf{s} \mid \|\mathbf{s} - \mathbf{x}_2\| - \|\mathbf{s} - \mathbf{x}_1\| = c \cdot t_{12}\}$ whose foci coincide with the receivers $\mathbf{x}_1$ and $\mathbf{x}_2$. In other words, if the measured TDOA across the two sensors is t_{12}, then $\mathbf{s} \in S_2$.

In the exceptional case where the source lies on the the perpendicular bisector of the receivers' axis, i.e. the source is exactly at 90° or 270°, the TDOA is zero, the distances to the two sensors are the same, $\|\mathbf{s} - \mathbf{x}_2\| = \|\mathbf{s} - \mathbf{x}_1\|$, and $\mathbf{s}$ cannot be found. In all other cases, the circle S_1 of constant amplitude ratio and the hyperbola S_2 of constant TDOA intersect in two points $\mathbf{s}$ and $\mathbf{s}^-$, who are symmetric with respect to the sensors axis (see Figure A.1). The source is found in one of the two points.

[1] https://en.wikipedia.org/wiki/Apollonian_circles

B Recovering distance information from multi-channel audio

In this appendix, it is shown how the distance of an audio source cannot be recovered from the cross-correlation between the signals received at the microphones. If the emitted signal $s(t)$ is unknown, the cross-correlation function does not contain enough information to fully localize a source in two dimensions. Consider the following example,



where two sensors at locations $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$ record sound in free-field condition, following the same model as in A.1. A speaker at position $\mathbf{s}$ emits a sound $s(t)$. The signal $s(t)$ is a wide-sense stationary (WSS) random process with zero mean and variance $E[s^2(t)] = \sigma^2$. Moreover, it is uncorrelated, with auto-correlation function

$$R_{ss}(\tau) = E[s(t)s(t+\tau)] = \sigma^2\delta(\tau).$$

The receivers measure attenuated and delayed copies of $s(t)$,

$$x_c(t) = \alpha_c\, s(t - \Delta_c), \quad c = 1, 2 \tag{B.1}$$

where α_c is the distance dependent attenuation factor at the c-th channel and Δ_c the time delay. The attenuation α_c is inversely proportional to the distance between source and receiver, $\alpha_c = 1/\|\mathbf{s} - \mathbf{x}_c\|$; the time delay Δ_c is calculated as $\Delta_c = \|\mathbf{s} - \mathbf{x}_c\|/c$, where

$c = 340\text{m/s}$ is the speed of sound. Therefore, the time difference of arrival (TDOA) between the received signals depends on the distances between the elements of the array and source, and it is defined as

$$t_{12} \triangleq \Delta_1 - \Delta_2 = \frac{\|\mathbf{s} - \mathbf{x}_1\| - \|\mathbf{s} - \mathbf{x}_2\|}{c}. \tag{B.2}$$

It follows that the cross-correlation function between signals received at locations $\mathbf{x}_1, \mathbf{x}_2$ will be

$$\begin{aligned} R_{x_1 x_2}(\tau) &= E[x_1(t)\, x_2(t+\tau)] && \text{(B.3)} \\ &= E[\alpha_1\, s(t-\Delta_1)\, \alpha_2\, s(t-\Delta_2+\tau)] && \text{(B.4)} \\ &= E[\alpha_1\, s(t-\Delta_1)\, \alpha_2\, s(t-\Delta_1+t_{12}+\tau)] && \text{(B.5)} \\ &= \alpha_1\, \alpha_2\, E[s(t-\Delta_1)\, s(t-\Delta_1+t_{12}+\tau)] && \text{(B.6)} \\ &= \alpha_1\, \alpha_2\, \sigma^2\, \delta(\tau+t_{12}). && \text{(B.7)} \end{aligned}$$

Now consider the case of a different speaker, who induces the same TDOA t_{12} at the receivers. As shown in section A.1, all the speaker locations $\mathbf{s}$ yielding a specific TDOA t_{12} belong to a hyperbola $S_2 = \{\, \mathbf{s} \mid \|\mathbf{s} - \mathbf{x}_2\| - \|\mathbf{s} - \mathbf{x}_1\| = c\, t_{12}\}$ whose foci $\mathbf{x}_1, \mathbf{x}_2$ correspond to the positions of the receivers. Let the speaker be at a location $\mathbf{s}' \in S_2, \mathbf{s}' \neq \mathbf{s}$ and the emitted sound $s'(t)$ be again WSS, with auto-correlation function $R_{s's'}(\tau) = (\sigma')^2 \delta(\tau)$. The received signals $x'_1(t), x'_2(t)$ will be attenuated with coefficients α'_1, α'_2. Then there exist a value for the variance of $s'(t)$ such that

$$(\sigma')^2 = \alpha_1\, \alpha_2\, \frac{\sigma^2}{\alpha'_1\, \alpha'_2}. \tag{B.8}$$

As a consequence,

$$\begin{aligned} R_{x'_1 x'_2}(\tau) &= \alpha'_1\, \alpha'_2\, (\sigma')^2\, \delta(\tau+t_{12}) && \text{(B.9)} \\ &= \alpha_1\, \alpha_2\, \sigma^2\, \delta(\tau+t_{12}) && \text{(B.10)} \\ &= R_{x_1 x_2}(\tau). && \text{(B.11)} \end{aligned}$$

In words, the channel-wise cross-correlations between recordings of source signals generated at different locations $\mathbf{s}, \mathbf{s}'$ are the same, $R_{x'_1 x'_2}(\tau) = R_{x_1 x_2}(\tau)$. Therefore, the mapping between cross-correlation function and emitter location is not unique.

This example shows how signals recorded at two different emitter locations can yield the same cross-correlation function, proving that if both the variance of the emitted signal and the position of the corresponding source are unknown, the cross-correlation function does not carry enough information to localize the speaker.

C Network architecture

This appendix details the architecture of the deep neural networks used throughout the experiments in chapter 4. This parameters were found through the hyperparameter optimization procedure described in section 4.3. Unless the description of the individual experiment specifies differently, $B = 16$, $F = 16$, $\mu = 2$, $T = 4$. Further, $L = 128$ for `NN128` or $L = 256$ for `NN256`.

DNN architecture			
Description	Number of filters	Filter size	Output shape
Feature extractor ($\times T = 4$)			(Q, 1, F, B)
Concatenate			(Q, T=4, F, B)
Conv + BN + LR + Max	16	(3, 1, 1)	(16, 3, F, B)
Conv + BN + LR + Max	16	(3, 1, 1)	(32, 1, F, B)
DB1 ($D = 8$, $k = 6$)			(80, F, B)
DB2 ($D = 8$, $k = 6$)			(128, F, B)
Reshape			(μ, L, L)
DB3 ($D = 4$, $k = 20$)			(82, L, L)
Conv + BN + GMax	44	(1, 1)	(L, L)
$\max(0, x)$			(L, L)
$\min(1, x)$			(L, L)

Table C.1: The number of latent maps is $\mu = 2$, and $L = 128$ is the dimension of the output likelihood map. ”Conv” denotes convolution. ”BN” denotes batch normalization. ”Max” denotes max pooling with kernel $(1, 3)$. ”LR” denotes leaky rectified linear unit with leakage 0.2. ”GMax” is global maximum across channel dimension. ”DB” is a dense block as described in Table C.3.

Feature extractor block			
Layer (type)	Number of filters	Filter size	Output shape
Input			(1, 1, C, 320)
Conv + BN	F	(1, N)	(1, F, C, 320)
SD + DepthwiseConv + BN + LR	$B \cdot F$	(C, P)	(1, F, B, 320)
Conv + BN + LR + Max	Q	(1, R)	(Q, F, B, 318)
Conv + BN + LR + Max	Q	(1, R)	(Q, F, B, 77)
Conv + BN + LR + Max	Q	(1, R)	(Q, F, B, 17)
Conv + BN + LR + Max	Q	(1, R)	(Q, F, B, 1)
Transpose			(Q, 1, F, B)
Output		(Q, 1, F, B)	

Table C.2: The feature extractor block. ”Conv” denotes convolution. ”DepthwiseConv” denotes depthwise convolution. ”BN” denotes batch normalization. ”SD” denotes spatial dropout with drop probability $p = 0.3$. ”Max” denotes max pooling with kernel $(1, 4)$ and padding. ”LR” denotes leaky rectified linear unit with leakage 0.2.

Dense block with $D = 4$ layers			
Layer (type)	Number of filters	Filter size	Output shape
Input			(1, x, L, L)
BN + LR + Conv	k	(3,3)	(1, k, L, L)
BN + LR + Conv	k	(3,3)	(1, k, L, L)
BN + LR + Conv	k	(3,3)	(1, k, L, L)
BN + LR + Conv	k	(3,3)	(1, k, L, L)
Concatenate		(1, $4k + x$, L, L)	

Table C.3: Dense block. ”Conv” denotes convolution. ”BN” denotes batch normalization. ”LR” denotes leaky rectified linear unit with leakage 0.2. ”Concatenate” denotes concatenation of all previous layers along channel dimension.

List of Figures

List of Tables

Bibliography

[1] P. Naylor and N. Gaubitch, *Speech Dereverberation*, vol. 59. Jan. 2011.

[2] A. Asaei, M. Golbabaee, H. Bourlard, and V. Cevher, “Structured Sparsity Models for Reverberant Speech Separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 620–633, Mar. 2014.

[3] M. Vorländer, *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality.* RWTHedition, Berlin Heidelberg: Springer-Verlag, 2008.

[4] H. Kuttruff, *Room Acoustics.* CRC Press, Oct. 2016.

[5] I. Dokmanic, Y. M. Lu, and M. Vetterli, “Can one hear the shape of a room: The 2-D polygonal case,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Prague, Czech Republic), pp. 321–324, IEEE, May 2011.

[6] D. Markovic´, F. Antonacci, A. Sarti, and S. Tubaro, “Estimation of room dimensions from a single impulse response,” in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, Oct. 2013.

[7] S. Tervo and T. Korhonen, “Estimation of reflective surfaces from continuous signals,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 153–156, Mar. 2010.

[8] M. Crocco, A. Trucco, V. Murino, and A. D. Bue, “Towards fully uncalibrated room reconstruction with sound,” in *2014 22nd European Signal Processing Conference (EUSIPCO)*, pp. 910–914, Sept. 2014.

[9] M. Crocco, A. Trucco, and A. Del Bue, “Uncalibrated 3D Room Reconstruction from Sound,” *arXiv:1606.06258 [cs]*, June 2016. arXiv: 1606.06258.

[10] S. Tervo and T. Tossavainen, “3D room geometry estimation from measured impulse responses,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 513–516, Mar. 2012.

[11] F. Antonacci, J. Filos, M. R. P. Thomas, E. A. P. Habets, A. Sarti, P. A. Naylor, and S. Tubaro, “Inference of Room Geometry From Acoustic Impulse Responses,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 2683–2695, Dec. 2012.

[12] L. Remaggi, P. J. B. Jackson, P. Coleman, and W. Wang, “Acoustic Reflector Localization: Novel Image Source Reversion and Direct Localization Methods,”

IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, pp. 296–309, Feb. 2017. arXiv: 1610.05653.

[13] I. Dokmanic, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proceedings of the National Academy of Sciences*, vol. 110, pp. 12186–12191, July 2013.

[14] I. Jager, R. Heusdens, and N. D. Gaubitch, "Room geometry estimation from acoustic echoes using graph-based echo labeling," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Shanghai), pp. 1–5, IEEE, Mar. 2016.

[15] M. Coutino, M. B. Møller, J. K. Nielsen, and R. Heusdens, "Greedy alternative for room geometry estimation from acoustic echoes: A subspace-based method," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 366–370, Mar. 2017.

[16] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, pp. 943–950, Apr. 1979.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[18] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-y. Chang, and T. Sainath, "Deep Learning for Audio Signal Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 206–219, May 2019. arXiv: 1905.00078.

[19] H. Gamper and I. J. Tashev, "Blind Reverberation Time Estimation Using a Convolutional Neural Network," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, (Tokyo), pp. 136–140, IEEE, Sept. 2018.

[20] R. Falcon Perez, "Machine-learning-based estimation of room acoustic parameters," Dec. 2018.

[21] W. Yu and W. B. Kleijn, "Room Geometry Estimation from Room Impulse Responses using Convolutional Neural Networks," *arXiv:1904.00869 [eess]*, Apr. 2019. arXiv: 1904.00869.

[22] L. Savioja and U. P. Svensson, "Overview of geometrical room acoustic modeling techniques," *The Journal of the Acoustical Society of America*, vol. 138, pp. 708–730, Aug. 2015.

[23] H. P. Tukuljac, A. Deleforge, and R. Gribonval, "MULAN: A Blind and Off-Grid Method for Multichannel Echo Retrieval," p. 12, 2018.

[24] S. Tervo, T. Korhonen, and T. Lokki, "Estimation of Reflections from Impulse Responses," *Building Acoustics*, vol. 18, pp. 159–173, Mar. 2011.

[25] M. Pollefeys and D. Nister, “Direct computation of sound and microphone locations from time-difference-of-arrival data,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2445–2448, Mar. 2008.

[26] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, “Auto-localization in ad-hoc microphone arrays,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 106–110, May 2013.

[27] K. Endoh, Y. Yamasaki, and T. Itow, “Grasp and development of spatial informations in a room by closely located four-point microphone method,” in *ICASSP ’86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11, pp. 909–912, Apr. 1986.

[28] I. Dokmanić, L. Daudet, and M. Vetterli, “From acoustic room reconstruction to slam,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6345–6349, Mar. 2016.

[29] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, pp. 276–280, Mar. 1986.

[30] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, pp. 320–327, Aug. 1976.

[31] M. S. Brandstein and H. F. Silverman, “A robust method for speech signal time-delay estimation in reverberant rooms,” in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 375–378 vol.1, Apr. 1997.

[32] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, “Robust Localization in Reverberant Rooms,” in *Microphone Arrays: Signal Processing Techniques and Applications* (M. Brandstein and D. Ward, eds.), Digital Signal Processing, pp. 157–180, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[33] Y. Wang, J. Li, P. Stoica, M. Sheplak, and T. Nishida, “Wideband RELAX and wideband CLEAN for aeroacoustic imaging,” *The Journal of the Acoustical Society of America*, vol. 115, pp. 757–767, Jan. 2004.

[34] F. Vesperini, P. Vecchiotti, E. Principi, S. Squartini, and F. Piazza, “A neural network based algorithm for speaker localization in a multi-room environment,” in *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Sept. 2016.

[35] W. He, P. Motlicek, and J.-M. Odobez, “Deep Neural Networks for Multiple Speaker Detection and Localization,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 74–79, May 2018. arXiv: 1711.11565.

[36] P. Pertila and M. Parviainen, "Time Difference of Arrival Estimation of Speech Signals Using Deep Neural Networks with Integrated Time-frequency Masking," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, United Kingdom), pp. 436–440, IEEE, May 2019.

[37] J. Dmochowski, J. Benesty, and S. Affes, "On Spatial Aliasing in Microphone Arrays," *IEEE Transactions on Signal Processing*, vol. 57, pp. 1383–1395, Apr. 2009.

[38] M. Lovedee-Turner and D. Murphy, "Application of Machine Learning for the Spatial Analysis of Binaural Room Impulse Responses," *Applied Sciences*, vol. 8, p. 105, Jan. 2018.

[39] N. Ma, J. A. Gonzalez, and G. J. Brown, "Robust Binaural Localization of a Target Sound Source by Combining Spectral Source Models and Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 2122–2131, Nov. 2018. arXiv: 1904.03006.

[40] J. Pak and J. Won Shin, "Sound Localization Based on Phase Difference Enhancement Using Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, pp. 1–1, May 2019.

[41] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, "CRNN-Based Multiple DoA Estimation Using Acoustic Intensity Features for Ambisonics Recordings," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 22–33, Mar. 2019.

[42] J. Pons and X. Serra, "Randomly weighted CNNs for (music) audio classification," *arXiv:1805.00237 [cs, eess]*, May 2018. arXiv: 1805.00237.

[43] M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, C.-A. Deledalle, and W. Li, "Machine learning in acoustics: a review," *arXiv:1905.04418 [physics]*, May 2019. arXiv: 1905.04418.

[44] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," *arXiv:1710.10059 [cs, eess]*, Oct. 2017. arXiv: 1710.10059.

[45] S. Adavanne, A. Politis, and T. Virtanen, "Localization, Detection and Tracking of Multiple Moving Sound Sources with a Convolutional Recurrent Neural Network," 2019.

[46] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 34–48, Mar. 2019. arXiv: 1807.00129.

[47] S. Chakrabarty and E. A. P. Habets, "Multi-Speaker Localization Using Convolutional Neural Network Trained with Noise," *arXiv:1712.04276 [cs, eess, stat]*, Dec. 2017. arXiv: 1712.04276.

[48] S. Chakrabarty and E. A. P. Habets, "Broadband DOA estimation using Convolutional neural networks trained with noise signals," *arXiv:1705.00919 [cs, stat]*, May 2017. arXiv: 1705.00919.

[49] S. Chakrabarty and E. A. P. Habets, "Multi-scale aggregation of phase information for reducing computational cost of CNN based DOA estimation," *arXiv:1811.08552 [cs, eess]*, Nov. 2018. arXiv: 1811.08552.

[50] S. Chakrabarty and E. A. P. Habets, "Multi-Speaker DOA Estimation Using Deep Convolutional Networks Trained with Noise Signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 8–21, Mar. 2019. arXiv: 1807.11722.

[51] J. M. Vera-Diaz, D. Pizarro, and J. Macias-Guarasa, "Towards End-to-End Acoustic Localization Using Deep Learning: From Audio Signals to Source Position Coordinates," *Sensors*, vol. 18, p. 3418, Oct. 2018.

[52] P. Vecchiotti, N. Ma, S. Squartini, and G. J. Brown, "End-to-end Binaural Sound Localisation from the Raw Waveform," *arXiv:1904.01916 [cs, eess]*, Apr. 2019. arXiv: 1904.01916.

[53] L. Perotin, A. Défossez, E. Vincent, R. Serizel, and A. Guérin, "Regression versus classification for neural network based audio source localization," Apr. 2019.

[54] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv:1503.02531 [cs, stat]*, Mar. 2015. arXiv: 1503.02531.

[55] A. Wabnitz, N. Epain, C. Jin, and A. van Schaik, "Room acoustics simulation for multichannel microphone arrays," p. 6, 2010.

[56] D. D. Carlo, A. Deleforge, and N. Bertin, "Mirage: 2D Source Localization Using Microphone Pair Augmentation with Echoes," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, United Kingdom), pp. 775–779, IEEE, May 2019.

[57] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel Signal Processing With Deep Neural Networks for Automatic Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 965–979, May 2017.

[58] "Conventional Beamforming Techniques," in *Microphone Array Signal Processing* (J. Benesty, J. Chen, and Y. Huang, eds.), Springer Topics in Signal Processing, pp. 39–65, Berlin, Heidelberg: Springer, 2008.

[59] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," *arXiv:1611.05267 [cs]*, Nov. 2016. arXiv: 1611.05267.

[60] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv:1803.01271 [cs]*, Mar. 2018. arXiv: 1803.01271.

[61] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient Object Localization Using Convolutional Networks," *arXiv:1411.4280 [cs]*, Nov. 2014. arXiv: 1411.4280.

[62] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning.* 2020.

[63] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," *arXiv:1611.09326 [cs]*, Nov. 2016. arXiv: 1611.09326.

[64] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv:1608.06993 [cs]*, Jan. 2018. arXiv: 1608.06993.

[65] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv:1611.03530 [cs]*, Nov. 2016. arXiv: 1611.03530.

[66] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* USA: Prentice-Hall, Inc., 1993.

[67] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167 [cs]*, Feb. 2015. arXiv: 1502.03167.

[68] I. Mccowan, *Microphone Arrays: A Tutorial.* 2001.

[69] E. Warsitz and R. Haeb-Umbach, "Blind Acoustic Beamforming Based on Generalized Eigenvalue Decomposition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1529–1539, July 2007.

[70] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding Convolution for Semantic Segmentation," *arXiv:1702.08502 [cs]*, May 2018. arXiv: 1702.08502.

[71] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017. arXiv: 1412.6980.

[72] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, (London, UK, UK), pp. 9–50, Springer-Verlag, 1998.

[73] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *The Journal of Machine Learning Research*, vol. 13, pp. 281–305, Feb. 2012.

[74] Y.-C. Lu and M. Cooke, “Binaural Estimation of Sound Source Distance via the Direct-to-Reverberant Energy Ratio for Static and Moving Sources,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 1793–1805, Sept. 2010.

[75] C. Payer, D. Štern, H. Bischof, and M. Urschler, “Integrating spatial configuration into heatmap regression based CNNs for landmark localization,” *Medical Image Analysis*, vol. 54, pp. 207–219, May 2019.

[76] A. Nibali, Z. He, S. Morgan, and L. Prendergast, “Numerical Coordinate Regression with Convolutional Neural Networks,” *arXiv:1801.07372 [cs]*, Jan. 2018. arXiv: 1801.07372.

[77] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, pp. 1256–1266, Aug. 2019. arXiv: 1809.07454.