Peer-reviewed version available at Sensors 2018, 18, 3418; doi:10.3390/s18103418

Article

Towards End-to-End Acoustic Localization using Deep Learning: from Audio Signal to Source Position Coordinates

Juan Manuel Vera-Diaz, Daniel Pizarro and Javier Macias-Guarasa*

Department of Electronics, University of Alcalá, Campus Universitario s/n, 28805, Alcalá de Henares, Madrid, Spain. manuel.vera@edu.uah.es, daniel.pizarro@uah.es, javier.maciasguarasa@uah.es
* Correspondence: javier.maciasguarasa@uah.es; Tel.: +34-91-885-6918

Abstract: This paper presents a novel approach for indoor acoustic source localization using microphone arrays and based on a Convolutional Neural Network (CNN). The proposed solution is, to the best of our knowledge, the first published work in which the CNN is designed to directly estimate the three dimensional position of an acoustic source, using the raw audio signal as the input information avoiding the use of hand crafted audio features. Given the limited amount of available localization data, we propose in this paper a training strategy based on two steps. We first train our network using semi-synthetic data, generated from close talk speech recordings, and where we simulate the time delays and distortion suffered in the signal that propagates from the source to the array of microphones. We then fine tune this network using a small amount of real d ata. Our experimental results show that this strategy is able to produce networks that significantly improve existing localization methods based on *SRP-PHAT* strategies. In addition, our experiments show that our CNN method exhibits better resistance against varying gender of the speaker and different window sizes compared with the other methods.

Keywords: acoustic source localization; microphone arrays; deep learning; convolutional neural networks

1. Introduction

The development and scientific research in advanced perceptual systems has notably grown during the last decades, and has experimented a tremendous rise in the last years due to the availability of increasingly sophisticated sensors, the use of computing nodes with higher and higher computational power, and the advent of powerful algorithmic strategies based on deep learning (all of them actually entering the mass consumer market). The aim of perceptual systems is to automatically analyze complex and rich information taken from different sensors, in order to obtain refined information on the sensed environment and the activities being carried out within them. The scientific works in these environments, cover research areas from basic sensor technologies, to signal processing and pattern recognition, and open the path to the idea of systems able to analyze human activities, providing them with advanced interaction capabilities and services..

In this context, localization of humans (being the most *interesting* element for perceptual systems) is a fundamental task that needs to be addressed so that the systems can actually start to provide higher level information on the activities being carried out. Without a precise localization, further advanced interaction between humans and their physical environment cannot be carried out successfully.

The scientific community has devoted a huge amount of effort to build robust and reliable indoor localization systems, based on different sensors [1–3]. Non-invasive technologies are preferred in this context, so that no electronic or passive devices need to be carried by humans for localization. The two non-invasive technologies that have been mainly used in indoor localization are those based on video systems and acoustic sensors.

This paper focuses on audio-based localization, with no previous assumptions on the acoustic signal characteristics nor in the physical environment, apart from the fact that unknown wide-band audio sources (e.g. human voice) are captured by a set of microphone arrays placed in known positions. The main objective of the paper is to directly use the signals captured by the microphone arrays to automatically obtain the position of the the acoustic source detected in the given environment.

Even though there are a lot of proposals in this area, Acoustic Source Localization (ASL) is still a hot research topic. This paper proposes a convolutional neural network (CNN) architecture that is trained end-to-end to solve the acoustic localization problem. To our knowledge, this is the first work in the literature that does not provide the network with feature vectors extracted from the speech signals, but directly uses the speech signal. Avoiding hand crafted features has been proved to increase the accuracy of classification and regression methods based on convolutional neural networks in other fields, such as in computer vision [4,5].

Our proposal is evaluated using both semi-synthetic and real data, outperforming traditional solutions based on Steered Response Power (*SRP*) [6], that are still the basis of state-of-the-art systems [7–10].

The rest of the paper is organized as follows. In Section 2 a review study of the state-of-the-art in acoustic source localization with special emphasis on the use of deep learning approaches. Section 3 describes the CNN based proposal, with details on the training and fine tuning strategies. The experimental work is detailed in Section 4, and Section 5 summarizes the main conclusions and contributions of the paper and gives some ideas for future work.

2. State of the Art

Many approaches exist in the literature to address the acoustic source localization (ASL) problem. According to the classical literature review in this topic, these approaches can be broadly divided in three categories [11,12]: time delay based, beamforming based, and high-resolution spectral-estimation based methods. This taxonomy relies in the fact that ASL has been traditionally considered a signal processing problem based on the definition of a signal propagation model [11–19], but, more recently, the range of proposals in the literature also considered strategies based on exploiting optimization techniques and mathematical properties of related measurements [20–24], and also using machine learning strategies [25–27], aimed at obtaining a direct mapping from specific features to source locations [28], area in which deep learning approaches are starting to be applied and that will be further described later in this section.

Time delay based methods (also referred to as *indirect methods*), compute the time difference of arrivals (TDOAs) across various combinations of pairs of spatially separated microphones, usually using the Generalized Correlation Function (GCC) [13]. In a second step, the TDOAs are combined with knowledge of the microphones' positions to generate a position estimation [11,29].

Beamforming based techniques [12,15,19,30] attempt to estimate the position of the source, optimizing a spatial statistic associated with each position, such as in the Steered Response Power (*SRP*) approach, in which the statistic is based on the signal power received when the microphone array is steered in the direction of a specific location. *SRP-PHAT* is a widely used algorithm for speaker localization based on beamforming that was first proposed in [6][1]. It combines the robustness of the SRP approach with the Phase Transform (PHAT) filtering, which increases the robustness of the algorithm to signal and room conditions, making it an ideal strategy for realistic speaker localization systems [16,17,32–34]. Other beamforming based methods such as the Minimum Variance Distortionless Response (MVDR) [18], exhibits problems when facing reverberant environments, because it introduces a new trade-off between dereverberation and noise reduction.

[1] Although the formulation is virtually identical to the *Global Coherence Field* (GCF) described in [31]

In what respect to spectral estimation based methods, the multiple signal classification algorithm (MUSIC) [35], has been widely used, but these methods, in general, tend to be less robust than beamforming methods [12], as they assume incoherent signals and are very sensitive to small modeling errors.

In the past few years, deep learning approaches [36] have taken the lead in different signal processing and machine learning fields, such as computer vision [37,38] and speech recognition [39–41], and, in general, in any area in which complex relationships between observed signals and the underlying processes generating them need to be discovered.

The idea of using neural networks for ASL is not new. Back in the early nineties and the first decade of the current century, works such as [25,42,43] proposed the use of neural network techniques in this area. However an evaluation on realistic and extensive data sets was not viable at this time, and the proposals were somehow limited in scope.

With the advent and huge increase on applications of deep neural networks in all areas of machine learning, and mainly due to the sophisticated capabilities and more careful implementation details of network architectures and the availability of advanced hardware architectures with increased computational capacity, promising works have been proposed also for ASL [44–58].

The main differences between the different proposals using neural networks for ASL reside in the architectures, input features, the network output (target), and the experimental setup (using real or simulated data).

Regarding the information given to the neural network, we can find several works using features physically related to the ASL problem. Some of the proposals use features derived from the GCC or related functions, which actually make sense as these correlation function is closely related to the TDOAs which are used in traditional methods to generate position estimations. The published works use either the GCC coefficients directly [50], features derived from them [45,55] or from the correlation matrix [47,49], or even combined with others, such as cepstral coefficients [53]. Other works are focused in exploiting binaural cues [44,46], features derived from convolving the spectrum with head related impulse responses [58] or even narrowband SRP values [56]. The latter approach goes one step further from correlation related values, as the SRP function actually integrates multiple GCC estimations in such a way that acoustic energy maps can be easily generated from it.

Opposed to the previously described works using refined features directly related to the localization problem, we can also find others using frequency domain features directly [48,52], in some cases generated from spectrograms of general time-frequency representations [51,54]. These approaches represent a step forward compared with the previous ones, as they give the network the responsibility of automatically learn the relationship between spectral cues and the location related information [57] kind of combines both strategies, as they use spectral features but calculating them in a cross-spectral fashion, that is, combining the values from all the available microphones in the so-called Cross Spectral Map (CSM).

In none of the referenced works, the authors try to make use of the raw acoustic signal directly, and we are interested in evaluating the capabilities of CNN architectures in directly exploiting this raw input information.

In what respect to the estimation target, most of the works are oriented towards estimating the Direction of Arrival (DOA) of the acoustic sources [45,50,51,55,56], or DOA related measurements such as azimuth angle [44,46,48], elevation angle [58], or position bearing+range [53]. Some of the proposals pose the problem not as a direct estimation (regression) but as a classification problem among a predefined set of possible position related values [47–49,52,54] (azimuth, positions in a predefined grid, etc.). Works with a very different target try to estimate a *clean* acoustic source map [57] or learn time-frequency masks as a preprocessing stage prior to ASL [59].

In none of the referenced works the authors try to directly estimate the coordinate values of the acoustic sources, and, again, we are interested in evaluating the capabilities of CNN architectures to directly generate this output information.

Finally, in what respect to the experimental setup, most works use simulated data either for training or for training and testing [44–52,54–59], usually by convolving clean (anechoic) speech with impulse responses (room, head related, or DOA related (azimuth, elevation)). Only some of them actually face real recordings [44,45,53,55,56], which in our opinion is a must to be able to assess the actual impact of the proposals in real conditions.

So, in this paper we describe, for the first time in the literature to the best of our knowledge, a CNN architecture in which we directly exploit the raw acoustic signal to be provided to the neural network, with the objective of directly estimating the three dimensional position of an acoustic source in a given environment. This is the reason why we refer to this strategy as end-to-end, considering the full coverage of the ASL problem. The proposal has been tested on both semi-synthetic and real data from a publicly available database.

3. System Description

3.1. Problem Statement

Our system obtains the position of an acoustic source from the audio signals recorded by an array of M microphones. Given a reference coordinate origin, the source position is defined with the 3D coordinate vector $\mathbf{s} = \begin{pmatrix} s_x & s_y & s_z \end{pmatrix}^\top$. The microphones positions are known and they are defined with coordinate vectors $\mathbf{m}_i = \begin{pmatrix} m_{i,x} & m_{i,y} & m_{i,z} \end{pmatrix}^\top$ with $i = 1, \ldots, M$. The audio signal captured from the i^{th} microphone is denoted by $x_i(t)$. This signal is discretized with a sampling frequency f_s and is defined with $x_i[n]$. We assume for simplicity that $x_i[n]$ is of finite-length with N samples. This corresponds to a small window of audio with duration $w_s = N/f_s$, which is a design parameter in our system. We denote as $\mathbf{x}_i$ the vector containing all time samples of the signal:

$$\mathbf{x}_i = \begin{pmatrix} x_i[0] & \ldots & x_i[N-1] \end{pmatrix}^\top. \tag{1}$$

The problem we seek to solve is to find the following regression function f:

$$\mathbf{s} = f\left(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{m}_1, \ldots, \mathbf{m}_M\right), \tag{2}$$

that obtains the speaker position given the signals recorded from the microphones.

In classical simplified approaches, f is found by assuming that signals received from different microphones mainly differ by a delay that depends on the relative position of the source with respect to the microphones. However, this assumption breaks in environments where the signal suffers from random noise and distortion, such as multi-path signals or microphone non-linear response.

Due to the aforementioned effects, and the random nature of the audio signal, the regression function of equation (2) cannot be estimated analytically. We present in this paper a learning approach for directly obtaining f using Deep Learning. We represent f using a Convolutional Neural Network (CNN) which is learned end-to-end from the microphone signals. In our system we assume that microphones positions are fixed. We thus drop the requirement of knowing the microphone's position from equation (2) which will be implicitly learned by our network with the following regression problem:

$$\mathbf{s} = f_{net}(\mathbf{x}_1, \ldots, \mathbf{x}_M), \tag{3}$$

where f_{net} denotes the function that we represent using the CNN and whose topology is described next.

3.2. Network Topology

The topology of our neural network is shown in figure 1. It is composed of five convolutional blocks of one dimension and two fully connected blocks. Following equation (3), the network inputs

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0570.v1

Peer-reviewed version available at Sensors 2018, 18, 3418; doi:10.3390/s18103418

are the set of windowed signals from the microphones and the network output is the estimated position of the acoustic source.

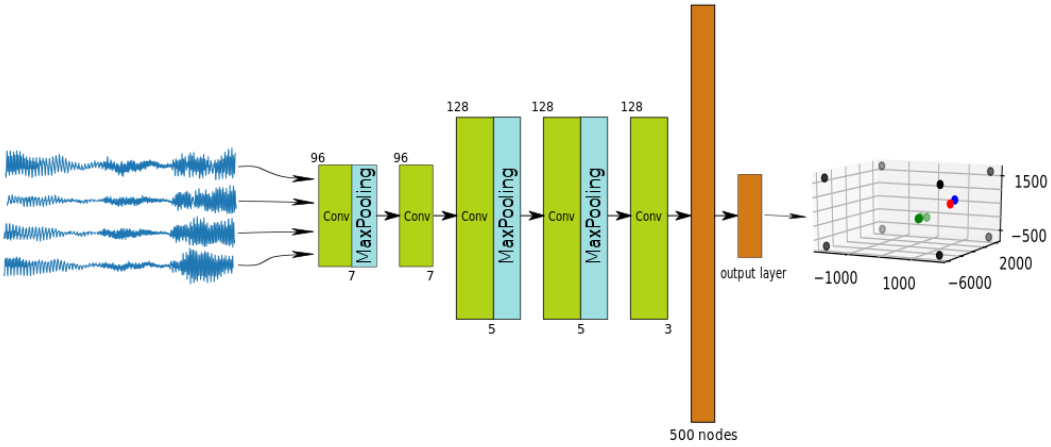

Figure 1. Used network topology

Table (1) shows the size and amount of convolutional filters in the proposed network. We use filters of size 7 (layers 1 and 2), size 5 (layers 3 and 4) and size 3 (layer 5). The number of filters is 96 in the first two convolutional layers and 128 in the rest. As seen in figure 1, some of the layers are equipped with *MaxPooling* filters with the same pool size as their corresponding convolutional filters. The last two layers are fully-connected layers, one hidden with 500 nodes and the output layer. All layer's activation functions are "ReLUs" with the exception of the output layer. During training we include dropout with probability 0.5 in the fully-connected layers to prevent overfitting.

Table 1. Network convolutional layers summary

Block	Filters	Kernel
Convolutional block 1	96	7
Convolutional block 2	96	7
Convolutional block 3	128	5
Convolutional block 4	128	5
Convolutional block 5	128	3

3.3. Training Strategy

The amount of available real data that we have in our experimental setup (see Section 4) will be, in general, limited for training a CNN model. To cope with this problem we propose a training strategy comprising two steps:

Step 1. Training the network with semi-synthetic data: We use close-talk speech recordings and a set of randomly generated source positions to generate simulated versions of the signals captured by a set of microphones that share the same geometry with the environment used in real data. Additional considerations on the acoustic behavior of the target environment (specific noise types, noise levels, etc.) is also taken into account to generate the data. This dataset can virtually be made as big as required to train the network.

Step 2. Fine tuning the network with real data: We train the network on a reduced subset of the database captured in the target physical environment using the weights obtained in Step 1 as initialization.

3.3.1. Semi-Synthetic Dataset Generation

In this step we extract audio signals from any available close-talk (anechoic) corpus, and use them to generate semi-synthetic data. There are many available datasets suitable for this task (freely of commercially distributed). Our semi-synthetic dataset can thus be made as big as required for training the CNN.

For this task, we randomly generate position vectors $\mathbf{q} = \begin{pmatrix} q_x & q_y & q_z \end{pmatrix}^\top$ of the acoustic source using a uniform distribution that covers the physical space (room) that will be used.

The loss function we use to train the network is the mean squared error between the estimated position given by the network ($\mathbf{s_i}$) and the target position vector ($\mathbf{q_i}$). It follows the expression:

$$\mathcal{L}(\mathbf{\Theta}) = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{q_i} - \mathbf{s_i}|^2, \tag{4}$$

where $\mathbf{\Theta}$ represents the weights of the network. Equation (4) is minimized in function of the unknown weights using iterative optimization based on the Stochastic Gradient Descent (SGD) algorithm [60]. We finally obtain the target weights $\theta \in \mathbf{\Theta}$ once a termination criterion is met in the optimization. More details are given in Section 4 about the training algorithm.

In order to realistically simulate the signals received in the microphones from a given source position we have to consider two main issues:

- Signal propagation considerations: This is affected by the impulse response of the target room. Different alternatives can be used to simulate this effect, such as convolving the anechoic signals with real room impulse responses such as in [47], that can be difficult to acquire for general positions in big environments; or using room response simulation methods such as the image method [61] used in [62] for this purpose.
- Acoustic noise conditions of the room and recording process conditions: These can be due to additional equipment (computers, fans, air conditioning systems, etc.) present in the room, and to problems in the signal acquisition setup. This can be addressed by assuming additive noise conditions, and selecting a noise type and acoustic effects that should be preferably estimated in the target room.

In our case, and regarding the first issue, we used an initial simple approach, just taking into account the propagation delay from the source position to each of the microphones, that depends on their relative position and the sound speed in the room.

We denote the number of samples we have to shift a signal to simulate the arrival delay suffered at microphone i by $N_{s_i} = f_s \frac{d_i}{c}$ where f_s is the sampling frequency of the signal, d_i is the euclidean distance between the acoustic source and the i microphone and c is the sound speed in air ($c = 343 m/s$ in a room at $20C^o$). In general N_{s_i} is not an integer number. We thus require a way to simulate sub-sample shifts in the signal. In order to implement the delay N_{s_i} on $\mathbf{x}_{pc}$ (the windowed signal of N samples from the close-talk dataset) to obtain $\mathbf{x}_i$ we use the following transformation:

$$\mathbf{X}_{pc} = \mathcal{F}\{\mathbf{x}_{pc}\} \quad \mathbf{x}_i = A_i \left(\mathcal{F}^{-1}\{\mathbf{X}_{pc}\, \mathbf{D}_{s_i}\} \right), \quad with\ \mathbf{D}_{s_i} = \left(1, e^{-j\frac{2\pi N_{s_i}}{N}}, e^{-j\frac{4\pi N_{s_i}}{N}}, \cdots, e^{-j(N-1)\frac{2\pi N_{s_i}}{N}} \right) \tag{5}$$

where we first transform $\mathbf{x}_{pc}$ into the frequency domain $\mathbf{X}_{pc}$ using the *Discrete Fourier Transform* operator $\mathcal{F}$. We then change its phase according to N_{s_i} by the phase vector $\mathbf{D}_{s_i}$ and transform the signal back into time domain $\mathbf{x}_i$, using the *Inverse Discrete Fourier Transform* operator $\mathcal{F}^{-1}$. A_i is an amplitude factor applied to the signal that follows a uniform random distribution, and it is different for each microphone, preventing the network from being affected by amplitude differences between the signals captured in different microphones ($A_i \in [0.01, 0.03]$ in the experimental setup described in Section 4).

Regarding the second issue, we simulate noise and disturbances in the signals arriving to the microphones so that the signal-to-noise ratio and the spectral content of the signals are as similar as

possible to those found in the real data. In order to provide an example of the methodology we follow, we refer in this section to the particular case of the IDIAP room (see Section 4.1.1) that will be used in our real data experiments, and the Albayzin Phonetic Corpus (see Section 4.1.2) that will be used for synthetic data generation.

In the IDIAP room, a spectrogram based analysis showed that the recordings are contaminated with a tone at around $25Hz$ in the spectrum which does not appear in anechoic conditions, probably due to room equipment of electrical noise generated in the recording hardware setup. We have determined that the frequency of this tone actually varies in a range between $20Hz$ and $30Hz$. So, in the synthetic data generation process, we have *contaminated* the signals from the phonetic corpus with an additive tone of a random frequency in this established range, and we have also added white gaussian noise following the expression:

$$x_{pc_{new}}[n] = x_{pc}[n] + k_s \sin(2\pi f_0 n / f_s + \phi_0) + k_\eta \eta_{wgn}[n], \tag{6}$$

where k_s is a scaling factor for the contaminating tone signal (similar to the tone amplitude found in the target room recordings, 0.1 in our case), $f_0 \in [20, 30]Hz$, $\phi_0 \in [0, \pi]rad$, η_{wgn} is a white gaussian noise signal, and k_η is a noise scaling factor to generate signals with a SNR which is similar to that found in the target room recordings.

After this procedure is applied, the semi-synthetic signal data set will be ready to be used in the neural network training procedure.

3.3.2. Fine Tuning Procedure

The previous step takes care of reproducing simple acoustic characteristics of the testing room such as the propagation effects and the presence of specific types and levels of additive noises, but there are other phenomena like multi-path and reverberation propagation which are more complex to simulate. In order to introduce these acoustic behaviors of the target physical environment, our proposal is to carry out a fine tuning procedure of the network model using a short amount of real recorded data in the target room

Although there are other methods such as the one proposed in [49], where an unsupervised DNN is implemented for the adaptation of parameters to unknown data, we believe that the fine tuning process implemented is adequate because, in the first place, it is a supervised process with which a better performance is expected to be obtained and, secondly, not all the sequences of the test data set are used, so that only a few are used for the fine tuning process, saving the rest for the test phase.

4. Experimental Work

In his section we describe the datasets used in both steps of the training strategy described in Section 3.3, and the details associated with it. We then define the experimental setup general conditions, and the error metrics used for comparing our proposal with other state-of-the-art methods and finally present our experimental results, starting from the baseline performance we aim at improving.

4.1. Datasets

4.1.1. IDIAP AV16.3 Corpus: for testing and fine tuning

We have evaluated our proposal using the audio recordings of the AV16.3 database [63], an audio-visual corpus recorded in the *Smart Meeting Room* of the IDIAP research institute, in Switzerland. We have also used the physical layout of this room for our semi-synthetic data generation process.

The *IDIAP Smart Meeting Room* is a $3.6m \times 8.2m \times 2.4m$ rectangular room with a rectangular table centrally located and measuring $4.8m \times 1.2m$. On the table's surface there are two circular microphone arrays of $0.1m$ radius, each of them composed by 8 regularly distributed microphones as shown in figure 2. The centers of both arrays are separated by a distance of $0.8m$. The middle point between

them is considered as the origin of the coordinate reference system. A detailed description of the meeting room can be found in [64].

The dataset is composed by several sequences of recordings, synchronously sampled at 16 KHz, which a wide range of experimental conditions in the number of speakers involved and their activity. Some of the available audio sequences are assigned a corresponding annotation file containing the real ground truth positions (3D coordinates) of the speaker's mouth at every time frame in which that speaker was talking. The segmentation of acoustic frames with speech activity was first checked manually at certain time instances by a human operator in order to ensure its correctness, and later extended to cover the rest of recording time by means of interpolation techniques. The frame shift resolution was defined to be 40 ms. The complete dataset is fully accessible on-line at [65].

Figure 2. (a) Simplified top view of the *IDIAP Smart Meeting Room*, (b) A real picture of the room extracted from a video frame, (c) Microphone setup used in this proposal

In this paper we will just focus on all the annotated sequences of this dataset featuring a single speaker, whose main characteristics are shown in Table 2. This allows us to directly compare our performance with the state-of-the-art method presented in [20]. Note that the firsts three sequences are performed by a speaker remaining static while speaking at different positions, and the last two ones by a moving speaker, being all of the speakers different. We will refer to these sequences as `s01`, `s02`, `s03`, `s11` and `s15` for brevity.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0570.v1

Peer-reviewed version available at Sensors 2018, 18, 3418; doi:10.3390/s18103418

Table 2. *IDIAP Smart Meeting Room* used sequences.

Sequence	Average speaker height (cm)*	Duration (seconds)	Number of ground truth frames	Description
seq01-1p-0000	54.3	208	2248	A single male speaker, static while speaking, at each of 16 locations. The speaker is facing the microphone arrays.
seq02-1p-0000	62.5	171	2411	A single female speaker, static while speaking, at each of 16 locations. The speaker is facing the microphone arrays.
seq03-1p-0000	70.3	220	2636	A single male speaker, static while speaking, at each of 16 locations. The speaker is facing the microphone arrays.
seq11-1p-0100	53.5	33	481	A single male speaker, making random movements while speaking, and facing the arrays.
seq15-1p-0100	79.5	36	436	A single male speaker, walking around while alternating speech and long silences. No constraints

* The average speaker height is referenced to the system coordinates and refers to the speaker's mouth height.

4.1.2. Albayzin Phonetic Corpus: for Semi-Synthetic Dataset Generation

The Albayzin Phonetic Corpus [66] consists of 3 sub-corpora of 16 kHz 16 bits signals, recorded by 304 Castilian Spanish speakers in a professional recording studio using high quality close talk microphones.

We use this dataset to generate semi-synthetic data as described in Section 3.3.1. From the 3 sub-corpora, we will be only using the so-called *phonetic corpus* [67], composed of 6800 utterances of phonetically balanced sentences. This phonetical balance characteristic makes this dataset perfect for generating our semi-synthetic data, as it will cover all possible acoustic contexts.

4.2. Training and Fine Tuning Details

In the semi-synthetic dataset generation procedure, described in Section 3.3.1, we generate random positions $\mathbf{q}$ with uniformly distributed values in the following intervals: $q_x \in [0, 3.6]m$, $q_y \in [0, 8.2]m$ and $q_z \in [0.92, 1.53]m$, which correspond to the possible distribution of the speaker's mouth positions in the *IDIAP* room [63].

Regarding the optimization strategy for the loss function described by equation (4) we employ the *ADAM* [68] optimizer (variant of SGD with variable learning rate) along 200 epochs with a batch size of 100 samples. 7200 different frames of input data per epoch are randomly generated during the training phase and other 800 for validation.

The experiments will be performed with three different window lengths ($80ms$, $160ms$ and $320ms$), so the training phase will be run once per window length, obtaining three different network models. In each training, 200 audio recordings are randomly chosen and 40 different windows are randomly extracted from each. In the same way, 200 acoustic source position $\mathbf{q}$ vectors are randomly generated so that each position generates 40 windows of the same signal.

For the fine tuning procedure described in Section 3.3.2, we will be mainly using sequences `s11` and `s15`, that features a speaker moving in the room while speaking, and also sequences `s01`, `s02` and `s03` in a final experiment.

As it will be described in Section 4.6, we will also address experiments trying to assess the relevance of adding additional sequences `s01`, `s02` and `s03` to complement the fine tuning data provided by `s11` and `s15`. We will also refer to gender and height issues in the fine tuning and evaluation data.

4.3. Experimental Setup

In our experiments, sequences s01, s02 and s03 are used for testing the performance of our network and, as explained above, to complement sequences s11 and s15 for fine tuning.

In this work, we are using a simple microphone array configuration, aimed at evaluating our proposal in a resource-restricted environment, as it was done in [20]. In order to do so, we are using 4 microphones (numbers 1, 5, 11 and 15, out of the 16 available in the AV16.3 data set), grouped in two microphone pairs. The selected microphone pairs configurations are shown in Figure 2.c, in which microphones with the same color are considered as belonging to the same microphone pair. We provide results depending on the length of the acoustic frame, for 80ms, 160ms and 320ms, to precisely assess to what extent the improvements are consistent with varying acoustic time resolutions.

The main interest of our experimental work is assessing whether the end-to-end CNN based approach (that we will refer to as CNN) is competitive as compared with state-of-the-art localization methods. We will compare this CNN approach with the standard *SRP-PHAT* method, and the recent strategy proposed in [20] that we will refer to as GMBF. This GMBF method is based on fitting a generative model to the GCC-PHAT signals using sparse constraints, and it reported significant improvements over *SRP-PHAT* in the *IDIAP* dataset [20,69].

After providing baseline results comparing *SRP-PHAT*, GMBF and our proposal without fine tuning procedure, we will then describe four experiments, that we briefly summarize here:

- In the first experiment, we will evaluate the performance improvements when using a single sequence for the fine tuning procedure.
- In the second experiment, we will evaluate the differences between the semi-synthetic training plus the fine tuning approach, versus just training the network from scratch.
- In the third experiment, we will evaluate the impact of adding an additional fine tuning sequence.
- In the last experiment, we will evaluate the final performance improvements when also adding static sequences to the refinement process.

4.4. Evaluation metrics

Our CNN based approach yields a set of spatial coordinates $\mathbf{s}_k = \begin{pmatrix} s_{k,x} & s_{k,y} & s_{k,z} \end{pmatrix}^\top$ that are estimations of the current speaker position as time instant k. These position estimates will be compared, by means of the Euclidean distance, to the ones labeled in a transcription file containing the real positions $\mathbf{s}_{k_{GT}}$ (*ground truth*), of the speaker.

We evaluate performance adopting the same metric used in [20] and developed under the CHIL project [70]. It is known as MOTP (*Multiple Object Tracking Precision*) and is defined as:

$$MOTP = \frac{\sum_{k=1}^{N_P} |\mathbf{s}_{k_{GT}} - \mathbf{s}_k|^2}{N_P}, \tag{7}$$

where N_P denotes the total number of position estimations along time, $\mathbf{s}_k$ the estimated position vector and $\mathbf{s}_{k_{GT}}$ the labeled ground truth position vector.

We will compare our experimental results, and that of the GMBF method, with that of *SRP-PHAT*, measuring the relative improvement in MOTP with method, that is defined as follows:

$$\Delta_r^{MOTP} = 100 \frac{MOTP_{SRP-PHAT} - MOTP_{proposal}}{MOTP_{SRP-PHAT}} [\%] \tag{8}$$

4.5. Baseline Results

The baseline results are shown in Table 3 for sequences s01, s02 and s03, and all the evaluated time window sizes (in all the tables showing results in this paper, **bold font** highlight the best ones for a given data sequence and window length). The Table shows the results achieved by the *SRP-PHAT*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0570.v1

Peer-reviewed version available at Sensors 2018, 18, 3418; doi:10.3390/s18103418

standard algorithm strategy (columns SRP), the alternative described in [20] (columns GMBF), and the proposal in this paper without applying the fine-tuning procedure (columns CNN). We also show the relative improvements of GMBF and CNN as compared with SRP-PHAT.

Table 3. Baseline results for the SRP-PHAT strategy (columns SRP); the one in [20] (columns GMBF), and the CNN trained with synthetic data without applying the fine-tuning procedure (columns CNN) for sequences `s01`, `s02` and `s03` for different window sizes. Relative improvements as compared to SRP-PHAT are shown below the MOTP values.

		80*ms*			**160***ms*			**320***ms*		
		SRP	GMBF	CNN	SRP	GMBF	CNN	SRP	GMBF	CNN
s01	$MOTP(m)$	1.020	**0.795**	1.615	0.910	**0.686**	1.526	0.830	**0.588**	1.464
	Δ_r^{MOTP}		**22.1**%	−58.3%		**24.6**%	−67.7%		**29.1**%	−76.4%
s02	$MOTP(m)$	0.960	**0.864**	2.124	0.840	**0.759**	1.508	0.770	**0.694**	1.318
	Δ_r^{MOTP}		**10.0**%	−121.3%		**9.6**%	−79.5%		**9.9**%	−71.2%
s03	$MOTP(m)$	0.900	**0.686**	1.559	0.770	**0.563**	1.419	0.690	**0.484**	1.379
	Δ_r^{MOTP}		**23.8**%	−73.2%		**26.9**%	−84.3%		**29.9**%	−99.9%
Average	$MOTP(m)$	0.957	**0.778**	1.763	0.836	**0.666**	1.481	0.760	**0.585**	1.385
	Δ_r^{MOTP}		**18.7**%	−84.3%		**20.4**%	−77.1%		**22.9**%	−82.3%

The main conclusions from the baseline results are:

- Best MOTP values for the standard SRP-PHAT algorithm are around 69*cm*, with averages between 76*cm* and 96*cm*. For the GMBF, best MOTP values are around 48*cm*, with averages between 59*cm* and 78*cm*.
- MOTP values improve as the frame size increases, as expected, given that better correlation values will be estimated for longer window signal lengths.
- The GMBF strategy, as described in [20], achieves very relevant improvements as compared with SRP-PHAT, with average relative improvements around 20%, and peak values of almost 30%.
- Our CNN strategy, which at this point is only trained with semi-synthetic data, is very far from reaching the SRP-PHAT or GMBF in terms of performance. This result leads us to think that there are other effects only present in real data, such as reverberation, that are affecting the network.

Given the discussion above, we decided to apply the fine tuning strategy discussed in Section 3.3.2, with the experimental details described in Section 4.2. So, the results shown in Table 3 will be compared with those obtained by our CNN method, under different fine tuning (and training) conditions, and will be described below.

4.6. Results and Discussion

The first experiment in which we applied the fine tuning procedure used `s15` as the fine tuning subset.

Table 4 shows the results obtained by GMBF (columns GMBF) and CNN with this fine tuning strategy (columns CNNf15). From the table results it can be seen that CNNf15 is, most of the times, better than the *SRP-PHAT* baseline (except in two cases for `s03` in which there was a slight degradation). The average performance shows a consistent improvement of CNNf15 compared with SRP-PHAT, between 1.8% and 11.3%. However CNNf15 is still behind GMBF in all cases but one (for `s02` and 80*ms*).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0570.v1

Peer-reviewed version available at Sensors 2018, 18, 3418; doi:10.3390/s18103418

Table 4. Results for the stratgy in [20] (columns GMBF); and the CNN fine tuned with sequence s15 (columns CNNf15).

		80*ms*		160*ms*		320*ms*	
		GMBF	CNNf15	GMBF	CNNf15	GMBF	CNNf15
s01	$MOTP(m)$	**0.795**	0.875	**0.686**	0.833	**0.588**	0.777
	Δ_r^{MOTP}	**22.1**%	14.2%	**24.6**%	8.5%	**29.1**%	6.4%
s02	$MOTP(m)$	0.864	**0.839**	**0.759**	0.801	**0.694**	0.731
	Δ_r^{MOTP}	10.0%	**12.6**%	**9.6**%	4.6%	**9.9**%	5.1%
s03	$MOTP(m)$	**0.686**	0.835	**0.563**	0.806	**0.484**	0.734
	Δ_r^{MOTP}	**23.8**%	7.2%	**26.9**%	-4.7%	**29.9**%	-6.4%
Average	$MOTP(m)$	**0.778**	0.849	**0.666**	0.813	**0.585**	0.746
	Δ_r^{MOTP}	**18.7**%	11.3%	**20.4**%	2.8%	**22.9**%	1.8%

Our conclusion is that the fine tuning procedure is able to effectively complement the trained models from synthetic data, leading to results that outperform SRP-PHAT. This is specially relevant as:

- The amount of fine tuning data is limited (only 36 seconds, corresponding to 436 frames, as shown in Table 2), thus opening the path to further improvements with a limited data recording effort.
- The speaker used for fine tuning was mostly moving while speaking, while in the testing sequences the speakers are static while speaking. This means that the fine tuning material include far more active positions than in the testing sequences, and the network is able to extract the relevant information for the tested positions.
- The speaker used for fine tuning is a male, and the obtained results for male speakers (sequences s01 and s03) and the female one (sequence s02) do not seem to show any gender-dependent bias, which means that the gender issue does not seem to play a role in the adequate adaptation of the network models.

When comparing the results of Table 3 and Table 4, and given the large improvement when applying the fine tuning strategy, we could think that the effect of the initial training with semi-synthetic data is limited. From this argument, we run an additional training experiment in which we just trained the network *from scratch* using s15, aiming at assessing the actual effect of semi-synthetic training+fine tuning versus just training with real room data.

Table 5 shows the comparison between these two options: training from scratch using s15 (columns CNNt15) and semi-synthetic training+fine tuning with s15 (columns CNNf15). The average improvement of the latter approach varies between 1.8% and 11.3% with an average improvement over all window lengths of 5.3%, while the training from scratch average improvement varies between -20.6% and 4.3% with an average value of -7.0%. These differences show that the training+fine tuning proposal outperforms training the network from scratch, thus validating our methodology.

Table 5. Results for the CNN proposal, either trained from scratch with sequence s15 (columns CNNt15) or fine tuned with sequence s15 (columns CNNf15).

		80*ms*		160*ms*		320*ms*	
		CNNt15	CNNf15	CNNt15	CNNf15	CNNt15	CNNf15
s01	$MOTP(m)$	1.009	**0.875**	0.949	**0.833**	1.0009	**0.777**
	Δ_r^{MOTP}	1.1%	**14.2**%	-4.3%	**8.5**%	-21.6%	**6.4**%
s02	$MOTP(m)$	**0.807**	0.839	**0.767**	0.801	0.807	**0.731**
	Δ_r^{MOTP}	**15.9**%	12.6%	**8.7**%	4.6%	-4.8%	**5.1**%
s03	$MOTP(m)$	0.935	**0.835**	0.911	**0.806**	0.936	**0.734**
	Δ_r^{MOTP}	-3.9%	**7.2**%	-18.3%	$-$**4.7**%	-35.7%	$-$**6.4**%
Average	$MOTP(m)$	0.915	**0.849**	0.875	**0.813**	0.916	**0.746**
	Δ_r^{MOTP}	4.3%	**11.3**%	-4.6%	**2.8**%	-20.6%	**1.8**%

In spite of the relevant improvements with the fine tuning approach, they are still far from making this suitable for further competitive exploitation in the ASL scenario (provided we have the GMBF alternative), so that we next aim at increasing the amount of fine tuning material.

In our third experiment, we applied the fine tuning procedure using an additional *moving speaker* sequence, that is, including s15 and s11 in the fine tuning subset.

Table 6 shows the results obtained by GMBF and CNN fine tuned with s15 and s11 (CNNf15+11 columns). In this case, we see additional improvements over using only s15 for fine tuning, and there is only one case in which CNNf15+11 does not outperforms SRP-PHAT (with a marginal degradation of -0.3%).

Table 6. Relative improvements over SRP-PHAT for the strategy in [20] (columns GMBF); and the CNN fine tuned with sequences s15 and s11 (columns CNNf15+11)

		80*ms*		**160***ms*		**320***ms*	
		GMBF	CNNf15+11	GMBF	CNNf15+11	GMBF	CNNf15+11
s01	$MOTP(m)$	**0.795**	0.805	**0.686**	0.750	**0.588**	0.706
	Δ_r^{MOTP}	**22.1**%	21.1%	**24.6**%	17.6%	**29.1**%	14.9%
s02	$MOTP(m)$	0.864	**0.809**	0.759	**0.716**	**0.694**	0.712
	Δ_r^{MOTP}	10.0%	**15.7**%	9.6%	**14.8**%	9.9%	7.5%
s03	$MOTP(m)$	**0.686**	0.792	**0.563**	0.732	**0.484**	0.692
	Δ_r^{MOTP}	**23.8**%	12.0%	**26.9**%	4.9%	**29.9**%	-0.3%
Average	$MOTP(m)$	**0.778**	0.802	**0.666**	0.732	**0.585**	0.703
	Δ_r^{MOTP}	**18.7**%	16.2%	**20.4**%	12.4%	**22.9**%	7.5%

The CNN based approach shows again an average consistent improvement compared with SRP-PHAT between 7.5% and 16.2%.

In this case, the newly added sequence (s11, with a duration of only 33 seconds) for fine tuning corresponds to a randomly moving male speaker, and the results show that its addition contributes to further improvements in the CNN based proposal, but it is still behind GMBF in all cases but two, but with results getting closer. This suggests that a further increment in the fine tuning material should be considered.

Our last experiment will consist of fine tuning the network including also additional static speaker sequences. To assure that the training (including fine tuning) and testing material are fully independent, we will fine tune with s15, s11 and with the static sequences that are not tested in each experiment run, as shown in Table 7.

Table 7. Fine tuning material used in the experiment corresponding to Table 8 columns CNNf15+11+st.

Test sequence	**Fine tuning sequences**
seq01	s15 + s11 + s02 + s03
seq02	s15 + s11 + s01 + s03
seq03	s15 + s11 + s01 + s02

Table 8 shows the results obtained for this fine tuning scenario, and the main conclusions are:

- The CNN based method exhibits much better average behavior than GMBF for all window sizes. Average absolute improvement against SRP-PHAT for the CNN is more than 10 points higher than for GMBF, reaching 31.3% in the CNN case and 20.7% for GMBF.
- Considering individual sequences, CNN is significantly better than GMBF for sequences s01 and s02, and slightly worse for s03.
- Considering the best individual result, maximum improvement for the CNN is 41.6% (s01, 320*ms*), while the top result for GMBF is 29.9% (s03, 320*ms*).
- The effect of adding static sequences is beneficial, as expected, provided that the acoustic tuning examples will be generated from positions which are similar, but not identical, as the speakers

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0570.v1

Peer-reviewed version available at Sensors 2018, 18, 3418; doi:10.3390/s18103418

have varying heights and their position in the room is not strictly equal from sequence to sequence.

- The improvements obtained are significant and come at the cost of additional fine tuning sequences. However, this extra cost is still reasonable, as the extra fine tuning material is of limited duration, around 400 seconds in average (6.65 minutes).

Table 8. Relative improvements over SRP-PHAT for the strategy in [20] (columns GMBF); and the CNN fine tuned with the sequences described in Table 7 (columns CNNf15+11+st)

		80*ms*		**160***ms*		**320***ms*	
		GMBF	CNNf15+11+st	GMBF	CNNf15+11+st	GMBF	CNNf15+11+st
s01	$MOTP(m)$	0.795	**0.607**	0.686	**0.540**	0.588	**0.485**
	Δ_r^{MOTP}	22.1%	**40.5%**	24.6%	**40.7%**	29.1%	**41.6%**
s02	$MOTP(m)$	0.864	**0.669**	0.759	**0.579**	0.694	**0.545**
	Δ_r^{MOTP}	10.0%	**30.3%**	9.6%	**31.1%**	9.9%	**29.2%**
s03	$MOTP(m)$	**0.686**	0.707	**0.563**	0.617	**0.484**	0.501
	Δ_r^{MOTP}	**23.8%**	21.4%	**26.9%**	19.9%	**29.9%**	27.4%
Average	$MOTP(m)$	0.778	**0.664**	0.666	**0.581**	0.585	**0.511**
	Δ_r^{MOTP}	18.7%	**30.6%**	20.4%	**30.6%**	22.9%	**32.8%**

Finally, to summarize, Figure 3 shows the average MOTP relative improvements over *SRP-PHAT* obtained by our CNN proposal using different fine tuning subsets, and its comparison with the GMBF results, for all the signal window sizes.

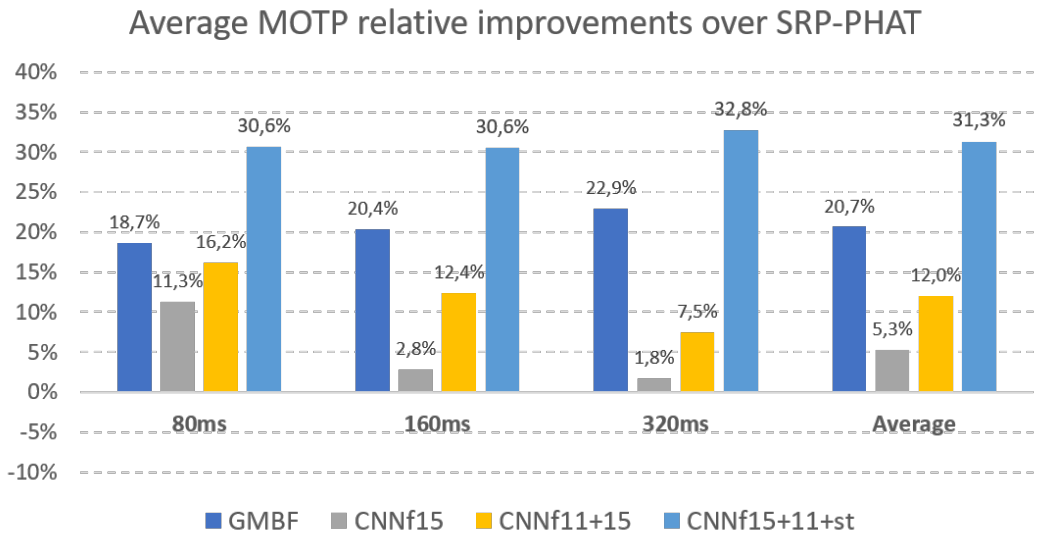

Figure 3. MOTP relative improvements over SRP-PHAT for GMBF and CNN using different fine tuning subsets (for all window sizes).

From the results obtained by our proposal, it is clear that the highest contribution to the improvements from the bare CNN training is the fine tuning procedure with limited data (CNNf15, comparing Tables 3 and 4), while the addition of additional fine tuning material consistently improves the results (Tables 6, and 8). It is again worth noticing that these improvements are consistently independent of the gender of the considered speaker and whether there is a match or not between the static or dynamic activity of the speakers being used in the fine tuning subsets. This suggest that the network is actually learning the acoustic cues that are related to the localization problem, so that we can conclude that our proposal is a suitable and promising strategy for solving the ASL task.

5. Conclusions

We have presented in this paper the first audio localization CNN that is trained end-to-end from the audio signals to the source position. We show that this method is very promising, outperforming the state-of-the-art methods [20,69] and those using *SRP-PHAT*, given that sufficient fine tuning data is available. In addition, our experiments show that the CNN method exhibits good resistance against varying gender of the speaker and different window sizes compared with the baseline methods. Given that the amount of data recordings for audio localization is limited at the moment, we have thus proposed in the paper to first train the network using semi-synthetic data followed by fine tuning using a small amount of real data. This has been a common strategy in other fields to prevent overfitting, and we show in the paper that it significantly improves the system performance as compared with training the network from scratch using real data.

In a future line of work we plan to improve the generation of semi-synthetic data including reverberation effects and testing in detail the effects of gender and language in the system performance. In addition we plan to include more real data by developing a large corpus for audio localization, that will be made available to the scientific community for research purposes. Also, an extensive evaluation will be carried out to asses the impact of the proposal with more complex acquisition scenarios (comprising a higher number of microphone pairs).

Author Contributions: Conceptualization, Daniel Pizarro; Methodology, Writing - review & editing and visualization, Daniel Pizarro, Juan Manuel Vera-Diaz and Javier Macias-Guarasa; Investigation, Juan Manuel Vera-Diaz; Writing - original draft, Juan Manuel Vera-Diaz; Software, Daniel Pizarro and Juan Manuel Vera-Diaz; Resources Javier Macias-Guarasa; Funding Acquisition, Daniel Pizarro and Javier Macias-Guarasa

Funding: Parts of this work were funded by the Spanish Ministry of Economy and Competitiveness under projects HEIMDAL (TIN2016-75982-C2-1-R), ARTEMISA (TIN2016-80939-R), and SPACES-UAH (TIN2013-47630-C2-1-R), and by the University of Alcalá under projects CCGP2017/EXP-025 and CCG2016/EXP-010. Juan Manuel Vera-Diaz is funded by Comunidad de Madrid and FEDER under contract reference number PEJD-2017-PRE/TIC-4626.

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

1. Torres-Solis, J.; Falk, T.H.; Chau, T. A review of indoor localization technologies: towards navigational assistance for topographical disorientation. In *Ambient Intelligence*; InTech, 2010.
2. Ruiz-López, T.; Garrido, J.L.; Benghazi, K.; Chung, L. A survey on indoor positioning systems: foreseeing a quality design. In *Distributed Computing and Artificial Intelligence*; Springer, 2010; pp. 373–380.
3. Mainetti, L.; Patrono, L.; Sergi, I. A survey on indoor positioning systems. Software, Telecommunications and Computer Networks (SoftCOM), 2014 22nd International Conference on. IEEE, 2014, pp. 111–120.
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, pp. 1097–1105.
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
6. DiBiase, J. A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays. PhD thesis, Brown University, 2000.
7. Nunes, L.O.; Martins, W.A.; Lima, M.V.; Biscainho, L.W.; Costa, M.V.; Goncalves, F.M.; Said, A.; Lee, B. A steered-response power algorithm employing hierarchical search for acoustic source localization using microphone arrays. *IEEE Transactions on Signal Processing* **2014**, *62*, 5171–5183.
8. Cobos, M.; García-Pineda, M.; Arevalillo-Herráez, M. Steered response power localization of acoustic passband signals. *IEEE Signal Processing Letters* **2017**, *24*, 717–721.
9. He, H.; Wang, X.; Zhou, Y.; Yang, T. A steered response power approach with trade-off prewhitening for acoustic source localization. *The Journal of the Acoustical Society of America* **2018**, *143*, 1003–1007.
10. Salvati, D.; Drioli, C.; Foresti, G.L. Sensitivity-Based Region Selection in the Steered Response Power Algorithm. *Signal Processing* **2018**.

11. Brandstein, M.S.; Silverman, H.F. A practical methodology for speech source localization with microphone arrays. *Computer Speech & Language* **1997**, *11*, 91–126. doi:10.1006/csla.1996.0024.
12. DiBiase, J.; Silverman, H.; Brandstein, M. Robust localization in reverberant rooms. *Microphone Arrays* **2001**, pp. 157–180.
13. Knapp, C.; Carter, G. The generalized correlation method for estimation of time delay. *Acoustics, Speech and Signal Processing, IEEE Transactions on* **1976**, *24*, 320 – 327. doi:10.1109/TASSP.1976.1162830.
14. Zhang, C.; Florencio, D.; Zhang, Z. Why does PHAT work well in low noise, reverberative environments? Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, 2008, pp. 2565 –2568. doi:10.1109/ICASSP.2008.4518172.
15. Dmochowski, J.P.; Benesty, J. Steered Beamforming Approaches for Acoustic Source Localization. In *Speech Processing in Modern Communication*; Cohen, I.; Benesty, J.; Gannot, S., Eds.; Springer Berlin Heidelberg, 2010; Vol. 3, *Springer Topics in Signal Processing*, pp. 307–337. 10.1007/978-3-642-11130-3_12.
16. Cobos, M.; Marti, A.; Lopez, J. A modified SRP-PHAT functional for robust real-time sound source localization with scalable spatial sampling. *Signal Processing Letters, IEEE* **2011**, *18*, 71–74.
17. Butko, T.; Gonzalez Pla, F.; Segura Perales, C.; Nadeu Camprubí, C.; Hernando Pericás, F.J. Two-source acoustic event detection and localization: online implementation in a smart-room. Proceedings of the 17th European Signal Processing Conference (EUSIPCO'11), 2011, pp. 1317–1321.
18. Habets, E.A.P.; Benesty, J.; Gannot, S.; Cohen, I. The MVDR Beamformer for Speech Enhancement. In *Speech Processing in Modern Communication*; Cohen, I.; Benesty, J.; Gannot, S., Eds.; Springer Berlin Heidelberg, 2010; Vol. 3, *Springer Topics in Signal Processing*, pp. 225–254. 10.1007/978-3-642-11130-3_9.
19. Marti, A.; Cobos, M.; Lopez, J.J.; Escolano, J. A steered response power iterative method for high-accuracy acoustic source localization. *The Journal of the Acoustical Society of America* **2013**, *134*, 2627–2630, [https://doi.org/10.1121/1.4820885]. doi:10.1121/1.4820885.
20. Velasco, J.; Pizarro, D.; Macias-Guarasa, J. Source Localization with Acoustic Sensor Arrays Using Generative Model Based Fitting with Sparse Constraints. *Sensors* **2012**, *12*, 13781–13812. doi:10.3390/s121013781.
21. Padois, T.; Sgard, F.; Doutres, O.; Berry, A. Comparison of acoustic source localization methods in time domain using sparsity constraints. 2015. cited By 0.
22. Velasco, J.; Pizarro, D.; Macias-Guarasa, J.; Asaei, A. TDOA Matrices: Algebraic Properties and Their Application to Robust Denoising With Missing Data. *IEEE Transactions on Signal Processing* **2016**, *64*, 5242–5254. doi:10.1109/TSP.2016.2593690.
23. Compagnoni, M.; Pini, A.; Canclini, A.; Bestagini, P.; Antonacci, F.; Tubaro, S.; Sarti, A. A Geometrical-Statistical Approach to Outlier Removal for TDOA Measurements. *IEEE Transactions on Signal Processing* **2017**, *65*, 3960–3975. doi:10.1109/TSP.2017.2701311.
24. Salari, S.; Chan, F.; Chan, Y.T.; Read, W. TDOA Estimation With Compressive Sensing Measurements and Hadamard Matrix. *IEEE Transactions on Aerospace and Electronic Systems* **2018**, pp. 1–1. doi:10.1109/TAES.2018.2826230.
25. Murray, J.C.; Erwin, H.R.; Wermter, S. Robotic sound-source localisation architecture using cross-correlation and recurrent neural networks. *Neural Networks* **2009**, *22*, 173 – 189. What it Means to Communicate, doi:https://doi.org/10.1016/j.neunet.2009.01.013.
26. Deleforge, A. Acoustic Space Mapping: A Machine Learning Approach to Sound Source Separation and Localization. Theses, Université de Grenoble, 2013.
27. Salvati, D.; Drioli, C.; Foresti, G.L. On the use of machine learning in microphone array beamforming for far-field sound source localization. 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), 2016, pp. 1–6. doi:10.1109/MLSP.2016.7738899.
28. Rascon, C.; Meza, I. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems* **2017**, *96*, 184 – 210. doi:https://doi.org/10.1016/j.robot.2017.07.011.
29. Stoica, P.; Li, J. Lecture Notes - Source Localization from Range-Difference Measurements. *IEEE Signal Processing Magazine* **2006**, *23*, 63–66. doi:10.1109/SP-M.2006.248717.
30. Cobos, M.; García-Pineda, M.; Arevalillo-Herráez, M. Steered Response Power Localization of Acoustic Passband Signals. *IEEE Signal Processing Letters* **2017**, *24*, 717–721. doi:10.1109/LSP.2017.2690306.
31. Omologo, M.; Svaizer, P. Use Of The Cross-Power-Spectrum Phase In Acoustic Event Location. *IEEE Trans. on Speech and Audio Processing* **1993**, *5*, 288–292.

32. Dmochowski, J.; Benesty, J.; Affes, S. A Generalized Steered Response Power Method for Computationally Viable Source Localization. *Audio, Speech, and Language Processing, IEEE Transactions on* **2007**, *15*, 2510 –2526. doi:10.1109/TASL.2007.906694.
33. Badali, A.; Valin, J.M.; Michaud, F.; Aarabi, P. Evaluating real-time audio localization algorithms for artificial audition in robotics. Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, 2009, pp. 2033 –2038. doi:10.1109/IROS.2009.5354308.
34. Do, H.; Silverman, H. SRP-PHAT methods of locating simultaneous multiple talkers using a frame of microphone array data. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, 2010, pp. 125 –128. doi:10.1109/ICASSP.2010.5496133.
35. Schmidt, R. Multiple emitter location and signal parameter estimation. *Antennas and Propagation, IEEE Transactions on* **1986**, *34*, 276 – 280. doi:10.1109/TAP.1986.1143830.
36. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep learning*; Vol. 1, MIT press Cambridge, 2016.
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, pp. 1097–1105.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2016**, pp. 770–778.
39. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; others. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* **2012**, *29*, 82–97.
40. Graves, A.; Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. International Conference on Machine Learning, 2014, pp. 1764–1772.
41. Deng, L.; Platt, J.C. Ensemble deep learning for speech recognition. INTERSPEECH, 2014.
42. Steinberg, B.Z.; Beran, M.J.; Chin, S.H.; Howard, J.H. A neural network approach to source localization. *The Journal of the Acoustical Society of America* **1991**, *90*, 2081–2090, [https://doi.org/10.1121/1.401635]. doi:10.1121/1.401635.
43. Datum, M.S.; Palmieri, F.; Moiseff, A. An artificial neural network for sound localization using binaural cues. *The Journal of the Acoustical Society of America* **1996**, *100*, 372–383, [https://doi.org/10.1121/1.415854]. doi:10.1121/1.415854.
44. Youssef, K.; Argentieri, S.; Zarader, J.L. A learning-based approach to robust binaural sound localization. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2927–2932. doi:10.1109/IROS.2013.6696771.
45. Xiao, X.; Zhao, S.; Zhong, X.; Jones, D.L.; Siong, C.E.; Li, H. A learning-based approach to direction of arrival estimation in noisy and reverberant environments. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2015**, pp. 2814–2818.
46. Ma, N.; Brown, G.; May, T., Exploiting deep neural networks and head movements for binaural localisation of multiple speakers in reverberant conditions. In *Proceedings of Interspeech 2015*; ISCA, 2015; pp. 3302–3306.
47. Takeda, R.; Komatani, K. Discriminative multiple sound source localization based on deep neural networks using independent location model. *2016 IEEE Spoken Language Technology Workshop (SLT)* **2016**, pp. 603–609.
48. Takeda, R.; Komatani, K. Sound source localization based on deep neural networks with directional activate function exploiting phase information. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2016**, pp. 405–409.
49. Takeda, R.; Komatani, K. Unsupervised adaptation of deep neural networks for sound source localization using entropy minimization. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 2217–2221. doi:10.1109/ICASSP.2017.7952550.
50. Sun, Y.; Chen, J.; Yuen, C.; Rahardja, S. Indoor Sound Source Localization With Probabilistic Neural Network. *IEEE Transactions on Industrial Electronics* **2018**, *65*, 6403–6413. doi:10.1109/TIE.2017.2786219.
51. Chakrabarty, S.; Habets, E.A.P. Multi-Speaker Localization using Convolutional Neural Network Trained with Noise. ML4Audio Workshop at NIPS, 2017.
52. Yalta, N.; Nakadai, K.; Ogata, T. Sound source localization using deep learning models. *Journal of Robotics and Mechatronics* **2017**, *29*, 37–48. doi:10.20965/jrm.2017.p0037.
53. Ferguson, E.L.; Williams, S.B.; Jin, C.T. Sound Source Localization in a Multipath Environment Using Convolutional Neural Networks. *CoRR* **2017**, *abs/1710.10948*, [1710.10948].

54. Hirvonen, T. Classification of Spatial Audio Location and Content Using Convolutional Neural Networks. 138th Audio Engineering Society Convention 2015, 2015, Vol. 2.
55. He, W.; Motlícek, P.; Odobez, J. Deep Neural Networks for Multiple Speaker Detection and Localization. *CoRR* **2017**, *abs/1711.11565*, [1711.11565].
56. Salvati, D.; Drioli, C.; Foresti, G.L. Exploiting CNNs for Improving Acoustic Source Localization in Noisy and Reverberant Conditions. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2018**, *2*, 103–116. doi:10.1109/TETCI.2017.2775237.
57. Ma, W.; Liu, X. Phased Microphone Array for Sound Source Localization with Deep Learning. *CoRR* **2018**, *abs/1802.04479*, [1802.04479].
58. Thuillier, E.; Gamper, H.; Tashev, I. Spatial audio feature discovery with convolutional neural networks. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.
59. Pertilä, P.; Cakir, E. Robust direction estimation with convolutional neural networks based steered response power. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 6125–6129. doi:10.1109/ICASSP.2017.7953333.
60. Le, Q.V.; Ngiam, J.; Coates, A.; Lahiri, A.; Prochnow, B.; Ng, A.Y. On optimization methods for deep learning. Proceedings of the 28th International Conference on International Conference on Machine Learning. Omnipress, 2011, pp. 265–272.
61. Allen, J.B.; Berkley, D.A. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America* **1979**, *65*, 943–950, [https://doi.org/10.1121/1.382599]. doi:10.1121/1.382599.
62. Velasco, J.; Martín-Arguedas, C.J.; Macias-Guarasa, J.; Pizarro, D.; Mazo, M. Proposal and validation of an analytical generative model of SRP-PHAT power maps in reverberant scenarios. *Signal Processing* **2016**, *119*, 209 – 228. doi:http://dx.doi.org/10.1016/j.sigpro.2015.08.003.
63. Lathoud, G.; Odobez, J.M.; Gatica-Perez, D. AV16.3: An Audio-Visual Corpus for Speaker Localization and Tracking. Procceedings of the MLMI; Bengio, S.; Bourlard, H., Eds. Springer-Verlag, 2004, Vol. 3361, *Lecture Notes in Computer Science*, pp. 182–195.
64. Moore, D.C. The IDIAP Smart Meeting Room. Technical report, IDIAP Research Institute, Switzerland, 2004.
65. Lathoud, G. AV16.3 Dataset. http://www.idiap.ch/dataset/av16-3/ (accessed on 11 october 2012), 2004.
66. Association, E.E.L.R. Albayzin corpus. http://catalogue.elra.info/en-us/repository/browse/albayzin-corpus/b50c9628a9dd11e7a093ac9e1701ca0253c876277d534e7ca4aca155a5611535/.
67. Moreno, A.; Poch, D.; Bonafonte, A.; Lleida, E.; Llisterri, J.; Mariño, J.B.; Nadeu, C. Albayzin speech database: design of the phonetic corpus. EUROSPEECH. ISCA, 1993.
68. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
69. Velasco-Cerpa, J.F. Mathematical Modelling and Optimization Strategies for Acoustic Source Localization in Reverberant Environments. PhD thesis, Escuela Politécnica Superior. University of Alcalá (Spain), 2017.
70. Mostefa, D.; Garcia, M.; Bernardin, K.; Stiefelhagen, R.; McDonough, J.; Voit, M.; Omologo, M.; Marques, F.; Ekenel, H.; Pnevmatikakis, A. Clear evaluation plan, document CHIL-CLEAR-V1.1 2006-02-21. http://www.clear-evaluation.org/clear06/downloads/chil-clear-v1.1-2006-02-21.pdf (accessed on 11 october 2012), 2006.