RAY SPACE TRANSFORM INTERPOLATION WITH CONVOLUTIONAL AUTOENCODER

L. Comanducci, F. Borra, P. Bestagini, F. Antonacci, A. Sarti, S. Tubaro

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

ABSTRACT

In this paper we propose an algorithm for the reconstruction of the Ray Space Transform (RST) through the use of neural networks. In particular, our aim is to reconstruct the magnitude of the RST acquired from a linear microphone array, as if the array were composed by a larger amount of microphones. This is useful for applications that need a higher RST resolution when only a limited amount of microphones can be used due to practical constraints or physical limitations. The proposed solution leverages recent advancements in deep learning as it is based on a fully convolutional autoencoder. To validate our method, we show through a simulative campaign that it is possible to improve sound source localization using the reconstructed RST compared to the use of the original RST.

***Index Terms*—** Ray space, source localization, deep learning, convolutional neural networks

1. INTRODUCTION

Space-time audio processing has gained large interest in the research community for the applications in virtual reality, hands-free communication, video-conferencing, as well as other fields. Within this context, in this paper we propose a deep learning solution aimed at interpolating the signals acquired by a linear microphone array where some microphones are missing, to obtain the RST that would have been estimated by the complete microphone array.

The *Ray Space* [1] is a space where a point corresponds uniquely to an acoustic ray (i.e. a straight line along which acoustic energy travels). If a proper parameterization of rays is adopted, rays departing from an acoustic source are mapped in the Ray Space onto linear patterns [2]. The parameters of these patterns are uniquely determined by the source location and the array geometry. The *Ray Space Transform* (RST) [3] is a linear operation that maps array data on the Ray Space in a sub-band-wise fashion.

The intuition behind the proposed work is that, as rays coming from the same acoustic object are naturally clustered in the ray space onto linear patterns, magnitudes of RSTs can be considered as images showing characteristic shapes and patterns that can be easily learned through Convolutional Neural Networks (CNNs). It is then possible to reconstruct RSTs by exploiting CNNs principles used to solve interpolation or super-resolution problems in the image processing literature.

The use of learning-based procedures for space-time processing is not novel in the literature. In one of the first applications of neural networks to the problem of source localization [4], the authors estimate the direction of a sound source from the signals detected by two directional, spatially separate receivers. In [5], the authors use an architecture based on deep neural networks (DNN) to localize a source in a room, possibly also keeping into account audio captured in other rooms. The input provided to the network consists of the Generalized Cross Correlations of the microphones within the arrays in use. In [6], the authors feed the phase component of the short-time Fourier transform coefficients of the received microphone signals into a CNN to estimate the broadband Direction Of Arrival (DOA). More in general, learning-based procedures are used, as an example, also in [7], where authors address the problem of single source localization in ad-hoc microphone networks through a Bayesian inference approach.

In this work, we aim at recovering the magnitude of the Ray Space Transform of a complete array (in the following *complete RST*) from the corrupted Ray Space Transform obtained from an array with missing microphones (*undercomplete RST*). This is useful for applications based on RST analysis, whenever a complete microphone array cannot be deployed in the environment due to physical limitations or cost constraints. We adopt a specific CNN architecture known as fully convolutional autoencoder [8, 9]. During training, the network receives as input pairs of images representing the magnitude of a simulated complete and undercomplete RSTs, and learns the correspondence between the two inputs for a set of training source positions. During training, only one source is active at any time. During test, the network is fed with the undercomplete RST where more than one source is active, and produces an estimate of the complete RST. The ability of the network to estimate the RST of multiple sources even if training has been conducted with only a single source active, demonstrates the capability of the network to generalize.

In order to evaluate the effectiveness of the proposed system, we show that by using the proposed RST reconstruction method it is possible to achieve higher localization accuracy than using the original RST acquired with fewer microphones.

2. SIGNAL MODEL AND PROBLEM STATEMENT

In this section we first define the array signal model adopted, then we review the Ray Space Transform and finally we formulate the problem.

The signal received at the lth microphone of the array can be modeled, in the frequency domain, as

$$P(\mathbf{r}_l, \omega) = \sum_{n=1}^{N} G(\mathbf{r}_l, \mathbf{r'}_n, \omega) S_n(\omega) + e_l(\omega), \tag{1}$$

where ω is the angular frequency, $\mathbf{r}_l = [x_l, y_l]^T$ is the position of the lth microphone, $\mathbf{r'}_n = [x'_n, y'_n]^T$ and $S_n(\omega)$ are the position and the signal emitted by the nth source, respectively, N is the total number of sources, $e_l(\omega)$ is the lth microphone additive noise, and $G(\mathbf{r}_l, \mathbf{r'}_n, \omega)$ is the channel frequency response between the lth source and the nth microphone. The signals acquired by an uniform linear array, can be conveniently mapped in a domain known as *ray space* through the Ray Space Transform [3]. The ray space consists in the (m, q) coordinates of lines on which acoustic rays lie, where

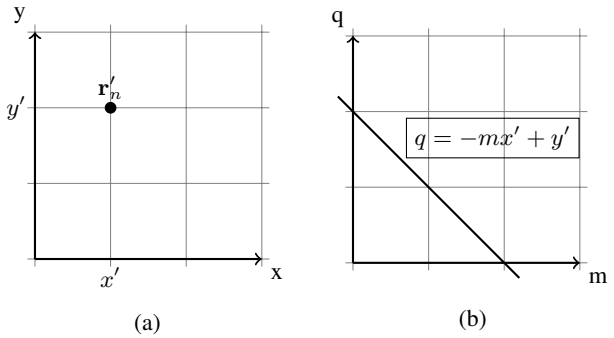

Fig. 1: Example of the representation of a point source $\mathbf{r}'_n$ in the geometric space (a) and in the ray space (b)

m and q are the slope and intercept of the line on the y axis, respectively. In a dual fashion, acoustic rays departing from a point (x', y') in the geometric space are mapped onto the line $q = -mx' + y'$ in the ray space, as shown in Fig. 1. If we have a uniform linear array of L sensors spaced by d and displaced along the y axis (i.e $x_l = 0 \quad \forall l$, $y_l = (l - (L-1)/2)d$), the Ray Space Transform is defined as

$$[\mathbf{Y}]_{i,w}(\omega) = d \sum_{l=0}^{L-1} P(\mathbf{r}_l, \omega) e^{-j\frac{\omega}{c}\frac{y_l m_w}{\sqrt{1+m_w^2}}} e^{-\pi \frac{(y_l - q_i)^2}{\sigma^2}}, \qquad (2)$$

where $[\cdot]_{i,w}$ indicates the (i,w)th element of a matrix, c is the speed of sound, while $q_i = \bar{q}(i - (I-1)/2)$, $i = 0, \ldots, I-1$, $m_w = \bar{m}(w - (W-1)/2)$, $w = 0, \ldots, W-1$ denote the samples on the m and q axes, respectively, I and W are the number of samples in the Ray Space and σ is the width of the Gaussian window. As we can see from (2), the RST employs a sliding Gaussian window modulated by a complex exponential to map the microphone array signal onto the (m, q) ray space domain. When dealing with wide-band signals, it is useful to define a wide-band extension $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I \times W}$ of the RST. In particular, it is computed as the geometric mean, over all the $K/2$ frequency bins, of the magnitude of the RSTs $|\mathbf{Y}(\omega_k)|$ [3].

In this paper, we deal with uniform linear array where random missing microphones are present. In this scenario, if we define $\mathcal{P}$ and $\mathcal{M}$ as the set of present and missing microphone index, respectively, the signal at the lth microphone is given by

$$P(\mathbf{r}_l, \omega) = \begin{cases} 0 & l \in \mathcal{M} \\ \sum_{n=1}^{N} G(\mathbf{r}_l, \mathbf{r'}_n, \omega) S_n(\omega) + e_l(\omega) & l \in \mathcal{P}. \end{cases} \qquad (3)$$

Given the model in (3), the RST and its wide-band extension are modified accordingly. In particular, let us call $\tilde{\boldsymbol{\mathcal{Y}}}$ the wide-band undercomplete RST. As an example, Fig. 2a) and Fig. 2b show the absolute value of complete and undercomplete wideband RSTs, respectively. Our goal is to find an operator $\mathcal{U}$ that, given as input $\tilde{\boldsymbol{\mathcal{Y}}}$ is able to reconstruct an estimate $\hat{\boldsymbol{\mathcal{Y}}}$ of the complete RST $\boldsymbol{\mathcal{Y}}$, as in Fig. 2c, i.e $\hat{\boldsymbol{\mathcal{Y}}} = \mathcal{U}\left(\tilde{\boldsymbol{\mathcal{Y}}}\right)$. In the following we propose a solution to find the operator $\mathcal{U}$ that adopts a learning-based approach.

3. PROPOSED SOLUTION

In order to reconstruct the information lost in the undercomplete RST, we propose a solution based on a specific CNN known as convolutional autoencoder, inspired by the application of such architecture to many image restoration problems [8, 10]. Other types of techniques, such as low-pass filtering, simply extrapolate from the available information, but in any case they do not take advantage of a-priori knowledge as reconstruction through CNNs does. In the following, we report all the details about the autoencoder architecture, its training strategy, and we explain how to deploy it at testing time.

3.1. CNN Architecture

We propose a CNN whose input is a 64×64 sample image representing the magnitude of a wide band undercomplete RST $\tilde{\boldsymbol{\mathcal{Y}}}$. The CNN output is a 64×64 sample image representing the estimated magnitude of the complete RST $\hat{\boldsymbol{\mathcal{Y}}}$. This complete RST estimate can then used for source localization, as well as for other problems tackled in the Ray Space.

The CNN we propose is a fully convolutional autoencoder [9]. This means that only convolutional layers and non-linear activations are used. As any autoencoder, the proposed architecture can be split into two parts known as *encoder* and *decoder*. The proposed encoder is composed by a series of five 2D convolutional layer with the following parameters: i) 256 filters of size 6×6 and stride 1×1; ii) 128 filters of size 5×5 and stride 2×2; iii) 128 filters of size 4×4 and stride 2×2; iv) 64 filters of size 4×4 and stride 2×2; v) 32 filters of size 3×3 and stride 2×2.

The structure of the decoder is the same one of the encoder where the numbers and sizes of the filters are reversed, and transposed convolutions are used instead of normal convolution. The only exception is the last layer, which makes use of a single filter followed by rectified linear unit (ReLU) non-linearity in order to return a one-channel 64×64 sample image. In preliminary simulations we tested that ReLU performs better with respect to other smoother activations functions such as the sigmoid.

The whole architecture implementing the operator $\mathcal{U}$ is the concatenation of the encoder and decoder. In order to choose the presented model, we preliminarily tested the performance of different types of architectures. In particular, we considered the same architecture presented here and all the possible combinations given by the addition of both a batch normalization and a ReLU layer after each convolutional one. The results showed that the chosen architecture exhibits the best trade-off between performance and simplicity.

Notice that, even by fixing the input resolution to 64×64, it is still possible to apply the proposed architecture to RSTs of different size. Indeed, in our experiments, we resize slightly larger RSTs to 64×64 sample using bilinear interpolation [11]. Then we re-scale the output back to its original size. However, working with a network with such a small input enables to greatly reduce training time, as well as the amount of needed training data.

3.2. Training strategy

In order to properly train the proposed architecture, we need a set of RST pairs representing $\tilde{\boldsymbol{\mathcal{Y}}}$ and $\boldsymbol{\mathcal{Y}}$ for different source positions. To this purpose, we consider a set of source positions $\mathcal{S}^{(\text{train})} = \{\mathbf{r}_n^{(\text{train})} | n = 1, \ldots, N^{(\text{train})}\}$. For each one of these sources, we generate two images: i) $\tilde{\boldsymbol{\mathcal{Y}}}_n^{(\text{train})}$, which represents the wide band extension of the undercomplete RST; ii) $\boldsymbol{\mathcal{Y}}_n^{(\text{train})}$, which represents the wide band extension of the complete RST. To generate the input RSTs, we consider a reverberant environment where additive noise is present at the microphones. The setting used to acquire the desired output images is a free-field environment where no additive noise is present at the microphones, allowing us to perform some kind of super-resolution reconstruction of the images.

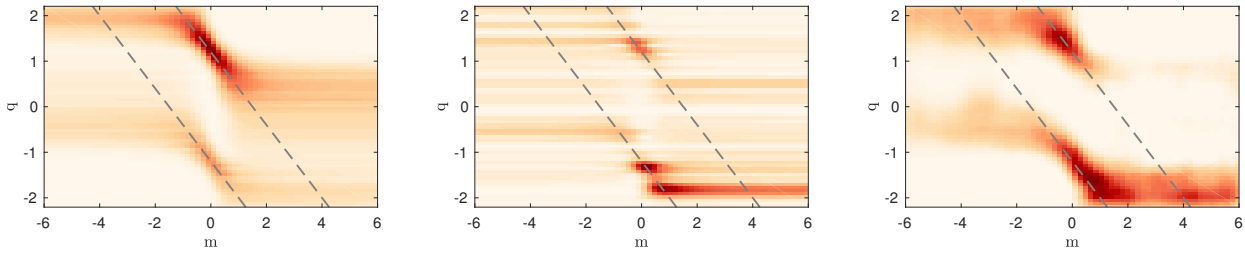

Fig. 2: Example of the wideband extension of the RST obtained with a uniform array without missing microphones $\mathcal{Y}$ (a), with missing microphones $\tilde{\mathcal{Y}}$ (b) and the decoded image $\hat{\mathcal{Y}}$ (c), when two sources are present in the acoustic scene. Dashed lines represent the theoretical position of lines due to the sources.

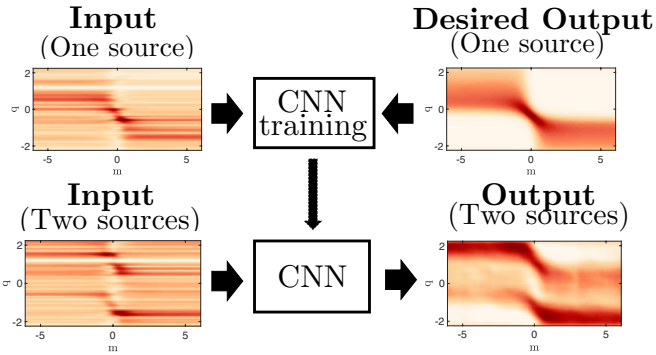

Fig. 3: Representation of the training and test procedures.

In order to train the network, we use as loss function the mean-squared error between the output of the autoencoder $\hat{\mathcal{Y}}_n^{(\text{train})}$ and the ideal version of it $\mathcal{Y}_n^{(\text{train})}$. Training is performed using Adam optimizer with default parameters given in [12]. The number of epochs (i.e the amount of times the optimizer iterates over the whole training set) is empirically set to 500 in order to ensure network convergence. The best model is picked as the one minimizing the loss over a small validation set of data. CNN convergence is typically reached before 100 epochs, though. Notice that, in order to ensure numerical stability in the optimization process, we normalize all input images in the range $[0, 1]$.

3.3. System deployment

Once the network is trained, we can use it to process new RSTs. Given a RST $\tilde{\mathcal{Y}}_n^{(\text{test})}$ acquired with missing microphones, we scale it using bilinear-interpolation to a size of 64×64 sample. Then, we normalize its sample values to span the range $[0, 1]$. We finally feed this modified RST to the CNN. The output is a 64×64 sample image that can be resized and rescaled back to the original size, thus obtaining $\hat{\mathcal{Y}}_n^{(\text{test})}$.

Notice that, even though training is performed considering the presence of a single source in the environment, test conditions can be different. Fig. 3 shows the training and test pipeline in case one source is considered in training, and two sources are considered for testing (i.e., the challenging scenario considered in our experiments). However, to augment the reconstruction capabilities of the network, it is better, but not strictly necessary, to use the same configuration of missing microphones $\mathcal{M}$ both during training and test.

4. SIMULATION RESULTS

In this section we present the simulation results regarding the developed model, in order to show its effectiveness. First we present the evaluation metrics, then we show the setup of the simulated acoustic scenes used for the tests. We finally present results regarding the reconstruction capabilities of our network, also in terms of source localization.

4.1. Evaluation metrics

In order to evaluate the quality of the reconstructed images we use the Structural Similarity Index (SSIM) as defined in [13]. This measures the similarity between two images, and ranges between 0 (i.e. very dissimilar images) and 1 (i.e. identical images).

As for the localization, we adopt the Root Mean Square Error (RMSE) between the localized and actual source positions. More specifically, let us consider N sources. The RMSE between the true sources position $\mathbf{r}_n$ and the estimated ones $\hat{\mathbf{r}}_n$ is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{N} ||\mathbf{r}_n - \hat{\mathbf{r}}_n||^2}{N}}. \tag{4}$$

4.2. Simulation setup

The setup considered in order to generate the train and test images is shown on the left of Fig. 4. In particular we consider $N^{(\text{train})} = 432$ train sources and $N^{(\text{test})} = 9$ test sources. Notice that training and test source positions never overlap.

The room configurations used in order to generate the input and desired output images are summarized in Tab. 1. As source signal, we used realizations of white Gaussian noise, considering a variance of $\sigma^{(\text{train})} = 0.05$ for training and a variance of $\sigma^{(\text{test})} = 1$ for test. In this way we show that our network is able to generalize to signals not seen during the training phase. Moreover, test signals are corrupted with a zero-mean Gaussian noise with fixed input signal-to-noise ratio (iSNR) of 20 dB.

In all our experiments we considered that the index of the missing microphones $\mathcal{M}$ is the same during training and test phase. In particular we considered settings with a number of missing microphone $|\mathcal{M}| \in \{10, 20, 30, 40, 50\}$. The training was performed by generating $N^{(\text{train})}$ RSTs, each one of which is built considering

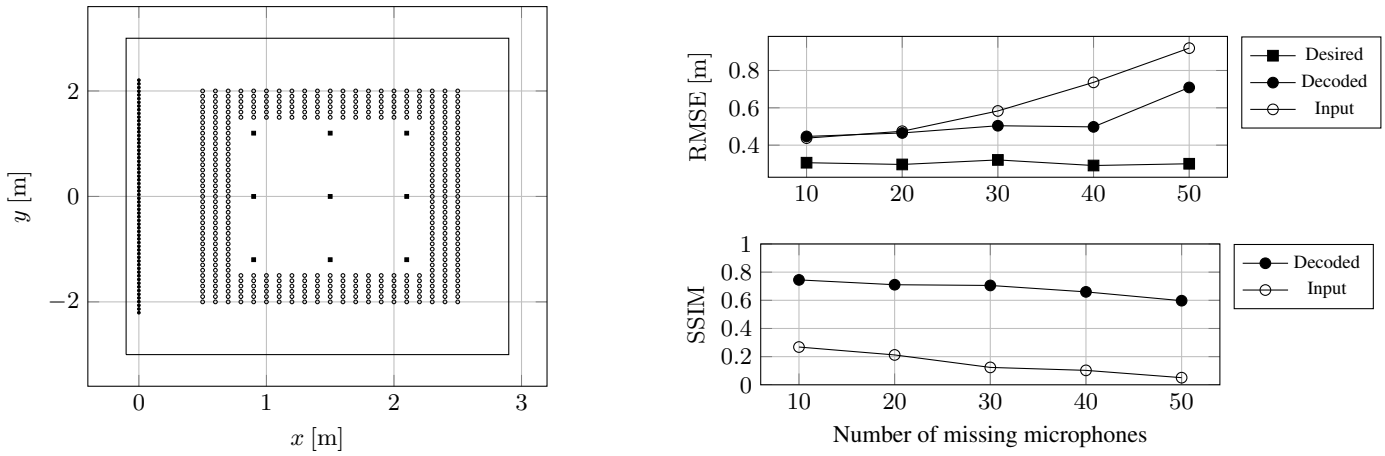

Fig. 4: (Left) Simulation setup showing the array (•), training sources (○) and test sources (■). (Right) RMSE and SSIM as a function of the number of missing microphones. SSIM compares desired RSTs with input (○) and decoded (•) ones.

Table 1: Room configurations used to generate the input and desired output images.

	Input	**Desired output**
Room size	(3×6)	∞
Reflection coefficient	$\rho = 0.8$	$\rho = 0$
iSNR	20dB	∞dB

only one emitting source placed in $\mathbf{r}_n^{(\text{train})}$. Conversely, in order to generate the test RSTs, we considered the case of two acoustic sources emitting at the same time, by enumerating all possible pairs of $N^{(\text{test})}$ sources that do not result in systematic errors in the ideal case (e.g. aligned sources). For each configuration of missing microphones, we considered 10 realizations of noise in order to generate multiple RSTs for the same pair of sources.

4.3. SSIM results

On the bottom right of Fig. 4 we present results about RST estimation in terms of SSIM. Specifically, we compare the SSIM obtained without using the undercomplete RST (i.e. $\tilde{\boldsymbol{\mathcal{Y}}}_n^{(\text{test})}$ compared to $\boldsymbol{\mathcal{Y}}_n^{(\text{test})}$), and the reconstructed RST (i.e. $\hat{\boldsymbol{\mathcal{Y}}}_n^{(\text{test})}$ compared to $\boldsymbol{\mathcal{Y}}_n^{(\text{test})}$)

The SSIM relative to the reconstructed images is higher if compared to the SSIM relative to the input images, for all the number of missing microphones considered. This shows that our network is able to reconstruct images that are similar to the desired ones. Clearly, as the number of missing microphones increases from 10 to 50, SSIM decreases. However, when the undercomplete RST shows SSIM smaller than 0.1, the recovered RST still has SSIM greater than 0.6.

4.4. Localization results

Source localization based on RST wideband extension is carried out following the technique proposed in [14]. In a nutshell, localization is accomplished by finding the peaks of the RST wideband extension under analysis, and then performing a clustering operation followed by linear regression. This enables to identify linear patterns corresponding to the acoustic sources (see Fig. 2a). The number of lines indicates the number of sources. The line equation is mapped into a source position. We set the number of sought lines to two. As proposed in [14], we used RANSAC algorithm [15] for fitting the lines.

On the top right of Fig. 4, we show results in terms of localization performances when using our network. We stress the fact that for each test image we localize the two emitting sources in a reverberant environment, which is a challenging task by itself. Both source positions are considered in order to compute the RMSE. Notice that the localization accuracy relative to the images reconstructed by our network $\hat{\boldsymbol{\mathcal{Y}}}_n^{(\text{test})}$ are similar to the ones relative to the input images $\tilde{\boldsymbol{\mathcal{Y}}}_n^{(\text{test})}$ only if the number of missing microphones is small. As the number of missing microphones increases, the reconstructed results show increasing better performances. Compared against the ideal case (complete RST), the localization error is almost constant, except for the very challenging case of 50 missing microphones over 64.

5. CONCLUSIONS

In this paper we have proposed a methodology that applies convolutional autoencoders to the Ray Space Transform, in order to reconstruct the information lost when using arrays with missing microphones for acquisition. The goal is to reconstruct the RST as if it were acquired with the complete array. This enables to simulate the RST of dense arrays in situations in which the only available microphone array is composed by a limited amount of microphones (e.g. due to costs, deployment limitations, array faults, etc.).

The capability of the network to reconstruct the RST both in terms of image quality, and in terms of source localization information demonstrate the feasibility of the presented approach. Indeed, we showed that it is possible to localize acoustic sources using a reconstructed RST with higher accuracy compared to the use of an undercomplete RST. The network also generalizes when it is tested on data unseen during training (i.e. multiple sources active at the same time, and source positions never used for training).

Results contained in this paper are useful not only for the presented scenario, but stimulate us in applying learning architectures for other more challenging RST reconstruction tasks. Future work will be devoted to RST reconstruction using multiple sparse arrays.

6. REFERENCES

[1] F. Antonacci, M. Foco, A. Sarti, and S. Tubaro, "Fast tracing of acoustic beams and paths through visibility lookup," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 812–824, May 2008.

[2] D. Markovic, F. Antonacci, A. Sarti, and S. Tubaro, "Soundfield imaging in the ray space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2493–2505, 2013.

[3] Lucio Bianchi, Fabio Antonacci, Augusto Sarti, and Stefano Tubaro, "The ray space transform: A new framework for wave field processing," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5696–5706, 2016.

[4] Michael S. Datum, Francesco Palmieri, and Andrew Moiseff, "An artificial neural network for sound localization using binaural cues," *The Journal of the Acoustical Society of America*, vol. 100, no. 1, pp. 372–383, 1996.

[5] Fabio Vesperini, Paolo Vecchiotti, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "Localizing speakers in multiple rooms by using deep neural networks," *Computer Speech and Language*, vol. 49, pp. 83 – 106, 2018.

[6] S. Chakrabarty and E. A. P. Habets, "Broadband doa estimation using convolutional neural networks trained with noise signals," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2017, pp. 136–140.

[7] B. Laufer-Goldshtein, R. Talmon, and S. Gannot, "Semi-supervised source localization on multiple manifolds with distributed microphones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1477–1491, July 2017.

[8] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[10] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros, "Context encoders: Feature learning by inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] Earl J. Kirkland, "Bilinear interpolation," in *Advanced Computing in Electron Microscopy*, pp. 261–263. Springer, Boston, MA, 2010.

[12] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[13] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[14] D. Marković, F. Antonacci, L. Bianchi, S. Tubaro, and A. Sarti, "Extraction of acoustic sources through the processing of sound field maps in the ray space," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2481–2494, Dec 2016.

[15] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.