Article

Sound Source Localization Using Graph Regularized Neural Network

Saulius Sakavičius[1,*], Andreas Brendel[2], Artūras Serackis[1] and Walter Kellermann[2]

[1] Vilnius Gediminas Technical University
[2] Friedrich–Alexander University Erlangen–Nürnberg
* Correspondence: saulius.sakavicius@vgtu.lt
‡ These authors contributed equally to this work.

Version October 21, 2020 submitted to Appl. Sci.

Abstract: In this article we present a data-driven approach for a single speech source localization within an acoustic enclosure. Our method consist of high-dimensional acoustic feature extraction, selection based on a fitness criterion, feature manifold learning and low-dimensional embedding, graph dataset construction and an application of a graph-regularized neural network (GRNN) to learn the mapping between the embedded feature coordinates and the Cartesian coordinates of the sound source. Our method relies on the assumption of the feature space spatial smoothness. We present the experimental results of the speech source localization in real acoustic enclosures using two compact circular microphone arrays. We compare the performance of the GRNN for single speech sound source localization to the performance of two baseline algorithms.

Keywords: sound source localization; array signal processing; manifold learning; graph regularization; artificial neural networks

1. Introduction

Sound source localization is an increasingly important component in teleconferencing, autonomous driving, security and human-computer interaction systems [1–3].

Here could be added a list/few examples of common/typical problems in particular situations/applications, that motivates us to keep investigating this field.

Do "camera aiming" is one of the classical tasks that has such name? Camera aiming can be inaccurate due to the parallax error *citation is needed here* if the camera and the microphone array are not rotating on the same axis. To accurately steer the camera to a certain point in space in such setup, a complete set of coordinates, either Cartesian or polar, of the target point is needed. Thus, knowing only the direction of arrival (DoA) of the sound source would not be sufficient. At least two DoAs are needed to describe the source position on a plane. There are plenty of source direction of arrival estimation methods utilizing such compact arrays [4–6], but only a few methods are offered for 2D or 3D source localization that can estimate both DoA and the distance or the Cartesian coordinates of the sound source, involving beamforming and particle filtering [7] averaged directivity patterns of blind source separation systems [8].

It might be desired to use compact microphone arrays that have as few microphone elements as possible. By "compact" we here imply that the dimensions of the microphone array are much smaller than the dimensions of the source localization space.

The Steered Response Power - Phase Transform (SRP-PHAT) algorithm [**?**] has been shown [9] to be one of the most robust sound source localization approaches operating in noisy and reverberant environments. However, its practical implementation is usually based on a costly fine grid-search procedure, making the computational cost of the method a real issue [1].

When invoking sound source localization algorithms, usually free-field and far-field assumptions are made which aren't always true, especially inside acoustic enclosures, in which the diffuse field is observed due to the acoustic reflections. For a particular acoustic enclosure, either of the assumptions might be too coarse and would not reflect the actual situation, which might be a particular combination of either of the acoustic models. This calls for a data-driven approach which can learn better models. Learning-based sound source localization methods might be further advantageous in such circumstances.

There are several learning-based source localization approaches, based on either semi-supervised [10] or supervised [11–13] learning paradigms. In both of these approaches to work, a set of acoustic features from known sound source positions (the labeled dataset) is needed.

Labeled feature acquisition is very costly. On the other side, it is relatively easy to obtain a large dataset of unlabeled audio features. Considering our setting, it is relatively easy to collect a large amount of acoustic features without labels, and it is very tedious to provide labels (in our case – the coordinates of the sound source) for such data. Such learning-based strategy is promising as users spend most of our time in the same environments (living room, office, etc.), where such training data can be obtained and used.

In this article we present a method to localize a sound source in two dimensions using a two compact microphone arrays.

2. Materials and Methods

In this section, we provide a theoretical background for our investigation.

ANN-based sound source localization can be performed using various acoustic features that depend on the sound source, either relative to the microphone array(s) or the acoustic enclosure.

It can be assumed, that acoustic features are spatially smooth, that is, features obtained at spatially close source positions are also close in feature space. The relative distance or affinity of the features can be simply estimated by calculating the Euclidean distance between feature vectors.

It can be assumed, that high dimensional acoustic features lie on a low-dimensional manifold, embedded in a high-dimensional feature space. Since the acoustic features are only dependent on the coordinates of the sound source, it is expected that the manifold would represent the spatial relations between the nearby acoustic features. We consider that that the affinity matrix of the low-dimensional embeddings of the acoustic manifold represent the *graph* of the acoustic features.

While the obtained embeddings of the acoustic manifold might represent the relative spatial relations between the acoustic features, it is not tied to physical properties and it also might be very non-linear. The translation of embedded space to physical space must be done in a separate step using a nonlinear regression method.

When ANN is utilized to obtain the sound source position via regression using acoustic features, it might be expected that the predictions of the ANN would also exhibit spatial smoothness. Feature manifold could be used during the training of the ANN to ensure that the ANN learns the relation between the acoustic features and the source positions while also retaining the spatial smoothness. This spatial smoothness as well as the awareness of the relative spatial positions of the acoustic features is especially important when a semi-supervised learning strategy is involved. Using a training dataset that contains both labeled and unlabeled samples, knowledge of the relative distances between unlabeled and labeled samples might help to train the regressor to predict source locations for the unlabeled samples based on their manifold distance to the labeled features.

Our method is comprised of two stages.

1. The low-dimensional embedding of an acoustic feature manifold is obtained from a combined dataset of labeled and unlabeled samples. This manifold represents relative distances between acoustic feature in a low-dimensional embedded space.
2. A neural network is trained on the combined dataset, using a loss function that consist of a supervised loss (calculated only for labeled samples) and a graph loss (calculated for all samples,

considering k nearest neighbors). Supervised loss ensures that the regressor is able to learn accurate relations between the acoustic features and the source positions. Graph loss ensures that the source position predictions remain spatially smooth.

Considering that the combined dataset consist of relatively low number of labeled samples and vast amounts of unlabeled samples, the supervised loss acts as a mean to "straighten" the manifold while the graph loss is used to infer the labels for the unlabeled samples based on their distance in feature (or embedded) space to the labeled samples.

2.1. Acoustic Features

We have considered several types of acoustic features, that are discussed further.

2.1.1. Time Difference of Arrival

Time Difference of Arrival (TDoA) is a trivial acoustic feature, that can be estimated using GCC-PHAT. Knowing the TDoA for several non-colinear (or non-parallel?) microphone pairs, it is possible to estimate the position of the sound source using triangulation (trilateration).

While this would be a simple and straightforward method, the accurate TDoA estimation becomes very tricky in reverberant or noisy environments. Moreover, the TDoA contains only very little information about the distance between the sound source and the microphone pair (just one value per pair). For a microphone array with 4 elements, that's only 6 values. TDoA does not explicitly contain any information about the structure of reflections withing the enclosure, nor the geometry or acoustic properties of the enclosure; it only depends on the relative source position with respect to the microphone array(s).

2.1.2. Room Impulse Response and Room Transfer Function

It is assumed that high-dimensional acoustic features, such as room impulse response (RIR) or room transfer function (RTF) contain a unique fingerprint of sound source and microphone positions within an enclosure. This is because the structure of room reflections is unique for every source position and every microphone position (theoretically, there might be some cases when same RIR is obtained for more than one combination of microphone and sound source positions, but this is probably possible in ideal room, which exhibit point symmetry around the center of the room; in real rooms this is impossible; also the microphones must be also placed symmetrically in the enclosure for this effect to occur).

While the RIRs and RTFs contain enough information to uniquely determine the position of the source within an enclosure, in practice it is impossible to obtain RIR without knowing the positions of the sound source and the microphone within the room beforehand. RTFs, on the other hand, is a viable option.

2.1.3. Steered Response Power

Steered Response Power (SRP) and SRP with Phase Transform (SRP-PHAT) vectors can be considered the middle ground between the trivial acoustic features like TDoA and ideal features, like RIR or RTF. SRP-PHAT features are obtainable in real world, are relatively high-dimensional and contain information about sound reflections within the room.

2.1.4. Properties of acoustic features

The most important property of all acoustic features in this investigation is the spatial smoothness of feature space. In other words, acoustic features are similar to each other for sound source positions that are close together.

In our investigation, we use the SRP-PHAT spatial spectra as acoustic features [14**?**].

2.2. Acoustic feature acquisition

Acoustic features were obtained within an acoustic enclosure using a single sound source, z coordinate was fixed at height $\mathrm{m_s}$. $\mathrm{N_M}$ circular microphone arrays were used for acoustic signal acquisition, each with $\mathrm{N_m}$ microphone elements and radius $\mathrm{m_M}$. Planes of the microphone arrays were parallel to the ground. Both arrays were held at a fixed height $\mathrm{m_M}$. Signals of the microphones are recorded at a fixed sampling frequency f_s and a fixed resolution Q.

2.2.1. Unlabeled dataset

The unlabeled dataset may be obtained from an array audio recording where the sound source is slowly moving inside the acoustic enclosure. The maximal speed of the sound source movement $v_{\mathrm{s\,max}}$ should be lower than the maximum expected localization error distance e_{max} per frame duration T_{fr}:

$$v_{\mathrm{s\,max}} = \frac{e_{\mathrm{max}}}{T_{\mathrm{fr}}} \tag{1}$$

2.2.2. Labeled dataset

The labeled dataset may be obtained from an array audio recordings where the sound source is stationed at a known position $\mathbf{s}_{(x,y,z)}$, described by coordinates (x, y, z) in Cartesian coordinate system within the acoustic enclosure and is producing signal (speech or noise) for a period of $T_\mathbf{s}$ seconds. A collection of $n \in \mathrm{N}_\mathbf{s}$ recordings at fixed source positions may be obtained.

2.3. Audio signal framing

Audio signals obtained from the microphone arrays are split into frames of duration T_{fr} seconds to obtain N_{fr} frames.

2.4. SRP-PHAT feature acquisition

For each audio frame $j \in N_{\mathrm{fr}}$ and for each microphone array $i \in \mathrm{N_M}$, a set of time-frequency representations of the microphone signals is calculated with $\mathrm{N_{FFT}}$ FFT points, without frame overlap and no windowing function.

A SRP-PHAT spatial spectrum $\mathbf{X}_{\text{SRP-PHAT}(j,i)}$ is obtained for each frame and for each array. $\mathbf{X}_{\text{SRP-PHAT}(j,i)}$ is a vector with $\mathrm{N}_\mathbf{X}$ elements, representing the WHAT at a particular DoA and covering an azimuth angle $\theta_\mathrm{M} \in [0°; 360°]$. SRP-PHAT spectra of all arrays are then concatenated per frame to obtain the acoustic feature $\mathbf{X}_j$ of $\mathrm{N_M} \cdot \mathrm{N}_\mathbf{X}$ elements.

If the audio recording has an associated location label (known coordinates), a frame is assigned the position label $\mathbf{s}_{(x,y,z)}$.

2.5. Acoustic features selection (thresholding)

It is considered that the sound source might not be active a all times, and that the signal is non-stationary (in case of speech signal, it might be considered quasi-stationary for frames that contain only one phoneme or a part of a phoneme). Thus, in case of an audio frame where the source is not active, the DoA of a sound source can not be determined, and the acoustic feature is considered to contain only noise. Such frames are to be discarded. For the selection of the audio frames in which the acoustic feature is usable, a thresholding algorithm was used. A metric $p_{i,j} = f(\mathbf{X}_{\text{SRP-PHAT}(j,i)})$ is calculated for and compared to the threshold level $L_{\mathrm{thr.}}$ which is the scaled mean of the metric of all obtained frames $L_{\mathrm{thr.}} = \mathrm{k}_p \frac{1}{\mathrm{N_{fr}}} \sum_{j \in N_{\mathrm{fr}}} p_j$, where k_p is the scaling coefficient used to control the threshold value. Metric $p_{i,j}$ is calculated per array to address a fact that the arrays might be not identical in terms of audio signal gain, the signal-to-noise ratio and frequency response. The metrics used to evaluate the fitness of the acoustic feature of the particular audio frame are:

1. Root-mean-square value of the SRP-PHAT spectrum, $p_{i,j}^{\mathrm{RMS},}(\mathbf{X}_{\text{SRP-PHAT}(j,i)}) = \sqrt{\langle \mathbf{X}^2 \rangle}$.

2. Crest factor of the SRP-PHAT spectrum, $p^{\text{CF}}_{i,j}(\mathbf{X}_{\text{SRP-PHAT}(j,i)}) = \frac{|\max(\mathbf{X}_{\text{SRP-PHAT}(j,i)})|}{p^{\text{RMS,}}_{i,j}(\mathbf{X}_{\text{SRP-PHAT}(j,i)})}$.

After determining $p_{i,j}$ of the $\mathbf{X}_{\text{SRP-PHAT}(j,i)}$ per array per frame, feature vectors $\mathbf{X}_j$ are selected of those frames j for which $p_{i,j} > L_{\text{thr.}}$ for all microphone arrays $i \in \text{N}_\text{M}$.

2.5.1. Training/testing dataset split

The labeled dataset is split into training and testing subsets by randomly selecting samples from $\text{N}_{\text{ts.}}$ source positions for training and the rest of the source positions $\text{N}_{\text{tr.}} = \text{N}_{\mathbf{s}} - \text{N}_{\text{ts.}}$ for testing from the entire set of labeled source positions. Following operations are performed separately for training ant testing labeled datasets.

2.6. Acoustic manifold embedding learning

Manifold embedding can be learned using a Nonlinear Dimensionality Reduction (NLDR) algorithm, such as isometric mapping (ISOMAP), t-distributed stochastic neighbor embedding (t-SNE) or localli linear embedding (LLE), among others. We have employed ISOMAP NLRD algorithm to obtain the high-dimensional feature embeddings in low-dimensional space, that is, learn the acoustic feature manifold.

2.6.1. ISOMAP embedding

One of the earliest approaches to manifold learning is the ISOMAP algorithm. Isomap can be viewed as an extension of Multi-dimensional Scaling (MDS) or Kernel Principal Component Analysis (PCA). Isomap seeks a lower-dimensional embedding which maintains geodesic distances between all points [15].

SRP-PHAT features from both labeled and unlabeled training datasets are embedded into $\text{D}_{\text{emb.}}$-dimensional embedded space using ISOMAP, with $\text{k}_{\text{emb.}}$ nearest neighbors considered. For each $\mathbf{X}_j$ feature, an embedding $\mathbf{Z}_j = [z_{d_1}, z_{d_2}, \ldots, z_{d_{\text{D}_{\text{emb.}}}}]$. This way, a low-dimensional representations of the high-dimensional acoustic features is obtained. Moreover, the learned manifold corresponds to the spatial structure of the acoustic feature space. Thus, the relative distances in the embedded space of unlabeled features to labeled features is known.

2.7. Graph dataset

2.7.1. Dataset preprocessing

The combined dataset for training the neural network is comprised of two datasets: N_u acoustic feature samples without source position labels (the unlabeled dataset) and N_l acoustic feature samples with source position labels (the labeled training dataset). Each sample feature $\mathbf{X}^{\text{u,l}}_j$ in the combined dataset also has a corresponding ISOMAP embedding $\mathbf{Z}^{\text{u,l}}_j$. In order to train the GRNN with graph regularization, the dataset must be preprocessed: for each sample, regardless of whether it is a labeled or an unlabeled sample, alongside the main feature, neighbor features $\mathbf{X}^n_j$ and their weights a^n where $n \in \mathbf{n}$ must be introduced. $\mathbf{n}$ denotes the neighborhood of the sample feature in the embedded space. This is done by first determining the k_G nearest neighbors of a particular sample in the embedded feature space and then appending those features as well as their weight coefficients to the training sample.

2.7.2. Affinity matrix calculation

In the embedded space, Euclidean distances are calculated between every feature. The distances between each data sample constitute the distance matrix $\mathbf{D}$, which is in turn used to calculate the affinity matrix. Affinity matrix $\mathbf{A}$ is calculated by subtracting $\mathbf{D}$ from the identity matrix: $\mathbf{A} = \mathbf{1} - \mathbf{D}$.

The distance matrix contains the Euclidean distances between each sample in the low dimensional embedded space:

$$\mathbf{D} = (d_{ij}); \tag{2}$$

$$d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|_2^2 \tag{3}$$

here $\mathbf{p}_i = (\alpha_i, \beta_i)$ is the point coordinate vector in the embedded space (in case of $N_{\text{ISO}} = 2$), α and β are the Cartesian coordinates in the embedded space.

Neighbor weights are inversely proportional to the Euclidean distances between the main feature and the neighbor features in the low-dimensional embedded space.

2.7.3. Neighbor samples and neighbor sample weights

For the training of the GRNN, each training sample must contain the main SRP-PHAT feature and k_G neighbor SRP-PHAT features (used for calculating the graph loss). Additionally, each neighbor feature is associated with its weight a, which is the corresponding element in the affinity matrix. To obtain the k_G neighbors of each sample, each row of the affinity matrix is thresholded so that only the k_G highest-valued elements remain their value, while other row elements are set to zero. The dataset is then expanded so that each sample now has associated neighbor SRP-PHAT features (indices of which are the non-zero elements in the rows of the affinity matrix).

2.7.4. Labeled/unlabeled sample marking

For the training dataset, a flag m denoting wether the sample is labeled or unlabeled is introduced. This flag holds value of either "True" of "False" (1 or 0). Content of this field is interpreted by the GRNN during the calculation of the loss function. Effectively, the supervised loss component is multiplied by the flag. In case of an unlabeled sample, the supervised loss is ignored, and only the graph loss is considered. In real-world scenario, GRNN expects all fields, including the target feature (the label, the coordinates of the source) to be passed during training. In case of the unlabeled sample (whether during the training phase or during the prediction phase), the supervised loss is not calculated, the label is ignored, and thus it can be set to random values or to zero.

2.7.5. Labeled samples repetition

We wish to train the GRNN using as few as possible labeled samples. It was found that the network is trained more effectively when the labeled samples are introduced more times (more often) than the unlabeled samples. It might be called "dataset balancing" []. Labeled samples (those with $m = 1$) are repeated N_R times ($N_R \in \{1, \ldots, 199\}$) and appended to the training data subset.

2.8. Graph-Regularized Neural Network

In our proposition, a neural network that is trained considering not only the labeled samples, but also neighboring labeled and unlabeled samples.

2.8.1. Neural network

Any neural network can be converted to graph-regularized neural network (GRNN) by introducing additional inputs for neighboring features as well as modifying the loss function to accommodate the graph loss.

A general architecture (one of possibilities) of a GRNNmodel is provided in Figure 1. In this figure, dotted lines encompasses the input vectors. Dahsed lines inside the GRNN block denote prediction (a forward pass). The loss function is given by $L = m(\hat{y}_0 - y) + \sum_{i \in k_g} a_i(\hat{y}_0 - \hat{y}_1)$. The loss function is discussed further in more detail.

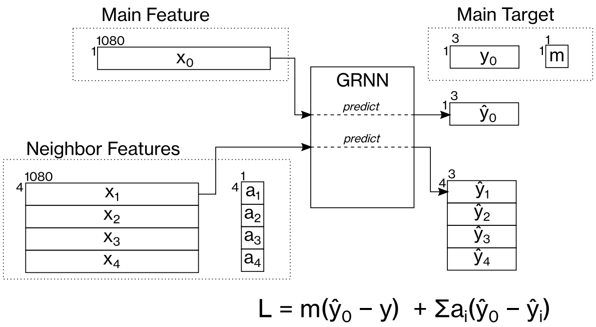


Figure 1. General architecture of a graph regularized neural network (considering 4 neighbor features). x_0 is the main input feature, $x_{1..4}$ are neighbor input features, $a_{1..4}$ are corresponding neighbor input feature weights, y_0 is the target feature, m is the labeled/unlabeled flag, $\hat{y}_0$ is the label prediction for main input feature, $\hat{y}_{1..4}$ are the label predictions for the neighbor input features

Table 1. Table of neural network architectures used in for the experimentation

	Architecture											
	1		2		3		4		5		6	
Layer	Activation	Size	Activation	Size	Activation	Size	Activation	Size	Activation	Size	Activation	Size
input	linear	720	linear	720	linear	720	linear	720	linear	720	linear	720
hidden 1	linear	14	linear	4	linear	10	linear	10	linear	10	Leaky ReLU	10
2	sigmoid	2	sigmoid	32	relu	31	ReLU	15	ReLU	15	Leaky ReLU	15
3	tanh	24	tanh	23			ReLU	15	ReLU	15	Leaky ReLU	15
4	sigmoid	33	sigmoid	54					ReLU	15	Leaky ReLU	15
5	linear	50	linear	37					ReLU	15	Leaky ReLU	15
output	linear	2	linear	2	linear	2	linear	2	linear	2	linear	2

2.8.2. Architecture

Apart from the introduction of additional inputs (neighbor features, weights and flags), the actual neural network is just a multilayer perceptron. During prediction phase, only the main input contributes to the prediction.

In this experiment, a several multilayer perceptron architectures were used. The summary of the architectures are presented in Table1.

This architecture was the found during previously performed hyperparameter optimization.

2.8.3. Loss function

Nearby source positions produce similar acoustic features. Therefore, the predicted source positions for the nearby acoustic features should also be similar If they are similar, the graph loss is small. If they are not similar, we need to penalize the predictor with a large graph loss

The loss function used for the GRNN training is comprised of two parts: the supervised loss (the difference between the ground truth label and the predicted label) and the graph loss (the difference bewteen the main input feature label prediction and the weighted sum of neighbor input features label predictions). It can be expressed as

$$L = \mu m \sum_{i \in N_b} (\hat{y}_i - y_i)^2 + (1 - \mu m) \sum_{i \in N_b} \sum_{j \in k_g} a_{ij} (\hat{y}_i - \hat{y}_j)^2 \tag{4}$$

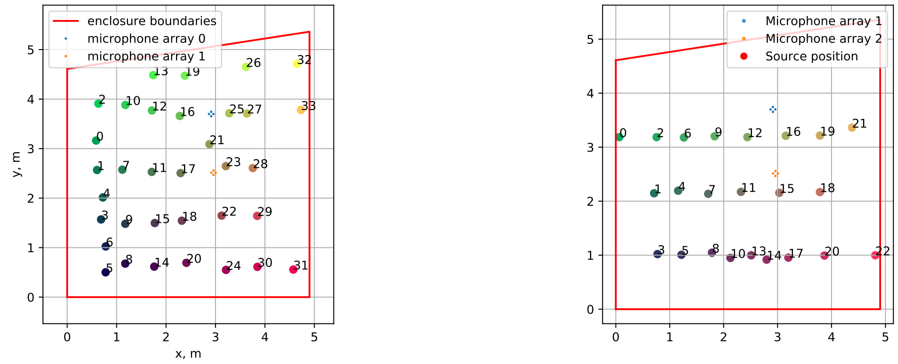
here N_b – number of samples in one training batch, k_g – size of the neighborhood, a_{ij} is the neighbor weight, equal to the corresponding element in the affinity matrix.

2.9. Experimental setup

We have evaluated the performance of our proposed method using a real-world microphone array audio dataset with speech signal as the sound source, which was recorded in a particular acoustic enclosure.

2.9.1. Enclosure

All audio data used for the experimentation was collected in an irregular shaped room with the side dimensions of 4.902 m in x axis, 5.361 m in y axis and 3.75 m in z axis. The geometry of the enclosure is presented in Figure 2.



(a) 1st dataset

(b) 2nd dataset

Figure 2. The geometry of the acoustic enclosure used for real-world dataset collection and labeled sound source positions (1st and 2nd datasets); height of the enclosure was 3.75 m

2.9.2. Sound source

Source signal

The signal of the sound source was a 1 min excerpt from the AMI Corpus [16], the mix of close-talking microphone signals, containing male and female speech samples.

Sound source signal was reproduced using a compact battery powered loudspeaker that was mounted on an adjustable height stand.

Two sets of labeled and unlabeled array audio recordings were recorded. [1]

Source positions

The first set of 34 array audio recordings was obtained with the sound source held stationary for 1 min in one of 34 positions within the enclosure. The second set of 23 array audio recordings was obtained in the same fashion as the first set.

[1] The sets were recorded on different days and a slight shift in microphone array positions might have had occurred. The first set have more labeled recordings (34), but the unlabeled recording is only around 8.5 minutes long, while the second recording has less labeled recordings (23), and source positions does not cover the entire area of the enclosure, but the unlabeled recording is around 40 minutes long.

The coordinates of the sound source were measured using a handheld laser distance measurement tool with accuracy of 1 mm. The locations of the known source positions are presented in Figure 2 as colored circles. The values of the red, green and blue components of each circle color are proportional to the x_s, y_s and z_s coordinates of the sound source position. The vertical coordinate, $z_\mathrm{s} = 1.9\,\mathrm{m}$, was constant for all source positions.

2.9.3. Unlabeled audio dataset

The unlabeled audio dataset was collected using the same microphone array setup and the same audio source and signal. The loudspeaker was moved manually at a reasonably constant speed of approximately $0.1\,\mathrm{m\,s^{-1}}$, scanning the entire floor area of the enclosure . The vertical coordinate of the sound source, $z = 1.9\,\mathrm{m}$, was held constant during the entire collection of the unlabeled dataset. The total duration of the recording in the first dataset was 511 s . In the second dataset, the duration of the unlabeled audio recording was 2420 s.

2.9.4. Microphone arrays

In our experimentation we have used $\mathrm{N_M} = 2$ radial microphone arrays with radius $\mathrm{m_M} = 0.045\,\mathrm{m}$, each consisting of $\mathrm{N_m} = 4$ microphone elements spaced at equal angles $\phi_\mathrm{m} = 90°$. The centers of the microphone arrays were placed at coordinates $\mathbf{M}_{C,1}^{(x,y,z)} = [2.913, 3.699, 1.313]$ m and $\mathbf{M}_{C,2}^{(x,y,z)} = [2.960, 2.512, 1.309]$ m.

First element of each array was oriented towards the positive x axis with respect to the microphone array center (the rotation of the elements of the microphone array relative to the x axis was 0°).

2.9.5. Acoustic feature acquisition

Audio signals obtained from the microphone arrays were split into frames with the duration of 0.05 s. This frame duration was chosen so that every audio frame would contain only one speech phoneme.

For each audio frame and for each array, a SRP-PHAT spatial spectrum with 360 elements, covering DoA of 360° (1 degree resolution) was calculated using `pyroomaoustics` *Python* implementation [14] of a method presented in [**?**]. During SRP-PHAT spectra calculation, $\mathrm{N_{FFT}} = 512$ FFT points were used, with 50 % overlap, for the STFT snapshot calculation. For each frame, SRP-PHAT spectra were concatenated to produce a single 720-dimensional acoustic feature.

2.9.6. Acoustic feature selection

Acoustic features for further processing were selected based on the RMS or CF metrics of the feature vector, thresholded by the scaled mean of the corresponding metrics of the entire dataset. Thresholding scaling coefficient k_p was selected from the range $\mathrm{k}_p \in [1.0, 1.1, \ldots, 1.5]$ for both RMS and CF based methods.

2.9.7. Acoustic feature manifold learning

Acoustic feature manifold embeddings were found using ISOMAP NLDR algorithm, implemented in [15]. We have chosen to embed the manifold into 2-dimensional embedded space, since in our experimentation, the only the x and y coordinate of the sound source was changing, thus, the acoustic feature are expected to rely only on two variables. The number of nearest neighbors considered for each sample was selected in the range $\mathrm{k_{emb.}} = 2^n; n \in [0, 1, \ldots, 6]$. The same settings were used for both unlabeled and labeled audio dataset. The acoustic manifold was first learned on unlabeled dataset and then the labeled dataset was transformed into low-dimensional embedded space using the already learned manifold.

2.9.8. Dataset construction

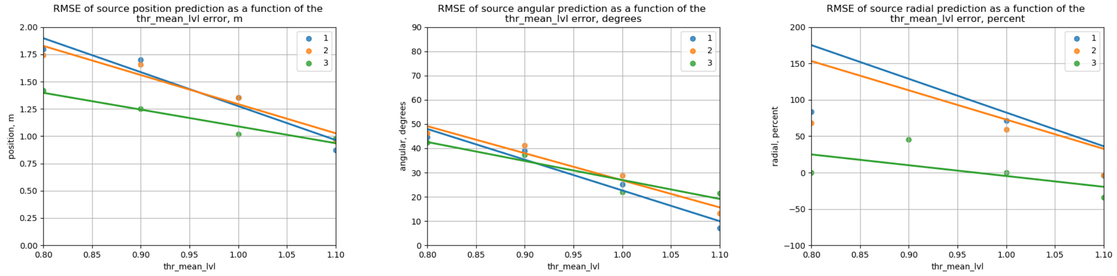
The training and testing datasets were constructed as described in Section 2.7. Number of nearest graph neighbors considered for each sample k_G was selected from a set $k_G \in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20]$.

2.9.9. GRNN training

We have trained the GRNN for $N_{ep} = 15$ epochs with each set of parameters. The sample batch size was $N_{bs} = 2^n; n \in [6, 7, \ldots, 13]$.

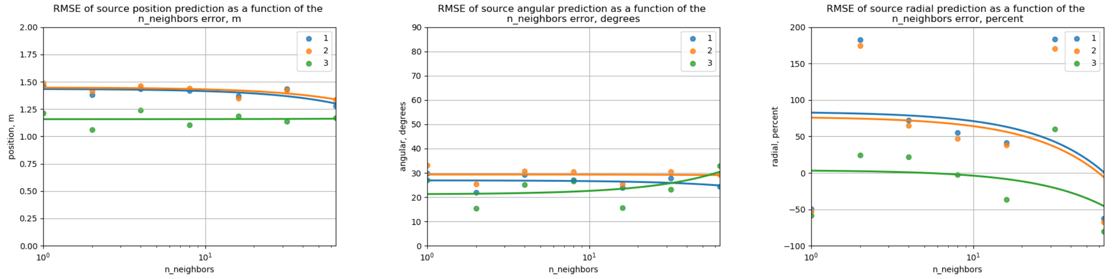
3. Results

We have investigated the influence of each of the parameters on the sound source localization error.



(a) position error (b) angular error (c) radial error

Figure 3. Dependency between the source position prediction, source DoA estimation and radial estimation RMSE and the acoustic feature thresholding level; linear regression model showed in solid line



(a) position error (b) angular error (c) radial error

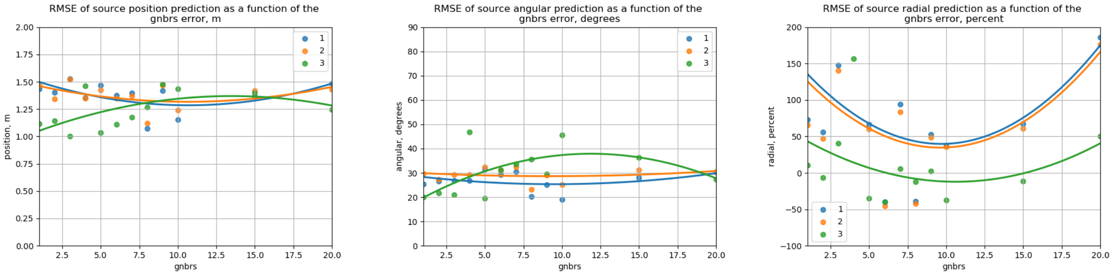
Figure 4. Dependency between the source position prediction, source DoA estimation and radial estimation RMSE and the number of nearest neighbors considered for ISOMAP embedding; linear regression model showed in solid line

The results of sound source localization using the geometric source localization approach are presented in Figure9(a) and Figure9(b).

Using such methods, the positions of the sound source were estimated using real-world speech audio data. The results of the localization are presented in Figure 10.

After assembling the training dataset with 2 nearest neighbors, and training the neural network for 50 epochs, the predictions of the source position were more accurate than using the baseline methods.

The summary of source location prediction errors for different localization methods are presented in Table 2. The results of the source location prediction using a GRNN are presented in Figure 12.

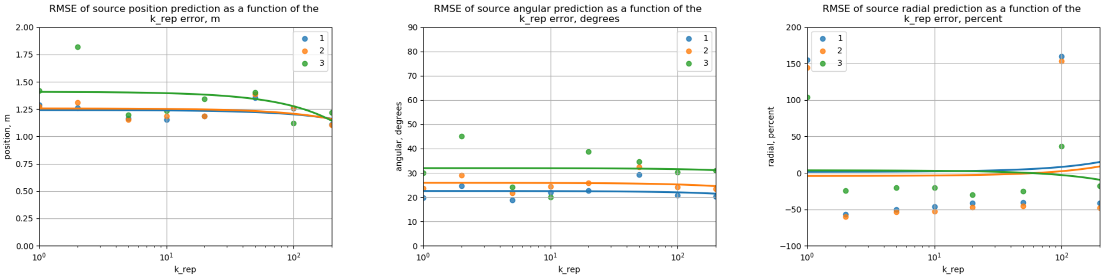


(a) position error

(b) angular error

(c) radial error

Figure 5. Dependency between the source position prediction, source DoA estimation and radial estimation RMSE and the number of nearest graph neighbors considered during the training dataset construction; 2nd order regression model showed in solid line

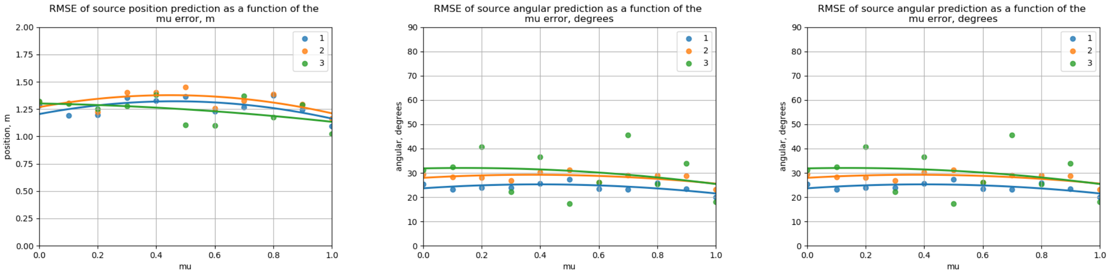


(a) position error

(b) angular error

(c) radial error

Figure 6. Dependency between the source position prediction, source DoA estimation and radial estimation RMSE and the labeled samples repetition rate during GRNN training; linear regression model showed in solid line



(a) position error

(b) angular error

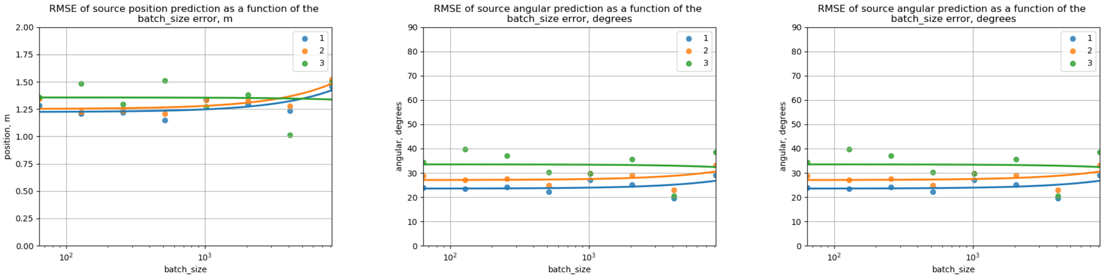
(c) radial error

Figure 7. Dependency between the source position prediction, source DoA estimation and radial estimation RMSE and the ratio between the supervised and unsupervised loses considered during the GRNN training; linear regression model showed in solid line

The distributions of the prediction errors for different source localization methods are presented in Figure 13.

4. Discussion

As can be seen from the experimental results, our method outperform the baseline methods for almost all parameter configurations. Our method produces position estimation error that is 24.2 % lower than using a geometrical source localization method, and 19.1 % lower than using the intensity map method at low feature fitness threshold levels. When the threshold is high, the performance of all methods becomes comparable. It needs to be addressed that it is impractical to use high threshold

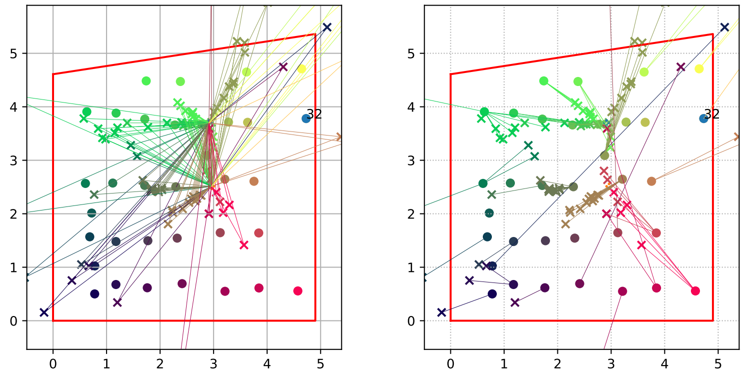


(a) position error

(b) angular error

(c) radial error

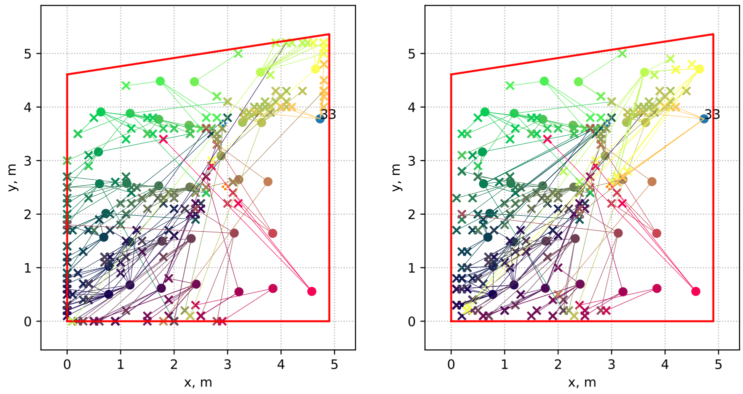
Figure 8. Dependency between the source position prediction, source DoA estimation and radial estimation RMSE and the GRNN training sample batch size; linear regression model showed in solid line



(a) lines represent the DoA radii of each array

(b) lines represent localization error

Figure 9. Results of real-world speech source localization using geometric localization algorithm



(a) source positions is the argument of the intensity map

(b) source positions is the location of the most prominent local peak of the intensity map

Figure 10. Predicted source positions using the intensity map approach

values because it is possible that the sound source would not be localized at all (all its features are below the threshold).

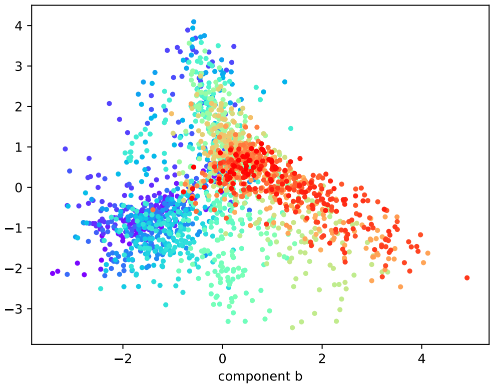


Figure 11. 2-dimensional ISOMAP embeddings of the SRP-PHAT features of the real-world unlabeled speech signal; point hue represents its

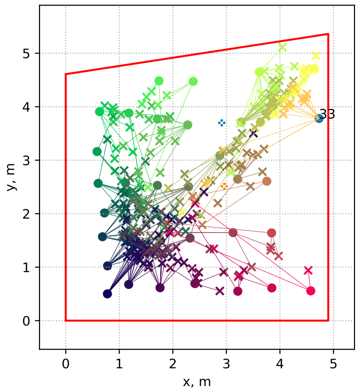


Figure 12. Prediction of sound source position using a GRNN trained for 50 epochs.

347 As seen from the Figure 4, parameter $N_{k_{emb.}}$ does not affect the performance of the baseline 348 algorithms. This is the expected case, since this parameter is not involved in obtaining the source 349 position estimations using the baseline methods. As for the GRNN approahc, we observe that $N_{k_{emb.}}$ 350 has little impact for the position estimation, and has only a marginal impact on the angular and radial 351 source position estimation errors. Nevertheless, in our method performed on average 20.3 % better 352 than the baseline methods when considering the source position estimation RMSE.

353 Considering the number of the nearest neighbors of the samples in the embedded space when 354 constructing the training graph dataset, Figure 5 shows that the smallest source position estimation 355 error is obtained when only a small number of graph nearest neighbors are selected. This might be due 356 to the nonlinearity of the embedded space. The larger the number of considered neighbors, the further 357 the samples are in the embedded space, and the larger the error. As can be seen from the radial error

Table 2. Summary of source location prediction errors for different localization methods

Method	MSE, m^2	MAE, m	RMSE, m
Geometric	8.37	1.92	2.89
Intensity Map, argmax	2.50	1.26	1.58
Intensity, peak location	0.91	1.06	1.37
GRNN	**0.86**	**0.84**	**0.93**

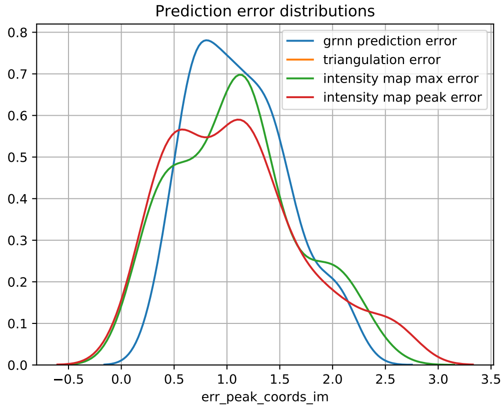


Figure 13. The distributions of the prediction errors for different source localization methods

plot, the number of the nearest graph neighbors considered is influencing the radial source position prediction error the most. The radial error is smallest when our algorithm considers 10 neighbors. Overall smallest source position error is achieved with 3 nearest graph neighbors.

As can be seen from the Figure 6, the labeled sample repetition rate is not influencing the source position estimation error much. The angular error is reduced at high repetition rates, but the radial error is increased. This might be due to the condition where the labeled sample positions are condensed around the center of the enclosure, and the supervised loss function forces the network to predict source positions towards the center. This produces large estimation errors for the sound sources that are further away from the center of the enclosure.

The most suitable ratio of supervised to unsupervised loss, as can be seen from the Figure 7, is $\mu = 0.5$.

The training batch size does not significantly affect the source position prediction error (Figure 8).

5. Conclusions

A novel data-driven sound source localization method was proposed. Compared to baseline sound source localization methods, the geometrical and the intensity map-based method, our method showed an improvement of 4.2 % of the source position estimation RMSE (0.86 m compared to 0.91 m), compared to the best performing baseline method (intensity map-based with local peak location). The largest angular error improvement was 20 % compared to the same baseline algorithm. The radial error improvement was 40 %. For all experiments, the reverberation time of the enclosure was $T_{60} = 0.37$ s. To conclude, our method showed to be a viable option when a data-driven approach is needed due to adversity of acoustic conditions within an enclosure.

Abbreviations

The following abbreviations are used in this manuscript:

GRNN	graph regulzarized neural network
SRP	steered response power
PHAT	phase transform
DoA	direction of arrival
RMS	root mean square
MSE	mean squared error
MAE	mean average error
RMSE	root mean squared error
ISOMAP	isometric mapping
NLDR	non-linear dimensionality reduction

References

1. Marti, A.; Cobos, M.; Aguilera, E.; Lopez, J.J. Speaker Localization and Detection in Videoconferencing Environments Using a Modified SRP-PHAT Algorithm. p. 8.
2. Lopatka, K.; Kotus, J.; Czyzewski, A. Detection, Classification and Localization of Acoustic Events in the Presence of Background Noise for Acoustic Surveillance of Hazardous Situations. *75*, 10407–10439. doi:10.1007/s11042-015-3105-4.
3. Valin, J.M.; Michaud, F.; Hadjou, B.; Rouat, J. Localization of Simultaneous Moving Sound Sources for Mobile Robot Using a Frequency-Domain Steered Beamformer Approach. pp. 1033–1038 Vol.1, [1602.08629]. doi:10.1109/ROBOT.2004.1307286.
4. Brutti, A.; Omologo, M.; Svaizer, P. Comparison Between Different Sound Source Localization Techniques Based on a Real Data Collection. 2008 Hands-Free Speech Communication and Microphone Arrays. IEEE, pp. 69–72. doi:10.1109/HSCMA.2008.4538690.
5. Weng, J.; Guentchev, K.Y. Three-Dimensional Sound Localization from a Compact Non-Coplanar Array of Microphones Using Tree-Based Learning. *110*, 310–323, [11508957]. doi:10.1121/1.1377290.
6. Awad-Alla, M.A.; Hamdy, A.; Tolbah, F.A.; Shahin, M.A.; Abdelaziz, M.A. A Two-Stage Approach for Passive Sound Source Localization Based on the SRP-PHAT Algorithm **2020/ed**. *9*. doi:10.1017/ATSIP.2020.6.
7. Valin, J.M.; Michaud, F.; Rouat, J. Robust 3D Localization and Tracking of Sound Sources Using Beamforming and Particle Filtering. 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Vol. 4, pp. IV–IV. doi:10.1109/ICASSP.2006.1661100.
8. Lombard, A.; Rosenkranz, T.; Buchner, H.; Kellermann, W. Multidimensional Localization of Multiple Sound Sources Using Averaged Directivity Patterns of Blind Source Separation Systems. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 233–236. doi:10.1109/ICASSP.2009.4959563.
9. Silverman, H.F.; Yu, Y.; Sachar, J.M.; Patterson, W.R. Performance of Real-Time Source-Location Estimators for a Large-Aperture Microphone Array. *13*, 593–606. doi:10.1109/TSA.2005.848875.
10. Laufer-Goldshtein, B.; Talmon, R.; Gannot, S. Semi-Supervised Sound Source Localization Based on Manifold Regularization. *24*, 1393–1407. doi:10.1109/TASLP.2016.2555085.
11. He, W.; Motlicek, P.; Odobez, J. Deep Neural Networks for Multiple Speaker Detection and Localization. 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 74–79. doi:10.1109/ICRA.2018.8461267.
12. He, W.; Motlicek, P.; Odobez, J.M. Adaptation of Multiple Sound Source Localization Neural Networks with Weak Supervision and Domain-Adversarial Training. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 770–774. doi:10.1109/ICASSP.2019.8682655.
13. Adavanne, S.; Politis, A.; Virtanen, T. Direction of Arrival Estimation for Multiple Sound Sources Using Convolutional Recurrent Neural Network. [arXiv:cs, eess/1710.10059].
14. Scheibler, R.; Bezzam, E.; Dokmanić, I. Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 351–355. doi:10.1109/ICASSP.2018.8461310.
15. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-Learn: Machine Learning in Python. *12*, 2825–2830.
16. Carletta, J.; .; others. The AMI Meeting Corpus: A Pre-Announcement. Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction. Springer-Verlag, MLMI'05, pp. 28–39. doi:10.1007/11677482_3.