FACULTY OF ENGINEERING AND ARCHITECTURE

"I can 't understand you, there is someone speaking over you!" Sound separation using distributed microphone arrays

Martijn Meeldijk
Student number: 02111587

Supervisor: Prof. dr. ir. Nilesh Madhu
Counsellors: Stijn Kindt, Alexander Bohlender

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Information Engineering Technology

Academic year 2022-2023

GHENT UNIVERSITY

Acknowledgements

Vul aan...

Toelichting in verband met het masterproefwerk

Deze masterproef vormt een onderdeel van een examen. Eventuele opmerkingen die door de beoordelingscommissie tijdens de mondelinge uiteenzetting van de masterproef werden geformuleerd, werden niet verwerkt in deze tekst.

Melding van vertrouwelijkheid (enkel indien van toepassing)

Bekijk hiervoor de informatie op de facultaire website - **Nota in verband met de vorm van de masterproef (alle opleidingen)**

Abstract

Meer informatie op https://masterproef.tiwi.ugent.be/verplichte-taken/ - Korte abstract (Nederlands en/of Engels)

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

List of Acronyms

ASN Acoustic Sensor Network.

FFT Fast Fourier Transform.

NMF Nonnegative Matrix Factorization.

WASN Wireless Acoustic Sensor Network.

List of Code Fragments

1

Introduction

1.1 Problem definition

Many devices such as Amazon's Alexa and Google Home require the processing of human speech to function. This is not always as straightforward as one would expect. Especially considering the input of said devices often contains a great deal of background noise, such as reverberation or other unwanted sound sources. One solution is to make use of microphone arrays in order to help extract the wanted speech component from the signal. A microphone array consists of multiple microphones placed close to each other, typically inside the same device. From the microphone array's signals, time-frequency masks can be created. These are subsequently used in combination with the short-time Fourier transform of the signal to extract the speech component.

To further improve on this solution, it is possible to make use of (ad hoc) distributed microphone arrays. This way, several microphones or microphone arrays are placed at different locations, opening up new possibilities for signal processing such as utilizing the different amplitudes of the signals received in different locations. Signal capture using ad hoc distributed microphones, or acoustic sensor networks (ASNs), is an active and rapidly expanding field of research. With the inclusion of microphones in an increasing variety of smart devices, distributed audio capture is becoming increasingly available - with potential for application in a wide range of fields such as surveillance for assisted living and healthcare, hearing aids, communications. The challenges, however, are also manifold. Compared to traditional, compact microphone arrays, where multiple microphones are placed close to each other with predefined geometries, the relative locations of sensors are not known a priori, and their placement with respect to audio sources of interest can be arbitrary. The processing power and bandwidth available to each node can also be limited - constraining on-edge processing and data communication with a central hub.

1.2 Motivation

1.3 Summary of results

2

Background

2.1 Signal Model

The acoustic environment considered in this thesis consists of N acoustic sources and D microphones which are distributed, typically in an unknown arrangement, within the boundaries of the environment. The signal received by a microphone d may be described in continuous time t as:

$$x_d(t) = \sum_{n=1}^{N} \int_0^{\infty} h_{nd}(\tau) s_n(t-\tau) d\tau \tag{2.1}$$

- $s_n(t)$: the n-th source signal
- $h_{nd}(t)$: the impulse response from source n to microphone d
- $x_d(t)$: the resulting microphone signal

This can displayed by making use of the convolution operator like so:

$$x_d(t) = \sum_{n=1}^{N} h_{nd}(t) * s_n(t) \tag{2.2}$$

The microphone signals are sampled and transformed to the short-time discrete Fourier domain.

$$X_d(k,b) = \mathrm{STFT}[x_d(l)] \tag{2.3}$$

- $x_d(l)$: the sampled signal of microphone d
- l: the time sample index
- k: the frequency bin index
- b: the time frame index

As of now, this model doesn't take into consideration possible interference or noise. This will be discussed in the next paragraph.

2.1.1 Interference

Taking into account an interference signal ... TODO

2.1.2 Modulation Mel-Frequency Cepstral Coefficients

Computing the inverse Fourier transform of the logarithm of the spectrum of a signal results in a cepstrum [1]. The mel-Frequency Cepstrum (MFC) is another way to represent the short-term power spectrum of a sound signal. It is essentially a way of representing a sound signal in a manner that approximates the human auditory system's response [2]. The MFC differs from the cepstrum due to the use of the Mel scale [3], which rescales the normal frequency scale so that pitches are perceived (by humans) to be equal in distance from one another. One way to convert f herts into m mels is:

$$m = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \tag{2.4}$$

//TODO

2.1.3 Speaker Embeddings

2.1.4 Cepstral mean normalization

2.2 Source separation

Source separation aims to obtain a signal containing only the target source, effectively suppressing all other sources except the target. The aforementioned principle is recognized as adaptive 'nulling'.[4] In this part, an emphasis will be put on source separation in ad hoc setups, so any techniques that require positional information on microphone nodes will not be discussed.

2.2.1 Time-Frequency masking

By assuming that sources are approximately disjoint in the short-time-frequency plane, it is reasonable to assume that only one source is dominant at any time-frequency point. This allows for the estimation of a spectral mask $\mathscr{M}_n(k, b)$ which suppresses time-frequency points that do not belong to the target source, effectively suppressing interferers. TODO

$$\mathscr{M}_n(k, b) = \begin{cases} 1 & \text{if source } n \text{ is dominant at } (k, b) \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

2.3 Clustering source dominated microphones

2.3.1 Fuzzy clustering

The first step in the fuzzy clustering procedure consists of extracting a feature set composed of Mel-frequency cepstral coefficients (MFCCs) and their modulation spectra. This is essentially a way of representing a sound signal in a manner that approximates the human auditory system's response to the signal. [2] This feature set is computed across signal segments of 4s, after which the effects of reverberation are reduced via cepstral mean normalization. For each signal segment, a feature vector is generated for each of the D microphones. These vectors, denoted by $\mathbf{v}_d$, consist of A features. [5].

The next step is to estimate clusters of microphones dominated by one of the sources. Several algorithms exist to estimate an optimal fuzzy partition of the set of observations. A well studied and popular method is the Fuzzy c-Means algorithm (FCM) , which evaluates the least-squared error functional:

$$J_m(\Delta, \mathbf{u}) = \sum_{d=1}^{D} \sum_{n=1}^{N} (\mu_{nd})^{\alpha} ||\mathbf{v}_d - \mathbf{u}_n||_{\beta}^2 \tag{2.6}$$

The value $\mu_{n,d}$ denotes the fuzzy membership value. The matrix ∇ contains all the $\mu_{n,d}$ and is iteratively estimated. An example is shown in table 2.1.

	Cluster n		
Mic. d	1	2	3
1	0.1	0.3	**0.6**
2	**0.5**	0.3	0.2
3	**0.7**	0.2	0.1
4	0.25	0.2	**0.55**
5	0.25	**0.6**	0.15
6	0.15	**0.65**	0.2
7	0.05	0.15	**0.8**

Tabel 2.1: Example fuzzy membership values for 7 microphones and 3 clusters

2.3.2 Fuzzy-membership value aware signal enhancement

2.3.2.1 Initial source signal estimation

It is possible to perform beamforming using the microphones of a source cluster if the relative delays between the microphones are known for that source. Since the locations of the microphones are unknown, the following needs to be done for each cluster n. An initial estimate of the source signal ($\hat{s}_{i_n}(l)$) at all microphones $d = i_n$ assigned to that cluster is obtained. Subsequently, a reference microphone is selected for each cluster. By performing correlation analysis of all other

microphone signals with respect to the reference microphone, time-differences-of-arrival (TDOAs) can be estimated. These TDOAs can afterwards be used in the beamforming stage.

By presuming that different sources are approximately disjoint in the time-frequency (T-F) plane, only one source may be assumed to be dominant at any one T-F point (k, b). This allows for the estimation of a spectral mask $\mathscr{M}_n(k, b)$ for each cluster. Applying this mask to the microphone signals of that cluster will provide an estimate of the underlying source signal with a reduced amount of interference by other sources. The simplest way to represent such a mask would be like so [6]:

$$\mathscr{M}_n(k,b) = \begin{cases} 1 & \text{if source } n \text{ is dominant at } (k,b) \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

To estimate this mask, the microphone $d = R_n$ with the highest FMV is selected as reference microphone. By computing the short-time Fourier transform (STFT) representation $X_{R_n}(k, b)$ from the signal of this microphone, the binary mask for cluster n can be acquired by the following:

$$\mathscr{M}_n(k,b) = \begin{cases} 1 & |X_{R_n}(k-b)| > \frac{1}{B}\sum |X_{R_j}(k,b)|, \\ & \quad j = 1,\ldots,N \text{ and } j \neq n \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

The parameter B is used to average the spectral amplitudes across time in order to reduce the effect of jitter induced by the large inter-microphone distances in ad-hoc arrays. The masks $\mathscr{M}_n(k, b)$ are applied to the spectra $X_{i_n}(k, b)$ of all microphones i_n assigned to cluster n.

$$\tilde{X}_{i_n}(k,b) = X_{i_n}(k,b)\mathscr{M}_n(k,b) \tag{2.9}$$

Afterwards, the inverse STFT of $\tilde{X}_{i_n}(k, b)$ is computed in order to reconstruct the time-domain signal $\hat{s}_{i_n}$ representing the initial estimate of the source signal. The estimate of the reference microphone is used for the correlation analysis, which yields TDOAs for all microphones of each cluster with respect to the reference microphone.

2.3.2.2 Clustering-steered beamforming

A generalized DSB can be formed in the time-domain using the relative TDOAs for a cluster.

$$\hat{s}_{n,\,\text{W-DSB}}(l) = \sum_{i_n} w_{n,i_n} x_{i_n}(l + D_{i_n}) \tag{2.10}$$

- l: The discrete time index
- D_{i_n}: The relative TDOA's
- w_{n,i_n}: The weights allocated to each microphone i_n of cluster n

In [6], all weights were set uniformly, but in [7] the weights are set proportional to the FMV. The latter approach yields better results, since the weighting makes it so that signals with a higher FMV are considered of higher importance, implying a higher signal-to-noise ratio (SNR) in said signals. By selecting the first I_n microphones with the highest FMV per cluster, the uncertainty introduced by microphones with a low FMV is reduced.

2.3.2.3 Mask re-estimation

After performing the previous, a post-filtering mask is computed in a similar manner to 2.8.

$$\mathscr{M}_{n,\text{DSB}}(k,b) = \begin{cases} 1 & |\hat{S}_{n,\text{FMVA-DSB}}(k,b)| > \frac{1}{B}\sum |\hat{S}_{j,\text{FMVA-DSB}}(k,b)| \\ & \quad j = 1,\ldots,N \text{ and } j \neq n \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

This mask is applied to $\hat{S}_{n,\text{FMVA-DSB}}(k,b)$, after which the time-domain signal is reconstructed. This results in a final, enhanced estimate of the source signal in each cluster.

2.3.3 Federated Learning

The increasingly declining cost of acoustic sensors and the rapid rise in popularity of wireless networks and mobile devices have aided in providing the technological infrastructure necessary for WASNs. Examples of scenarios where these can be beneficial range from smart-homes and ambient-assisted living to machine diagnosis and surveillance. ASN applications typically have to deal with multiple simultaneously active sound sources, resulting in the fact that an exchange of information about microphone positions or information-rich signal representations is often necessary. Transferring such data over a potentially insecure wireless network carries a considerable amount of risk regarding privacy. Even in the context of a small-scale environment such as a smart-home, the act of eavesdropping by an unauthorized user who gains access to the network can result in significant privacy risks through the interception of sensitive data. Furthermore, as the importance of privacy in our world increases, and privacy regulations such as the European Union General Data Protection Regulation (EU GDPR) arise, the need for a more privacy-aware solution becomes almost undeniable.

Clustered federated learning (CFL) [8] aims to provide such a solution. Instead of using feature representations derived from raw audio data, this methods solely requires ASN nodes to share locally learned neural network parameter updates with a central node. So far, federated learning has only been used in (semi-) supervised learning applications where (weak) classification labels were available, with its intended practical use consisting of massively distributed systems that handle large amounts of data [9]. The task of adapting CFL to an unsupervised scenario and implementing it effectively within the context of ASNs is a complex and challenging endeavor.

2.3.3.1 Federated learning

Federated learning operates by following a three-step iterative procedure over a certain amount of communication rounds τ. The initial stage involves the synchronization of clients with the server, accomplished by downloading the most recent model parameters represented by the column vector θ. Secondly, each client i improves its own model parameters $\boldsymbol{\theta}_i^\tau$ independently with stochastic gradient descent (SGD) on their data $\boldsymbol{D}_i$. Finally, each client uploads their model parameters updates $\boldsymbol{\Delta\theta}_i^\tau$ to the server, where they are aggregated according to

$$\boldsymbol{\theta}^{\tau+1} = \sum_{i=1}^{M} \frac{|\boldsymbol{D}_i|}{|\boldsymbol{D}|} \boldsymbol{\Delta\theta}_i^\tau \tag{2.12}$$

- M: The number of clients
- $|\boldsymbol{D}_i|$ The cardinality of the dataset of the i-th client
- $|\boldsymbol{D}|$: The cardinality of the total dataset

In cases where the clients' data originates from incongruent distributions, it is shown [10, 11] that there exists no single set of parameter updates able to optimally minimize the loss of all clients simultaneously. The suggested approach is to cluster clients following similar distributions, and training separate server models for each cluster. The first step in the procedure involves calculating the cosine similarity $a_{i,j}$ between the nodes' update vectors.

$$a_{i,j} = \frac{\langle \boldsymbol{\Delta\theta}_i, \boldsymbol{\Delta\theta}_j \rangle}{||\boldsymbol{\Delta\theta}_i, \boldsymbol{\Delta\theta}_j||} \tag{2.13}$$

- $\langle \cdot \rangle$ The inner product
- $||\cdot||$ The L_2 norm

Subsequently, these cosine similarities $a_{i,j}$ are collected in the symmetric matrix $\boldsymbol{A}$

Algorithm 1 Unsupervised CFL for the estimation of source-dominated microphone clusters in ASNs

Input: Pre-trained autoencoder h, thresholds ϵ_1, ϵ_2 and ϵ_3,
maximum no. of rounds $\max_\tau$
freeze all parameters of h except $\boldsymbol{\theta}$
while audio buffer != empty **do**
  read audio $\boldsymbol{D}$ of M clients
  initialize cluster list $C = \{\{1, \dots M\}\}$ with a single
  cluster element that contains all M clients
  $C' = \{\}$
  $\boldsymbol{\theta}_c \leftarrow$ random initialization
  for $\tau = 1$ **to** $\max_\tau$ **do**
    for $c \in C$ **do**
      for $i \in c$ **do**
        $\boldsymbol{\Delta\theta}_i^\tau \leftarrow \mathrm{SGD}(h_{\boldsymbol{\theta}_c}(\boldsymbol{D}_i))$
      end for
      $\Delta\overline{\theta}_c = \| \frac{1}{|c|} \sum_{i \in c} \boldsymbol{\Delta\theta}_i \|$
      $\Delta\hat{\theta}_c = \max_{i \in c}(\|\boldsymbol{\Delta\theta}_i\|)$
      if $\Delta\overline{\theta}_c \leq \epsilon_1$ & $\Delta\hat{\theta}_c \geq \epsilon_2$ & $|\nabla\Delta\overline{\theta}_c| \leq \epsilon_3$ **then**
        $a_{i,j} = \frac{\langle \boldsymbol{\Delta\theta}_i, \boldsymbol{\Delta\theta}_j \rangle}{||\boldsymbol{\Delta\theta}_i, \boldsymbol{\Delta\theta}_j||}$
        $c_1, c_2 \leftarrow$ bi-partition$(\boldsymbol{A})$
        $\boldsymbol{\theta}_{c_1}^{\tau+1} = \boldsymbol{\theta}_c^\tau + \sum_{i \in c_1} \frac{|\boldsymbol{D}_i|}{|\boldsymbol{D}_{c_1}|} \boldsymbol{\Delta\theta}_i^\tau$
        $\boldsymbol{\theta}_{c_2}^{\tau+1} = \boldsymbol{\theta}_c^\tau + \sum_{j \in c_2} \frac{|\boldsymbol{D}_j|}{|\boldsymbol{D}_{c_2}|} \boldsymbol{\Delta\theta}_j^\tau$
        $C' = C' + \{c_1, c_2\}$
        $\tau = \max_\tau + 1$
      else
        $\boldsymbol{\theta}_c^{\tau+1} = \boldsymbol{\theta}_c^\tau + \sum_{i \in c} \frac{|\boldsymbol{D}_i|}{|\boldsymbol{D}_c|} \boldsymbol{\Delta\theta}_i^\tau$
        $C' = C' + \{c\}$
      end if
    end for
    $C = C'$
  end for
end while

2.3.4 Coherence-based clustering

A relatively novel and slightly different approach [12] to clustering in ad-hoc microphone arrays proposes a method based on the magnitude-squared-coherence between microphones' observations, which measures their degree of linear dependency by analyzing similar frequency components.

Subsequently, a non-negative matrix (NMF) based approach is utilized, with the goal of obtaining an optimal clustering, whereby nodes are assigned into subnetworks based on their respective microphone observations.

The suggested method offers the capability to dynamically perform clustering while imposing a low computational burden, rendering it highly applicable to various audio signal processing applications. Consequently providing a notable advantage in terms of processing efficiency, making the method an attractive option for real-world scenarios where computational resources may be limited or where rapid processing is required.

2.3.4.1 Signal model

To include interference in the signal model, the observed signal $x_d(t)$ at microphone d can be represented like so:

$$x_d(t) = s_d(t) + v_d(t) \tag{2.14}$$

Where $v_d(t)$ denotes the noise signal plus interference at time instant t. The linear signal model in equation 2.14 can be conveniently restated to denote the collection of a frame of samples into a vector form:

$$\begin{aligned} \mathbf{x}_d(t) &= [x_d(t)x_d(t-1)\cdots x_d(t-T+1)]^T \\ &= \mathbf{s}_d(t) + \mathbf{v}_d(t) \end{aligned} \tag{2.15}$$

- T: frame size
- $\_^T$: matrix transpose
- $\mathbf{x}_d(t)$: observed signal vector
- $\mathbf{s}_d(t)$: clean speech vector
- $\mathbf{v}_d(t)$: noise signal vector

2.3.4.2 Clustering algorithm

Magnitude-squared coherence

By utilizing the magnitude squared coherence, it is possible to conduct an analysis of the linear relationship between two signals $x(t)$ and $y(t)$. First, the Fast Fourier Transform (FFT) of both signals is computed. After which the coherence is measured as a function of the center frequency of the filter. The magnitude-squared coherence can be obtained with the following formula [13]:

$$\Gamma_{xy}(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)} \tag{2.16}$$

- f: The center frequency of the filter
- S_{xx}: The auto spectral density of x
- S_{yy} The auto spectral density of y
- S_{xy} The cross-spectral density

The power spectra $S_{xx}(f)$ and $S_{yy}(f)$ describe the distribution of power into frequency components composing the signals $x(t)$ and $y(t)$. [14] To compute the cross-spectral density, the following equation can be used:

$$S_{xy}(f) = \sum_{k=1-T}^{T-1} R_{xy}(k)e^{-i2\pi fk} \tag{2.17}$$

- $R_{xy}(k)$: The cross-correlation between $x(t)$ and $y(t)$
- T: The frame size

For the special case $x(t) = y(t)$, equation 2.17 reduces to $S_{xx}(f)$, $R_{xy}(k)$ can be estimated with:

$$R_{xy}(k) = \begin{cases} \frac{1}{T}\sum_0^{T-1-k} x(t)y(t+k) & k = 0, \ldots, T-1 \\ R_{xy}(-k) & k = -(T-1), \ldots, -1 \end{cases} \tag{2.18}$$

Now, sufficient information is provided to be able to compute $\Gamma_{xy}(f)$. This calculation yields a value between 0 and 1, with higher values denoting a stronger linear correlation. By calculating

$$C_{xy} = \frac{\sum_{f=0}^{F} \Gamma_{xy}(f)}{F} \in [0, 1] \tag{2.19}$$

all frequency bins are assigned the same weight regardless of their power. By arranging all coherence measures C_{xy} between the audio signals, a non-negative coherence matrix $\boldsymbol{C}$ can be obtained.

//moet M1 zijn TODO

$$\boldsymbol{C} = \begin{bmatrix} 1 & \cdots & \cdots & C_{1M} \\ C_{12} & 1 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ C_{1M} & C_{2M} & \cdots & 1 \end{bmatrix} \in \mathbb{R}_+^{M \times M} \tag{2.20}$$

2.3.4.3 Non-negative matrix factorization

The matrix $\boldsymbol{C}$ contains values that represent the degree of correlation between the signals observed by each combination of microphones, meaning that each row (or column) j of $\boldsymbol{C}$ represents the degree of correlation between the j-th microphone signal and all other signals. As a result, groups of microphones close to a specific source will be highly correlated.

The next step consists of exploiting the inherent clustering property of NMF [15]. $\boldsymbol{C}$ can be considered as a linear subspace of dimension M. By downgrading this subspace into a linear subspace with dimension corresponding to the amount of sources K, a clustering can be achieved. The matrix $\boldsymbol{C}$ is non-negative and can be modelled as:

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{B}^T \odot (\boldsymbol{1} - \boldsymbol{I}) + \boldsymbol{I} \tag{2.21}$$

- $\boldsymbol{B} \in \mathbb{R}^{M \times K}$: The cluster matrix, where K is the amount of speakers (the amount of clusters)
- $\odot$: Element-wise product
- $\boldsymbol{I}$: The identity matrix
- $\boldsymbol{1}$: The all-ones matrix

The latter two are introduced because the main diagonal of $\boldsymbol{C}$ does not provide any relevant information in the learning process of $\boldsymbol{B}$. Because $\boldsymbol{C}$ is symmetric, we model it as $\boldsymbol{B}\boldsymbol{B}^T$. It is possible to estimate $\boldsymbol{B}$ using iterative multiplicative update rules based on Euclidian divergence [16]:

$$\boldsymbol{B} \leftarrow \boldsymbol{B} \odot \frac{(\boldsymbol{C} \odot (\boldsymbol{1} - \boldsymbol{I}))\boldsymbol{B}}{(\boldsymbol{B}\boldsymbol{B}^{\boldsymbol{T}} \odot (\boldsymbol{1} - \boldsymbol{I}))\boldsymbol{B}} \tag{2.22}$$

Now each column of $\boldsymbol{B}$ contains the contribution of a microphone to each cluster. We can obtain the clustering result with:

$$\gamma_m = \{j \in [1, K] : B_{mj} \geq B_{mk}, \forall k \in [1, K]\} \tag{2.23}$$

The value γ_m denotes the cluster assigned to the m-th microphone. This is simply the largest value of column m.

2.4 Proposed methods

3

Methods

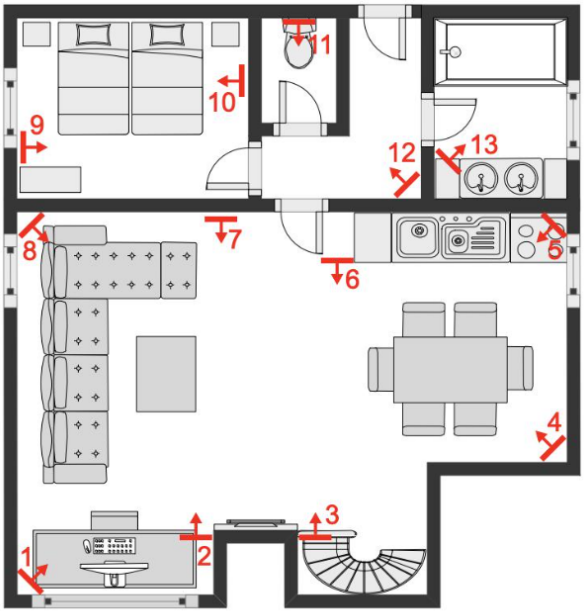
3.1 Pyroomacoustics

Pyroomacoustics is an open-source software package designed to streamline the development and evaluation of audio array processing algorithms. It provides a set of powerful tools for simulating and analyzing acoustic environments, including the generation of room impulse responses, the simulation of sound propagation, and the generation of synthetic audio signals. With its user-friendly interface and intuitive programming API, Pyroomacoustics is an accessible and versatile tool for researchers, students, and practitioners working in the field of audio processing. Its modular design and extensive documentation make it easy to extend and adapt to a wide range of research applications, from speech enhancement and source localization to sound event detection and acoustic scene analysis. Overall, Pyroomacoustics represents a valuable resource for anyone interested in exploring the fascinating and rapidly-evolving field of audio processing.

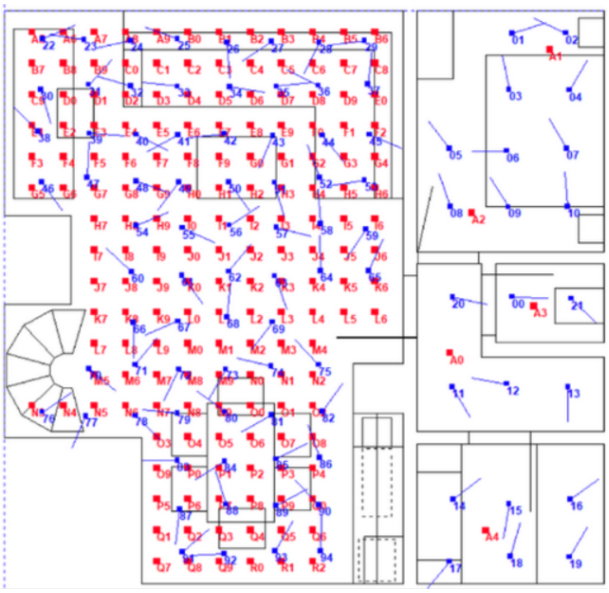
In the context of this thesis, Pyroomacoustics is used for the evaluation and comparison of different clustering methods.

3.2 SINS dataset

SINS is a collection of audio recordings that were captured in a real-life setting of a vacation home, where one individual lived for a duration of over a week. The audio was captured using a network of 13 microphone arrays that were strategically placed across multiple rooms. Each microphone array was composed of four microphones that were arranged linearly. The recordings were labeled according to the different levels of daily activities that were performed in the environment.



Figuur 3.1: Setup of the vacation home.



Figuur 3.2: Different options of the SINS dataset.

4

Results

Conclusion

References

[1] A. Oppenheim and R. Schafer, "From frequency to quefrency: a history of the cepstrum," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 95–106, 2004.

[2] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, nov 2001. [Online]. Available: https://link.springer.com/article/10.1007/BF02943243

[3] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937. [Online]. Available: https://doi.org/10.1121/1.1915893

[4] N. Madhu, "Acoustic source localization: Algorithms, applications and extensions to source separation," Ph.D. dissertation, Ruhr-Universität Bochum, 2009.

[5] R. M. Sebastian Gergen, Anil Nagathil, "Classification of reverberant audio signals using clustered ad hoc distributed microphones," pp. 1–12, 2014.

[6] N. M. Sebastian Gergen, Rainer Martin, "SOURCE SEPARATION BY FEATURE-BASED CLUSTERING OF MICROPHONES IN AD HOC ARRAYS," pp. 1–5, 2018.

[7] ——, "Source separation by fuzzy-membership value aware beamforming and masking in ad hoc arrays," pp. 1–5, 2018.

[8] R. M. Alexandru Nelus, Rene Glitza, "ESTIMATION OF MICROPHONE CLUSTERS IN ACOUSTIC SENSOR NETWORKS USING UNSUPERVISED FEDERATED LEARNING," pp. 1–5, 2021.

[9] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016.

[10] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, 2021.

[11] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, "On the byzantine robustness of clustered federated learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8861–8865.

[12] M. G. C. Antonio J. Munoz-Montoro, Pedro Vera-Candeas, "A Coherence-based Clustering Method for Multichannel Speech Enhancement in Wireless Acoustic Sensor Networks," pp. 1–5, 2018.

[13] W. A. Gardner, "A unifying view of coherence in signal processing," *Signal Processing*, vol. 29, no. 2, pp. 113–140, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0165168492900150

[14] P. Stoica and R. Moses, *Spectral analysis of signals*. Prentice Hall, 2004. [Online]. Available: https://user.it.uu.se/~ps/SAS-new.pdf

4 References

[15] H. D. S. Chris Ding, Xiaofeng He, “On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering,” p. 606–610, 2005.

[16] D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf

Appendices

Bijlage A

Toelichting bijlage.

Bijlage B

Toelichting bijlage.