Structure Image Method for Simulating Multi-Room Acoustics and Applications

Erez Shalev

Structure Image Method for Simulating Multi-Room Acoustics and Applications

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering

Erez Shalev

Submitted to the Senate
of the Technion — Israel Institute of Technology
Chislev 5782 Haifa November 2021

This research was carried out under the supervision of Prof. Israel Cohen, in the Faculty of Electrical & Computer Engineering.

Some results in this thesis have been published as articles by the author and research collaborators in journals during the course of the author's research period, the most up-to-date versions of which being:

E.Shalev and I.Cohen. Multiroom speech emotion recognition. Submitted to 47th IEEE Internat. Conf. Acoust. Speech Signal Process., ICASSP-2022, Singapore, May 22-27,2022.

Erez Shalev, Israel Cohen, and Dmitri Lvov. Indoors audio classification with structure image method for simulating multi-room acoustics. *The Journal of the Acoustical Society of America*, 150(4):3059–3073, 2021.

Acknowledgements

This work marks a significant milestone and represents the end of a period full of learning experiences and challenges.

I would like to express my gratitude to several people that helped and supported me during this research.

First, I would like to express my gratitude to Prof. Israel Cohen for the supervision, guidance, and support throughout this research. I couldn't have done it without you, and for this I am grateful.

Second, to my family. To my dear sister and brother Inbal and Eyal, for their support, help, and ideas. To my love Oriya Turgeman, who was patient enough to travel this journey with me. She supported me through the day-to-day, through the total length of the process, and also helped with some of the technical work. To my amazing parents, Sara and Ofer. their love and support were the keystone for this achievement, as well as, their involvement, read-review, and comments.

Last but not least, to Dimitry Lvov and DSPG. The collaboration with Dimitry was essential and he always had a fresh perspective on the subject matter.

I was fortunate to have all of you along this way and you are a big part of this achievement.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

Abstract

Can computers identify what the source of a sound in another room is? The task of audio classification is the task of recognizing the sound source within an audio segment. It is a very wide and hard task for machines. The performance of an audio classification system is very sensitive to environmental factors such as reverberation, and a system trained using common methods will perform poorly, when the sound source does not reside in the same room as the receiver.

Currently, the problem of sensitivity to environmental factors is solved by utilizing room impulse response generators. The output of these generators is convolved with the sound examples, in order to simulate the environment during the system training. However, for a cross-room task, the generator should be capable of simulating a cross-room audio transition. Moreover, when a large number of examples is needed, the generator should be capable of generating the examples fast enough, in a reasonable time frame. Current state-of-the-art room impulse response generators are either too slow, or incapable of generating a cross-room impulse response.

In this thesis, we present a new generator for cross-room impulse response. Our generator can simulate sound, travelling between two adjacent rooms, with a low computational complexity, suitable for a large scale simulation. We use the generated impulse responses in order to augment the dataset, by convolving each of the dataset examples with a number of audio environment examples. First, we define the data augmentation method, and show how this method improves the results of an audio classification system, for our generator as well as other generators. Then, we proceed to evaluate the audio classification systems on real-recorded cross-room examples, and show that our generator outperforms the current generator significantly.

Speech emotion recognition is the task of recognizing a speaker's emotional state, based on a speech audio segment. In order to show that our augmentation method can improve the performances of any cross-room audio task, for any model, we evaluate the first coupled-room speech-emotion-recognition system. We train three different models with two different architectures. We perform the training with and without our augmentation method, and show a performance improvement for both architectures. We use real-recorded examples of cross-room speech, to show that all models trained with our simulator, significantly outperforms models trained without our augmentation, or with a different generator.

This thesis introduces three novel contributions: (1) A new generator capable of simulating a cross-room audio transition. This simulator introduces a performance improvement for any of the tested architecture and for both of the tested cross-room audio tasks. (2) a methodology for augmentation of dataset, using audio-environment simulators. We show how using this methodology improves the performances of data-based method for audio classification tasks. It

is independent of the acoustical simulator method. (3) The first evaluation and assessment, to the best of our knowledge, on the tasks of audio classification and speech emotion recognition from another room. We train models using our defined methodology and simulator, as well as without, and show that our simulator out performs current existing methods, on data recorded in a real environment between rooms.

Glossary

Abbreviations

AR-GRU	attention ReLU GRU.
BA	Balanced accuracy.
CNN	Convolutional neural network.
CRNN	Convolutional recurrent neural network.
DCASE	Detection and classification of acoustic scenes and event.
DNN	Deep neural network.
DRN	dilated residual networks.
EmoDB	The Berlin emotional dataset.
ESC-50	Environmental sound classification dataset.
FN	False negative.
FP	False positive.
GMM	Gaussian mixture model.
GRU	gated recurrent units.
HMM	Hidden Markov model.
IM	Image Method.
IoT	internet of things.
k-NN	K-nearest neighbors.
LSTM	long short term memory.
NNMF	non-negative matrix factorization.
RAVDESS	The Ryerson audio-visual database of emotional speech and song.
ReLU	Rectified linear units.
ResNet	residual networks.
RIR	room impulse response.
RNN	Recurrent network.
SED	Sound event detection.
SELD	Sound event localization and detection.
SER	Speech emotion recognition.
StIM	Structure image method.
SVM	Support vector machine.
TDNN	time-delay neural networks.
TESS	The Toronto emotional speech database.

TN	True negative.
TP	True positive.
UA	Un-balanced accuracy.
WGN	white Gaussian noise.

Operators and known functions

$*$	The convolution operator.
$I_{(\cdot)}$	The indicator function, equals 1 if the condition in $(\cdot)$ is fulfilled, and 0 otherwise.
$\sum$	Summation function.
l_2	The 2-norm infinite-dimensional vector-space.

Notations

c	One of the classes to be classified.
$\hat{c}$	The class estimated by the model.
c_w	The speed of sound within a given wall.
C	The speed of sound.
$\mathcal{C}$	The set of all classes.
d	The spatial distance traveled by the sound signal.
f_c	Cut-off frequency of a wall, when it is considered as a low-pass filter.
F	The number of a feature map's frequency bins.
$h(t)$	A general representation of a room impulse response.
h_v	An RIR with respect to a virtual receiver location. Represents a repletion path.
j	y axis imaging variable.
(H, D, W)	height depth and width of a room.
k	z axis imaging variable.
K	Number of random room examples for augmentation.
$L = (L_x, L_y, L_z)$	3-D room dimensions.
L_w	The thickness of a give wall.
$\vec{m} = (m_x, m_y, m_z)$	A room replication vector on all three axes.
M	Maximal replication number.
$\hat{M}$	A classification model.
$\mathcal{M}$	The set of all possible room replication vectors with respect to M
n	Dataset's example index.
N	Dataset size.
$\mathbb{N}$	The set of natural numbers.
$O_{p,m}$	Reflection order with respect to the imaging vector $\vec{p}$ and the replica vector $\vec{m}$
$\vec{p} = (x_p, y_p, z_p)$	An imaging vector for low order imaging on all three axes.
P_{BA}	Balanced classification accuracy measure of the model.
P_{UA}	Unbalanced classification accuracy measure of the model.
P_{F1}	F1 score measure of the model.

$\mathcal{P}$	The set of all possible imaging vectors.
q	x axis imaging variable.
$\vec{r} = (x_r, y_r, z_r)$	A vector representing the 3-D coordinate location of the original receiver.
R_m	A vector representing the 3-D coordinate location of the replicated room.
R_p	A vector representing the 3-D coordinate location of the imaged source.
$s(t)$	The original audio signal to be classified.
$\vec{s} = (x_s, y_s, z_s)$	A vector representing the 3-D coordinate location of the original source.
t	Time variable.
T	The number of a feature map's time bins.
T_{60}	The time interval in which the decay level drops down by 60 dB.
$\vec{v} = (x_v, y_v, z_v)$	A vector representing the 3-D coordinate location of the imaged receiver.
$w(t)$	A time signal representing a random process noise.
(x, y, z)	Spatial 3D point location.
$y(t)$	The input signal to the classification model.
α_i	Transition coefficient with respect to wall i.
β_i	Reflection coefficient with respect to wall i.
δ	Dirac's delta function.
τ	The travelling time of the sound signal over the distance d.

Chapter 1

Introduction

1.1 Background and Motivation

Sound can travel through walls and between rooms. Information on the source producing that sound, can be extracted even between rooms. The task of identifying the sound event, which produced the audio, is called audio classification or sound event detection (SED). This task has been extensively researched in the past decade, and the technology that is able to perform it successfully is highly desired.

1.1.1 Sound event detection

There are many applications to SED. For example, smart cities can make use of SED to map noise sources in the city in order to improve citizen lives [1]. Cities can also utilize the classified noise, to alert emergency centers in cases of emergency, thus, improving citizens' safety. SED could be utilized for the benefit of smart homes as well. With the advancement of communication today, each and every home device or appliance can be connected to the internet. This concept is called the internet of things (IoT) [2]. IoT enables small programs to follow a person's behavior, and react with respect to his patterns. Examples are, notifying when the washing machine has finished working, alerting if the television was left on, or detecting anomal sounds in the childrens' room. The field of robotics can also utilise SED capabilities. The word robota is derived from the Slavic languages and generally refers to work. A robot is a machine that does the work for us, or assists in our work. Hence, human-robotic communication is an important part of robotics. Considering the fact that audio is a major human communication channel, it is highly beneficial for us to have robots which are capable of understanding audio cues. Security, alert, and alarm systems can also improve by using SED. Such systems can, for example, recognize a person falling, and alert in cases where an elderly person falls down. It can alert for a crying baby in the nursery or listen to the breathing of a baby and alert in case of breathing distress. SED capabilities can help raise an alert, by recognizing a "breaking window" sound or anomalies. In a long video surveillance recording, it can help index an event, and save us the time of watching redundant video segments. In general, audio is chipper to acquire than video, and insusceptible to lighting errors, making for a new brand of security systems, cheaper than the existing video systems.

Historically, specific cases of the SED task, such as human voice activity detection, music detection, or bird singing detection, were separately handled by different rule-based methods. General SED was studied using several data-driven methods, including support vector machines (SVM) and k-nearest-neighbors (k-NN) [3, 4], hidden Markov models (HMM) [5], Gaussian mixture models (GMM) [6], non-negative matrix factorization (NNMF) [7], Adaboost [8], and random forests [9], amongst others. However, the SED task has recently achieved impressive performances, with the advancements of deep-learning, and deep neural networks (DNN). Bilen et al. [10] explored a robust method for the evaluation of the task. Cakir et al. [11], Young et al. [12] and Adavanne et al. [13], among others, each experimented with their own DNN for polyphonic sound signals, be it a feed-forward DNN, a convolutional neural network (CNN), or a recurrent neural network (RNN). Adavanne also combined convolutional-recurrent architectures in a neural network [14] (CRNN), as a baseline model for the third task in detection and classification of acoustic scenes and event, 2019 (DCASE2019) [15]. Specifically for indoors audio classification, Bai et al. [16] have explored audio enhancement, using beamformers as a mean to improve classification performances. Their approach aims at eliminating reverberations, background interference, and noises.

Despite the high motivation for development, given by the possible applications, and the excessive research in the field, the state-of-the-art on SED still faces several challenges. A target source could be recorded in an environment with interference and other strong, undesired sounds. These sounds can be even louder than the target source. While our brains can overcome such interferences, to some extent, they majorly reduce the performances of current automated SED systems. Audio segments can contain more than one audio target source, in addition to interferences. These sources of interest can be time-wise consequent or overlapping, and their number could be unknown. Each of the sources has a different volume level, and they may be coming from different directions. Humans' auditory system can focus on a single source, in a loud environment, and even alter the attention between sources. The DCASE2019 challenge [17] issued a dataset with up to two time-wise overlapping classes, and evaluated the participants on the joint task of sound event localization and detection (SELD). While this is an improvement on the existing datasets, a real life audio may contain much more than two overlapping sources. Conversely to an image classification task, the SED task has much fewer available datasets. Moreover, fewer of them are hierarchically labeled. Hierarchic tagging is important, as source of sound can belong to several classes. For example, a 'car' label usually indicates that the sound belongs in addition to a superlabel, called 'vehicle'. This superlabel may also contain airplanes, bicycles, and boats. The Audioset [18] dataset is an example of a hierarchically labeled dataset, but the audio in each segment is recorded in uncontrolled environments. This means that the examples in the dataset cannot be combined to simulate time overlapping of sources, because they were recorded in different environments. Hence, a simple addition of the examples does not represents a real use-case. While DCASE2019 challenge [17] does offers us up to two overlapping classes, it falls short on the labeling, which is not hierarchical. The lack of control over the audio environment in the dataset, is a weak point of current datasets in general. Even though the DECASE2019 dataset is simulated over 5 different locations (different rooms), we have no control over these locations. The given dataset is already convolved with

the audio, and has the influence of the environment integrated into the recordings, without any possibility to change it. Audioset [18] presents audio from online videos, which means that the influence of the environment in the video is inherent to the recording, and can not be easily removed. In general, there is no single dataset yet with control over environmental factors, such as reverberations.

Data-driven methods have a high, inherent dependency on the dataset used to train them. For instance, if we want the system to be able to deal with polyphonic sounds, we need our dataset to have good representation of polyphonic examples. As mentioned before, there is a dataset shortage concerning environmental factors. There are two possibilities to cover environmental factors when training a data-driven method. The first is recording a new dataset, containing sounds from a large verity of environments. This option is time consuming and laborious. Moreover, it is task specific, and each environmental feature alteration and class addition requires re-recording. The second is simulating the environment, using an audio environment simulation method. This is the favorable solution, since for a new set of classes, you simply have to repeat the process with another dataset. In cases where environmental features alterations or classes addition are required, they are easily achievable by parameter changes and re-simulation. By quickly generating room impulse responses (RIR), this solution enables research and fine tuning with considerably less effort over the first option.

1.1.2 Acoustic environment simulation

In order to address the lack of control over the audio environment in current datasets, we can use acoustic environment simulators. An audio simulator can generate a room impulse response (RIR), which represents the room's reverberations, with respect to the locations of the source and receiver. There are three main disciplines for simulating an acoustic environment, namely, statistical methods, wave-based methods and geometrical methods, sometimes called ray-based. Statistical methods, such as statistical energy analysis [19, 20], do not concern with the temporal dynamic of a sound source, and are aimed for steady state information. Such methods are more suitable for constant noises, and are problematic when addressing audio events of a short duration. In the SED task, we anticipate signals which can be short timed, such as speech, or human distress such as a falling person or a baby's cry. Since speech is a transient audio signal, statistical methods are unfit for the SED task. The wave-based methods, such as finite element method [21, 22], boundary element method [23, 24, 25] and finite-difference time-domain [26], represent the acoustic of a structure by solving the wave equation. Such methods are highly descriptive in terms of room structure, although the solution is computationally demanding. Moreover, for some of these methods ,which are grid-based (such as the aforementioned wave-based methods), the computational complexity rises with the modeled frequency. Accordingly, these methods are more suitable for lower frequencies, and therefore they are limited when a large number of examples is required. While wave-based methods can be sufficient for classic algorithms, most of them are too slow for data-driven methods, where a high number of examples is required. One could generate such a large number of example once, and publish the results for the use of others. However, there are many different parameters for such a simulation, and a dataset covering all the possible combinations will be huge. The benefit of a fast simulation is the

ability to experiment with different ranges and combinations of parameters, without the need of a huge dataset. Geometrical, or ray-based methods, use rays or particles, which are reflected at the surface of the room, to model the travelling sound wave. Ray-tracing [27], is an example of a ray based method, which is primarily focused at the signal's energy, rather than the pressure wave [28]. The computational complexity of such methods is dependent on the reflection order and not on the frequency. Thus, these methods are faster than the aforementioned wave-based methods, for modeling higher frequency ranges. A commonly used simulation method, is Allen's and Berkley's Image Method [29] (IM). The IM is shown by Allen and Berkley to be a solution to the wave equation, given the source and receiver locations inside a rectangular enclosure with rigid walls . This approach is based on modeling waves reflected from walls by image sources in free field. IM provides this solution with a relatively low computational complexity, typical to ray-based methods, even when modeling higher frequencies. However, IM is limited to single-space square-shaped rooms.

Many expansions and additions where made, in order to improve IM: a highly efficient implementation of the IM method was published by Habets [30], Borish [31] introduced an expansion of IM to non-square rooms, Voländer [32] offered a ray-tracing combination, to obtain longer RIRs, with low computational complexity, incident-angle dependency was researched by Rindel [33], while Lam [34] studied adjustments for frequency dependent representation. When considering any indoors audio processing task, the source can sometimes be located in another room from the sensor. Non of these expansions has yet to explore IM's capabilities to address a source arriving from another room.

1.1.3 Speech emotion recognition

Speech is a specific class of sound, which can travel through walls, and contains plenty information. For example, a speech signal contains cues to the emotional state of the speaker. The task of recognizing this emotional state using the speech segment is called speech emotion recognition (SER). In general, such technology can help in customer support call review and analysis, human-machine interaction, mental health surveillance, etc. Mental health detection is becoming even more important with COVID-19 influences [35].

The vast research explores many aspects of SER. [36, 37, 38], among others, explored the input features for SER classifiers. Several supporting modalities were examined, such as visual-cues [39, 40, 41], bio-signals [42], textual information [43], and others. Many classification models were tested, such as, HMM [44, 45], GMM [46, 47], SVM [48, 47], and more.

Recently, deep learning methods have shown promising results for the field. Several DNN architectures were offered, among which are CNN, RNN, using long short term memory (LSTM) or gated recurrent units (GRU), time-delay neural networks (TDNN), residual networks (ResNet), and dilated residual network (DRN), to name a few. Some combinations of these architectures were tested as well [49, 50, 51, 52, 53]. Further, knowledge transfer between models was also studied, for different data and different domains, by [54]. Additional architectures are constantly tested, such as graph convolution networks based architecture and attention ReLU GRU (AR-GRU) [55, 56]. [57] explored methods to reduce the computational complexity of SER, and additional aspects of SER are being explored and researched constantly. The full scope of

aspects of SER is too vast to cover in this single work.

Despite vast research done in the SER field, none of the existing methods considers the speaker to be in a different room. With the increasing distress situations which were exposed by COVID-19's social isolation policy, this issue is becoming vital. A rise in domestic violence has been documented [58], as well as higher suicide rates [59] and other emotional responses with children, the elderly community, people coping with mental conditions, and the generally lonely persons. In such cases, cross-room emotion recognition can be very beneficial. Imagine a parent using such a system to recognize sadness or depression in their child in a different room, or a person detects the distress of a neighbor. Such a system can classify, for instance, aggressive behavior by noticing anger in one side of the conversation and sadness or fear in the other side, thus, alarming against domestic abuse. Sadly, the existing available datasets for SER do not support such audio-scenarios. The existing datasets are either recorded in a clean environment, or have a single room reverberation characteristics, which are uncontrollable to the researcher.

1.2 Research Overview

In this research, we wish to study the capability of audio classifiers, such as SED and SER, to classify audio indoors, when the signal is arriving from another room. Such capability is appealing to many applications, such as:

- Smart home systems [2] - can have economical benefits from such capabilities, saving redundant microphones.
- Non stationary robots, such as assistance robots [60] and surveillance robots [61] can benefit from the capability of classifying audio signals, which arrives from another room.
- Mental-health condition monitoring for isolation - In light of the experience acquired from COVID-19 isolation policy, this capability can improve life quality of patients in pandemic situations.
- Aggression detection [62] alert systems do not always have the privilege of residing in the same room where the aggression may occur. The ability to classify aggression beyond a wall can help alert, in cases of abuse and help with cases of domestic violence.

Accounting for the recent advancement of DNN-based classifiers, we wish to study and improve DNN audio classifiers, in cases where the sound arrives from another room. However, as mentioned before, DNNs are dependent on the training dataset. Existing datasets do not allow control over the environment they were recorded in. Hence, we wish to use a simulation method. The lack of simulation methods which can generate a high number of examples of cross-room audio scenarios, requires a development of a new method. The simulator should have a low computational complexity algorithm, which would be capable of simulating adjacent rooms audio environments in a large scale. Since Allen's and Berkley's IM is the most suitable existing method for a single room, we follow the principles of IM, while making adjustments for cross-room capabilities. We explore and define a framework for using simulated acoustic environments during the training of a DNN audio classifier. We evaluate the SED task on three different CNN

classifiers, using the environment simulation augmentation, and assess their performances when using either IM, our proposed cross-room method, or without any environmental representation at all. We proceed to evaluate the SER task for one of the previous CNN classifiers, and add two more RNN classifiers, to establish robustness of the DNN architecture, and independence of the specific task. In addition, we detail a methodology of integrating the RIR simulation method as an augmentation method for audio datasets.

1.3 Thesis Structure

This thesis is organized as follows: Chapter 2 contains mathematical background, formulation of the problem and necessary background materials for the following chapters. Chapter 3 discusses the problem of adapting the existing methods into coupled rooms. We suggest a novel solution and analyze the performance on the SED, with respect to three CNN classifiers. Chapter 4 evaluates the SER task, and compares the CNN and RNN performances, given augmentation from our simulator. Lastly, we conclude our discussion in Chapter 5 along with possible directions for future research and open questions.

Chapter 2

Background

In this chapter we provide background for reading this thesis. Section 2.1 formulates the problem. Section 2.2 overviews important highlights of the original IM.

2.1 Problem Formulation

2.1.1 Sound event classification formulation

We consider an audio signal $s(t)$ which is a recording of the audio class for SED or speech for SER, in a clear recording environment with minimum reverberation. When training without the environmental influences, we consider the input signal to the system, simply as,

$$y(t) = s(t). \tag{2.1}$$

For the simulation of the acoustic environment, we produced an RIR, $h(t)$. In this case, the input to our model is given by

$$y(t) = h(t) * s(t) + w(t), \tag{2.2}$$

where $s(t)$ is the target source, $*$ is the convolution operation, $h(t)$ represents the system through which the signal is deformed on its way to the receiver, and $w(t)$ is white Gaussian noise (WGN). We note that $w(t)$ has an important role in the integration of $h(t)$ into the dataset. We used a WGN signal, but an argument could be made for $w(t)$ being, not necessarily, white or stationary. We leave the research of other possible $w(t)$ for further study. We focus this work on $h(t)$ representing a sound segment which travels between two rooms.

Given a closed dictionary of classes $c \in \mathcal{C}$, we wish to classify the sound in $y(t)$ using a classification model $\hat{M}$ s.t. $\hat{c} = \hat{M}\{y(t)\}$. The balanced accuracy performance of a model P_{BA}, for a test set of size N, is simply the percent of correct classifications:

$$P_{BA} = 100\frac{\sum_{n=1}^{N} \mathbb{I}_{c=\hat{c}}[n]}{N} \tag{2.3}$$

where n is the sample index, $\mathbb{I}$ is the indicator function which equals 1 if $\hat{c} = c$, and 0 otherwise. This accuracy measure is simple and informative for a balanced dataset, such as the SED

dataset. In light of the results of challenges such as DCASE (Mesaros et al.[63],Kong et al.[64]), environmental sound classification (ESC-50) [65], the majority of SED models, and usually the most successful methods, are DNN. Before the emergence of deep learning, other classical methods were used for audio classification tasks such as HMM (Mesaors et al. [5]), GMM (Zhuang et al. [6]), NNMF (Gemmeke et al. [7]) and SVM (Temko et al. [4]). These methods achieve non-compatible results when compared with DNN models as shown by DCASE and ESC-50. Therefore, we limit our model discussion to DNN models.

2.1.2 Speech emotion recognition formulation

For the evaluation of SER models, we use three metrics. Given the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions, we first define the balanced accuracy (BA), P_{BA} by,

$$P_{BA} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.4}$$

Note that the BA measure is essentially the same as 2.3, written in the format of TP,FP,TN, and FN. Since the SER dataset is not balanced, we use two additional measures for the evaluation of the SER classifiers. For the second measure, we define the unbalanced accuracy (UA), P_{UA} given by,

$$P_{UA} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right). \tag{2.5}$$

Finally we define the F1 measure, P_{F1} given by,

$$P_{F1} = \frac{2TP}{2TP + FP + FN}. \tag{2.6}$$

The two last measures add more information when looking at a classification problem of an un-balanced dataset. We will use all three measures in the SER task.

2.2 Highlights on original image method

In ray-based simulation methods, a source $s(t)$ is modeled as a singular point emitting rays in all directions (similarly to a light source). These rays are reflected off surfaces and reverberate from the room's interior surfaces. The reverberations are represented as delayed, attenuated replicas, of $s(t)$, and are summed together with the direct path. Given a room architecture and information on the source and receiver locations, the result is an RIR $h(t)$ which can be convolved with any source $s(t)$, to simulate the reverberations. Note that this resulting RIR is specific to the source and receiver locations within the room. Such RIR represents the information on reverberations, by adequately attenuating each delayed replica with respect to time-decay, the order of reflections and the absorption of the room's surfaces.

The original IM is simply a way of tracing these routes, that is, calculating the delay and the reflections with low complexity. This method converges to the solution of the wave equation for a rectangular room, as shown by Allen and Berkley [29]. Hence, it is also considered to be a wave-based method as well as a ray-based method. We follow an efficient implementation of

the IM by Habets [30], and keep notations wherever possible. Figure 2.1 illustrates a low-order imaging, using the IM on a 2-D case, which is scalable for the 3-D case. Figure 2.1(a) represents the imaging (blue asterisk) of the original source (green asterisk) on the y axis, x axis, and a second order imaging on both x and y axes. A first order reflection path is represented by the orange line, whereas a second order reflection is represented by the green line. The first order reflection is demonstrated to create an isosceles triangle with base angles on the original and virtual (imaged) sources, and its obtuse angle on the left wall, where the dashed and solid lines meet. Due to the isosceles triangle, the path from the virtual source to the receiver is the same length as the path of the original source, when it is reflected off of the left wall. Given the length of the path d and the known speed of sound C, the delay can be calculated by,

$$\tau = \frac{d}{C}. \tag{2.7}$$

Using this calculated delay, the decay of the signal due to the distance, can be accounted for by

$$h_v(t) = \frac{\delta(t-\tau)}{4\pi d}, \tag{2.8}$$

where $h_v(t)$ represents the impulse response of such a virtual source. The wall's reflection coefficient, also has to be accounted for, as some of the sound is absorbed by the wall. This will be discussed later and is omitted in (2.8).

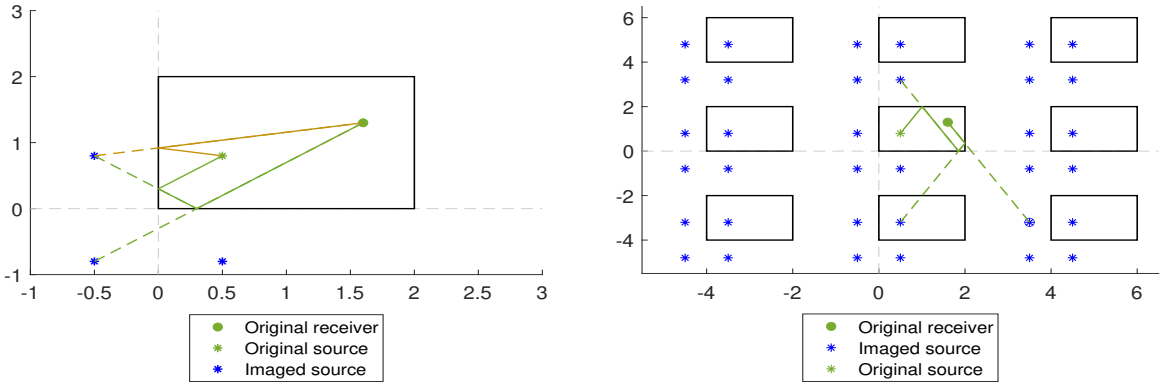
The imaging on the x axis creates an additional isosceles triangle on the lower wall, which is not traced in the illustration. The imaging on both axes represents a second order reflection on both axes, where the absorption of both the lower and the left walls, has to be accounted for. Isosceles triangles can be traced for higher order, and are demonstrated in Figure 2.1(b) using the green lines. The circled, blue asterisks, is the imaged-source being traced, and the solid lines represents the real path inside the room.

For a general 3-D source located at $\vec{s} = (x_s, y_s, z_s)$ we define the imaging vector $\vec{p} = (q, j, k)$. We follow Habets [30], and represent the imaging using $R_p = (x_p, y_p, z_p)$ as

$$R_p = (x_s(1-2q), y_s(1-2j), z_s(1-2k)), \tag{2.9}$$

where $q, j, k \in \{0, 1\}$. For example, a 3-D replication on two axes x and y which is not replicated on z, will be represented by setting $(q, j, k) = (1, 1, 0)$. Hence, the set of vectors $\vec{p} \in \mathcal{P}$ controls the low order reflections (up to 3 walls).

Let the room dimensions be defined by $L = (L_x, L_y, L_z)$. The original IM now proceeds and allocates replicas of this room, where the lower left corner of the room is allocated to all combination of $R_m = (2m_xL_x, 2m_yL_y, 2m_zL_z)$ for each replica. Here, $m_x, m_y, m_z \in \mathbb{N}$ are ranging in some range $-M < m_i < M$. These replicas of the original room and the sources ,(both original source and images), represent higher order reflections. The path traced in Figure2.1(b) represents a path for a virtual source, where $M = 1$. The figure illustrates the original source (green asterisk), the receiver (green circle), un-traced virtual sources (blue asterisks) and the traced virtual source (circled blue asterisks). Here, the parameters for the traced virtual source are $(q, j) = (1, 0)$ and $(m_x, m_y) = (1, -1)$. A similar example for such illustration can be found in Room Acoustics, Section 4.1, by Kuttruff [66].



(a) (b)

Figure 2.1: Original image method's illustration in 2-D case. The room's walls and the duplicates are marked in black rectangles. (a) Low order imaging using $\vec{p} \in \mathcal{P} = (q, j, k)$. (b) High order imaging using both $\vec{m} \in \mathcal{M} = (m_x, m_y, m_z)$ and $\vec{p} \in \mathcal{P} = (q, j, k)$.

Finally, given the receiver location $\vec{r} = (x_r, y_r, z_r)$, the path length d between the virtual source and the receiver is calculated by

$$\begin{aligned} R_p &= [(x_s(1-2q) - x_r, y_s(1-2j) - y_r, z_s(1-2k) - z_r] \\ R_m &= [2m_x L_x, 2m_y L_y, 2m_z L_z] \\ d &= \|R_p + R_m\|, \end{aligned} \tag{2.10}$$

which we can plug into (2.7) and account for the decay. The number of replications M of the room is either set by calculation or with respect to a user input value. When calculated, the constraint for enough replications is that the longest delay time T_{60} is the time interval in which the decay level drops down by 60 dB. This is commonly known as T_{60} (Room Acoustics, Section 3.8, by Kuttruff [66])

The reflection order of a source, located at distance $d = \|R_P + R_m\|$ with respect to the receiver, is given by

$$O_{p,m} = |2m_x - q| + |2m_y - j| + |2m_z - k|, \tag{2.11}$$

and the full impulse response is given by

$$\begin{aligned} &h(\vec{s}, \vec{r}, t) = \\ &= \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \beta_1^{|m_x - q|} \beta_2^{|m_x|} \beta_3^{|m_y - j|} \beta_4^{|m_y|} \beta_5^{|m_z - k|} \beta_6^{|m_z|} \frac{\delta(t - \tau)}{4\pi d}. \end{aligned} \tag{2.12}$$

The attenuation effect due to each single reflection from a face i, is accounted for by multiplying once with the respective reflection coefficient β_i.

Chapter 3

Structure Image Method

Introduction

Expanding the IM into coupled rooms, may seem as simple as replacing the first reflection coefficient with a transition coefficient. This transition coefficient represents the sound arriving from outside the room, transitioning through the wall, rather than reflected off of it. However, applying the image method without any additional alterations will cause phantom reflections, which do not represent any real path traveled by the sound signal. In this chapter we will first demonstrate how does the phantom, un-viable paths occur, when using the original IM as it is, in Section 3.1. We will analyze the case of a source which is located in a void outside a shoe-box room. Then, we suggest a new approach called the receiver IM in Section 3.2, which can produce the same results as the original IM for a single room. In Section 3.3, we will proceed to analyze the new method developed in Section 3.2 with respect to the problems, presented in Section 3.1. Section 3.4 systematically define the method in Section 3.3 for a general case. After analyzing the shoe-box room case, using this approach, we proceed to expand the method, and add the influence of an adjacent room in Section 3.5. Finally, we will present our evaluation of the method in Section 3.6.

3.1 Allocation of a source outside a room

We start by considering the original IM and the efficient implementation by Habets [30], when the source is outside the room. Figure 3.1(a) first raises a problem with some low order reflections. The original source (green asterisk) has a direct path which goes through the wall. Relying on Fundamentals of Acoustics, Section 6.2, By Kinsler [67], we consider the wall as a thin transition layer. This assumption means that the ray has the same direction of movement after transitioning through the wall. There is a contrition between the functionality of the wall as a rigid wall and a thin transition layer. The former, is derived under the assumptions of the IM, while the latter is a necessary assumption for the ray to travel straight through the wall. The rigid wall assumption is implying an infinit mass for the wall and zero elasticity, while the transition layer assumption implies a zero mass and infinit elasticity on the wall. One way to think about this physical model is assuming that the wall is rigid, and is vibrating as a single unit with respect to the impinging audio waves, transferring them to the other room. Also, we

note that the assumption is frequency dependent, as is the original IM, and holds for frequencies below the limit $\frac{c_w}{2L_w}$, where c_w is the speed of sound within the wall's material, and L_w is wall's thickness. The difference is that the signal, now, has to be multiplied by a transfer coefficient, rather than a reflection coefficient, and the wall acts as a low-pass filter, with a cutoff frequency f_c, of $f_c = \frac{c_w}{2L_w}$, attenuating high frequencies.

The direct path is a viable path, which has to be multiplied by the respective coefficient. However, when considering the virtual imaged sources, we can see in the example shown in Fig. 3.1(a) that only the imaging on y axis (green plus) has a viable reflection from the bottom wall. This virtual source needs to be multiplied with both the transition coefficient of the left wall, and the reflection coefficient of the bottom wall. The virtual source inside the room (red x) has no wall in its path to be reflected from the real source, hence, there is no such path through which the source can travel to the receiver. The imaging on x axis (also red x) uses the virtual source inside the room in the path reflection analysis, which practically does not exist. Therefore, this source also does not have a viable path. This problem is intensified when the original source is located in the lower left corner (green plus). In this case, only the direct path is a viable path, while all three images do not represent a real reflection path.

This problem complicates further when we scale up the reflection order. Let us assume that we could define a method for resolving which of the low order reflections are viable when given a source location, thus, eliminating the low order difficulties. The next step in IM would be to simply duplicate the room while only considering these viable images. Figure 3.1(b) illustrates the validity of paths with higher order reflections. The original source (green asterisk) is located with relatively close proximity to the left wall, with respect to the room dimensions. All the blue asterisks represent virtual sources with viable paths, while the virtual sources in red x do not have viable paths. It is clear from the scattering of the viable sources that there is no original pattern of a low order case, which we can replicate, in order to scale up. Moreover, there is no apparent relation between the parameters q, j, k, m_x, m_y and m_z to the validity of the path, and no apparent order can be found, even in terms of the drawing (note the virtual source at $(q, j, m_x, m_y) = (0, 1, 0, -1)$). In other words, there is no way to translate this back to an efficient algorithm, which would drop paths that are not viable and take into account only viable ones.

The comfortable example of a source located with relatively close proximity to the wall is an easier case. A more difficult case to analyse, would be of a source outside the room, which is located at a very far point from the wall, or at a point that has direct paths to two walls, meaning, rays can enter the room through each wall. When scaling up to 3D, we may even have a three wall penetration case. These two possibilities complicate further the allocation of a source outside the room with the existing method. We have yet to attend to the consideration of multiplying the high order reflections, by the correct coefficients with respect to the walls. Figure 3.1(c) illustrates a source where two walls are in the source's direct path, and its distance from the wall is greater than any one dimension of the room. The original source is represented by a green asterisk, virtual sources without a viable path are represented by red x symbols, and blue asterisks represent virtual sources with viable paths. Figure 3.1(c) further emphasises the difficulty of translating the existing IM into a fast and efficient algorithm for a source outside a

shoe-box room. This difficulty poses a limitation for generating a large number of RIRs with low computational complexity. Such limitation, induce the mentioned challenges of training a DNN model on a large number of simulated RIRs, which represent sources arriving from different rooms, thus, improving the classification's robustness to the environmental factors.

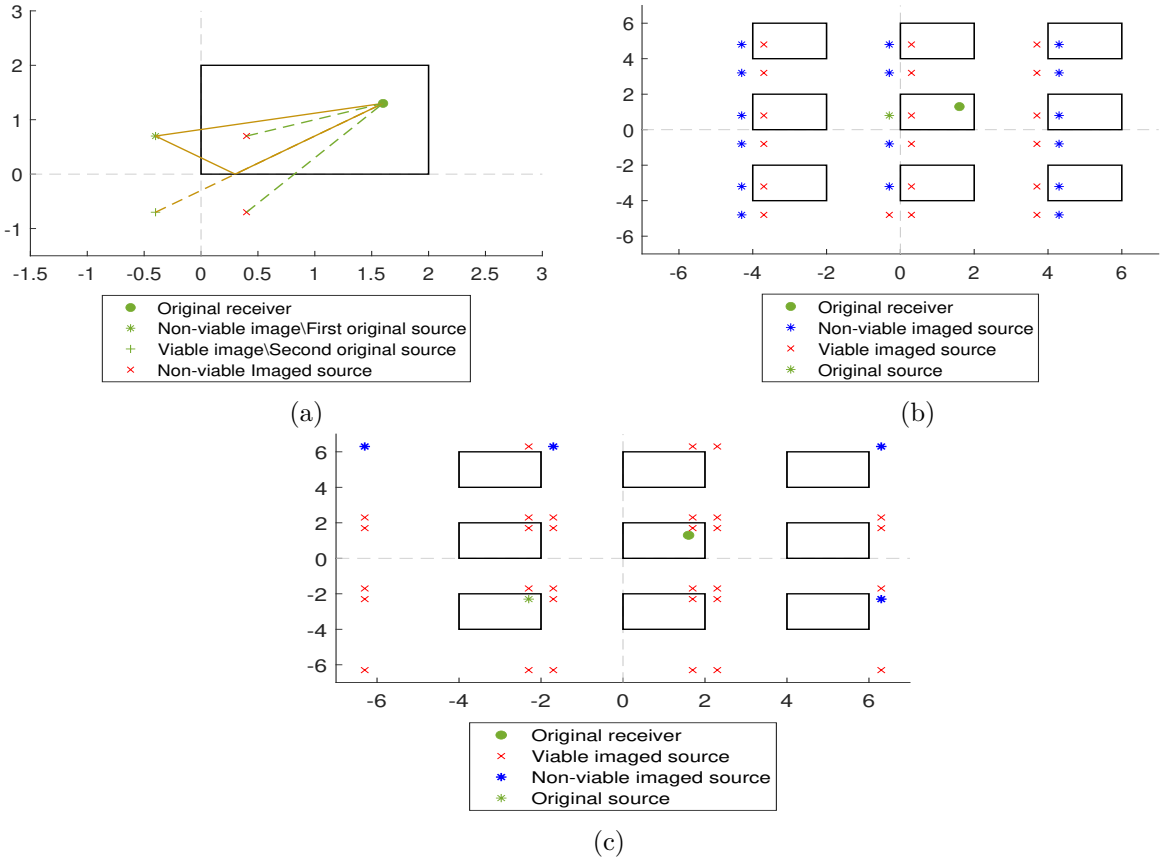


Figure 3.1: Original image method's 2D illustration for a source outside a shoe-box room.(a) Low order imaging of a source located outside the shoe-box room. Tow examples are given where the original source is different. (b) High order imaging of a source located outside a shoe-box room with relatively close proximity and with direct path to a single wall. (c) High order imaging of a source located outside a shoe-box room, relatively far from the room walls, with respect to the room's dimensions and with direct paths to two walls.

3.2 Receiver imaging

We, first, suggest a minor alteration to the original image method. Instead of imaging the source, we choose to image the receiver. Imaging of the receiver and the source is demonstrated in Figure 3.2. The original source (green asterisk) is imaged on the y axis to create a virtual source (blue asterisk). The original receiver (green circle) is also imaged on the y axis to create a virtual receiver (blue circle). The lines crossing from the imaged source to the real receiver, and from the imaged receiver to the real source, are obviously equal. Note that these lines create two isosceles triangles connected by their obtuse angles. This minor alteration may be achieved by modifying (2.10) to

$$
\begin{aligned}
R_p &= [(x_r(1-2q) - x_s, y_r(1-2j) - y_s, z_r(1-2k) - z_s] \\
R_m &= [2m_x L_x, 2m_y L_y, 2m_z L_z] \\
d &= \|R_p + R_m\| \ .
\end{aligned} \tag{3.1}
$$

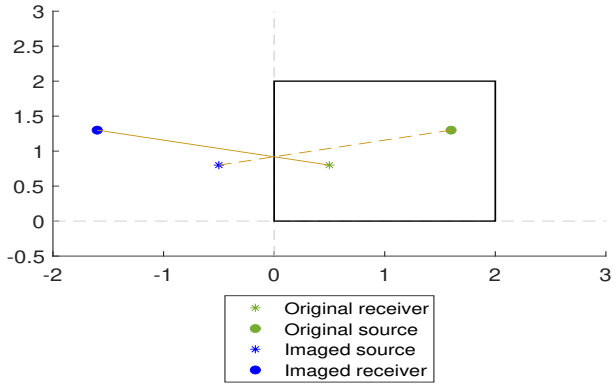


Figure 3.2: Imaging of both the source and receiver on the y axis.

We have generated two RIRs using the original IM and the receiver IM. The first pick, which is the loudest and represents the direct path, had a real mean square of 0.05. A comparison of the two showed a maximal error of $\sim 10^{-15}$ order of magnitude, which could be assigned to a numerical error. Hence, we can say that the methods are identical. Figure 3.3, compares an RIR generated by the receiver image method and a real RIR measured in a real room with objects and some furniture. Figure 3.3(a) depicts the room of sizes $(L_x, L_y, L_z) = (2.55, 3.3, 2.5)$ and a source and receiver located at $(x_s, y_s, z_s) = (1.25, 2.7, 0.55)$ and $(x_r, y_r, z_r) = (0.95, 0.3, 1.2)$ respectively. The real RIR was measured using a sine sweep [68]. It is clear from the measurement, that both the original and receiver image methods are not an accurate representation of a real environment. The receiver image method is capable of generating an RIR which is identical to the original image method, given a small alteration. This small alteration has no cost for the computation complexity over the original method and is very easy to implement.

3.3 Allocation of a source outside a room with receiver imaging

Now let us analyze a source allocated in a void outside of a shoe-box room, using the receiver image method. It is easy enough to skip the low order case, and move directly to a high order reflection example in Figure 3.4. All problems presented in Section 3.1 for the original IM, are reduced, and all that is left is to determine whether the line of sight between the source and the virtual receiver goes through the original room or not. This is demonstrated in Figure 3.4(a). All blue circles receivers and their paths (green dotted lines) go through the original room (the room located at $(0, 0)$), while all other receivers are red squares (paths in orange dotted lines). This is also true for the low order case which we skipped.

Figure 3.4(b) shows a path tracing of one such virtual receiver. Note that even though the method drops the irrelevant receivers, it still uses them in order to trace the paths of relevant receivers.

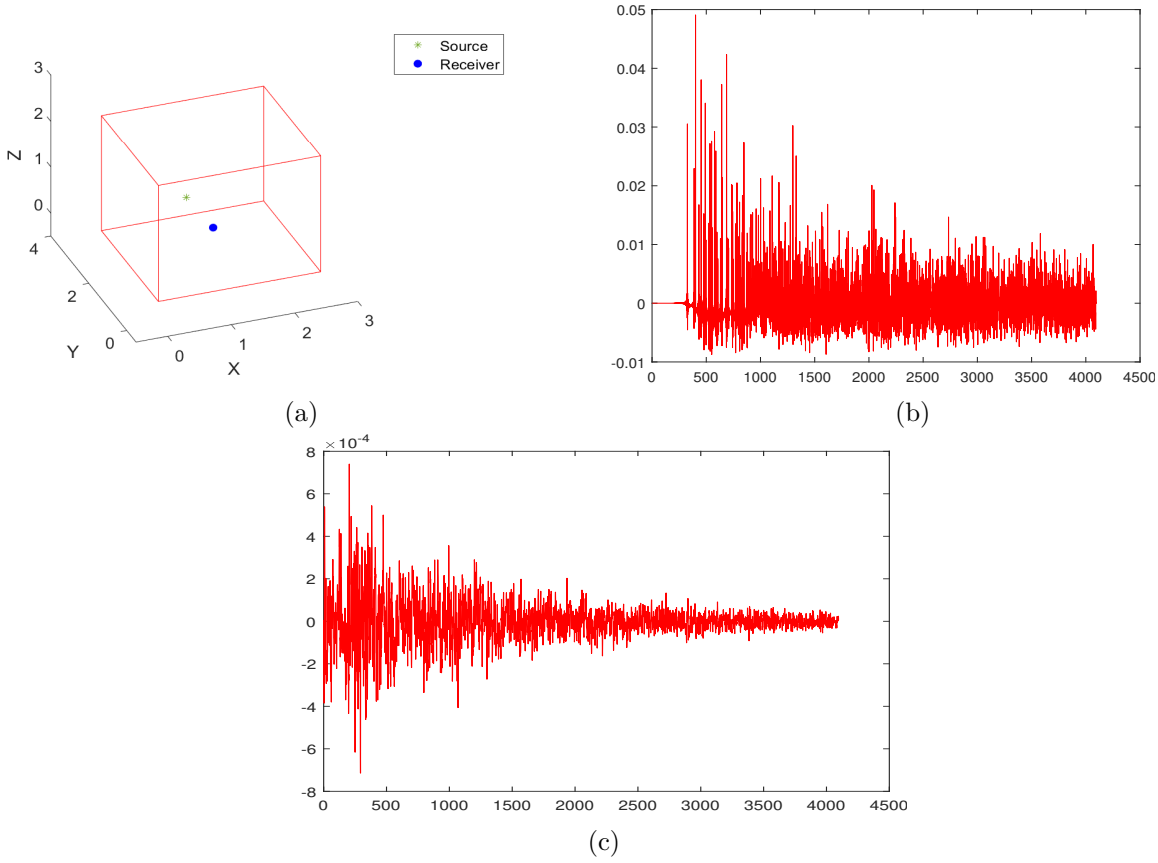


Figure 3.3: Original Receiver image method vs. a measured image method. (a) Room setup, (b) Receiver image method, (c) measured image method.

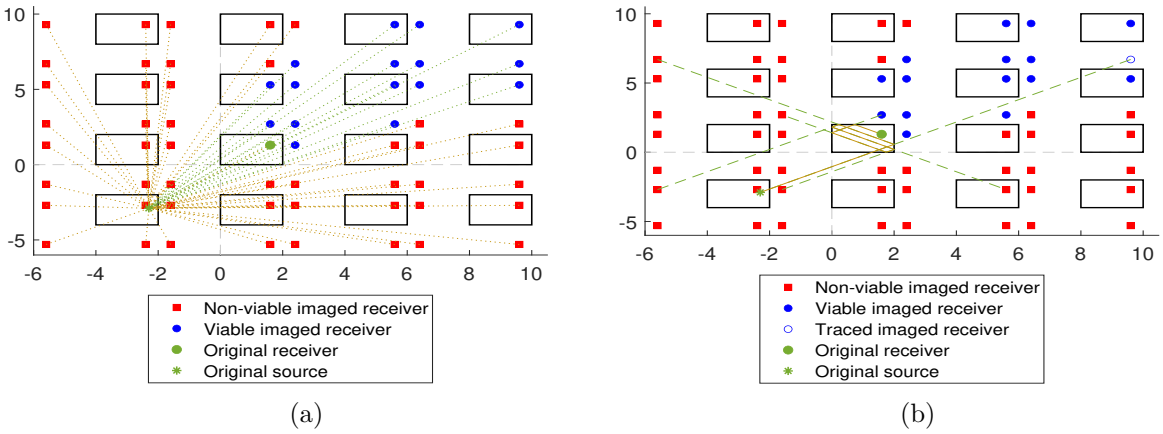


Figure 3.4: High order receiver imaging with a source outside a shoe-box room. (a) High order mapping of viable sources using rays. (b) Illustration of a reflection path for the virtual imaged source at $(m_x, m_y) = (2, 2)$ and $(q, j) = (0, 1)$. The real path in solid orange and the tracing in shredded green line.

3.4 Intersection with walls for a given virtual source

Given a virtual receiver, located at $\vec{v} = (x_v, y_v, z_v)$, and a source at $\vec{s} = (x_s, y_s, z_s)$, we wish to find if the room is located on the direct line between the points, and through which wall does the line penetrate the room. The line between the points is given by

$$l(u) = \vec{s} - u(\vec{v} - \vec{s}), \tag{3.2}$$

in a parametric form, using the parameter u. We constraint the location of the room to $0 \leq u \leq 1$, so that it is not located on the infinite line, inside the segment, between the two points.

Let a shoe-box room of size (L_x, L_y, L_z) be located at the origin. We calculate the parameter u for which the line intersects with each wall, and we infer the 3-D point of the intersection (x, y, z). Considering the wall on the x axis, for example, we first require the points to be on opposite sides of the wall. Then, we also require $0 \leq y \leq L_y$ and $0 \leq z \leq L_z$, to ensure that the point is within the wall's limits. All intersections with respect to the faces of the room are pre-calculated in Table 3.1. Walls, which are parallel to an axis, received, in the table, either the values 0s, when they are on the axis, or L_i if they are located on L_i, with respect to the axis i.

Table 3.1: Intersection of the Parameterized Line $l(u)$ with each of the Room's faces.

Face	u	x	y	z
1	$\frac{x_v}{x_v - x_s}$	0	$\frac{x_v y_s - x_s y_v}{x_v - x_s}$	$\frac{x_v z_s - x_s z_v}{x_v - x_s}$
2	$\frac{x_v - L_x}{x_v - x_s}$	L_x	$\frac{(L_x - x_s) y_v - (L_x - x_v) y_s}{x_v - x_s}$	$\frac{(L_x - x_s) z_v - (L_x - x_v) z_s}{x_v - x_s}$
3	$\frac{y_v}{y_v - y_s}$	$\frac{x_s y_v - x_v y_s}{y_v - y_s}$	0	$\frac{z_s y_v - z_v y_s}{y_v - y_s}$
4	$\frac{y_v - L_y}{y_v - y_s}$	$\frac{(L_y - y_s) x_v - (L_y - y_v) x_s}{y_v - y_s}$	L_y	$\frac{(L_y - y_s) z_v - (L_y - y_v) z_s}{y_v - y_s}$
5	$\frac{z_v}{z_v - z_s}$	$\frac{x_s z_v - x_v z_s}{z_v - z_s}$	$\frac{y_s z_v - y_v z_s}{z_v - z_s}$	0
6	$\frac{z_v - L_z}{z_v - z_s}$	$\frac{(L_z - z_s) x_v - (L_z - z_v) x_s}{z_v - z_s}$	$\frac{(L_z - z_s) y_v - (L_z - z_v) y_s}{z_v - z_s}$	L_z

At this point, for implementation efficiency, we define $\bar{u} = u|(x_v - x_s)(y_v - y_s)(z_v - z_s)|$ to eliminate the denominators. We pre-calculate the following 13 helper variables for the conditioning phase:

$$\begin{aligned} s_{xy} &= (x_v - x_s) L_y \\ s_{xz} &= (x_v - x_s) L_z \\ s_{yx} &= (y_v - y_s) L_x \\ s_{yz} &= (y_v - y_s) L_z \\ s_{zx} &= (z_v - z_s) L_x \\ s_{zy} &= (z_v - z_s) L_y \end{aligned} \tag{3.3a}$$

$$
\begin{aligned}
a_{xyz} &= |(x_v - x_s)(y_v - y_s)(z_v - z_s)| \\
a_{xy} &= |(x_v - x_s)(y_v - y_s)| \\
a_{xz} &= |(x_v - x_s)(z_v - z_s)| \\
a_{yz} &= |(y_v - y_s)(z_v - z_s)|
\end{aligned}
\tag{3.3b}
$$

$$
\begin{aligned}
c_{xy} &= x_s y_v - x_v y_s \\
c_{xz} &= x_s z_v - x_v z_s \\
c_{yz} &= y_s z_v - y_v z_s
\end{aligned}
\tag{3.3c}
$$

The conditions for each face of the room are summarized in Table 3.2. Note that the condition cannot be met for both face 1 and face 2, and the same goes for any two parallel faces. In terms of efficiency, in the worst case scenario we go through 12 conditions with 4 additional comparisons per axis. Algorithm 3.1 utilizes the sources, the pre-calculated variables in (3.3), the virtual receiver locations, the room dimensions and the conditions from Table 3.2. Using these, Algorithm 3.1 can determine if the ray between the receiver and the source crosses parallel faces respectively to an axis, and returns the face which is crossed first. Given the room's dimensions and the locations of both source and virtual receiver, Algorithm 3.2 uses Algorithm 3.1 to determine the face that is first crossed amongst all faces. An output of 0 from Algorithm 3.2 represents that the ray does not go through the original room at all (not a viable path).

Table 3.2: Face Intersection Conditions.

Face	$\bar{u}$	points on both sides of the face	Face limits first dimension	Face limits second dimension
1	$-x_v a_{yz}$	$x_v < 0 < x_s$	$0 \leq c_{xy} \leq -s_{xy}$	$0 \leq c_{xz} \leq -s_{xz}$
2	$(x_v - L_x)a_{yz}$	$x_v < L_x < x_s$	$s_{yx} - s_{xy} \leq c_{xy} \leq s_{yx}$	$s_{zx} - s_{xz} \leq c_{xz} \leq s_{zx}$
3	$-y_v a_{xz}$	$y_v < 0 < y_s$	$s_{yx} \leq c_{xy} \leq 0$	$0 \leq c_{yz} \leq -s_{yz}$
4	$(y_v - L_y)a_{xz}$	$y_v < L_y < y_s$	$-s_{xy} \leq c_{xy} \leq s_{yx} - s_{xy}$	$s_{zy} - s_{yz} \leq c_{yz} \leq s_{zy}$
5	$-z_v a_{xy}$	$z_v < 0 < z_s$	$s_{zx} \leq c_{xz} \leq 0$	$s_{zy} \leq c_{yz} \leq 0$
6	$(z_v - L_z)a_{xy}$	$z_v < L_z < z_s$	$-s_{xz} \leq c_{xz} \leq s_{zx} - s_{xz}$	$-s_{yz} \leq c_{yz} \leq s_{zy} - s_{yz}$

Algorithm 3.1 Determines penetration through original room, with respect to 1D.

Input: $a, L, x_s, x_v, face, face\_num, face\_u\_bar$

Output: $face\_num, face\_u\_bar$

```
if x_s < 0 < x_v then
  u_bar ← −x_s a
  if u_bar < face_u_bar then
    if first and second conditions of face then
      face_u_bar ← u_bar
      face ← face + 1
    end if
  end if
else if x_s < L < x_v then
  u_bar ← (x_s − L)a
  if u_bar < face_u_bar then
    if first and second conditions of face + 1 then
      face_u_bar ← u_bar
      face_num ← face + 1
    end if
  end if
end if
```

Algorithm 3.2 Determines through which of the faces does the ray between the source and receiver penetrate (if any).

Input: L, s, v

Output: $face\_num$

```
Initialize:
calculate all 13 variables
face_num ← 0
face_u_bar ← a_xyz
face_num, face_u_bar ← Alg1(a_yz, L_x, x_s, x_v, 1, face_u_bar)
face_num, face_u_bar ← Alg1(a_xz, L_y, y_s, y_v, 3, face_u_bar)
face_num, face_u_bar ← Alg1(a_xy, L_z, z_s, z_v, 5, face_u_bar)
```

3.5 Structure Image Method (StIM)

We now have to take into account the fact that the source is also located within a room. Let the receiver room be defined by $L_r = (L_{x,r}, L_{y,r}, L_{z,r})$ and the source room be defined by $L_s = (L_{x,s}, L_{y,s}, L_{z,s})$. Let the source and receiver be located at $\vec{s} = (x_s, y_s, z_s)$ and $\vec{r} = (x_r, y_r, z_r)$, respectively. Figure 3.5 shows a 2D such scenario, where the source's room is replicated. The original source in green asterisks. The receiver room's is not replicated, and the original receiver is in a green circle. First, we simply apply the receiver image method, using the original source location. Next, we continue by applying the original IM to the source's room,

in order to generate virtual source locations. Similarly to the problem presented in Section 3.1, we need to decide which of the virtual sources has a viable path. Mathematically, this is the same problem of finding out which virtual receiver is viable, when the original source is outside the room, only now, the receiver is the one outside the room. Hence, the solution is also similar to Section 3.2, with the exception that here we stay loyal to the imaging of the sources. For each virtual source, we can use Algorithms 3.1 and 3.2 to determine if the ray between the virtual source and the original receiver goes through the original source room. If it does, we apply the receiver image method on the receiver room, using that virtual source's location (blue asterisks in Figure 3.5). Otherwise, we omit the virtual source as the path is not viable (red x symbols in Figure 3.5).

We are now left with the last consideration of adjusting the reflection attenuation, to account for the wall transitions. In the implementation by Habets [30], the impulse response for such virtual source $\vec{s}$, given the receiver location $\vec{r}$ was already introduced by (2.12). We now adjust these equations for the case of StIM.

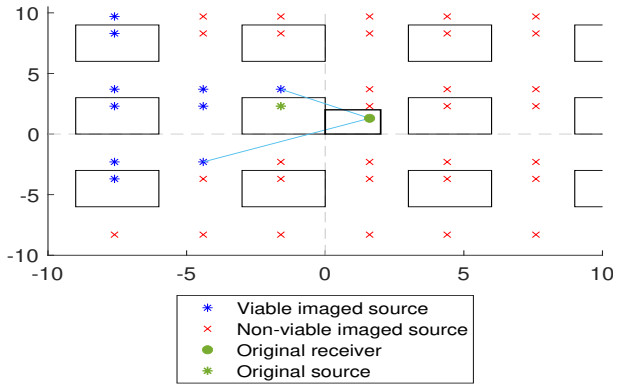


Figure 3.5: Structure image method illustration.

We note that for adjacent rooms, the rays can only be transferred through the joint wall. This can happen in two ways as presented in Figure 3.5 with the light blue lines. The ray either penetrates through the cross section between the joint wall, in which case, the transition coefficient is with respect to that wall, or it travels out of the source's room and into the receiver's room through another wall. Let us split the impulse response into two factors: h_1 and h_2, where h_1 contains all the rays penetrating the inner wall, and h_2 contains the rays that travel out of the source's room. For the h_2 case, we have to use Algorithms 3.1 and 3.5 to find the face of the receiver's room, through which the ray enters. The transition coefficient is then the multiplication of the transition coefficient for that face, and the joint face. Higher order cases, of a ray which travels more than once between the rooms, exist as well. We do not handle these cases in this paper, as they are usually attenuated below 60 dB. To adjust for the attenuation of a joint room impulse response, we gather the reflection coefficients from the source room $\beta_{i,s}$ and from the receiver room $\beta_{i,r}$. We denote the transition and reflection coefficients of the face through which the ray exits the source room and enters the receiver room $\bar{\beta}_s, \bar{\alpha}_s, \bar{\beta}_r$ and $\bar{\alpha}_r$, where β represents reflection and α represents transition. The impulse response $h_1(\vec{s}, \vec{r}, t)$ for the first case, for a virtual source $\vec{s}$ given $p_s \in \mathcal{P}_s = (q_s, j_s, k_s)$, $m_s \in \mathcal{M}_s = (m_{x,s}, m_{y,s}, m_{z,s})$ and a receiver $\vec{r}$ given $p_r \in \mathcal{P}_r = (q_r, j_r, k_r)$, $m_r \in \mathcal{M}_r = (m_{x,r}, m_{y,r}, m_{z,r})$ can be written by

$$
\begin{aligned}
&h_1(\vec{s},\vec{r},t) = \\
=&\frac{\bar{\alpha}_s}{\bar{\beta}_s\bar{\beta}_r}\sum_{p_s\in\mathcal{P}_s}\sum_{m_s\in\mathcal{M}_s}\sum_{p_r\in\mathcal{P}_r}\sum_{m_r\in\mathcal{M}_r} \\
&[\beta_{1,s}^{|m_{x,s}-q_s|}\beta_{2,s}^{|m_{x,s}|}\beta_{3,s}^{|m_{y,s}-j_s|}\beta_{4,s}^{|m_{y,s}|}\beta_{5,s}^{|m_{z,s}-k_s|}\beta_{6,s}^{|m_{z,s}|}] \\
&[\beta_{1,s}^{|m_{x,r}-q_r|}\beta_{2,s}^{|m_{x,r}|}\beta_{3,s}^{|m_{y,r}-j_r|}\beta_{4,s}^{|m_{y,r}|}\beta_{5,s}^{|m_{z,r}-k_r|}\beta_{6,s}^{|m_{z,r}|}]\frac{\delta(t-\tau)}{4\pi d}.
\end{aligned}
\tag{3.4}
$$

Here, we note that $\bar{\beta}_s = \bar{\beta}_r$. We simply divide by both to omit a single reflection for each method, as it becomes a transition. We then represent the transition by multiplying by the respective transition coefficient, $\bar{\alpha}_s$. For the second case, the impulse response $h_2(\vec{s},\vec{r},t)$, we are only required to account for the additional multiplication, by the transition coefficient of the receiver room's face through which the ray enters. Hence, the impulse response can be written by

$$
h_2(\vec{s},\vec{r},\tau) = \bar{\alpha}_r h_1(\vec{s},\vec{r},\tau). \tag{3.5}
$$

This complete analysis is dubbed the structure image method (StIM). StIM can produce a structural RIR with a relatively small overhead, while maintaining the capability of producing an inside room RIR using the receiver image method, without any overheads. It enables generating a large number of RIRs, describing sources arriving from an adjacent room, with a low computational complexity. StIM enables the training and evaluation of DNN models for classification of sound, with a higher diversity of environmental representation.

We note that in an indoor environment, sound may travel in other ways between rooms. StIM can model a pathway between coupled rooms by assigning a transition coefficient of 1 to a segment of the wall, representing a door or a window. However, it cannot model a transition of sound from distant rooms (non coupled) via an open pathway. An expansion of StIM to a higher number of rooms may be able to cope with such a problem, by assigning an adequate transition coefficients to specific segments. We leave this subject to future work.

Figure 3.6 depicts an example of a StIM RIR, compared with a measured RIR. The measured RIR, in Figure 3.6(c), was measured in two identical coupled rooms, both of size $(L_x, L_y, L_z) = (2.55, 3.3, 2.5)$ meters. Figure 3.6(a) details the measuring setup, where the source's room is allocated between -2.55 and 0 on the x axis, while the receiver's room is allocated between 0 and 2.55 on the x axis. The source and receiver locations are $(x_s, y_s, z_s) = (-0.4, 1.3, 0.9)$ and $(x_r, y_r, z_r) = (1.28, 1.75, 0.81)$ respectively (in meters). Both rooms are real room with various objects and furniture. Figure 3.6(b) is the StIM measured RIR, of the same setup. Similar to the receiver and original IM, StIM is not an accurate representation of a real environment.

We have measured the generation time of IM RIR. The average over three RIR measurements, is 0.3743 seconds. Measuring the StIM RIR generation, the average of three measurements yielded 2.73 seconds for a single RIR. It is important to note, that if we ignore the reflections from the source room, and simply account for the reflections from the receiver room, the StIM RIR is generated in an average of 0.3726 seconds. This shows that the longer calculation time

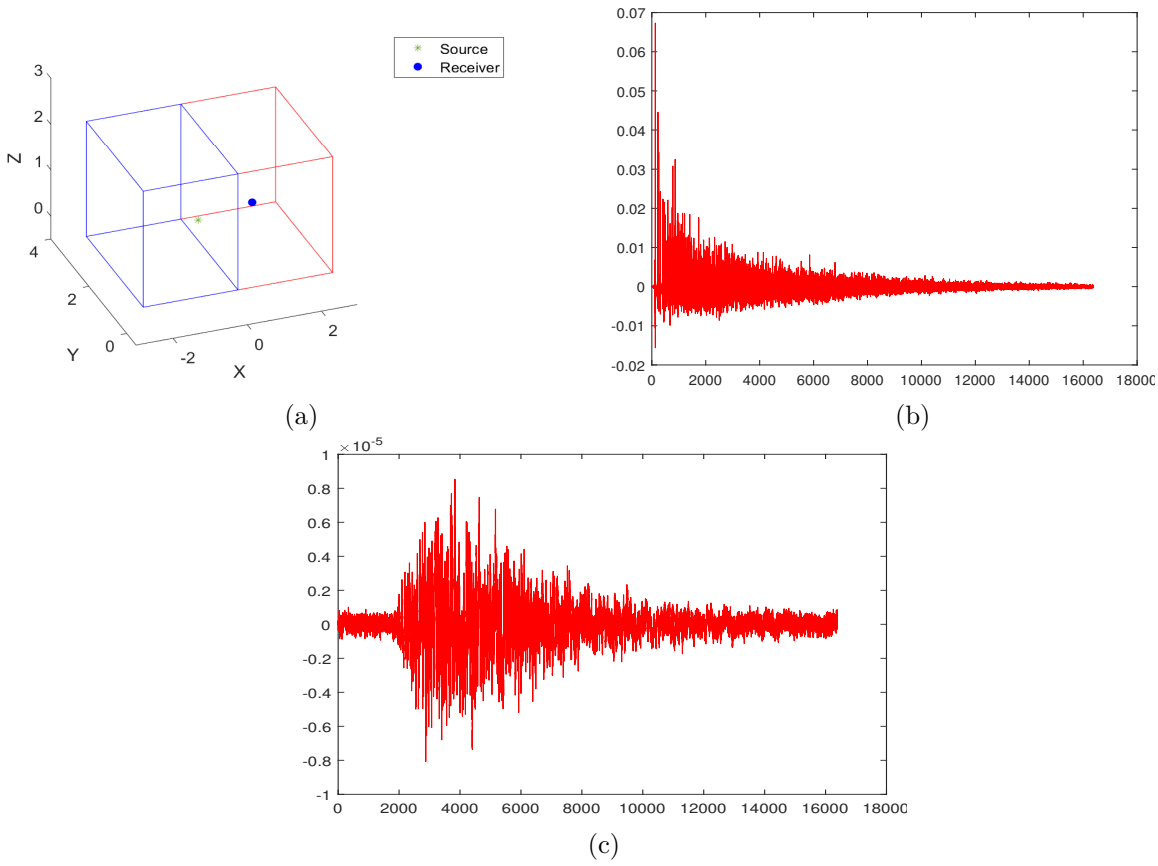


Figure 3.6: An example of StIM with a coupled room audio scene vs. a measured image method. (a) Coupled rooms setup, (b) StIM, (c) measured image method.

is only due to taking the reflections from the receiver room into account.

3.6 Experimental Framework and Results

3.6.1 RIR simulation

For simulation purposes, we have generated 1000 location samples. Each location is composed of two adjacent rooms of random sizes. For simplicity, all the reflection and transition coefficients are equal on all faces. The height, depth and width of the rooms (H, D, W) are randomized under the constraints $2 \leq H \leq 6$,$1.5 \leq D \leq 4$,$1.5 \leq W \leq 4$. The ceiling and floor of both rooms are aligned to create the same height in both rooms, as well as an alignment with respect to a single wall, as presented in Figure 3.5. The height and wall alignment assumptions are often the case also with common structures. In each location simulated, we randomize a receiver location in one of the rooms. Two sources locations are then randomized, one in each room. Thus, every location contains two RIRs, one inside the receiver's room and the second in the adjacent room. The first RIR represents the existing IM and the second RIR is calculated using StIM. All RIRs are of length 4,096 samples, with 44.1 KHz sample rate.

3.6.2 Dataset, features and pre-processing

For audio classification, we have used the ESC-50 dataset [65]. We are focusing on three indoors associated classes, namely, 'crying_baby', 'coughing' and 'toilet_flush', yielding a dictionary of size $|\mathcal{C}| = 3$. This dataset contains very clean samples, which were recorded in an interference free, quiet environment. Each sound is a 5-second segment, and was sampled with 44.1 KHz sample rate. In the pre-processing phase, we used three different processes to represent three different scenarios, using the RIRs produced in Section 3.6.1. The first scenario is the plain sound, as it is (clean). The second scenario includes convolving samples with IM RIRs from Section 3.6.1, that represent the source and receiver being inside the same room (inside_room). The third scenario simulates the sound arriving to the receiver from an adjacent room (outside_room), by convolving with StIM RIRs from Section 3.6.1. After simulating the scenario by convolving with the respective RIR, each sample is padded with zeros up to 6 seconds. Mel-frequency cepstral coefficient (MFCC) features are extracted for each sample using 40 coefficients calculated over mel-spectrogram, with 2,048 FFT bins, a 2,048 Hanning window, and 512 samples hop length. We have tested STFT and mel-spectogram features, as well. MFCC features gave the best representation out of the three, however, this may not be the best feature for other SEC tasks, such as polyphonic audio or a time-varying label. Finally, a random WGN $w(t)$, with signal to noise ratio of 30 dB is added, as in (2.2), and the sample is normalized with respect to root mean square (RMS) of 1. Ko et al. [69], who studied the effects of IM RIRs simulation on DNN models training for a speech recognition task, showed that the transition to real environment, when training with IM RIRs, can be problematic. In order to mitigate this problem, they have developed a full algorithm, which uses randomized WGN point sources and ambient noise. Adding such randomness to the data deals with the fact that StIM and IM are not accurate representations of a real environment with objects, as shown in Figures 3.3 and 3.6. Though we do not follow their full algorithm, we show that simply adding a WGN $w(t)$ to each sample is enough, in this case, to make a smooth transition. Adding randomized WGN point sources, helps to better represent a real environment, as reflections from stationary objects such as a furniture, and moving objects such as people, are not well represented in the IM. From a deep-learning point of view, the addition of WGN can be considered as a form of vicinal risk minimization. We train on similar but different examples by adding the random vector, which is also know as data augmentation (Chapelle et al.[70], Zhang et al. [71]). Following this pre-processing methodology, we augment the dataset with K RIR instances for each audio example, in the cases of inside_room and outside_room. The value of K was empirically tested with the values $\{1, 5, 10, 20, 50, 100, 200\}$.

3.6.3 Deep neural network model and training

For a classification model, we used 3 different models. All models are CNN, we assume the label is constant throughout each sample, and we treat the input as a single image. One could use an RNN\CRNN model and treat the input MFCC as a time series. We are dealing with RNN architectures in Chapter 4. We chose a simple, generic, CNN as a baseline. We compare the results against an Alexnet classifier [72] and a VGG16 classifier [73]. Both, Alexnet and VGG16 are common classifiers and we show that our data-augmentation method, improves the results

for all three methods. All models are implemented using keras.

Baseline:

The input to the baseline model is a tensor of size $1 \times 40 \times 517$, where 1 is the number of channels (a gray-scale image), 40 is the number of MFCC coefficients, and 517 is the resulting time bins from Section 3.6.2. The baseline method is composed of 4-layered CNN blocks followed by a dense soft-max layer. Each CNN block has a first CNN layer with a fixed kernel size of 2, relu activation functions, and the number of filers is $2^i \cdot 16$, where $i \in 0, 1, 2, 3$, is the index of the layer. The CNN layer is followed by a max-pooling layer with a pool-size of 2. The block is concluded with a dropout layer with a dropout-rate of 0.1. After all 4 blocks, we add a 2D global-average-pooling, followed by a dense, soft-max classifier of size $1 \times |\mathcal{C}|$, with an l_2 regularization, where $|\mathcal{C}|$ is the number of classes to be classified. The model was trained using an Adam optimizer, with a learning rate of 0.001, and categorical cross-entropy loss.

Alexnet:

For the Alexnet classifier, we used the original implementation with 3 alterations. We have adjusted the input to the size of $1 \times 65 \times 479$ tensor. Here 1 is the number of channels (a gray-scale image), 65 is the number of MFCC coefficients, and 479 represents a time segment of 5.77 seconds. The cropping in time is done in order to fit the image to the Alexnet classifier, and is still within the zero-padding limit of Section 3.6.2. The second alteration is the step size of the first layer alone. We set the step-size to $(1, 9)$ so that the second layer of Alexnet receives the expected size. Finally, we have altered the last soft-max layer to the required number of classes $|\mathcal{C}|$. The model was trained using an Adam optimizer, with a learning rate of 0.001, and categorical cross-entropy loss.

VGG16:

For the VGG16 classifier, we have also changed the size of the input size to the first layer, so that we avoid further changes to the architecture. The input is a tensor of size $1 \times 65 \times 479$. In this model we had to lower the learning rate to 0.0001. Otherwise, the model did not learn. We used an Adam optimizer, with categorical cross-entropy loss.

3.6.4 Augmentation:

Prior to training, we divided the dataset into train, validation and test sets, with equal distribution of classes in each set. For the cases of inside_room and outside_room, the 1,000 RIRs produced in Section 3.6.1 are also divided into train, validation and test sets. This division allows us to test the robustness of the resulting system, with respect to the location. We randomise K RIRs for each sample with respect to the division, such that RIRs in the test set were never seen in the training phase (same for validation). The samples are augmented by convolving with each of the K RIRs, as described in (2.2). We then proceed to extract features, according to the process described in Section 3.6.2. We trained each model for 100 epoches, using early stopping

Table 3.3: Evaluation of architectures' performances, trained without any augmentation. The training, validation, and testing results are presented for each architecture. The most right column is the performance on real sound recorded by us in a real environment.

Network	Training	validation	Test	Real data
Baseline	100%	93.33%	100%	56.66%
Alexnet	100%	93.33%	96.66%	40%
VGG	100%	96.66%	100%	53.33%

with patience of 5. The number of epoches and patience, were empirically chosen. We assess the results using (2.3).

3.6.5 Experimental results analysis

We used the scenarios described in Section 3.6.2, in order to train and asses numerous models of each network. The training procedure of each model represents the scenario which was used to train it, and the respective chosen K for inside_room and outside_room. For example, a training procedure named 'inside_room_K_20' represents a model trained using IM RIRs with $K = 20$ impulse response augmentation, while 'outside_room_K_100' represents a model trained using StIM RIRs with $K = 100$ impulse response augmentation. We started by training a clean model for each network and evaluate the results. For the evaluation, we use the test set, as well as, real sound recorded by us in a real environment. We have recorded a total of 30 recordings, 10 for each class, using two laptops and one cellphone. The recordings were made in two of the researches' residential environment due to Covid-19 isolation limitations. 'toilet_flush' was recorded in two different locations. For the first set, the source's room sizes where approximately $H = 2.5_m, D = 1.1_m, W = 0.7_m$, and for the second location $H = 2.75, D = 1.2_m, W = 1.75_m$. In booth cases, the receivers location was in a hall, with many openings to many adjacent rooms. The source room's door was kept close and other doors from the receiver's hall were either close or open. The 'coughing' class was also recorded in two sets of adjacent rooms. In the first set, the source's room is of size $H = 2.5_m, D = 2.5_m, W = 3.1_m$, and the receivers room is $H = 2.5_m, D = 2.5_m, W = 3.4_m$. In the second set, the source's room is of size $H = 2.75_m, D = 5.2_m, W = 3.3_m$, and the receivers room is $H = 2.75_m, D = 3.3_m, W = 3_m$. In both cases, the doors where kept closed. The 'crying_baby' was recorded in a third, single location, where the source's room dimensions were $H = 2.1_m, D = 3_m, W = 2.75_m$ and the receiver's room is $H = 2.1_m, D = 3_m, W = 2.5_m$. These locations, present rectangular rooms (excluding halls) with various furniture, objects and openings (doors and windows). The receiver location was altered between recordings, as well as, the 'coughing' class's source location. Table 3.3, presents the training, validation, and testing results for all three networks on the clean datasets, as well as, performances on the real recordings. The clean models seem to achieve very good results on the simulation phase, without the help of any augmentation. However, Table 3.3 reveals that these results can be degraded, when environmental features are present.

In order to improve the environmental robustness, we have proceeded to train 7 additional models for each network, with RIR augmentations. At first stage, we have generated 1000 such location RIRs, and divided them into train, validation and test sets. Then, we chose

a different number of RIRs, K, to represent the location for each of the 7 models, where $K = \{1, 5, 10, 20, 50, 100, 200\}$. Out of the respective set, we have randomized K locations for each sample. This procedure represents the sound over a large variety of locations while avoiding representing each sample with all location examples, which reduces the complexity while maintaining performances. Figure 3.7 shows the accuracy results for the testing set, as a function of K, for all three classifiers, for both inside_room and outside_room scenarios. The results show that using a higher order of augmentation, K, can mitigate over-fitting towards the training set's RIRs. For both IM and StIM cases, we observe a saturation of the performances using $K = 100$ RIR examples. This generally makes sense, as a higher number of K better represents the distribution of source and receiver allocations and structure of the room. The results justify the requirement for a large number of RIRs. StIM is essential in order to produce such a large number of simulated RIRs quickly. It can produce both structure RIRs, while maintaining the ability to generate an IM RIR using the receiver imaging.

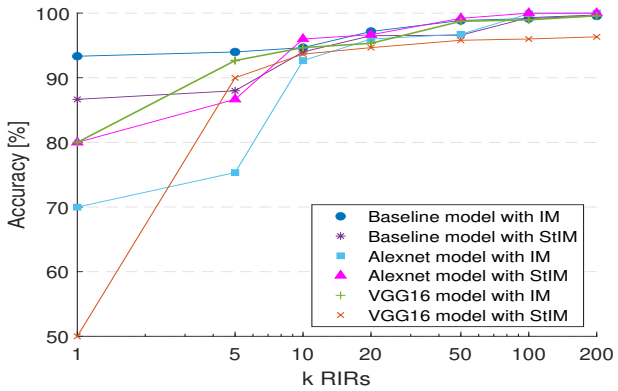


Figure 3.7: Accuracy of all three classifiers with respect to K randomly chosen RIRs.

In order to study the performances of each classifier when presented with a different scenario than the one presented in the training process, we proceed to evaluate the models using specific pre-processing scenarios. Table 3.4 is divided into two parts, the first one shows the results of classifiers when presented with data from the 'inside_room' training scenario, and the second part show the results of classifiers when presented with data from the 'outside_room' training scenario. We have evaluated only the $K = 200$ models, as these are the best performing models for each training scenario.

At first glance in Table 3.4, the results can be deceiving. The 'clean' models seem to achieve similar performances when tested against scenarios they did not train with. However, as was shown in Table 3.3, these results are only true for clean samples, recorded in a quiet and clean-of-interference environment. When we introduce the effect of reverberations to these models, the results are not as promising. Table 3.4 shows that models trained in different incompatible scenarios can cope with other scenarios to some extent. On the first part of Table 3.4, the models trained under the 'outside_room' scenario achieves bellow 90% for all networks. On the second part, the models trained under the 'inside_room' scenario, also achieves bellow 90% for all networks. Not surprisingly, the models trained to fit the specific scenario achieve the best results.

These results show the benefits of training with generators which are capable of simulating

Table 3.4: Evaluation of architectures' performances, trained without augmentation (clean), with IM augmentation (inside_room), or with StIM augmentation (outside_room). The performance in the higher table, 'Inside room' are with respect to a test set, pre-processed using IM. The performance in the lower table, 'outside room' are with respect to a test set, pre-processed using StIM.

Inside room			
training procedure	Baseline Accuracy	Alexnet Accuracy	VGG Accuracy
inside_room_K_200	**99.71%**	**99.94%**	**99.58%**
outside_room_K_200	73.03%	86.88%	86.53%
Clean	94.53%	84.78%	98.81%
Outside room			
training procedure	Baseline Accuracy	Alexnet Accuracy	VGG Accuracy
inside_room_K_200	75.43%	78.01%	89.51%
outside_room_K_200	**98.8%**	**99.81%**	**95.85%**
Clean	94.96%	89.3%	**96.68%**

many structural examples to augment the data, such as adjacent rooms. Since both IM and StIM are not an accurate representations of a real room, the transition from simulation into a real environment is not trivial. This is true especially for StIM, given the contradictions in the assumption of the physical model mentioned in 3.1. In order to study this transition, we proceeded to evaluate the three different models for each network, with our real recorded samples. Table 3.5 details the performances of all 9 models on the real-recorded examples. The middle column in Table 3.5 shows the results for models that were trained without adding $w(t)$ in the training process. Such models achieved similar performances to the ones where $w(t)$ was added on the simulation phase. However, when transitioning to real examples, recorded in a real structure, there is an obvious degradation in the performances of all models trained without $w(t)$ (middle column of Table 3.5). From the right-most column in Table 3.5, it is clearly visible that the models trained with RIRs of any kind, when adding $w(t)$, are performing better in real life recordings. The introduction of some randomization to an RIR can improve the results. We have added $w(t)$, following the procedure in Section 3.6.2. Alternatively,it is also possible to implement on StIM the full algorithm proposed by Peddinti et al. [69], to deal with the transition to real-environment. Table 3.5 shows that all 3 classifiers benefit from StIM, when presented with a recording from another room.

Adding too much noise (low SNR) can degrade the performances simply by masking the original signal with noise. too little sound, however, may result in an inaccurate representation of a real environment. Thus, a very high SNR is also a problem. Hence, we proceed to test each of the three networks, using $K = 200$, StIM RIRs augmentation, while altering the SNR of the $w(t)$ randomization factor. Figure 3.8 shows the results for different SNRs. When we look at the higher end of the SNR, where the randomization is relatively lower and has a lower impact, we observe a decrease in the performances. This is to be expected, since the results converges to the results where we do not add any $w(t)$ in Table 3.5. On the lower end, we also see a drop in performances, converging into the region of 30% to 40%. These results are close to guessing when dealing with a model, classifying between 3 classes. On $SNR = 0$, the signal and the random noise contribute the same mean square power to the total signal. This means

Table 3.5: Evaluation of architectures' performances, trained without augmentation (clean), with IM augmentation (inside_room), or with StIM augmentation (outside_room). The performance are with respect to a test set recorded in a real environment of coupled rooms. The middle column represents the results of models trained without the addition of a WGN $w(t)$, while the left column, represents the results of models trained with the addition of a WGN $w(t)$.

model	Accuracy without $w(t)$	Accuracy with $w(t)$
Baseline_Clean	66.66%	56.66%
Baseline_inside_room_K_200	63.33%	83.33%
Baseline_outside_room_K_200	60%	**93.33%**
AlexNet_Clean	53.33%	40%
AlexNet_inside_room_K_200	63.33%	69.99%
AlexNet_outside_room_K_200	76.66%	**83.33%**
VGG16_Clean	50%	53.33%
VGG16_inside_room_K_200	63.33%	76.66%
VGG16_outside_room_K_200	76.66%	**90%**

that the noise obscures the signal. Hence, the classifiers are converging to guessing.

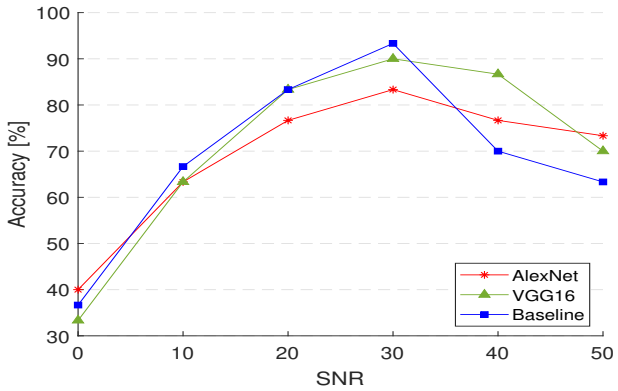


Figure 3.8: Accuracy of all three classifiers, trained using StIM, with $K = 200$, with respect to different SNR levels. The performance are with respect to a test set recorded in a real environment of coupled rooms.

The fine-tuning of the SNR level is, probably, classifier and task dependent. Figure 3.8 shows that for VGG, the optimal SNR is somewhere between 30 dB and 40 dB, where for the baseline, it is much closer to 30 dB. It is possible that the tuning is also depends on the dataset and involved classes. Broad spectrum signals, will probably need more caring in the additive noise level, as they are more similar in nature to the noise. Colored noises can be considered in such cases, to differ the noise's spectral characteristics from the target class's spectral characteristics. We leave this subject for future study.

Despite the assumptions' contradiction mentioned in 3.1, this augmentation method is evidently beneficial to improve the results. Diving dipper into the physical model of StIM may improve the performances even more. We live this subject for future research as well.

Chapter 4

Speech emotion recognition using StIM

4.1 Introduction

We have demonstrated the benefits of using StIM augmentation, for the audio classification task using CNNs, in chapter 3. We were mainly concerned with establishing the contribution of StIM to a single case of a data-driven method for a specific task. We wish to show that the contribution is not task-limited or architecture-limited. Hence, this chapter will demonstrate the benefits of StIM for a second task, SER, while utilizing other architectures, specifically RNN. We will evaluate a SER classifier with the previously tested CNN, and compare to an evaluation on two RNNs.

4.2 Datasets

We are using a combination of three datasets for the training and evaluation of the SER models: The Berlin emotional datase (EmoDB) [74]. The Toronto emotional speech database (TESS) [75], and the Ryerson audio-visual database of emotional speech and song (RAVDESS) [76]. For simplicity, we limit the datasets to the specific emotion labels ['angry', 'sad', 'neutral', 'ps', 'happy'], yielding a dictionary of size $|\mathcal{C}| = 5$. Each audio file was loaded with sample rate of $48KHz$, and each sample is zero-padded up to $309,500$ samples (approximately 6.448 seconds). The zero padding is used in order to fix the input length, and the number of time bins T. We proceed to follow the procedure in 3.6.2 for environmental representation, and extract both MFCC and mel-spectrogram coefficients, which are extracted from an STFT. The STFT was calculated with $2,048$ frequency bins, using a $2,048$ samples hanning window, with 512 samples hop length. MFCC was calculated using 45 coefficients. Both MFCC and mel-spectrogram features were concatenated to create a $(T \times F) = (605 \times 173)$ feature map. We have extended a third dimension with size 1 to represent a single channel, where the model requires a channel dimension.

In addition to the three datasets, we have recorded sounds in a real, coupled-room, audio-scene. This real-test set is composed of 75 samples, 15 for each of the 5 labels. In this real-test, the receiver room was of size $(H \times D \times W) = (2.75_m \times 1.5_m \times 2.5_m)$, and the source room was

of size $(H \times D \times W) = (2.75_m \times 3.1_m \times 2.5_m)$. Both rooms are real adjacent rooms, containing various furniture, and other absorbing and reflecting objects. Between recordings, we altered the locations of the source and receiver within their respective rooms. We used two common cellular phone devices for our recordings, which should better represent the data quality that such a system is most likely to encounter in real life. We kept the sampling rate, and used mono-recordings, in MP4 file format.

4.3 Models

We used three DNN architectures, namely a CNN AlexNet model, and two custom RNN models. All three were trained with and without augmentation, and evaluated using the matrices defined in Section 2.1.

4.3.1 Architecture discussion

AlextNet is a common classifier, and was also used in 3 for SED. A CNN architecture considers the input as an image, where one axis represents the time bins and the other represents the frequency bins. Since the convolution is two dimensional, this allows the architecture to extract both inter-frequency and sequential data, as well as the connections between them. However, the filter size is fixed, which limits the length of the sequential feature extraction, in terms of time-length.

Conversely to CNNs, RNNs are specifically tailored for the extraction of sequential data. The architecture's advantage is in learning the length of contextual sequence relevant for the task at hand. Considering our task, we note that emotion in speech normally has a sequential factor and dependency between consequent words and tonality. Even the tempo of the sentence can contain valuable classification information. Therefore RNN architectures are also prime candidates for this task, due to their relative benefits in classifying sequential data. For the purpose of evaluating adjacent room SER, we designed two custom bi-directional RNN models. We have to keep in mind that the RNN units are incapable of extracting inter-frequency information, and the architecture is dependent on the dense units for that purpose.

Current literature is indecisive with regards to the optimal architecture for audio tasks, RNN or CNN [77]. While both architectures have their pros and cons, we have to test all three of our models, in order to find the optimal method for SER in general, and for SER from an adjacent room in particular.

4.3.2 AlexNet

The original AlexNet is designed for a square, 224×224 image as an input. In order to fit our feature map into the network, we have altered the first layer, so that it will receive a single channel, $(T \times F)$, image. We kept the original $(11, 11)$ filter size, but altered the stride to $(11, 3)$, so that it fits our $(T \times F)$ size. As a result, the rest of the model remains unchanged. When generating the dataset as described in Section 4.2, we added an additional channel dimension. The AlexNet was trained using 150 epochs, using an Adam optimizer, with early stopping.

4.3.3 RNN

We created 2 RNN customized networks. All the parameters were empirically chosen, with respect to the datasets and task use-case. In both networks, the first two layers are RNN layers with 128 bi-directional units. The difference between the RNNs is the type of RNN units, namely GRU or LSTM. Each RNN layer is followed by a dropout layer with the value 0.3 and 0.2, for GRU and LSTM, respectively. The next two layers are dense, fully connected layers, with rectified linear units (ReLU) activation, followed by the same dropout layer as the respective RNN unit. Lastly, we added a final dense layer with soft-max activation, and a one-hot labeling output of size $|\mathcal{C}| = 5$. The RNN models were trained for 70 epochs, using Adam optimizer, with early stopping.

4.4 Augmentation method

It is possible to train a simple model for an SER task using the existing datasets, and then evaluate its performance when the audio arrives from another room. However,the results presented in Section 3.6.5 suggests that it is highly beneficial to integrate a simulation of the audio environment into the training phase. We are using the set of a $1,000$ generated cross-rooms RIRs, produced by StIM, from Section 3.6.2. The only difference is the set of $k = \{10, 20, 50, 100\}$ values.

4.5 Experimental Results

We start by training all three models without augmentation. We used 20% of the clean data as an evaluation test-set. The performances of all three models are presented in Table 4.1. The three columns represent the three measures with respect to Section 2.1.2. The GRU-RNN model seems to perform slightly better than other architectures with compatible results. Such small differences may be caused by neglectable factors such as weight initialization or similar training parameters, thus, we claim all three model to perform equally.

Table 4.1: Architecture performances on the combined datasets for a 'Clean' trained model. All three models reach compatible performances on the test-set.

Architecture	**BA**	**UA**	**F1**
AlexNet	84.18%	84.44%	84.45%
LSTM-RNN	83.74%	83.04%	83.54%
GRU-RNN	85.94%	85.61%	85.86%

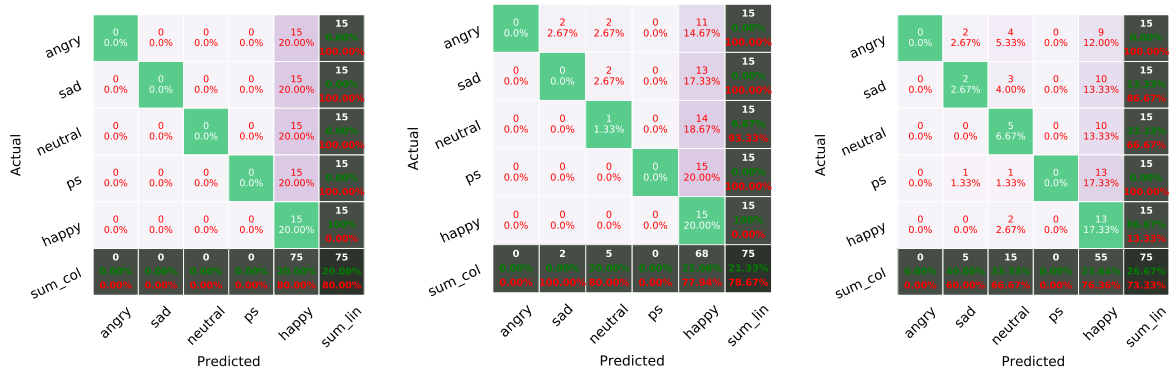
Following the results of 3 on real environments, we follow up with an evaluation of the performances on real data. This real-test set is composed of 75 samples, 15 for each of the 5 labels. The receiver room was of size $(H \times D \times W) = (2.75 \times 1.5 \times 2.5)$ m, and the source room was of size $(H \times D \times W) = (2.75 \times 3.1 \times 2.5)$ m. Both rooms are real adjacent rooms, containing various furniture, and other absorbing and reflecting objects. The rooms are connected with a

common corridor and an adjacent wall. The doors to the corridor where kept close during the recordings. Between recordings, we altered the locations of the source and receiver within their respective rooms. We used two common cellular phone devices for our recordings, which should best represent the data quality that such a system is most likely to encounter in real life. We kept the sampling rate, and used mono-recordings, in MP4 file format.

We proceed to evaluate these models using our real data. The results, presented in Table 4.2, show a significant degradation of performances for all three models. As evident by the confusion matrices, presented in Figure 4.1, all three models predict mostly the same single label for all the test samples, reducing the performances to a level of guesses rather than classifications.

Table 4.2: Architecture performances on the real data for a 'Clean' trained model. All three models' performances are degraded to a level of guessing.

Architecture	BA	UA	F1
AlexNet	20%	20%	6.66%
LSTM-RNN	21.33%	21.33%	9.22%
GRU-RNN	26.66%	26.66%	18.09%



(a) (b) (c)

Figure 4.1: Confusion matrices of classifiers' performance, on real data, when trained without augmentation. (a) AlexNet, (b) RNN with LSTM, and (c) RNN with GRU. The three models all guess a constant label ('happy') in most cases.

These low performances emphasise the necessity of the augmentation method for the cross-room SER task. Therefore, we proceed to train all three models using K augmentation folds. The results, with respect to K, are presented in Figure 4.2.

The performances on the real data of each of the models, trained with augmentation is listed in Table 4.3. The most left column is the architecture, while the second left column is the value of K. The three most right columns are the measures from Section 2.1.2. The confusion matrices for these models can be found in Appendix A. The table implies that all three architectures greatly benefit from integrating the augmentation into the data. It seems that the best results are achieved using $K = 50$ folds of augmentation, and are saturated for higher values of K. In

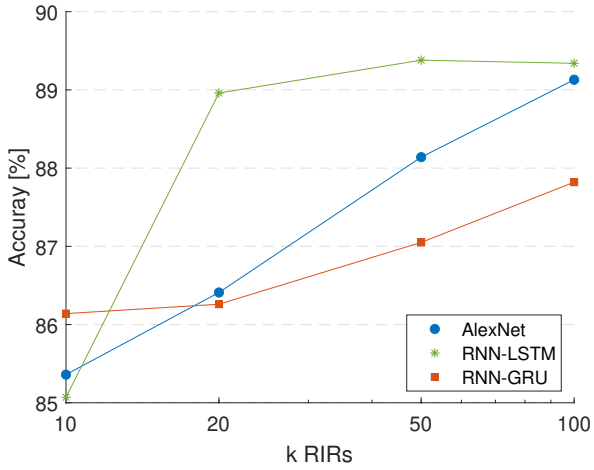


Figure 4.2: Performances on real data with respect to K on the test set. The performances are not saturated, however, they do saturate on the real recorded examples, as demonstrated in Table 4.3.

Chapter 3, we have tested the saturation in K using the test-set, and not the real recordings. Here, we test the saturation on the real recorded environment, and as evident by Figure 4.2 the test-set results have not yet reached a saturation. Empirically setting the level of K with respect to the saturation on the test-set may not be good enough, as the transition to a real environment is problematic. However, evaluating on real recordings requires a bigger dataset with a higher variety. It is an interesting point to learn which method is better. Sadly, we don't have enough real recording to asses this with good statistics. We leave this as an open question for further study.

Table 4.3: Architecture performances on the real data, trained with augmentation. The performance improve, even for a small value of $K = 10$. The performance saturate around $K = 50$.

Architecture	**k**	**BA**	**UA**	**F1**
AlexNet	10	33.33%	33.33%	23.45%
	20	38.66%	38.66%	35.79%
	50	50.66%	50.66%	43%
	100	**49.33%**	**49.33%**	**47.73%**
LSTM-RNN	10	42.66%	42.66%	37.68%
	20	57.33%	57.33%	54.45%
	50	61.33%	61.33%	58.23%
	100	**60%**	**60%**	**58.8%**
GRU-RNN	10	44.44%	44.44%	38.84%
	20	46.66%	46.66%	40.01%
	50	52%	52%	42.74%
	100	**53.33%**	**53.33%**	**50.12%**

It is evident by Table 4.3, that even a small augmentation of $K = 10$ greatly improves the

performances from the real-recorded set. The sequential models show better results over the CNN for all K values, while the best performing model is the LSTM-RNN. This could imply that the sequential characteristics of the data in the SER task, from adjacent room, have more impact on the performances than the inter-frequency features. This makes sense, as reverberations are a product of delaying and summing the original signal. Such a process has a sequential nature, with a varying window of time delay, depending on the room shape. large rooms will have longer delays, while in small rooms, the receiver will experience more frequent reflection incidents. The RNN architecture is able to learn which sequence length is optimal to observe. In the case of SER in a coupled-room audio-scene, the sequence length seem to have an important effect on the overall performances.

Chapter 5

Conclusions

5.1 Research Summary

Current simulation methods for acoustical environment are not capable of describing an audio travelling between adjacent-room, with the required low computational complexity for data-driven methods. In this work, we introduced StIM, a new method capable of simulating an adjacent-room audio environment. We have shown the benefits of using such an augmentation for audio tasks, developed for the purpose of targeting use-cases of coupled rooms.

StIM offers a performance improvement for any DNN architecture aimed at any cross-room audio task. It inherits the computational complexity characteristics of the original IM, with a small overhead due to the additional conditions of adjacent rooms. This low computational complexity stands up to the requirements of generating a large-scale RIR dataset. A generation time of merely several seconds for each example is adequate for development and research purposes. It allows the generation of our 1,000 RIR dataset, in well-under 45 minutes.

Our experiments show that for a cross-room audio task system, it is imperative to use such an augmentation. Without StIM, a model which performed quite well in the design and simulation phase, can suffer a serious performance degradation on a real cross-room audio environment.

The results indicate that data-driven method requires a simulation, that is as close to reality as possible. Current audio environment simulation methods do not describe a real environment accurately enough. They focus on the computational model of wall reflections, and do not account for the randomness of objects and their movement inside a real room. As evident by our results, without the addition of some randomization to the impulse response function, models fail to make the transition into a real environment.

Even StIM is limited in the descriptive capabilities, without the introduction of some randomization. However, the added WGN is not necessarily the best choice for all cases. The type of random process and the level of augmentation are treated as hyper parameters, which could be task-specific or architecture-dependant. There could be another, more analytic way, to choose either of them. This subject remains to be researched.

The fact that DNNs are so dependent on the descriptive capability of a simulation method, reveals the existing methods' weakness in describing a real environment. Other works, such as Ko et al. [69], tried to address this issue in a different approach of a full algorithm, randomly allocating white noise point sources within the room, while we have omitted the use of their full

algorithm, and used instead a simple addition of a WGN vector to address the issue. StIM is not limited in that aspect, and a full implementation of the algorithm suggested in [69] can be integrated into StIM.

Despite the possibility of using either approach, WGN or the algorithm in [69], to deal with the transition to a real environment, some additional solutions may come to mind. For example, a fourth discipline of RIR generators based on the generative power of data-driven methods, rather than their classification capabilities, in some paradigm.

StIM is aimed at DNN methods and was tested using a verity DNN architectures, and two different tasks. DNN has shown promising results for tasks that were formerly un-achievable by classical model-based methods. The overwhelming improvement in a DNN's performance in the use-case of cross-room tasks, due to using StIM, may duplicate to other data-driven methods as well. Furthermore, it can even benefit model-based methods, saving time in the simulation phase, when modeling cross-room audio environments.

The implications of our results open exciting new technological capabilities. For example, non-static robots can now have the ability to capture the emotional state of a target speaker. We can save money on redundant microphones in smart houses and security systems. Even surveillance and alarm systems can benefit from the new possibility of performing tasks from beyond walls.

5.2 Future Research

While working on this thesis a couple of ideas for further research emerged:

A wider exploration of the physical model While StIM improves the results as an augmentation method, there is a contradiction within the assumptions of the supporting model physical. A deeper research and understanding of the physical support may improve the performances even more.

The influence of $w(t)$ In Section 3.6.5 we have mentioned that $w(t)$ could be a colored noise. An open question remains, regarding the nature of $w(t)$ as a random vector. What is the best distribution of $w(t)$? What is the influence of using a colored noise?

Higher order of transmissions between rooms In Section 3.5, we only deal with sound transitioning once between the rooms. While this is a good assumption for most cases, in some cases, like a very loud source, this assumption needs to be re-checked. The question of how to adapt IM and StIM to a higher number of transitions between the rooms, remains open.

Setting up the K parameter In the worst case scenario, setting up K, with respect to the test-set saturation, yields good performances. However, it may be possible that the saturation on the real-recorded examples is good enough. Relying on real recordings will require some examples of real data, which may be harder to obtain than simulated test-set. How to evaluate a lower value of K without relying on the test-set, remains an open question.

DNN as RIR simulators As mentioned in Section 5.1, we suggest a research on a DNN based method, for RIR generation. Such a simulator can relay on a combination of real measured RIR dataset, and IM generated dataset. It can generate either time-domain-represented RIRs or frequency-domain-represented RIRs. Many generative architectures are potential candidates, such as generative-adversarial-networks and variational-auto-encoders. Generative DNNs are known for their ability to generate natural-like signals. The benefits of such a generator is the low computational time. One disadvantage could be a lack of parameter control.

CRNN models and distilling We have evaluated and compared the contribution of both CNN and RNN architectures. A combined CRNN model is a very common DNN model solution for audio tasks, building on the benefits of both CNNs and RNNs. However, such model is relatively big, which translates to longer training time, higher computational complexity in the inference stage, a higher resource requirement from its environment (in a real product), and a better generalization capability. Distilling the knowledge of a neural network [78], could be a nice solution to some of these. Here, we offer to test the performances of a large CRNN model, and then use the result as a teacher to train a smaller, student model. The student model will learn how to generalize from the teacher, rather than learning directly from the dataset alone. Such a small model will require lower computational complexity in the inference stage, and generally lower resources from its environment. Despite the small size of this model, it will still maintain higher generalization capabilities, compared to a model of the same size, trained directly on the dataset. The interesting case here is to study where the simulation integration is required in this process. Should the RIRs be integrated only to the teacher training process? Should the teaching be on the real data?

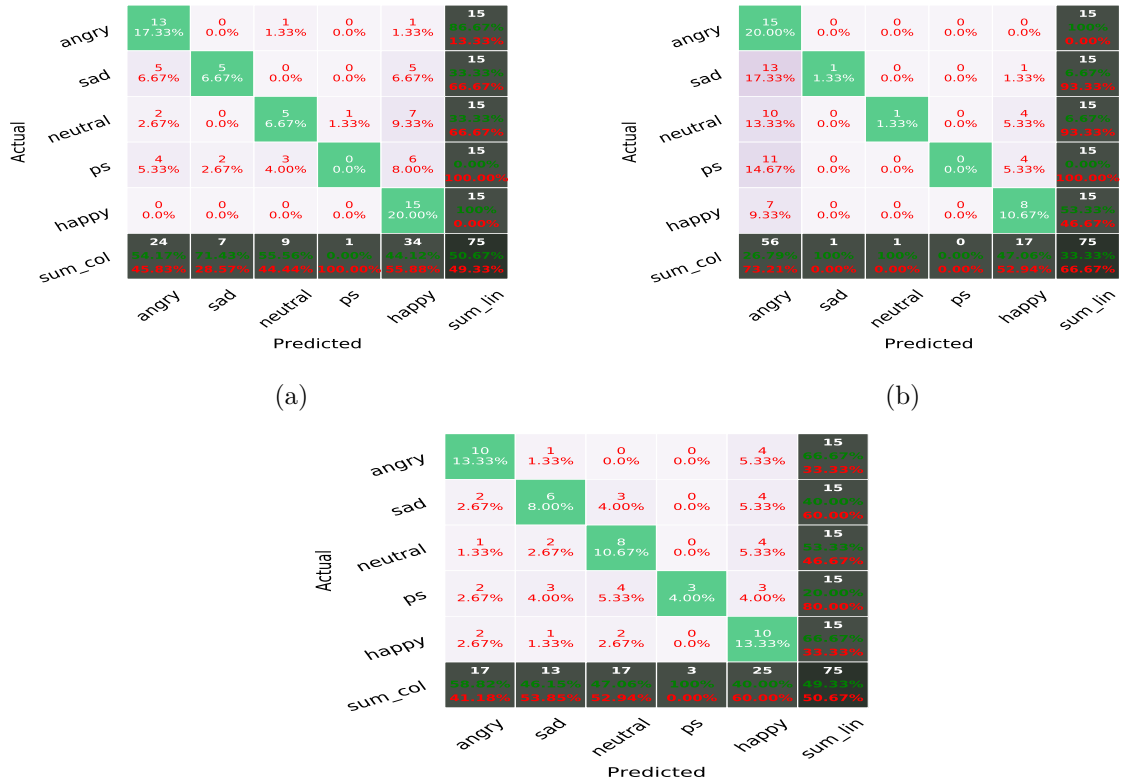
Interlacing SER datasets for aggressive behaviour detection The existing SER datasets are aimed at general emotion recognition. If we target the use case of a binary classification task for aggression detection, we can use a new type of augmentation. In an aggressive interaction, one side's emotional state should be more prominent to anger, while the other side could be prominent to sadness, fear, or anger as well. Interlacing the speakers in a SER dataset into such a description, can improve the performances of an aggressive detection classifier.

Appendix A

Appendix

A.1 AlexNet

Confusion matrix for AlexNet classifier. The confusion matrices represent the AlexNet model performances, as a SER classifier. The performances are evaluated using the real-recorded examples, described in Section 4.2.



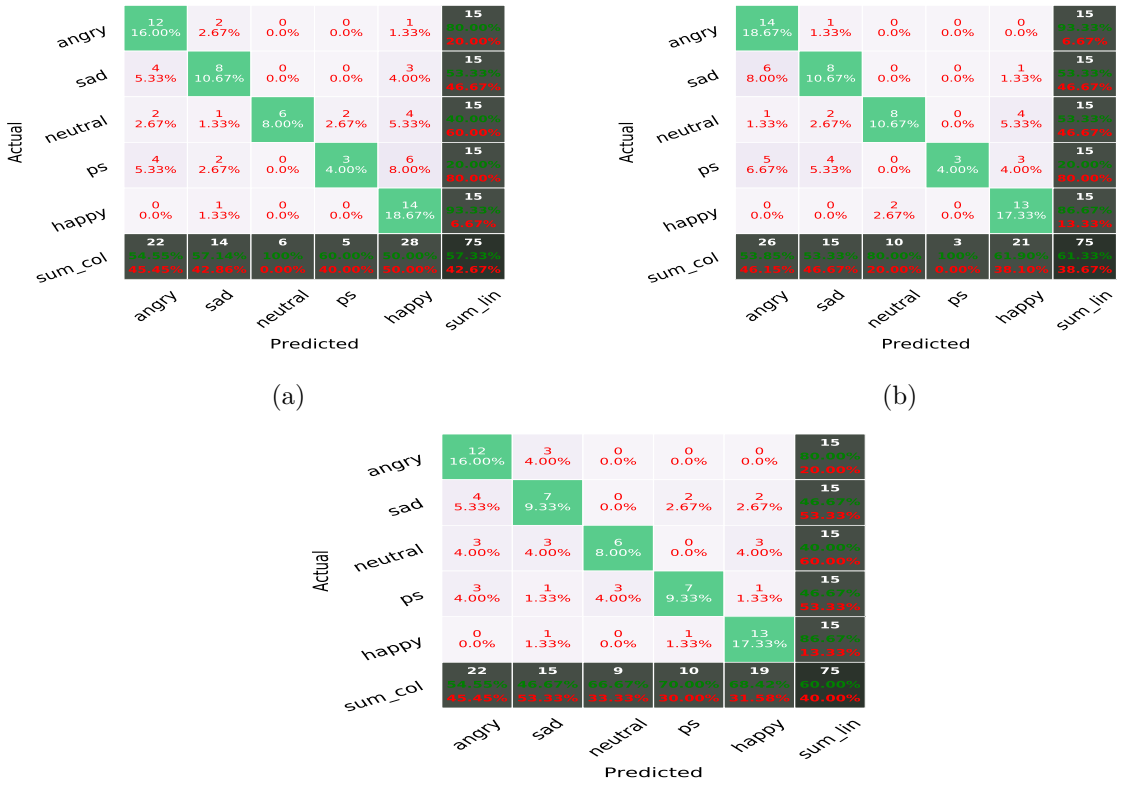
(a) (b)

(c)

Figure A.1: Confusion matrices for AlexNet Classifier. The confusion matrices represents the performance evaluation of real recorded examples for a SER task. (a) AlexNet trained with $k = 10$ folds of augmentation. (b) AlexNet trained with $k = 50$ folds of augmentation. (c) AlexNet trained with $k = 100$ folds of augmentation.

A.2 LSTM-RNN

Confusion matrix for LSTM-RNN classifier. The confusion matrices represent the LSTM-RNN model performances, as a SER classifier. The performances are evaluated using the real-recorded examples, described in Section 4.2.



(a)

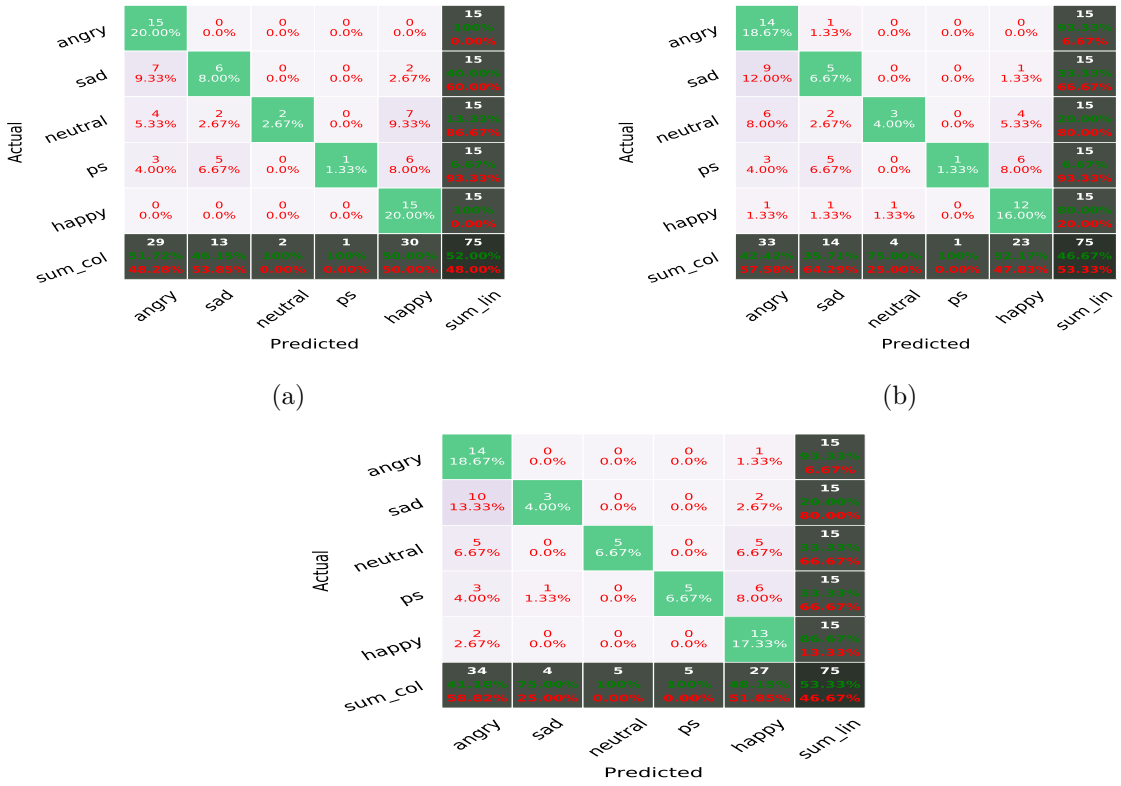
(b)

(c)

Figure A.2: Confusion matrices for LSTM-RNN Classifier. The confusion matrices represents the performance evaluation of real recorded examples for a SER task. (a) LSTM-RNN trained with $k = 10$ folds of augmentation. (b) LSTM-RNN trained with $k = 50$ folds of augmentation. (c) LSTM-RNN trained with $k = 100$ folds of augmentation.

A.3 GRU-RNN

Confusion matrix for GRU-RNN classifier. The confusion matrices represent the GRU-RNN model performances, as a SER classifier. The performances are evaluated using the real-recorded examples, described in Section 4.2.



(a) (b)

(c)

Figure A.3: Confusion matrices for GRU-RNN Classifier. The confusion matrices represents the performance evaluation of real recorded examples for a SER task. (a) GRU-RNN trained with $k = 10$ folds of augmentation. (b) GRU-RNN trained with $k = 50$ folds of augmentation. (c) GRU-RNN trained with $k = 100$ folds of augmentation.

Bibliography

[1] Y. Alsouda, S. Pllana, and A. Kurti, "A machine learning driven IoT solution for noise classification in smart cities," *arXiv preprint arXiv:1809.00238*, 2018.

[2] S. Krstulović, "Audio event recognition in the smart home," *Computational Analysis of Sound Scenes and Events*, pp. 335–371, 2018.

[3] P. Janik and T. Lobos, "Automated classification of power-quality disturbances using SVM and RBF networks," *IEEE Transactions on Power Delivery*, vol. 21, no. 3, pp. 1663–1669, 2006.

[4] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "CLEAR evaluation of acoustic event detection and classification systems," in *Proc. International Evaluation Workshop on Classification of Events, Activities and Relationships*, pp. 311–322, Springer, 2006.

[5] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Proc. 2010 18th European Signal Processing Conference*, pp. 1267–1271, IEEE, 2010.

[6] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543–1551, 2010.

[7] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, *et al.*, "An exemplar-based NMF approach to audio event detection," in *Proc. 2013 IEEE workshop on applications of signal processing to audio and acoustics*, pp. 1–4, IEEE, 2013.

[8] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[9] M. E. Niessen, T. L. Van Kasteren, and A. Merentitis, "Hierarchical modeling using automated sub-clustering for sound event recognition," in *Proc. 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, IEEE, 2013.

[10] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *Proc. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 61–65, IEEE, 2020.

[11] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. 2015 international joint conference on neural networks (IJCNN)*, pp. 1–7, IEEE, 2015.

[12] I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, "Audio event detection using multiple-input convolutional neural network," *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[13] S. Adavanne, G. Parascandolo, P. Pertilä, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," *arXiv preprint arXiv:1706.02293*, 2017.

[14] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," *arXiv preprint arXiv:1905.08546*, 2019.

[15] S. Adavanne, A. Politis, and T. Virtanen, "A Multi-room Reverberant Dataset for Sound Event Localization and Detection," in *Proc. Submitted to Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019.

[16] M. R. Bai, S.-S. Lan, J.-Y. Huang, Y.-C. Hsu, and H.-C. So, "Audio enhancement and intelligent classification of household sound events using a sparsely deployed array," *The Journal of the Acoustical Society of America*, vol. 147, no. 1, pp. 11–24, 2020.

[17] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in DCASE 2019," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[18] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, IEEE, 2017.

[19] P. Smith Jr, "Response and radiation of structural modes excited by sound," *The Journal of the Acoustical Society of America*, vol. 34, no. 5, pp. 640–647, 1962.

[20] R. H. Lyon and G. Maidanik, "Power flow between linearly coupled oscillators," *The journal of the Acoustical Society of America*, vol. 34, no. 5, pp. 623–639, 1962.

[21] A. Craggs, "The use of simple three-dimensional acoustic finite elements for determining the natural modes and frequencies of complex shaped enclosures," *Journal of sound and vibration*, vol. 23, no. 3, pp. 331–339, 1972.

[22] G. Gladwell, "A variational formulation of damped acousto structural vibration problems," *Journal of Sound and vibration*, vol. 4, no. 2, pp. 172–186, 1966.

[23] A. Burton and G. Miller, "The application of integral equation methods to the numerical solution of some exterior boundary-value problems," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 323, no. 1553, pp. 201–210, 1971.

[24] D. Colton and R. Kress, *Integral equation methods in scattering theory*. SIAM, 2013.

[25] T. Walsh, L. Demkowicz, and R. Charles, "Boundary element modeling of the external human auditory system," *The Journal of the Acoustical Society of America*, vol. 115, no. 3, pp. 1033–1043, 2004.

[26] L. Savioja, J. Backman, A. Järvinen, and T. Takala, "Waveguide mesh method for low-frequency simulation of room acoustics," in *Proceedings of the 15th International Conference on Acoustics (ICA-95), Trondheim, Norway*, pp. 637–640, 1995.

[27] A. Krokstad, S. Strom, and S. Sørsdal, "Calculating the acoustical room response by the use of a ray tracing technique," *Journal of Sound and Vibration*, vol. 8, no. 1, pp. 118–125, 1968.

[28] S. Siltanen, T. Lokki, and L. Savioja, "Rays or waves? understanding the strengths and weaknesses of computational room acoustics modeling techniques," in *Proc. Int. Symposium on Room Acoustics*, 2010.

[29] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[30] E. A. Habets, "Room impulse response generator," *Technische Universiteit Eindhoven, Tech. Rep*, vol. 2, no. 2.4, p. 1, 2006.

[31] J. Borish, "Extension of the image model to arbitrary polyhedra," *The Journal of the Acoustical Society of America*, vol. 75, no. 6, pp. 1827–1836, 1984.

[32] M. Vorländer, "Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm," *The Journal of the Acoustical Society of America*, vol. 86, no. 1, pp. 172–178, 1989.

[33] J. H. Rindel, "Modelling the angle-dependent pressure reflection factor," *Applied Acoustics*, vol. 38, no. 2-4, pp. 223–234, 1993.

[34] Y. W. Lam, "Issues for computer modelling of room acoustics in non-concert hall settings," *Acoustical science and technology*, vol. 26, no. 2, pp. 145–155, 2005.

[35] M. Czerwinski, J. Hernandez, and D. McDuff, "Building an AI that Feels: AI systems with emotional intelligence could learn faster and be more helpful," *IEEE Spectrum*, vol. 58, no. 5, pp. 32–38, 2021.

[36] D. Bitouk, R. Verma, and A. Nenkova, "Class-level spectral features for emotion recognition," *Speech communication*, vol. 52, no. 7-8, pp. 613–625, 2010.

[37] N. Sato and Y. Obuchi, "Emotion recognition using mel-frequency cepstral coefficients," *Information and Media Technologies*, vol. 2, no. 3, pp. 835–848, 2007.

[38] K. S. Rao, S. G. Koolagudi, and R. R. Vempada, "Emotion recognition from speech using global and local prosodic features," *International journal of speech technology*, vol. 16, no. 2, pp. 143–160, 2013.

[39] N. Sebe, I. Cohen, T. Gevers, and T. S. Huang, "Emotion recognition based on joint visual and audio cues," in *Proc. 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1, pp. 1136–1139, IEEE, 2006.

[40] C.-H. Wu, J.-C. Lin, and W.-L. Wei, "Two-level hierarchical alignment for semi-coupled HMM-based audiovisual emotion recognition with temporal course," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1880–1895, 2013.

[41] P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. W. Schuller, and S. Zafeiriou, "End-to-end multimodal emotion recognition using deep neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1301–1309, 2017.

[42] J. Kim, "Bimodal emotion recognition using speech and physiological changes," *Robust speech recognition and understanding*, vol. 265, p. 280, 2007.

[43] S. Yoon, S. Byun, and K. Jung, "Multimodal speech emotion recognition using audio and text," in *Proc. 2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 112–118, IEEE, 2018.

[44] B. Schuller, G. Rigoll, and M. Lang, "Hidden Markov model-based speech emotion recognition," in *Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 2, pp. II–1, IEEE, 2003.

[45] O.-W. Kwon, K. Chan, J. Hao, and T.-W. Lee, "Emotion recognition by speech signals," in *Proc. Eighth European Conference on Speech Communication and Technology*, 2003.

[46] C. Busso, S. Lee, and S. Narayanan, "Analysis of emotionally salient aspects of fundamental frequency for emotion detection," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 4, pp. 582–596, 2009.

[47] L.-S. A. Low, N. C. Maddage, M. Lech, L. B. Sheeber, and N. B. Allen, "Detection of clinical depression in adolescents' speech during family interactions," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 574–586, 2010.

[48] P. Shen, Z. Changjun, and X. Chen, "Automatic speech emotion recognition using support vector machine," in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, vol. 2, pp. 621–625, IEEE, 2011.

[49] Z. Peng, Y. Lu, S. Pan, and Y. Liu, "Efficient Speech Emotion Recognition Using Multi-Scale CNN and Attention," in *Proc. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3020–3024, 2021.

[50] S. Zhang, S. Zhang, T. Huang, and W. Gao, "Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching," *IEEE Transactions on Multimedia*, vol. 20, no. 6, pp. 1576–1590, 2017.

[51] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers, and B. Schmauch, "CNN + LSTM architecture for speech emotion recognition with data augmentation," *arXiv preprint arXiv:1802.05630*, 2018.

[52] H. Meng, N. Bianchi-Berthouze, Y. Deng, J. Cheng, and J. P. Cosmas, "Time-delay neural network for continuous emotional dimension prediction from facial expression sequences," *IEEE transactions on cybernetics*, vol. 46, no. 4, pp. 916–929, 2015.

[53] R. Li, Z. Wu, J. Jia, S. Zhao, and H. Meng, "Dilated residual network with multi-head self-attention for speech emotion recognition," in *Proc. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6675–6679, IEEE, 2019.

[54] Y. Gao, J. Liu, L. Wang, and J. Dang, "Domain-Adversarial Autoencoder with Attention Based Feature Level Fusion for Speech Emotion Recognition," in *Proc. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6314–6318, 2021.

[55] A. Shirian and T. Guha, "Compact Graph Architecture for Speech Emotion Recognition," in *Proc. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6284–6288, 2021.

[56] S. T. Rajamani, K. T. Rajamani, A. Mallol-Ragolta, S. Liu, and B. Schuller, "A Novel Attention-Based Gated Recurrent Unit and its Efficacy in Speech Emotion Recognition," in *Proc. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6294–6298, 2021.

[57] S. Kwon *et al.*, "A CNN-assisted enhanced audio signal processing for speech emotion recognition," *Sensors*, vol. 20, no. 1, p. 183, 2020.

[58] B. Boserup, M. McKenney, and A. Elkbuli, "Alarming trends in US domestic violence during the COVID-19 pandemic," *The American Journal of Emergency Medicine*, vol. 38, no. 12, pp. 2753–2755, 2020.

[59] M. D. Griffithsa and M. A. Mamunb, "COVID-19 suicidal behavior among couples and suicide pacts: Case study evidence from press reports,"

[60] I. Rodomagoulakis, N. Kardaris, V. Pitsikalis, E. Mavroudi, A. Katsamanis, A. Tsiami, and P. Maragos, "Multimodal human action recognition in assistive human-robot interaction," in *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2702–2706, IEEE, 2016.

[61] X. Wu, H. Gong, P. Chen, Z. Zhong, and Y. Xu, "Surveillance robot utilizing video and audio information," *Journal of Intelligent and Robotic Systems*, vol. 55, no. 4, pp. 403–421, 2009.

[62] H. Sinha, V. Awasthi, and P. K. Ajmera, "Audio classification using braided convolutional neural networks," *IET Signal Processing*, vol. 14, no. 7, pp. 448–454, 2020.

[63] A. Mesaros, A. Diment, B. Elizalde, T. Heittola, E. Vincent, B. Raj, and T. Virtanen, "Sound event detection in the DCASE 2017 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 992–1006, 2019.

[64] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep neural network baseline for DCASE challenge 2016," *Proceedings of DCASE 2016*, 2016.

[65] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018, ACM Press.

[66] H. Kuttruff, *Room acoustics*. Crc Press, 2016.

[67] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of acoustics*. 1999.

[68] A. Farina, "Advancements in impulse response measurements by sine sweeps," in *Proc. Audio Engineering Society Convention 122*, Audio Engineering Society, 2007.

[69] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224, IEEE, 2017.

[70] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," *Advances in neural information processing systems*, pp. 416–422, 2001.

[71] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[73] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[74] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proc. Ninth European Conference on Speech Communication and Technology*, 2005.

[75] M. K. Pichora-Fuller and K. Dupuis, "Toronto emotional speech set (TESS)," 2020.

[76] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PloS one*, vol. 13, no. 5, p. e0196391, 2018.

[77] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[78] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

(2) שיטת עבודה לשימוש בסימולטורים לסביבה קולית, לצורך אוגמנטציה. אנו מגדירים את האופן שבו יש להשתמש בתוצר של גנרטורים לסביבה אקוסטית, בכדי לבצע אוגמנטציה. אנו מראים כיצד שימוש בשיטת עבודה זו משפר את הביצועים של שיטות מבוססות נתונים עבור משימות סיווג קול עבור כל גנרטור, ועבור הגנרטור שלנו בפרט. (3) הערכת ביצועים, הראשונה מסוגה למיטב ידיעתנו, של משימות של סיווג מקורות קול וזיהוי רגש בדיבור מעבר לקיר. אנו מאמנים מודלים עם ובלי השימוש בסימולטור ובשיטת העבודה שלנו, ומראים כי הסימולטור שלנו מגיע לביצועים טובים בהרבה מהסימולטורים הקיימים, עבור דגימות שהוקלטו בסביבה אמיתית בין חדרים.

תקציר

האם מחשבים יכולים לזהות מהו מקור הקול בחדר אחר. המשימה של זיהוי מקור בתוך מקטע קול נקראת סיווג מקור אודיו, והיא משימה רחבה ומורכבת למכונות בפני עצמה. מערכות מבוססות נתונים הן המערכות המתאימות ביותר למשימה שכזו, אך הן תלויות במידע שמוצג להן בזמן האימון. בכדי לבצע סיווג מעבר לקיר, על המידע לייצג בצורה טובה קול המגיע מעבר לקיר. מהביצועים של מערכת לסיווג קול רגישים מאוד לשינויים בין המידע שהוצג באימון למידע שעליו הן צריכות לפעול, כמו הדים, החזרים וגורמים סביבתיים דומים. לכן מערכת המאומנת בשיטות מקובלות לא תצליח להגיע לביצועים גבוהים, במקרים בהם מקור האודיו לא נמצא באותו חדר של המקלט.

כיום, בכדי להתגבר על הרגישות של מערכות מסוג זה לגורמים סביבתיים, נהוג לעשות שימוש בגנרטורים המייצרים פונקציות תגובה להלם של חדר. בכדי לדמות את סביבת האודיו, דגימת הסאונד עוברת קונבולוציה עם התוצר של הגנרטור. לעומת זאת, לצורך סיווג קול בין חדרים, נידרש לגנרטור המסוגל לדמות מעבר קול בין חדרים. יתרה מזאת, במקרים בהם נדרש מספר רב של דוגמאות, על הגנרטור להיות מהיר מספיק ולייצר את הדוגמאות בזמן סביר. הכלים הקיימים כיום ליצירת תגובה להלם של חדר, לרוב אינם מסוגלים לתאר מעבר קול בין חדרים, ואלו שכן מסוגלים, אינם מסוגלים לבצע את המשימה בקצב מהיר מספיק.

בעבודה זו נציג גנרטור חדש ליצירת תגובה להלם לחדר. הגנרטור שלנו מסוגל לדמות אודיו, הנע בין שני חדרים מצומדים, בסיבוכיות חישובית נמוכה, ובמהירות המספיקה לסימולציה בסדר גודל גבוה. על ידי קונבולוציה של הדגימות עם התגובה להלם, אנו מבצעים אוגמנטציה של הדאטה-סט. ראשית, אנו מגדירים שיטה לאוגמנטציה בעזרת תגובה להלם, ומראים כיצד שיטה זו משפרת את ביצועי המסווגים המשתמשים בגנרטור שלנו, כמו גם בגנרטורים אחרים. לאחר מכן אנו ממשידים להראות שיפור ביצועים של מסווג, על דגימות אמת שהוקלטו מעבר לקיר, בשימוש בגנרטור שלנו בהשוואה לגנרטורים אחרים.

זיהוי רגש בדיבור היא המשימה של זיהוי המצב הרגשי של דובר, מתוך מקטע דיבור. בכדי להראות ששיטת האוגמנטציה שלנו מסוגלת לשפר את הביצועים של כל משימת זיהוי אודיו, ובעבור כל מודל במעבר קול בין חדרים, אנו בוחנים את הביצועים של המערכת הראשונה מסוגה לזיהוי רגש בדיבור בין חדרים. אנו מאמנים שלושה מודלים שונים, בשני סוגי ארכיטקטורות, ומבצעים את שלבי האימון עם ובלי שיטת האוגמנטציה שלנו. אנו מראים שיפור בביצועים עבור שני סוגי הארכיטקטורות. גם כאן, אנו משתמשים בדגימות דיבור אמיתיות שהוקלטו בין חדרים, בכדי להראות שמודלים שאומנו בשימוש בסימולטור שלנו, מגיעים לביצועים גבוהים באופן משמעותי בהשוואה למודלים שאומנו עם גנרטור אחר, או ללא גנרטור כלל.

עבודה זו מציגה שלוש תרומות חדשניות: (1) גנרטור חדש, המסוגל לדמות מעבר קול בין חדרים. גנרטור זה מסוגל לשפר את הביצועים לכל ארכיטקטורה, עבור כל משימת סיווג אודיו בין חדרים. הגנרטור הוא בעל סיבוכיות חישובית נמוכה, המאפשרת סימולציה של מספר גבוה של חדרים במהירות גבוה. סימולטור מסוג כזה מתאים במיוחד לשיטות אלגוריתמיות מבוססות נתונים, הדורשות מספר גבוה של דוגמאות. בנוסף, שיטות אלו תלויות במידע ולכן נדרש גנרטור המסוגל לדמות תנאים הדומים למה שהמערכת תפגוש בזמן פעולתה.

המחקר בוצע בהנחייתו של פרופסור ישראל כהן, בפקולטה להנדסת חשמל ומחשבים.

חלק מן התוצאות בחיבור זה פורסמו כמאמרים מאת המחבר ושותפיו למחקר בכתבי-עת במהלך תקופת המחקר של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

E.Shalev and I.Cohen. Multiroom speech emotion recognition. Submitted to 47th IEEE Internat. Conf. Acoust. Speech Signal Process., ICASSP-2022, Singapore, May 22-27,2022.

Erez Shalev, Israel Cohen, and Dmitri Lvov. Indoors audio classification with structure image method for simulating multi-room acoustics. *The Journal of the Acoustical Society of America*, 150(4):3059–3073, 2021.

תודות

עבודה זו היא אבן דרך משמעותית ומייצגת את סופה של תקופה מלאה בחוויות למידה ואתגרים.

אני רוצה להביע הכרת תודה למספר אנשים שעזרו לי במהלך מחקר זה.

ראשית, אני רוצה להביע את תודתי לפרופסור ישראל כהן, על ההנחיה, ההדרכה, והתמיכה במהלך מחקר זה. לא הייתי מגיע להישג זה בלעדיך, ואני מודה לך על כך.

שנית, למשפחה שלי. לאחותי ואחי ענבל ואייל, על התמיכה, עזרה, ועל הרעיונות שלהם. לאהובתי ובת הזוג שלי, אוריה תורג'מן, שסבלה אותי ברגעים הקשים במסע הארוך הזה. היא תמכה בי לכל אורך התהליך, בקשיי היום יום, ואפילו עזרה בחלק מהעבודה הטכנית של ההקלטות. להורים המדהימים שלי, שרה ועופר. האהבה והתמיכה שלהם הם אבן הראשה עבור הישג זה, כמו גם המעורבות שלהם, והמשוב והערות.

ולבסוף אך לא פחות, תודה לדמיטרי לבוב וחברת DSPG. שיתוף הפעולה עם דמיטרי היה הכרחי והוא תמיד הצליח לספק רעיונות יצירתיים ומבט מרענן על הנושא.

היה לי המזל הגדול לקבל את עזרתכם לאורך דרך זו וכולכם חלק גדול מהישג זה.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

שיטת דימות מבנית לסימולציה אקוסטית בין-חדרית וישומיה

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים בהנדסת חשמל

ארז שלו

הוגש לסנט הטכניון — מכון טכנולוגי לישראל
כסלו התשפ"ב חיפה נובמבר 2021

שיטת דימות מבנית לסימולציה אקוסטית בין-חדרית וישומיה

ארז שלו