MSc thesis in Computer Science

Fusing Bird's Eye View Map Encoding With Simulated Sounds for Generalizable Non-Line-Of-Sight Vehicle Detection

Boriss Bērmans

January 2024

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Boriss Bērmans: *Fusing Bird's Eye View Map Encoding With Simulated Sounds for Generalizable Non-Line-Of-Sight Vehicle Detection* (2024)

The work in this thesis was carried out in the:

Pattern Recognition group
Delft University of Technology

Supervisors: Dr. Hayley Hung
Dr. Julian Kooij
Shiming Wang

Abstract

Detecting nearby vehicles involves utilizing data from various sensors installed on a car as it moves. Common sensors for identifying nearby vehicles include LiDAR, cameras, and RADAR. However, all of these sensors suffer from the same issue – they cannot detect an approaching vehicle that is not yet visible. Hence, this thesis explores the potential of using a microphone array – an array of sensors capable of detecting vehicles that are out of sight. Exploring prior research on detecting obstructed vehicles using sound reveals an existing model capable of detecting nearby vehicles approaching from behind blind corners. However, as the local geometry around the ego vehicle affects the perceived sound patterns, this model was only designed to work within a specific set of T-junctions. Therefore, the thesis aims to take a step further and develop a detection model capable of detecting vehicles behind blind corners in environments not included in the training set of the deployed model. This is challenging for multiple reasons. First, literature review revealed a lack of suitable datasets comprising sounds from approaching vehicles behind blind corners within various road junctions. In addition, microphones, like other sensors, come with limitations. Sound inherently provides less spatial information compared to commonly used sensors in autonomous driving, such as LiDAR or cameras. Considering sound propagation variations in different road junction geometries, building a model adaptable across diverse junction types presents a challenge. To overcome the data scarcity and sound's inherent spatial limitations, the study investigates the potential of employing simulated acoustic responses within artificial road environments as training data for real-world vehicle detection. Simultaneously, to complement the sounds inherent advantage of detecting objects that are out of sight, the thesis proposes to use a Bird's Eye View (BEV) encoding of the top-down map from the driving vehicle's perspective. Having an encoding of the top-down map of the current driving environment would allow a detection model to expect sound signatures commonly observed within a given setting. Overall, the assessment of acoustic simulations could not outline a singular configuration of simulation properties allowing realistic sound propagation for any kind of considered junctions when hearing an approaching vehicle. However, it was observed that the utilization of specific simulation parameters can result in realistic sound propagation within the given junction. Subsequently, evaluating a novel BEV encoding within the newly proposed acoustic detection pipeline demonstrated either equivalent or superior performance compared to a model relying solely on sound. Overall, this research underscores the potential of incorporating BEV encoding in non-line-of-sight acoustic detection and suggests the promise of acoustic simulations within the field. This study contributes to advancing the integration of sound as an additional data modality in vehicle detection.

Preface

The completion of this thesis signifies the end of my master's degree in Computer Science at Delft University of Technology. It encapsulates eight months of work, spanning from May to December of 2023.

Working on the thesis has been incredibly rewarding. I learned a lot about the field of autonomous driving, signal processing, and vehicle localization. Being involved in a community that explores a multitude of captivating topics and endeavors to broaden the scope of existing knowledge has been very exciting. In addition, I definitely strengthened my scientific skills – writing, conducting experiments, and brainstorming about potential research solutions. I am immensely grateful to my supervisors, whose support and invaluable feedback have contributed significantly to my growth and learning. Their consistent availability and willingness to assist made the thesis journey a very enjoyable experience.

Lastly, my heartfelt appreciation goes to my girlfriend, family, and friends. Their unwavering support, whether through encouraging words, willingness to spend time together after a hard day of work, or offering interesting ideas for this thesis, has been invaluable. Their support meant the world to me.

Boriss Bērmans
Delft, January 2024

Contents

List of Figures

List of Tables

1. Introduction

1.1. Problem statement

Human perception allows us to detect presence of objects in our surroundings even without directly seeing them. One of the reasons why this is possible is due to our capability to hear. The ability to detect sounds emerging from occluded sources is especially valuable in circumstances where a source is obstructed by a wall or located behind, providing essential information to avoid potential risks like collisions. Unlike with humans, usage of audio for detecting nearby vehicles in the context of autonomous driving has been limited. Nevertheless, incorporating audio as an additional data source holds great promise for vehicle detection as well for multiple reasons.

As previously illustrated with an example involving humans, perception with sound excels in detecting objects that are out of sight. Naturally, this is an inherent and crucial advantage of sensing with sound compared to common sensors employed in autonomous driving, such as LiDAR or cameras (illustrated with Figure 1.1). Moreover, microphones are less affected by specific lighting or weather conditions like snow or fog, granting an advantage in challenging environments. On the other hand, the audio recorded by microphones typically conveys less spatial information compared to LiDAR or cameras. For instance, data derived from LiDAR point clouds can not only help with establishing the distance to the sound source, but also its shape. This inherently presents a more formidable challenge when attempting to extract such spatial data from audio input.

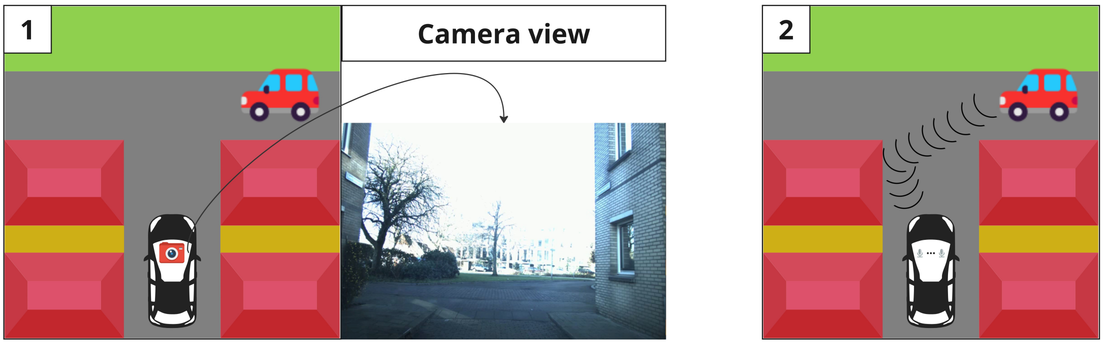


Figure 1.1.: Illustration of an inherent advantage of using microphones for detecting nearby vehicles. With the first part of the figure, the camera fails to see an incoming vehicle. In the second part of the figure, microphones are able to hear an approaching vehicle.

In the domain of acoustic vehicle detection, distinct research trajectories have emerged, delineating two primary approaches: detecting nearby vehicles within direct line of sight and identifying vehicles that are not visible from the perspective of the ego vehicle (i.e., the vehicle containing the sensors that perceive the surrounding environment). In the realm of non-line-of-sight detection, which is the primary focus of this thesis, earlier study (Schulz et al., 2020) proposed a direction of arrival classification model. Namely, the model is able to classify which direction an incoming vehicle is arriving from. The results suggested that utilizing sound grants invaluable reaction time to the driver to recognize that there is an incoming vehicle.

Nevertheless, previous work in non-line-of-sight detection faces notable challenges:

1. **Lack of generalizability to many types of driving environments.**. Previously proposed model in the recent study for this detection type works in a fixed number of outdoor environment types (two types of T-junctions). The model's emphasis on T-junctions is a deliberate design choice, driven by the distinctive variations in sound propagation and reflections encountered as a vehicle approaches different types of intersections like T-intersections, Y-intersections, or others. Hence, the model inherently lacks generalizability across various road junction settings not included in the training dataset, severely limiting its real-world applicability;
2. **Scarcity of available data from diverse driving scenarios for detecting vehicles using sound.** The literature review performed in the thesis revealed that most popular datasets in autonomous driving, such as nuScenes (Caesar et al., 2020), Waymo (Sun et al., 2020), and KITTI (Geiger, Lenz, Stiller, & Urtasun, 2013) do not provide acoustic data, as there were no microphones mounted on a driving ego vehicle while recording the respective datasets. In addition, only three datasets could be found for the purposes of detecting vehicles that are either in-line-of-sight (Chakravarthy et al., 2023; Valverde et al., 2021) or out-of-sight (Schulz et al., 2020). However, upon reviewing the datasets, it became evident that the in-line-of-sight detection datasets either lacked the reported data types as described in the respective papers or had data quality issues. In addition, the available dataset for non-line-of-sight detection included sound recordings limited to specific environmental types. Hence, the scarcity of applicable data poses a constraint on developing a model with the capacity to generalize to scenarios the model was not trained on.

As such, these challenges present exciting research opportunities and shape the research questions of the thesis.

1.2. Research questions

The challenges evident in previous non-line-of-sight detection studies, coupled with the inherent difficulty of extracting spatial details from sound, present several promising research opportunities:

Utilizing simulation software for simulating sound propagation in driving scenarios

The absence of diverse sound data in various driving scenarios hinders the development of a detection model generalizable to different types of road junction settings. In addition, obtaining new data within the autonomous driving domain is both time consuming and expensive. The limited availability of relevant data leads to an intriguing possibility: utilizing simulated acoustic responses within artificial road environments as training data for vehicle detection in the real-world setting. The use of simulations is not new to the autonomous driving domain. For example, there have been developments in simulation software for autonomous vehicle planning (Gulino et al., 2023), and there exists software for imitating real-world driving scenarios, including multi-sensory data perception from the ego vehicle (NVIDIA, 2023). In addition, the use of simulations is also not limited to the autonomous driving domain. For example, Generative Adversarial Networks (GANs) have proven effective in generating simulated images, thereby augmenting training data and enhancing image classification and recognition (Fang, Zhang, Sheng, & Ding, 2018; Frid-Adar, Klang, Amitai, Goldberger, & Greenspan, 2018).

Overall, the research opportunity can be summarized with the following question:

> *To what extent is acoustic simulation software capable of realistically simulating sound propagation in artificial driving settings, and what is the gap between the simulated sound responses and real-world recordings?*

In order to evaluate the simulation software's effectiveness, the thesis compares the acoustic responses between the available real-world dataset (Schulz et al., 2020) and the simulated responses generated within the imitated settings.

Using Bird's Eye View encoding of the surrounding driving environment

Given sound's inherent limitation in providing spatial information compared to other sensors in autonomous driving, utilizing some prior knowledge about the surrounding environment can be one way to deal with the limitation. Modern vehicles often come equipped with GPS that can provide real-time top-down map of the surrounding environment. In addition, modern vehicles can also have front-view cameras, which provide the view in front of the vehicle while it is in motion. Both the map and the front view data can offer valuable cues about the surroundings – the shape of the junction that is being approached, number of nearby walls or vehicles, etc. This thesis explores utilizing one type of the available data – top-down maps, for improving non-line-of-sight vehicle detection.

Utilizing top-down maps while the ego vehicle is in motion can help anticipate sound signatures commonly associated with similar settings. For example, sound propagation and reflection can differ based on whether the vehicle approaches a T-intersection, Y-intersection, or a cross-intersection. Therefore, the thesis explores encoding top-down maps into a compact Bird's Eye View (BEV) representation, aiming to incorporate this information as an additional feature for non-line-of-sight vehicle classification using sound. The subsequent research question becomes:

> *How can a top-down map of the surrounding environment, centered around the ego vehicle, be encoded to be used as an additional feature for the direction of arrival classification?*

Developing a new classification model

Previous research lacks a classification model capable of generalizing to driving environments beyond its training set. This motivates the need to develop a model with broader generalization capabilities. Following the previous two research questions, the remaining question arises:

> *Does training a vehicle detection model on acoustic features and BEV encodings allow it to more accurately classify a direction of arrival of an occluded vehicle in the driving environments not covered by the model's training dataset?*

1.3. Contributions

Having presented the challenges within the relevant research domain and defined the research questions the thesis is approaching, the following are the contributions of this work:

- **New direction-of-arrival classification pipeline** for non-line-of-sight vehicles that incorporates sound simulations and BEV encodings as complementary information to the acoustic features. To the best of our knowledge, this thesis presents the first work in utilization of acoustic simulations for detecting nearby vehicles. Given the limited availability of applicable data, the utilization of simulations in the proposed pipeline marks a crucial advancement in acoustic vehicle detection. In addition, this is the first work that aims to fuse prior knowledge about the surrounding driving environment in the form of the BEV encoding together with acoustic features for non-line-of-sight vehicle detection.
- **Simulator setup.** The proposed pipeline required acoustic simulations to generate training data for direction of arrival classification. However, no existing simulator was suitable for this purpose. Therefore, a key contribution of this thesis is the development of a simulator setup enabling the modeling of sound propagation from an approaching vehicle to the ego vehicle at a road intersection.
- **Compact BEV encoding** to encode the information about the surrounding environment in a simulated setting. Given the environment's influence on sound propagation, this thesis suggests using a specialized BEV encoding of the surrounding environment while the ego vehicle is in motion. Utilizing BEV encoding in the classifier enables generalization across different junction types that exhibit similar encodings to those the model was trained on, owing to the similar propagation of sound in analogous road environments.

These contributions get presented and thoroughly described in Chapter 3. Afterwards, new approaches get tested in Chapter 4.

1.4. Document structure

This document is structured as follows. First, in order to have a better understanding about the rest of the thesis, Chapter 2 presents background information and prior work on traditional sound source localization methods, acoustic vehicle detection and acoustic simulation software. Then, Chapter 3 covers the methods used for answering the previously posed research questions. Consequently, Chapter 4 presents the experimental results from the devised method. Lastly, Chapter 5 reflects on the obtained results, discusses limitations, explores potential avenues for future work, and draws final conclusions based on the research findings.

2. Theoretical background and related work

This chapter describes the research conducted to explore applicable literature and datasets. It initiates by presenting the efficacy of sound in localization using traditional techniques to motivate its role as a data modality in vehicle detection. Subsequently, it offers an overview of studies leveraging audio within the autonomous driving context, as well as motivates the choice of using BEV encoding of the top-down map of the surrounding environment. The chapter proceeds with an overview of the available datasets with road acoustics for vehicle detection. Then, it continues with exploration of non-line-of-sight detection – a pivotal task addressed within this thesis. Lastly, the chapter concludes with an overview of the acoustic simulation software.

2.1. Acoustic source localization

First and foremost, it is essential to outline why sound can be a valuable and informative modality for localizing nearby cars. Localizing objects using sound waves is called *acoustic source localization*. Utilizing sound for object localization has been a well-established practice for a long time. The topophone, one of earlier devices for acoustic source localization, was invented more than a century ago. When being worn by people, it allowed to detect ships in foggy conditions by orienting the device toward a sound source, which amplified the incoming sound (Yangfan Liu and J. Stuart Bolton and Patricia Davies, 2021). Subsequently, acoustic source localization played an important role in the First World War (Voort & Aarts, 2009). Since then, the relevance of acoustic source localization has extended widely, finding applications in the military domain (Baron, Bouley, Muschinowski, Mars, & Nicolas, 2019), automotive domain (Kim et al., 2005), wildlife tracking (Rhinehart, Chronister, Devlin, & Kitzes, 2020), and other areas.

Acoustic source localization can be either active or passive:

- **Active** acoustic location entails emitting sound to create an echo, which is subsequently analyzed to derive the sounding object's location;
- On the other hand, **passive** acoustic location involves detecting sound or vibrations emitted by the sounding object. The sound emitted by the object of interest gets analyzed to determine the position of the object.

This thesis delves into the utilization of microphones for acoustic source localization. Considering that microphones do not emit sound but only capture it, they are classified as passive sensors. Therefore, the thesis focuses on passive sound source localization within the domain of autonomous driving. Leveraging microphones, several traditional approaches exist for detecting sound-emitting objects. Before delving into acoustic source localization in the context of autonomous driving (Section 2.2), this section offers an outline of conventional approaches used for localizing sound-emitting objects with microphones.

2.1.1. Triangulation

Triangulation is the process of determining the location of a point by forming triangles to the point from known points (The MathWorks, 2023b). In the context of acoustic perception, this would imply that knowing (1) the distance between two microphones and (2) the angles between the sounding object to the first and the second sensor allows to calculate the distance d, as shown in Figure 2.1. Namely, by measuring α, β (i.e., angles formed by a baseline between the two microphones and the sounding object), and the distance L between the two microphones, the distance from the baseline d to the sounding object can be expressed by the following formula:

$$d = \frac{L}{\frac{1}{\tan \alpha} + \frac{1}{\tan \beta}} \tag{2.1}$$

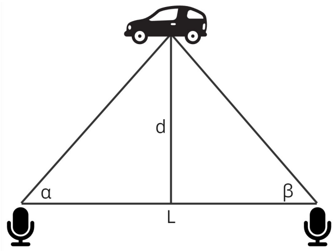


Figure 2.1.: Visualization of a triangle formed between the sensors and the object of interest.

2.1.2. Acoustic beamforming

Acoustic beamforming is a traditional signal processing method that improves the quality of signals originating from specific directions, while reducing unwanted noise and interference from other directions (Johnson & Dudgeon, 1993). This is a very well recognized method for acoustic localization, and several methods build on top of beamforming, such as SRP-PHAT (gets described in Section 2.1.3). This technique can be applied using a single or multiple sensor arrays. When employing microphones, arrays usually consist of omnidirectional (i.e., receiving signals in all directions) microphones, directional microphones, or a combination of both distributed around the perimeter of a space. These microphones are connected to a computing unit that records the results.

One of the simplest beamformers is a delay-and-sum (DAS) beamformer (Johnson & Dudgeon, 1993). The DAS beamformer uses a set of delays and weights to steer the array to different points or directions in a measurement plane (next to the sensor array). The delays are selected to optimize the array's sensitivity specifically to waves propagating from a particular direction. The amplitude weights allow to adjust the individual contributions of each sensor, thus changing the shape of the beam and reducing sidelobes (i.e., local maximas in the beam pattern, different from the beam with the highest power). After applying a weight and a delay to each signal from the microphone array, the DAS beamformer sums the resulting signals. The formula for the beamformer's output in the time domain is as follows:

$$y(t) = \sum_{i=1}^{N} w_i \cdot x_i(t - \Delta_i) \tag{2.2}$$

In this equation, $x_i \cdot (t - \Delta_i)$ represents the delayed signal from the ith sensor at time t after applying the appropriate delay Δ_i, N is the number of sensors, and w_i represents the weight or coefficient applied to the delayed signal before summation.

The resulting beam can be visualized by displaying the spatial sensitivity pattern or the directional response of the array to an incoming signal. As shown in Figure 2.2 (based on the work done by Schulz et al. (2020)), the beam can be visualized and overlaid with the image depicting a sounding vehicle.

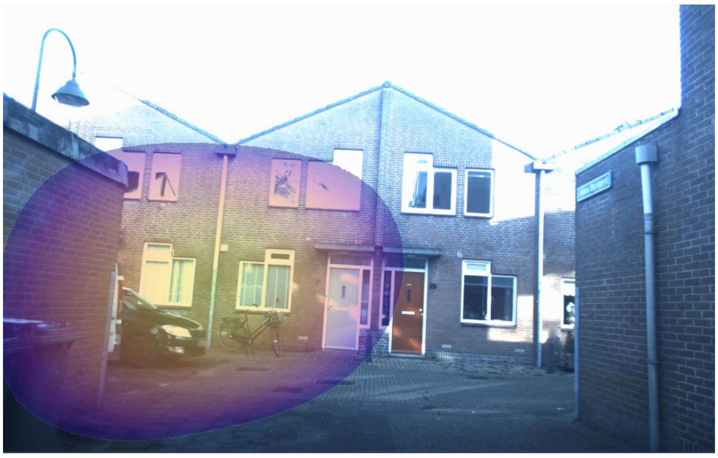


Figure 2.2.: Visualization of a beam over an incoming car. The beam is overlaid on an image captured by the front camera mounted on an ego vehicle.

2.1.3. Direction of arrival with SRP-PHAT

Several methods in acoustic localization handle a more specific task – Direction of Arrival (DoA) estimation. A DoA represents the direction from which a propagating wave arrives at a receiver or the receiver array. One of the more complicated methods for DoA estimation is Steered-response power with phase transform (SRP-PHAT) (Dibiase, 2000). Steered Response Power comprises a range of acoustic source localization algorithms. These algorithms can be understood as utilizing a beamforming-based approach, seeking the position or direction that maximizes the output of a steered delay-and-sum beamformer (Johnson & Dudgeon, 1993). SRP-PHAT is one of the methods within the family of SRP algorithms that uses phase transform (gets defined later in this section), which makes the algorithm more applicable to a variety of different acoustic environments. As SRP-PHAT DoA outputs are the acoustic features employed in the method of this thesis, the following presents an outline of how the algorithm works. Lastly, the section is concluded with a practical explanation of how to interpret the resulting feature vector, as it appears many times later in the thesis.

Outline of the algorithm

Before using the algorithm, a multi-channel audio input captured with multiple microphones is required. Then, each signal needs to be converted into the time-frequency domain by using Short Time Fourier Transform (STFT). The resulting spectrograms get processed by the algorithm as follows. One of the ways to express the SRP is by using the sum of the generalized cross-correlations (GCCs) of all possible microphone pairs $\hat{M} = \binom{M}{2} = M \cdot \frac{(M-1)}{2}$, weighted with a phase transform function $\Psi_{m1,m2}(f)$. This is a multi-step process (Dibiase, 2000):

First, the algorithm defines a generalized cross-correlation (GCC) for the difference between the two *steering delays* (i.e., the delays chosen to steer the array to the source's spatial location) of any two microphones in the array:

$$R_{m1,m2}\left(\Delta_{m1,m2}\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi_{m1,m2}(f) \cdot X_{m1}(f) \cdot X_{m2}^{*}(f) \cdot e^{j \cdot f \cdot \Delta m_1, m_2} df \tag{2.3}$$

- $\Delta_{m1,m2}$ is the difference between the two steering delays, $\Delta_{m1,m2} = \Delta_{m1} - \Delta_{m2}$;
- f denotes the frequency;
- $X_{m1}(f)$ is the spectrogram from the STFT applied to the signal from microphone m_1;
- $X_{m2}^{*}(f)$ is the complex conjugate of the STFT spectrogram applied to the signal from microphone m_2;

The crucial part of the SRP-PHAT approach is to weight the GCCs with the phase transform function. Effectively, the phase transform is used to whiten the cross-spectrum $X_{m1}(f) \cdot X_{m2}^{*}(f)$ (i.e., make it more uniform) between the two microphone signals. Phase transform function is defined as follows:

$$\Psi_{m1,m2}(f) = \frac{1}{|X_{m1}(f) \cdot X_{m2}^{*}(f)|} \tag{2.4}$$

Lastly, steered response power (the objective function of the SRP-PHAT algorithm) can be expressed as a function of the generalized cross-correlation:

$$SRP(\Delta_1 \ldots \Delta_M) = 2\pi \sum_{m_1=1}^{M} \sum_{m_2=1}^{M} R_{m1,m2}\left(\Delta_{m1,m2}\right) \tag{2.5}$$

- $\Delta_1 \ldots \Delta_M$ are the steering delays for each microphone M;
- m_1 and m_2 are two microphones from a set of all possible pairs $\hat{M}$;
- $\Delta_{m1,m2}$ is the difference between the two steering delays, $\Delta_{m1,m2} = \Delta_{m1} - \Delta_{m2}$.

SRP-PHAT algorithm consists in a grid-search procedure that evaluates the objective function $SRP(\Delta_1 \ldots \Delta_M)$ on a grid of candidate source locations. Hence, the output is dependent on the grid it is sampled on. Within the context of the thesis, the grid that is used is the azimuth direction around the microphone array. The range is limited to $[0, \pi]$, centered around the driving direction. For a better interpretation, the output of the algorithm gets visualized in the following section.

Interpretation of the algorithm's output

With the previously outlined algorithm, SRP-PHAT calculates the DoA energy for a fixed number of azimuth angles λ. Within the context of the thesis, these angles are in the range $[0, \pi]$. Effectively, the output of SRP-PHAT contains intensities over λ angles, adopting a polar coordinate representation. With favorable conditions, the peak intensity matches the location or direction of the sound source. For example, in Figure 2.3, the first image indicates that there is a sound source in front of the receiving microphone array from the top-down perspective. The last two images provide a top-down view illustrating a sound source's location relative to the microphone array, indicating whether it is coming from left or right. Importantly, while the peak intensity can point to the left, the sound can actually be originating from the opposite direction (i.e., right), and vice-versa. This can be possible in a scenario when a vehicle is approaching from behind a blind corner, and the sound gets reflected from the wall close to the ego vehicle (as demonstrated with Figure 1.1, Image 2).

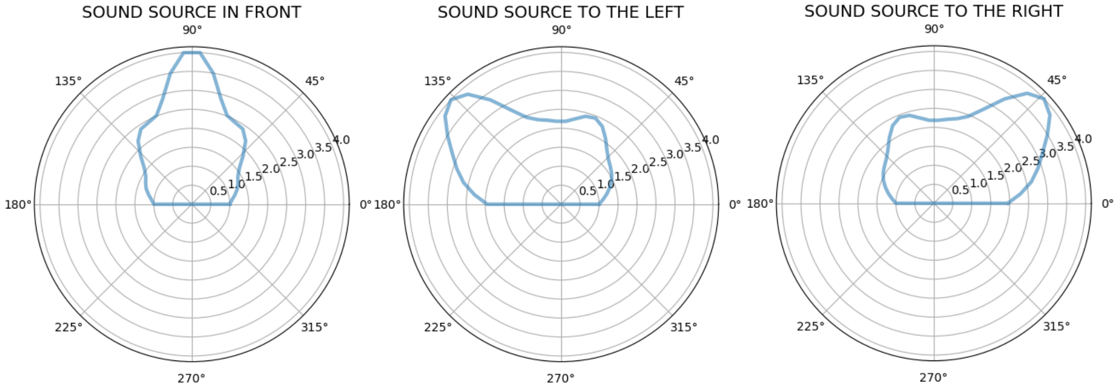


Figure 2.3.: Sample DoA features for three classes. From left to right: DoA intensities for a sound source located in front of the receiving microphone, to the left of it, or to the right of it. The direction can be deduced by identifying where the peak intensity points to.

2.2. Acoustic localization in autonomous driving

Audio is utilized in various research domains within the context of autonomous driving. There is a variety of tasks being addressed (Marchegiani & Fafoutis, 2021):

- **Acoustic object classification**, which involves classifying sounds with types of vehicles and siren/horn detection;

- **Acoustic object localisation**, which involves car localization using audio, and localization of sirens and horns;
- **Surface classification** of the surface type a vehicle is driving on;
- **Self-noise modelling** for anomaly detection of the driving vehicle, amongst others.

Given the project's specific emphasis on acoustic traffic detection, this section provides an overview of relevant works within that field. Namely, the section describes recent works in non-line-of-sight vehicle detection, pertaining to scenarios where approaching vehicles are not visible, as well as works that consider situations when the nearby vehicles are within direct line of sight.

2.2.1. Works in non-line-of-sight acoustic vehicle detection

In the context of non-line-of-sight vehicle detection, Schulz et al. (2020) devise a method to recognize an approaching car's direction at a road intersection when it arrives behind a blind corner. The researchers equipped a vehicle with a roof-mounted microphone array and positioned it at various locations near the intersection of T-junctions in Delft, The Netherlands. Subsequently, they recorded the ambient sounds of the environment, capturing instances with approaching vehicles as well as periods when there were no approaching vehicles at all. The multi-channel audio obtained was subjected to a standard audio processing algorithm to extract features indicating the direction from which an incoming vehicle approached. The processed data served as training input to train a Support Vector Machine, classifying the direction of the incoming vehicle into four categories: $\{left, front, right, none\}$, with *none* class indicating no approaching vehicle. The proposed pipeline was compared to the state-of-the-art visual object detection network *Faster R-CNN*, which can locate incoming cars when they are already visible. Thanks to the proposed model's capability to detect cars with sound, the pipeline achieves the same accuracy more than a second in advance, providing crucial reaction time to the driver. With this research, the authors provided pioneering work in non-line-of-sight vehicle detection using a microphone array.

2.2.2. Works in line-of-sight acoustic vehicle detection

Line-of-sight vehicle detection commonly involves training models on visual inputs like images and LiDAR point clouds, addressing tasks of 2D or 3D object detection. In computer vision, object detection focuses on identifying object instances within images or videos (The MathWorks, 2023c). For 2D object detection, a detection model typically outputs bounding boxes with width, height, x, and y coordinates within the image, along with class confidence scores and labels (Figure 2.4). In 3D object detection, additional parameters like an object's orientation in 3D space and corresponding bounding boxes get estimated as well. Traditionally, object detection primarily processes visual inputs such as images, videos, and LiDAR point clouds. However, there have been studies exploring the use of acoustic features for this task as well.

One of the pioneering works in object detection using sound comes from Gan, Zhao, Chen, Cox, and Torralba (2019). In this work, authors propose a model that allows to uses audio and camera meta-data information alone for detecting instances of a "car" class. Namely, using only acoustic features during inference, the model has the capability to output predictions about the surrounding vehicles on an image, similarly to the outputs depicted in Figure 2.4.

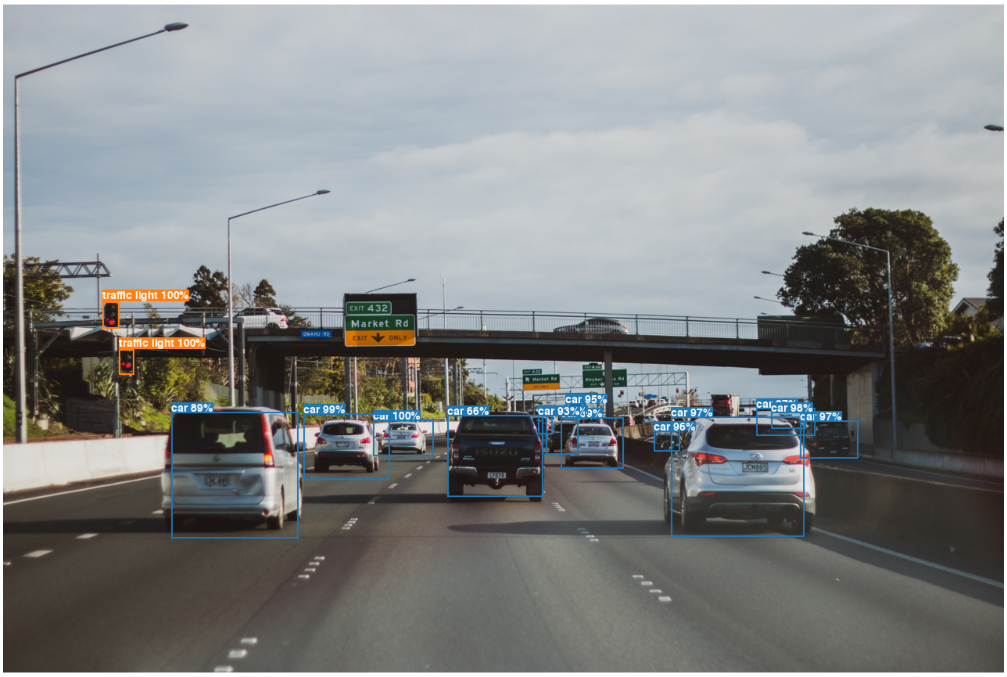


Figure 2.4.: Example of object detection using pre-trained YOLOv3 detection model. Image taken from Ethan Hooson, Unsplash, and processed with the detection model.

This gets achieved with a self-supervised model that uses knowledge distillation technique. The proposed pipeline includes a single vision (RGB) YOLOv2 teacher network to train the audio student subnetwork to regress the bounding boxes of the nearby vehicles by providing bounding box annotations. This study was pioneering in suggesting the potential of knowledge distillation for object detection using sound. The proposed model enabled 2D object detection of vehicles using a data modality that is not conventionally employed for object detection.

Another study conducted by Valverde et al. (2021) tackles the task of 2D object detection by making use of knowledge distillation as well. However, the researchers employ multiple teacher networks – depth, thermal, and RGB – to generate bounding box annotations for the audio student network. Unlike in the aforementioned study (Gan et al., 2019), the researchers do not utilize camera metadata as an input for the audio student network. The utilization of multiple teacher networks and the introduction of a novel loss function, specifically designed to facilitate information distillation from multiple modalities, led to improved performance compared to Gan et al. (2019). This enhancement was evidenced through the experiments described in the paper. In addition, Valverde et al. (2021) introduced a large-scale driving dataset, in which an ego vehicle drove in urban environments and collected multi-modal data. Training the proposed model on this dataset facilitated object detection across images captured in motion, presenting an advancement over Gan et al. (2019), wherein the model was solely trained and tested using data captured from static positions.

One of the recent studies (Chakravarthy et al., 2023) proposes to use long-range acoustic beamforming as a complementary modality to RGB for improved vehicle detection, especially in the scenarios with visible artifacts (e.g., glares), where audio information helps with forming more

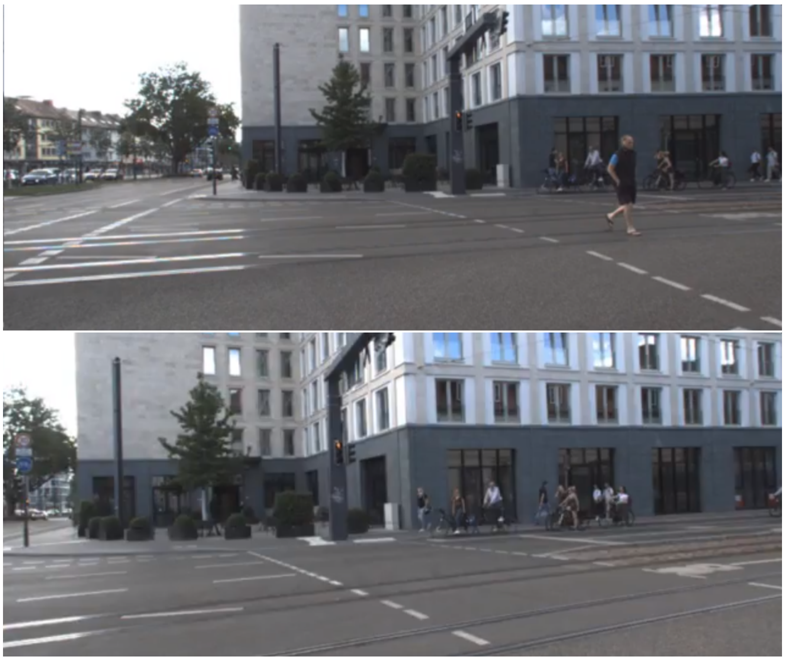


Figure 2.5.: Two different images from the same driving sequence. While being two different image files, these images have a lot of visual overlap.

precise bounding boxes. The authors' approach can either use acoustic beamforming features alone, or in combination with RGB images. Notably, the proposed method surpasses the performance of the work conducted by Gan et al. (2019). Furthermore, unlike the aforementioned studies conducted by Gan et al. (2019); Valverde et al. (2021), the evaluation is carried out on images and sounds of driving environments not included the training dataset, compared to using frames from the same sequences for both train, test and validation sets. This implies that the validation of the trained model is more fair, as images and sounds from the same driving sequence can be very similar to each other. This is demonstrated by Figure 2.5 – while two images come from two different files, their contents are very similar. This observation can also be applicable to overlapping sounds when recorded in identical locations or with minimal time intervals between recordings. The resulting similarity makes the train, test, and validation sets dependent, which makes the model validation results less reliable.

Among the referenced papers, the contributions from Chakravarthy et al. (2023) and Valverde et al. (2021) held significant relevance to the research objectives of the thesis. Hence, in the thesis's early stages, the initial work involved conducting reproducibility experiments and exploring the published datasets from both papers. However, several challenges emerged when attempting to reproduce the findings from the two papers. Chakravarthy et al. (2023) did not disclose the utilized code, making it inherently difficult to reproduce their outcomes. Furthermore, despite the publication of complete experiment code by Valverde et al. (2021), reproducing their results proved unattainable. Additionally, the method outlined in their paper did not generalize to alternative training data. Lastly, the datasets from both studies proved to be different from their descriptions in the respective papers, directing the focus of the thesis towards non-line-of-sight detection instead. The approach for assessing both papers and respective conclusions get described in Appendix B.

2.3. Bird's eye view perception in autonomous driving

Bird's eye view (BEV) perception is an emerging field of study that involves transforming perspective-view inputs into BEV features and performing different perception tasks, such as 3D object detection or semantic map generation of the surroundings in the bird's eye view (Lang et al., 2019; Liu et al., 2022; Wang et al., 2021). There has been an increasing amount of works in the field that encode LiDAR/image information into top-down BEV structure, due to the method's remarkable performance and straightforward interpretability as a unified representation for multiple sensors.

Another alternative view to the BEV can be *a perspective view* – the view that is typically orthogonal to BEV, and which can be retrieved by using a camera. When comparing the two views, BEV representation holds several advantages over the perspective view:

1. Detecting occluded vehicles (e.g., occluded by trees, other vehicles) with BEV representation is easier than in perspective view (Li et al., 2023) – objects in BEV are viewed from above, allowing a clear view of their tops and reducing occlusion caused by nearby objects or obstacles in the perspective view;

2. Perspective view is a subject to the scale problem – there can be many variations in the size of identical objects within the view. For example, the same car can be approaching from either 10 meters or 100 meters. While it is the exact same car, it appears to be much smaller in perspective view when being further away from the camera. On the other hand, BEV provides a consistent scale across the scene, facilitating uniform object size estimation;

3. Cars typically do not ascend or move vertically, given that urban roads generally lack steep inclines. Thus, the cars' movement limited to forward, left, right, and backward directions makes the BEV representation a more suitable and representative choice;

4. The top-down BEV representation can be easily obtained by the ego vehicle equipped with GPS, just like the perspective view can also be obtained on a vehicle equipped with a front-view camera. However, unlike the perspective view, the top-down map offers significantly more insights into the approaching junction. First, it allows to evaluate the relative widths of junction segments. In addition, it allows to see the outline of the road junction being approached (e.g., Y-junction, T-junction, cross-junction, illustrated with Figure 2.6).

Therefore, the aforementioned advantages make BEV applicable for non-line-of-sight vehicle detection scenarios as well. Incorporating data about the current driving environment in the form of a BEV encoding should guide the detection model, conditioned on sound features, to anticipate sound signatures commonly associated with similar road settings.

2.4. Datasets for acoustic vehicle detection

Initially, the goal of the thesis was to use acoustic information for detecting vehicles that are not necessarily out of sight. However, based on available data and initial work done in the thesis (described in Appendix B), the decision was made to confine the research scope. Nonetheless, this section describes the datasets containing acoustic information for both line-of-sight and non-line-of-sight detection.

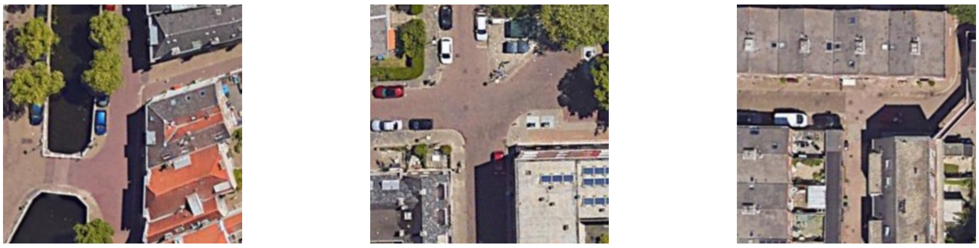


Figure 2.6.: An overview of several top-down junction maps in Delft. From left to right: Y-junction map, cross-junction map, T-junction map. Images retrieved using Google Earth by the author of the thesis.

In order to use acoustic information for detecting nearby vehicles, an extensive audio dataset captured by multiple microphones mounted on an ego vehicle is necessary. When examining some of the most prominent self-driving datasets available, such as nuScenes (Caesar et al., 2020), KITTI (Geiger et al., 2013), and Waymo (Sun et al., 2020), it became evident that they did not contain audio data captured by microphones. Instead, they included other modalities typically used for vehicle detection – LiDAR point clouds, RADAR sweeps, multi-view camera images, as well as sensor calibration and positioning information. Hence, the search involved finding non-conventional datasets in the context of autonomous driving. The search requirements were as follows:

1. As shown in Section 2.1.1, estimating the direction a vehicle is coming from is already possible with two microphones by using triangulation. The same requirement is applicable for beamforming, which requires signals from at least two microphones. Hence, the most important search requirement was finding a dataset for which a microphone array was used to record the sounds of approaching vehicles behind blind corners or vehicles in direct line of sight;

2. Inclusion of RGB images or a video feed synchronized with the microphone array. Having this type of data would allow to visually confirm the presence of the nearby vehicles, facilitate knowledge distillation approach as in Gan et al. (2019); Valverde et al. (2021), and allow for a beamforming overlay to visualize the acoustic localization features (like in Figure 2.2);

3. Inclusion of class annotations for sound samples. These would facilitate supervised learning approaches for detecting instances of multiple different classes (e.g., car, truck, motorcycle). In the context of non-line-of-sight detection, direction-of-arrival annotations (e.g., left, front, right) would enable classification with these classes;

4. Inclusion of sensor calibration and positioning information: necessary for accurate computation of acoustic features with a microphone array, and crucial for determining the relative sensor positions to each other and their placement in the world (i.e., sensor extrinsics).

Overall, it was only possible to find two applicable datasets for line-of-sight detection and one for non-line-of sight detection. Table 2.1 presents an overview of the available datasets.

Authors	Task	Modalities	Dataset annotations	Dataset Description
Valverde et al. (2021)	Line-of-sight detection	RGB, RGB-D, thermal, audio	Bounding boxes, class labels and probabilities for a single class "car"	The multi-modal dataset comprises of two recording types captured via a microphone array mounted on the ego vehicle: during the vehicle's movement and while stationary. In total, it encompasses 114380 entries of synchronized RGB, RGB-D, thermal, and audio frames.
Chakravarthy et al. (2023)	Line-of-sight detection	RGB, LiDAR point clouds, audio	Multi-class image annotations (car, van, pedestrian, bus, amongst others), 11 classes for sounds (small vehicle, horn, emergency vehicles, amongst others)	This dataset, reportedly, comprises of LiDAR point clouds, RGB images and sound recordings from a microphone array, captured in urban Montreal setting. The dataset spans 66 km of urban roads, amounting to 14 TB of storage.
Schulz et al. (2020)	Non-line-of-sight detection	Audio, video	Direction-of-arrival classes for a single incoming vehicle (left, right, front, none)	The dataset provides one-second audio recordings gathered by a vehicle's microphone array, captured during both stationary periods and while in motion. Moreover, the dataset includes video recordings of approaching vehicles.

Table 2.1.: Dataset descriptions

2.5. Non-line-of-sight acoustic vehicle detection

At the time of writing the thesis, Schulz et al. (2020) remains the sole published work on non-line-of-sight vehicle detection utilizing sounds captured by a microphone array. Hence, this thesis extends upon the work done by Schulz et al. (2020), and tries to address the inherent challenges associated with their proposed method. Before delving into the methodology employed in this thesis, it is essential to outline the previous method to gain a comprehensive understanding of the challenges associated with it.

2.5.1. Outline of the method from Schulz et al. (2020)

In this research paper, the authors want to predict if and from where another vehicle is approaching, both when the vehicle is in direct line of sight and when it is behind a blind corner. Namely, the work distinguishes three situations (as stated in the paper):

- an occluded vehicle approaches from behind a corner on the left, and only moves into view last-moment when the ego-vehicle is about to reach the junction;
- same, but a vehicle approaches behind a right corner;
- no vehicle is approaching.

Therefore, the paper proposes to develop a classifier designed to analyze audio samples captured by a microphone array. The primary objective of this classifier is to differentiate among four distinct categories. Namely, whether a vehicle is approaching from left or right, whether it is within the line of sight (implying a frontward direction of arrival), or whether there is no approaching vehicle at all.

When the vehicle is in direct line of sight, the method relies on using a conventional Direction-of-Arrival (DoA) algorithm for localizing sound sources. Namely, the implementation uses Steered-Response Power-Phase Transform (SRP-PHAT) algorithm, which was described in Section 2.1.3. Given continuous synchronized signals, SRP-PHAT computes the DoA energy for any given azimuth angle around the vehicle. Utilizing a DoA feature vector facilitates computation of the azimuth angle associated with the highest DoA energy. Through the establishment of a predefined threshold, the implementation straightforwardly assigns anticipated direction of arrival by comparing whether the angle surpasses or falls below the threshold angle. This process ultimately categorizes the perceived sounds into classes, namely $left, front, right$.

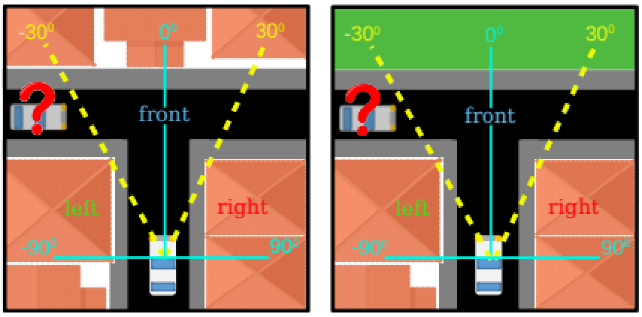
As demonstrated in the paper, direct line-of-sight classification does not pose a significant challenge, as a simple implementation that computes DoA and does threshold checks already performs well. With non-line-of-sight detection, solely using a DoA algorithm becomes insufficient, as salient sound sources produce sound wave reflections on surfaces, such as walls. This implies that, for instance, although the algorithm may indicate that the highest energy is originating from the right, the actual source of the sound can be coming from the left. Hence, the authors propose to use a data-driven approach, in which DoA features get used to train a Support Vector Machine (SVM).

As sound reflection patterns are observably different for different junction geometries, it was proposed to distinguish two types of junctions: "A" (completely walled junction) and "B" (walled exit junction), which get demonstrated with Figure 2.7. With the predefined junction types, the authors recorded a dataset in Delft. Data were collected from five distinct locations, where three of these locations fell under the category of type B, and the remaining two under type A. The recordings got further divided into static data, made while the ego-vehicle was in front of the junction but not moving, and dynamic data, where the ego-vehicle was reaching the junction at $\sim$ 15 km/h. After collecting the dataset, the authors proceeded to train a SVM model, which yielded favorable results in accurately classifying the direction of arrival of out-of-sight vehicles within the specified locations. Overall, the authors presented a pioneering approach, being the first to use passive acoustic perception for non-line-of-sight vehicle detection.

2.5.2. Main challenges of the method

Having identified the main contributions of the paper, the following observation becomes evident. The proposed model is not designed to be generalizable across different environments, primarily because it was exclusively trained on data from the specific locations used in the study. However, considering the myriad variations in real-world junctions and driving scenarios, deploying a model operating exclusively for a specific set of junctions would not be practical in real-world scenarios.

Another observation is that predicting direction of arrival for an obstructed vehicle in diverse driving environments in inherently difficult, attributed to the diverse sound reflection patterns in these environments. This is why the approach only considers two types of T-junctions, in



(a) **Type A**: completely walled (b) **Type B**: walled exit

Figure 2.7.: Schematics of considered environment types. The ego-vehicle approaches the junction from the bottom. Another vehicle might approach behind the left or right blind corner. Dashed lines indicate the camera field of view. Adapted from "Hearing What You Cannot See: Acoustic Vehicle Detection Around Corners" by Schulz et al., 2020, IEEE Robotics and Automation Letters, vol. 6, p. 2587 - 2594. ©2021 IEEE. Adapted with permission.

which the sound propagation patterns are very different compared to each other. Even with the potential to amass a dataset inclusive of extensive sound recordings across diverse junction types and subsequently train a classification model conditioned solely on DoA features for these variants, potential challenges emerge. First, the resulting training dataset would contain numerous DoA features, potentially exhibiting considerable feature overlap. Within the context of a classification task involving only four classes (front, left, right, none), such feature overlap could feasibly diminish the model's accuracy (Almutairi & Janicki, 2020). Moreover, achieving comprehensive coverage of various junction types within the dataset would be exceedingly challenging. This limitation implies that the fundamental issue of the model's inability to generalize to new environments remains unaddressed.

2.5.3. Proposed improvements over the method

In order to address the first challenge with the approach, collecting a more extensive dataset would still be pivotal. However, gathering a new dataset necessary for the task posed requires several time-consuming steps:

1. Getting an ego vehicle with a sensor arrangement;
2. Deciding on new locations with new types of intersections;
3. Getting the vehicle to the locations and recording new data. This is especially challenging, as there are several latent variables that can influence recorded sounds, such as weather conditions, ambient noise from crowds, and construction activities. Recording a new dataset would necessitate ensuring consistency in these factors across different recording sessions;
4. Post-processing the acquired data.

Hence, collecting a new dataset for the posed classification task is a labor-intensive endeavor. However, there is a way to address the challenge – using simulation software for simulating sounds in driving environments. As such, the thesis suggests utilizing acoustic simulations, and aims to assess the representativeness of such simulations for imitating sounds in driving environments. Having a more extensive dataset would enable training a model that possesses greater suitability for application across a broader range of locations, making it more adept for various driving scenarios.

Despite possessing a more extensive dataset, the challenge of potential feature overlap and the model's inability to generalize across even more diverse junction types and geometries persists. Nevertheless, there is a way to address these concerns. A modern vehicle is typically equipped with GPS, so a top-down view of the surrounding environment is readily available while the ego vehicle is moving. Leveraging these top-down maps during vehicle movement may help in anticipating sound signatures commonly associated with similar environments. However, there remains of question of how to use the maps for classification. In terms of representation, top-down maps are essentially arrays of pixel values. Compared to DoA, a pixel array is a much more higher dimensional feature vector. As such, training a classifier on both DoA vectors and the raw top-down maps would considerably diminish the representation of DoA acoustic features. Therefore, the thesis proposes a more condensed BEV encoding in Section 3.4. Then, it aims to determine whether using acoustic features and BEV encodings can improve classification accuracy of direction of arrival of an obstructed vehicle in driving scenarios not included in the model's training dataset. This proposal aims to directly tackle concerns about the model's generalizability to new environments.

2.6. Acoustic simulation software

As the choice of the acoustic simulation software was crucial for the method of the thesis, this section provides an overview of software / packages that were under consideration. Table 2.2 presents the overview of the applicable software.

Overall, the search for solutions in programming languages revealed two packages – one for indoor acoustics simulations and another one for outdoor road acoustics simulations. Search in Matlab packages revealed no software that would allow simulating sound propagation in outdoor scenarios. While some packages for outdoor acoustics were available, they lacked the capability to model rooms with non-rectangular shapes, a crucial necessity for representing road settings accurately. This aligns with the assertion made in the *pyroomacoustics* paper (Scheibler, Bezzam, & Dokmanic, 2018), which stated that a key motivation behind developing *pyroomacoustics* was the absence of Room Impulse Response generators accommodating rooms beyond rectangular shapes. In addition, search for packages in other programming languages yielded no packages for outdoor simulations or packages similar to *pyroomacoustics* that would both allow for configuration of non-rectangular rooms and have a realistic sound simulation model. Lastly, the search for GUI applications revealed two solutions. Notably, it was not possible to investigate a complete feature set of COMSOL (Multiphysics, 1998), as the software is not free. Hence, only limited information could be retrieved about that package.

When comparing the packages, one of the most important factors is the utilized sound propagation model. The models mentioned in Table 2.2 compute Room Impulse Response (RIR). In essence, a RIR is an audio recording of what it would sound like in a given room. The following provides descriptions about the mentioned RIR modelling techniques:

- **Image Source Model (ISM)**. ISM is a geometric simulation method that models specular sound reflection paths between the source and receiver (The MathWorks, 2023a). It assumes that sound travels in straight lines (rays) and undergoes perfect reflections when it encounters an obstacle. ISM has an important adjustable parameter — "order". This parameter signifies the level of sound reflections to be taken into account, with higher values entailing a greater number of mirrored sound reflections in the modeling process (illustrated with Figure 2.8);

- **Ray Tracing (RT)**. This model assumes that sound energy travels around the room in rays. From a sound source, a multitude of rays gets emitted and gets traced until the energy of each ray diminishes below a specified threshold. Modeled as a receiver volume, each microphone collects specular rays intersecting with it, contributing to the resulting RIR (Bezzam, Scheibler, Cadoux, & Gisselbrecht, 2020). The method requires a number of hyperparameters to be fine-tuned: (1) number of rays to be shot from the sound source, (2) receiver volume radius, (3) energy threshold at which rays get stopped, (4) maximum time of flight for rays. In addition, the technique is fundamentally different from ISM, as it allows to model diffuse reflections (visualized with Figure 2.9). This capability allows RT to capture the behavior of sound when reflecting off irregular surfaces (e.g., brick walls, compared to glass that exhibit specular reflections). These diffuse reflections contribute to creating a more realistic simulation of how sound propagates in an environment with more realistically textured walls.

- **Hybrid simulator (ISM + RT)**. The hybrid simulator uses ISM for modelling the early reflections and RT for the diffuse tail. Namely, the algorithm applies ISM of order N for specular reflections. Then, it applies RT for later reflections and late reverberation (Bezzam et al., 2020);

- **Sound-particle tracing (SPPS)** involves simulating movement of particles representing sound waves within an acoustic environment (Picaut & Fortin, 2012b). It traces the paths and interactions of these particles as they propagate and reflect within a room. In contrast to RT where reflection, wall absorption, diffusion, and atmospheric absorption are managed by applying weightings to the sound intensity carried by a ray, the SPPS approach adopts a probabilistic consideration of these physical phenomena. For instance, when encountering a wall with an absorption coefficient, the particle in this approach may have probabilities assigned to being reflected or absorbed. This method introduces a more realistic propagation of sound compared to the traditional RT approach by probabilistically addressing these physical interactions;

- **Simulation using variable length delay lines**. This is a specific simulation technique for modelling sound propagation on a non-urban road and was proposed in *pyroadacoustics* (Damiano & van Waterschoot, 2022). Namely, the model works with a static microphone array M and a single moving sound source S. The microphone array M receives the direct signal originating from S, and the sound that is a specular reflection off the road. In addition, the implementation uses variable length delay lines – signal processing units for introducing a variable delay in a signal path to simulate the time delay caused by the sound traveling a certain distance. The advantage of the approach lies in its consideration of road reflectivity and its capability to model the Doppler effect. This feature is essential for accurately capturing driving acoustics, as it accounts for changes in sound frequency caused by the relative motion between the source and the receiver.

Overall, ISM, RT, Hybrid simulator and SPPS are the simulators listed in ascending order of realism. The remaining simulation approach utilizing variable delay lines offers an advantage by accurately modeling the Doppler effect and considering road reflectivity. However, it lags behind the preceding approaches as it lacks the capability to include multiple sound sources and receivers.

Having outlined the available packages and their characteristics, Section 3.2 in the Method Chapter motivates the final choice of the acoustic simulation software that was employed for creating a necessary simulation pipeline.

Name	Type	Free to use	Supported environment types	Support for custom room shapes	Sound propagation model	Important features
Pyroomacoustics (Scheibler et al., 2018)	Python Package	Yes	Indoor	Yes	Image Source Model (ISM), ISM and Ray Tracing Hybrid	Ability to define multiple sound sources and microphones, support for saving propagated sound (as perceived by the microphones)
Pyroadacoustics (Damiano & van Waterschoot, 2022)	Python Package	Yes	Outdoor	No (i.e., only allows rectangular, shoebox-like rooms)	Simulation using variable length delay lines	Ability to define road materials and moving sound sources, support for Doppler effect
I-SIMPA (Picaut & Fortin, 2012a)	GUI app	Yes	Indoor and outdoor	Yes	Ray Tracing, Sound-particle tracing	Ability to model background noise, ability to define occlusions within the room other than walls
COMSOL (Multiphysics, 1998)	GUI app	No	Indoor and outdoor	Yes	Ray Tracing	Ability to define occlusions within the room other than walls

Table 2.2.: Available software for acoustic simulations.

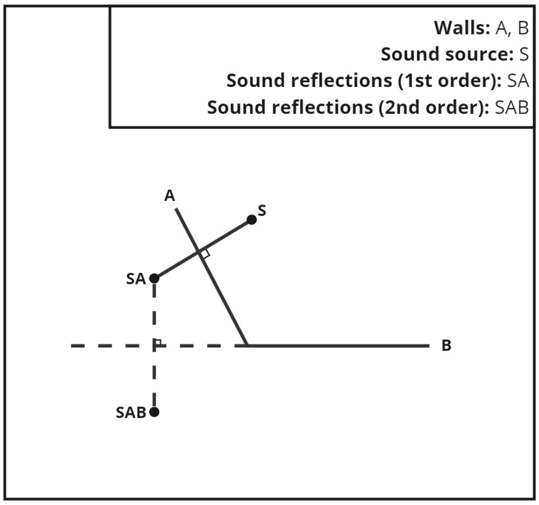

Walls: A, B
Sound source: S
Sound reflections (1st order): SA
Sound reflections (2nd order): SAB
A
S
SA
B
SAB

Figure 2.8.: Illustration of an order parameter in the ISM.

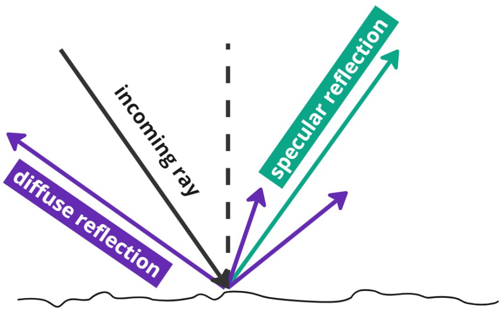


Figure 2.9.: Basic illustration showing diffuse reflections and a specular reflection off an irregular surface.

3. Methodology

Having provided essential background context in the preceding chapter, this chapter introduces the detection task we aim to address. Then, the chapter continues with the motivation regarding the choice of the simulation software designed to generate sounds audible to a microphone array within road junction settings. After motivating the choice of the software, the chapter presents a simulator pipeline that allows to generate sounds in the configured environments. The proposed pipeline enables the simulation of sounds in geometries resembling the shapes of real-world junctions. This facilitates the experiments in Chapter 4 to address the stated research questions. Afterwards, a novel BEV encoding technique gets presented. With both synthetic data generation and the new BEV encoding method established, this chapter introduces a new direction of arrival classification pipeline. Finally, the chapter outlines a method used to simulate junction scenarios from prior research, enabling a direct assessment of the simulations' representativeness compared to real-world settings in Chapter 4.

3.1. Formulation of the detection task

The task that the thesis is tackling is an online classification problem – prediction of a direction-of-arrival class of a single approaching vehicle. As the method of the thesis is an extension of the method devised by Schulz et al. (2020), the following first explains how the previous work approaches the stated classification problem.

3.1.1. Complete pipeline from Schulz et al. (2020)

In summary, the authors propose use DoA acoustic feature vectors computed with SRP-PHAT (defined in Section 2.1.3) for non-line-of-sight acoustic vehicle detection. These DoA features are effective in line-of-sight acoustic sensing, as they allow to determine the direction of the arriving vehicle by determining the peak intensity in the vector. However, solely determining the peak in non-line-of-sight scenario is not sufficient. The sound reflects from various occluders in the scene, bringing ambiguity about the direction of arrival of an approaching vehicle (as shown in Section 2.1.3). Recognizing that a straightforward rule system is not adequate for the task at hand, the authors opt for a data-driven approach instead. Namely, the researchers take the complete DoA feature vectors as features for a SVM classifier. As these features provide a signature of the surrounding sound propagation, the idea is that the classifier would be able to distinguish what kind of DoA patterns correspond to a discrete arrival class.

Figure 3.1 visualizes the approach taken by Schulz et al. (2020). The detection pipeline starts with the microphones that are mounted on an ego vehicle and are actively recording the sounds of the surrounding environment. Then, spectrograms for each of the individual microphone's sound recordings get computed. Afterwards, in order to capture temporal changes in the reflection pattern, these spectrograms get divided into a fixed number of L segments (in the case with the Figure 3.1, the number of segments is equal to two). Consequently, the authors compute DoA vectors from the spectrogram segments which serve as the sole features for training a SVM classifier. The output of the classification is a class probability distribution for the four classes, denoted as $C = \{left, front, right, none\}$. The labels “left”/“right” signify an occluded (not in direct line-of-sight) approaching vehicle from left/right, while “front” indicates that a vehicle is already in direct line-of-sight. The label “none” implies no approaching vehicle.

In the end, the detection task from the previous work can be concisely formulated as follows:

Given DoA vectors γ at train time, output one of the four discrete classes $\{left, front, right, none\}$.

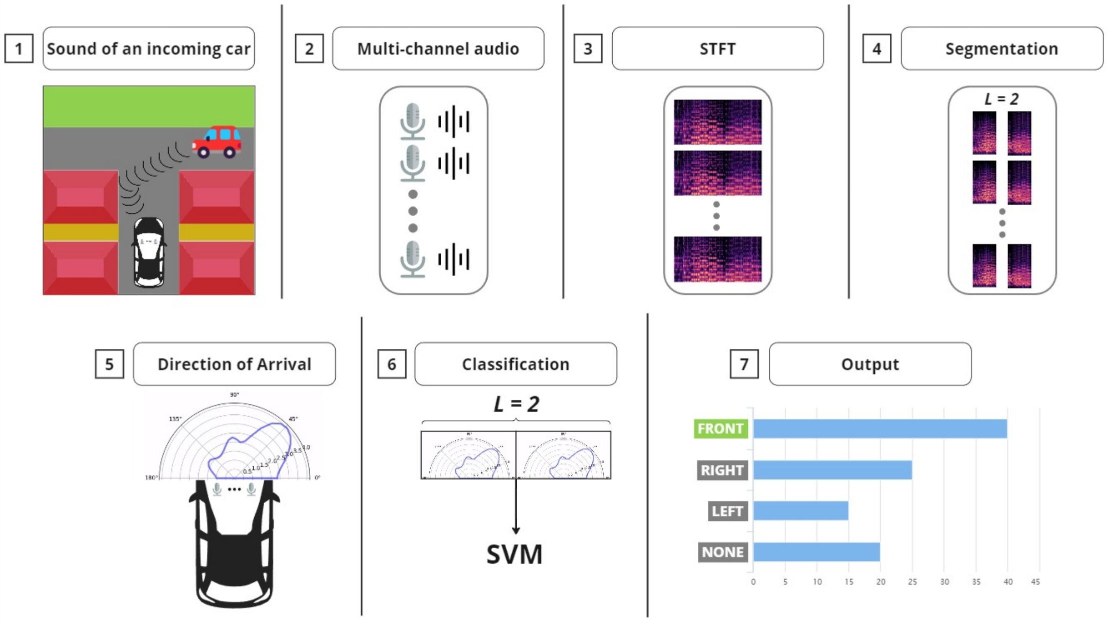


Figure 3.1.: Complete non-line-of-sight detection pipeline from Schulz et al. (2020). Images in the figure depict the following: (1) Microphone array records the sounds of the surrounding environment, (2) Multi-channel sound of an approaching vehicle, (3) STFT computation for each of the microphone signals, (4) Spectrogram segmentation in time dimension into L pieces, (5) Computation of Direction of Arrival, (6) Classification using the concatenated DoA vector, (7) Output of the classification, which is a class probability distribution.

3.1.2. Formulation of the detection task in this thesis

There are several changes to the formulation from previous work in the context of the thesis. Firstly, due to the constraints of the simulation software *pyroomacoustics*, modeling background noise is impossible. This implies that there are insufficient grounds to simulate the “none” class, leading the thesis approach to predict only the remaining three classes. In addition, as the primary goal of the thesis is to train a classifier that is generalizable to the environments not covered by the training dataset (e.g., sounds from a Y-junction when the classifier was only trained on sounds from junctions of different shapes), the input to the classifier is a concatenation of a DoA vector γ and a BEV representation vector β. Hence, the detection task in the thesis becomes as follows:

Given DoA vectors γ and BEV encodings β at train time, output one of the three discrete classes $\{left, front, right\}$.

The design of the BEV encoding gets discussed in Section 3.4, and the complete proposed detection pipeline gets presented in Section 3.5.

3.2. Choice of acoustic simulation software

Following the overview of the available acoustic simulation software in Section 2.6, this section outlines the reasoning behind selecting *pyroomacoustics* for constructing a simulation pipeline. Overall, having concluded the search for acoustic simulation software, the decision was made to opt for *pyroomacoustics*. The remaining packages were not selected for the following reasons:

- **Pyroadacoustics**: Inability to define any occlusions – walls, trees, etc, which is necessary for a proper modelling of the road junction settings;
- **I-SIMPA**: Inability to save the Room Impulse Response (i.e., the sound perceived by the microphone array in the given room), which is crucial subsequent processing with SRP-PHAT or other algorithms;
- **COMSOL**: The simulation package is not free, which can greatly hurt the reproducibility of the experiments. In addition, it was not possible to get the full feature specification of the product.

In contrast, *pyroomacoustics* has the necessary minimal requirements for collecting sounds in simulated junction environments, it allows to: (1) save audio captured by simulated microphone arrays of arbitrary shapes, (2) define room materials impacting sound propagation (3) define non-rectangular rooms. Lastly, it includes beamforming and direction-of-arrival algorithms out of the box. As such, *pyroomacoustics* was utilized for creating a simulator pipeline described in the following section.

3.3. Simulator setup

As one of the goals of the thesis is to use acoustic simulations for simulating sounds in driving scenarios, this section presents a description of the simulator setup.

In summary, the simulator is utilized to create the scenario wherein the ego vehicle approaches the beginning of the junction, along with the obscured sound source that remains out of sight (Figure 3.1, first image). To simulate this particular scenario, the procedure outlined below is executed within *pyroomacoustics* (visualized with Figure 3.2). Each step of the procedure refers to the respective numbered part of the Figure 3.2.

1. Initially, a set of walls is defined by their initial 2D coordinates. For each wall, the definition includes either the specification of its energy absorption and sound scattering coefficients, or the assignment of a material name (e.g., brickwork) that automatically configures these coefficients;
2. Subsequently, the position of the sound source gets specified, along with the emitted sound. For the simulator, a one-second recording of a functioning engine was selected. Following this, a microphone array gets defined within the room settings, described by its position, shape, and the overall number of microphones;
3. After finishing the previous steps, *pyroomacoustics* package simulates the microphone signal at every microphone in the array using a predetermined sound propagation model.

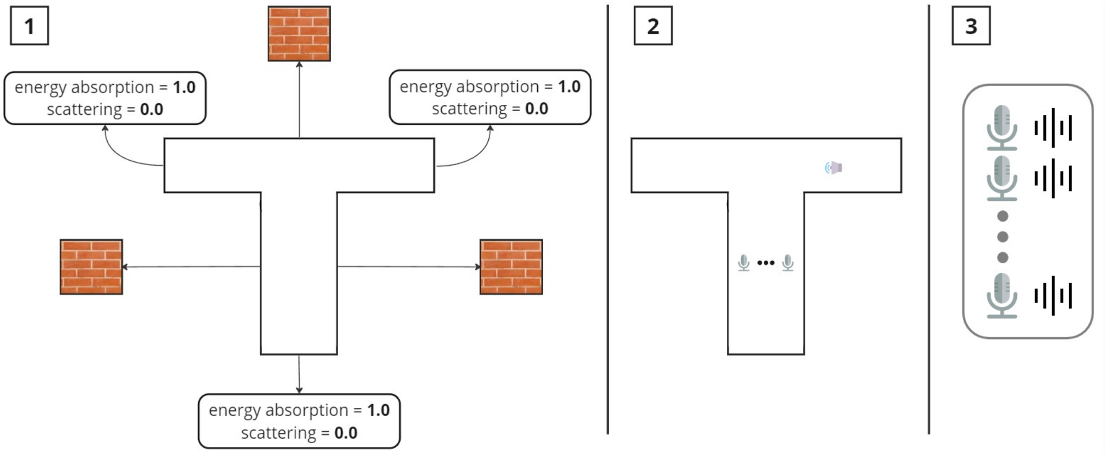


Figure 3.2.: Illustration of the simulator setup. From left to right: first, walls and their material properties get defined. Then, a sound source and a microphone array get created. After finishing the previous steps, it is possible to conduct an acoustic simulation, result of which are simulated sounds for each microphone.

One of the important requirements for the simulator is the ability to model junctions with some walls being absent, presenting an open space in front of the ego vehicle (as shown with some junctions in Figure 3.7). However, this is not possible in *pyroomacoustics* by default, as it requires definition of an enclosed space to run acoustic simulations. To address this, the simulation employs walls at the junction ends configured with sound energy absorption set to one (maximal absorption, which removes reflectivity) and sound scattering set to zero (no diffuse reflections). This configuration mimics sound propagation through open space, which allows modeling non-enclosed junctions within a 2D simulated setting.

3.3.1. Choice of the sound propagation model

Under the hood, in order to compute how sound waves interact with the space, *pyroomacoustics* uses an ISM simulator and a hybrid simulator based on ISM and Ray Tracing (as described in Section 2.6).

During this project, both hybrid and ISM simulators were used. In empirical observations, the DoA features from the sounds generated with a hybrid model exhibited high variability with each run due to the stochastic nature of the approach. This can be illustrated with an example. Consider the environment from Figure 3.4. Here, there is a sound source located to the right of the microphone array positioned at the bottom part of the road junction. Using this environment, it is possible to run acoustic simulations and compute DoA features from the simulated sounds. This can done with either ISM or the Hybrid model. Figure 3.3 presents the computed DoA features from four distinct runs in the aforementioned environment. Namely, the figure illustrates three images of visualized DoA features by using sounds generated with RT simulation model and one image for the sound generated with ISM. Overall, significant variability can be observed while using the RT model. This results from different seed initializations during each simulation round, as no simulation parameters were changed. Importantly, there was no way to fix the random seed, as in *pyroomacoustics* random seeds cannot be controlled for the RT model. In contrast, as ISM is deterministic, each acoustic simulation run within this model results in the same produced sound. Observations from this experiment and other alike tests suggest that using RT approach would add a layer of variability to the simulations for all junction environments, which can potentially hurt explainability of DoA patterns between different junctions. In addition, RT approach requires a greater number of hyperparameters to tune for effective operation, including parameters like the radius of the sphere around the microphone for energy integration, the energy threshold at which rays are terminated, the maximum time of flight for rays, among others. Conversely, ISM only depends on the "order" parameter.

In the end, owing to the simplicity of the ISM model and its interpretability observed in initial experiments, it was concluded that the ISM model would be the preferred choice.

3.4. Description of the proposed BEV encoding

Sound inherently conveys less spatial information compared to other sensors commonly employed in autonomous driving. Consequently, training a DoA classification model based on the features from sounds in multiple environments may not make the model generalizable to the environments not included in the training dataset. In addition, the resulting model may suffer performance issues due to feature overlap (as discussed in Section 2.5.2).

Importantly, sound propagates differently based on the environment – its geometry, number of occluders, wall materials, etc. As such, a geometric prior of the surrounding environment can guide the model to anticipate sound propagation associated with similar settings. As mentioned in Section 2.5.3, a modern vehicle is typically equipped with GPS, so a top-down view of the surrounding environment is readily available while the ego vehicle is moving. These maps provide important prior information about the driving surroundings, as they can depict the junction the vehicle is approaching (Y-junction, T-junction, etc.). However, there remains of question of how to use the maps for classification. In terms of representation, top-down maps are essentially arrays of pixel values. Compared to DoA acoustic features, an array of pixels is a much more higher dimensional feature vector. Hence, one of the research objectives of the thesis involves a design of a compact BEV encoding that will be used in conjunction with DoA features during the training process of the classification model.

As such, this section presents a novel encoding of the surrounding environment, which gets applied in the simulated settings. However, the approach can be generalized to encode a regular top-down map. The approach is heavily inspired by the DoA representation in sound processing. As described in section 2.1.3, the output of running SRP-PHAT on the sounds from

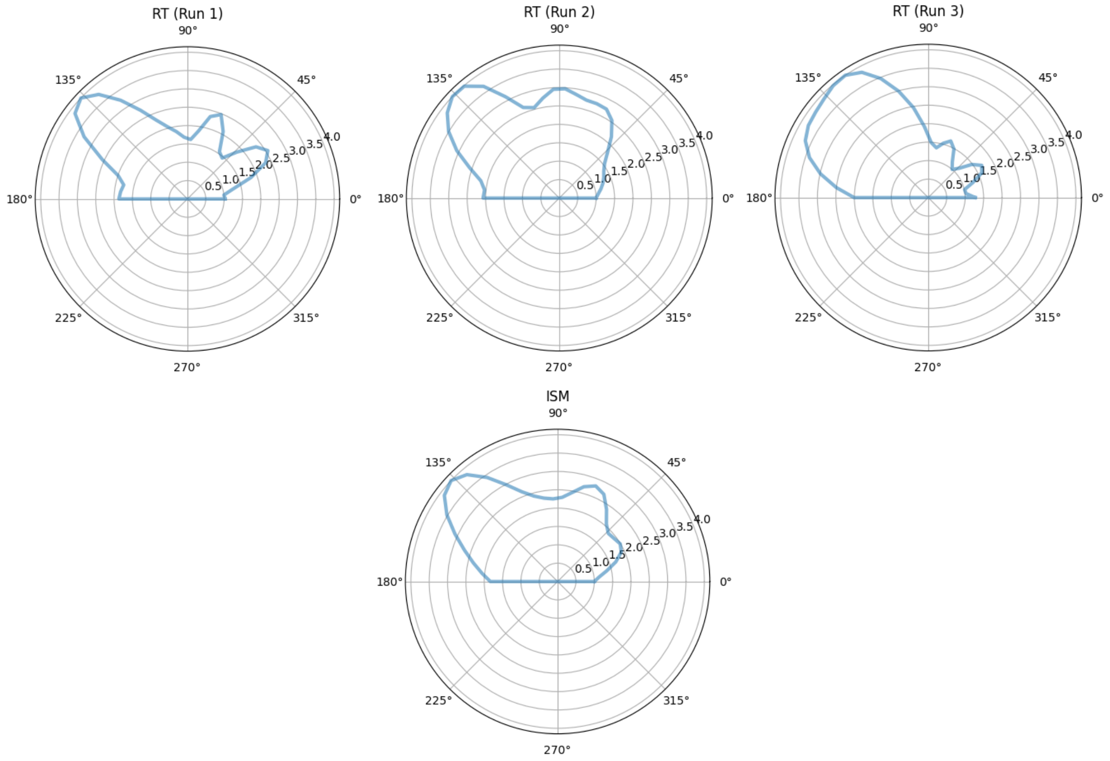


Figure 3.3.: Visualized DoA features for a vehicle approaching from the right behind the blind corner. From left to right: first three pictures show DoA intensities computed with sounds from three distinct simulation runs using RT. The remaining picture shows DoA intensities for any simulation round using ISM, which is deterministic.

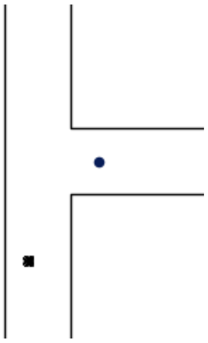


Figure 3.4.: Top-down view of the considered environment type in the simulator. In the image, a microphone array is located at the bottom of the image (black square), and the sound source is positioned behind a blind corner (black dot).

a microphone array is a DoA vector comprising sound intensities for a fixed number of azimuth angles. One could consider devising a BEV encoding represented with polar coordinates as well. Specifically, the approach in this thesis involves compressing the map of the surrounding environment with the ego vehicle situated at the center into a polar representation. This is achieved by emitting rays from the ego vehicle's position and sampling points along these rays. These sampled points are assigned values of either **0** or **1**, depending on whether they fall within the boundaries of the junction's roads. The outcome is a concise fingerprint of the surrounding environment.

Consistent with the aforementioned approach, the following procedure gets used for generating BEV encodings 3.5:

1. Shoot **n** rays of length **l**, uniformly separated between each other with an azimuth angle ϕ;
2. Sample **p** points along the ray, by identifying whether a point is inside the junction (0) or outside (1);
3. The result is $n \times p$ matrix.

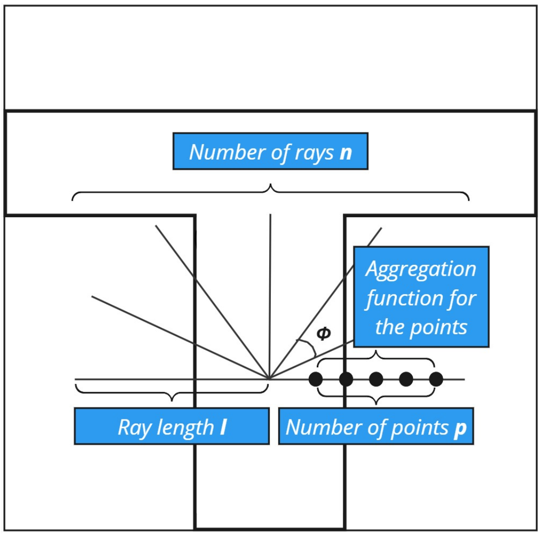


Figure 3.5.: Illustration of the proposed BEV encoding and its parameters.

The resulting matrix is a representation of the environment that encodes information about the nearby walls, encompassing information about the distance between the ego vehicle and these walls or indicating the absence/presence of a wall in particular directions. This encoding can be flattened and concatenated with the DoA acoustic feature, creating a unified feature vector. This vector can be subsequently utilized as training data for a classifier. However, the proposed approach with the encoding takes an additional step by aggregating p points for each of the n rays, resulting in a vector of size n. This approach is advantageous for several reasons:

1. Initially, the resulting matrix possesses significantly higher dimensionality compared to the acoustic DoA feature array. As a consequence, concatenating the acoustic and BEV

features yields a vector that primarily comprises the BEV feature. This dominant presence of the top-down map feature over the acoustic one is undesirable, as most of the resulting concatenated feature would consist of non-acoustic features;

2. In addition, current representation overlooks slight variations in room dimensions. Employing an aggregation function to aggregate sampled point features would consider the variations in room geometry. This adjustment would result in a more generalized encoding, reducing overfitting to specific room geometries.

Consequently, the following aggregation approaches get considered: **(1)** downsample the matrix by averaging p values n times (results in a vector of size **n**), **(2)** downsample the matrix by calculating the first principal component with Principal Component Analysis (PCA) (results in a vector of size **n**). Aggregating with average values addresses the aforementioned concerns by accommodating potential variability in room geometries. Regarding PCA, the first principal component is expected to capture the most variance for each ray, effectively addressing these concerns too.

Importantly, **l, p, n and aggregation function** are hyperparameters that can influence the performance of the model utilizing both DoA features and BEV encodings. Hence, Section 4.3.3 provides a study of these parameters.

3.5. Proposed detection pipeline

Having proposed a new BEV encoding and a simulator setup, this section describes a new detection pipeline – an extension of the method from Schulz et al. (2020) that integrates the newly proposed components. Figure 3.6 presents the proposed detection pipeline:

1. The pipeline begins with defining a road junction environment using the previously described simulator setup (Section 3.3);
2. Using the environment with a microphone array and a sound source removed, a BEV encoding gets constructed using the method from the previous Section 3.4;
3. The acoustic simulation gets executed;
4. Results of the simulation is simulated multi-channel sound (as perceived with each microphone);
5. Following the approach from the previous work, the sounds from all channels get converted into spectrograms with STFT;
6. The spectorograms get used as inputs for the DoA algorithm SRP-PHAT, just as in previous work;
7. The resulting DoA vector gets concatenated with the BEV vector for the respective room and serves as an input to the SVM classifier;
8. Output of the classification is a confidence distribution for the direction of arrival classes $\{left, front, right\}$.

Overall, several differences can be observed with respect to the pipeline proposed by Schulz et al. (2020). First, contrary to the previous method, the STFT spectrograms do not get divided into L segments and get directly processed by SRP-PHAT to compute the DoA vector. The underlying reason for segmenting spectrograms in previous work was to capture temporal changes in the produced sound reflection pattern. However, with *pyroomacoustics*, it is not possible to define moving sound sources, which renders the segmentation approach redundant. In addition, the output of the classifier considers only three classes, due to the inherent limitation of the simulator setup. The simulator does not allow modelling background noise, and as such, class *none* (i.e., no approaching vehicle) does not get predicted.

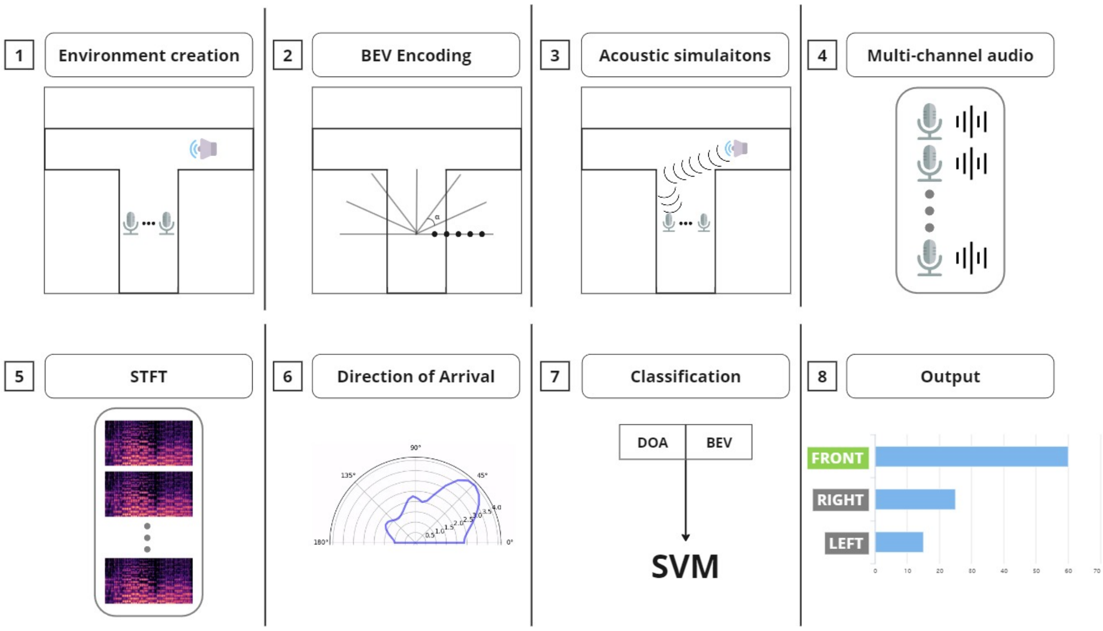


Figure 3.6.: Illustration of the proposed detection pipeline. Images in the figure depict the following: (1) Configuration of the simulated environment, (2) Creation of the BEV encoding of the environment, (3) Running acoustic simulation, (4) Microphone array records the sounds of the surrounding environment, (5) STFT computation for each of the microphone signals, (6) Computation of Direction of Arrival, (7) Classification using the concatenated DoA and BEV vector, (8) Output of the classification, which is a class probability distribution.

3.6. Generation of the simulated dataset

One of the goals of the thesis is to assess the effectiveness of acoustic simulations in simulating sound propagation in a driving scenario. Namely, we want to understand whether the proposed simulator setup allows to produce sound reverberation similar to what can be observed in the real-world settings. In the thesis, we approach this task by simulating the settings from Schulz et al. (2020). In the paper, the authors provide a detailed specification of the environments the dataset was recorded at – ego vehicle distances to the intersections, location coordinates, amongst others. Based on the provided information, it was possible to create

Location name	Location Abbreviation	Coordinates (latitude, longitude)
Anna Boogerd	SA1/DA1	52.01709, 4.3556
Kwekerijstraat	SA2/DA2	52.0087, 4.3530
Willem Dreeslaan	SB1/DB1	51.9812, 4.3670
Vermeerstraat	SB2/DB2	52.0165, 4.3618
Geerboogerd	SB3/DB3	52.0173, 4.3540

Table 3.1.: Overview of the dataset locations.

a dataset of sounds generated within the simulated settings of the paper. Having both the real-world dataset and the corresponding simulated version allowed to conduct an experiment (Section 4.2) that aimed to assess the effectiveness and realism of acoustic simulations. Hence, this section describes how the simulated dataset was generated, laying the groundwork for the subsequent experiment.

The real world dataset at hand consists of audio recorded at the road junctions displayed in Table 3.1. Here, the recordings are divided into static data (S), made while the ego-vehicle was in front of the junction and not moving, and dynamic (D) data when the ego-vehicle was approaching the junction at $\sim$ 15 km/h. The recordings are also divided in two categories of sound patterns (as show in Figure 2.7, type A and B). Then, for each type of junction, there are one second sound samples, where each sample is annotated with one of the four arrival classes $\{left, front, right, none\}$.

As such, the dataset creation involved simulation of the aforementioned settings. However, the limitations of the utilized simulation software limited the extent to which the settings could be simulated. Namely, as the package did not allow to simulate background noise, it was determined to exclude the *none* class and train a classifier for the prediction of three classes. In addition, the package did not have support for moving sound sources, which was necessary to appropriately model the incoming cars, as well as the moving ego-vehicle in *dynamic* setting. Subsequently, the decision was made to omit the *dynamic* junction type and focus on simulating the *static* class.

In order to simulate the junction settings with the devised simulator setup, the simulator setup requires definition of an environment's walls, as well as the creation of a sound source and a microphone array. Overall, five environments were defined with similar dimensions and shapes to the real world junctions (as shown in Figure 3.7). While creating the new room geometries, the following procedure was conducted:

1. Each junction was simulated as a walled room, with each wall having sound energy absorption and sound scattering coefficients assigned;
2. As the simulator only allows to define a room geometry and not a geometry of open space, having no wall at some parts of the junction was simulated as having a wall with sound energy absorption set to **one** and scattering set to **zero** (as described in Section 3.3);
3. Distances between the walls were approximated based on the information from Google Maps and the provided coordinates (Table 3.1).

To faithfully simulate the positions of the sound sources behind blind corners, it was essential to revisit the configuration outlined in the paper. In the paper, for $left$ and *right* classes within the *static* category, 1-second sound samples are extracted using a time window of $[t_0 - 1s, t_0]$.

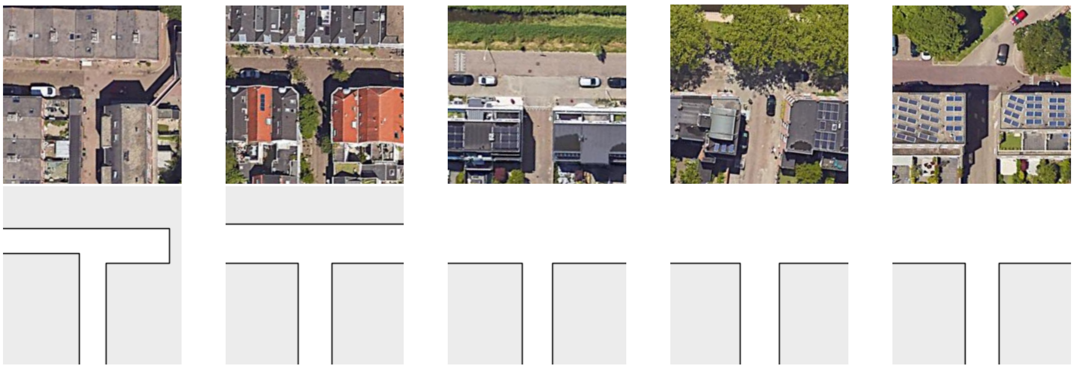


Figure 3.7.: An overview of the top-down junction images and the respective simulated versions. From left to right: SA1, SA2, SB1, SB2, SB3. Real-world images retrieved using Google Earth by the author of the thesis.

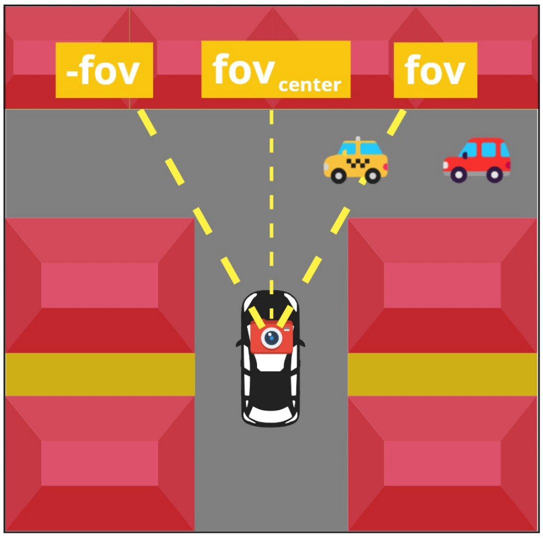


Figure 3.8.: Illustration of the ego vehicle's field of view and its center. The approaching red vehicle is out of sight from the ego vehicle's perspective, and the yellow car is visible.

Here, t_0 denotes the moment when an approaching vehicle enters the direct line of sight (i.e., ego vehicle camera's field of view fov, as demonstrated with Figure 3.8). Then, $front$ class gets extracted 1.5s after the $left/right$ samples. Given that the dataset was recorded in neighborhoods of Delft with a low speed limit, the approaching vehicles were likely moving at ~ 15 km/h speed (roughly 4 m/s). Hence, the sound source positions within the simulations were chosen to be in $[-fov - 4, -fov]$ for the $left$ class and $[fov, fov + 4]$ for the $right$ class. For the front class, static approaching vehicle was placed at fixed positions within the range $[fov_{center} - 2, fov_{center} + 2]$, with fov_{center} being the center of the field of view. On the contrary, generating microphone positions was a simpler process, and in the simulations, a fixed coordinate $[x_{mic}, y_{mic}]$ was employed for this purpose.

In the end, generation of three new samples of three classes followed a specific procedure:

1. Set three unique sound source positions. For $left$ and $right$ classes, set it within the ranges of $[-fov-4, -fov]$ and $[fov, fov+4]$. For $front$ class, define a sound source position in the range of $[fov_{center}-2, fov_{center}+2]$;
2. For each of the sound source positions, define an equal emitting sound (1s sample of a running engine);
3. For each of the sound positions, set an ego vehicle position to the fixed coordinates $[x_{mic}, y_{mic}]$;
4. Run the simulation, process the 3 resulting sound waves with SRP-PHAT algorithm.

In the end, this procedure was used multiple times to generate a complete dataset within each of the simulated intersections from Figure 3.7. The resulting dataset comprised of 96 sound samples (32 for each DoA class) for every type of junction {SimSA1, SimSA2, SimSB1, SimSB2, SimSB3}, as well as the corresponding class annotations. The dataset got further used in one of the experiments (Section 4.2). In addition, for the other experiment, the dataset got expanded by applying the aforementioned procedure for different kinds of junctions (e.g., Y-junction, cross-junction).

4. Experiments

In the preceding chapter, we detailed a simulator setup to generate sounds within simulated road junctions. Alongside this, we put forth a compact BEV encoding and introduced a novel detection pipeline. Now, in this chapter, we delve into experiments that strive to answer the research questions posed earlier in Section 1.2.

The chapter is divided into three experiments. With the first experiment, the goal is to understand the characteristics of DoA features that can be derived from the utilized real-world dataset (Schulz et al., 2020). This sets the expectations for the respective DoAs derived from the sounds in simulated settings. Then, the second experiment aims to assess the effectiveness of using simulations by training a classification model on simulated DoAs and testing on the DoAs from the respective real-world junctions. With the third experiment, the goal is to assess the effectiveness of the proposed BEV encoding for training a generalizable model, effectively addressing the last research question from Section 1.2.

4.1. Experiment 1: Exploring characteristics of the DoA acoustic features derived from the real-world dataset

Before diving into the exploration of DoA features from simulated sounds, this section provides an overview of how these features look like when computed from the real-world sounds. As a reminder, the real-world dataset consists of sound samples recorded in five different locations, with the locations belonging to two different types: *A* and *B* (Figure 2.7) With type *A*, there is a wall in the direction that the ego vehicle is facing, and type *B* there is no wall. Schulz et al. (2020) make an a crucial assumption that the sound propagation is different depending on the junction geometry. As such, this experiment aims to visualize the difference between the DoA features that can be computed from two types of junctions, setting the expectations for the respective simulations.

4.1.1. Experimental setup

To evaluate the assumption of sound propagation dissimilarity in different road junction settings, this experiment visualizes DoAs for sounds in different locations. Furthermore, the DoA features get visualized using the first two principal components of the Principal Component Analysis. Ultimately, the objective is to identify the specific DoA distributions that should be approximated by the simulation software.

4.1.2. Results

Figure 4.1 visualizes two principal components for each junction category (SA and SB). Overall, it is evident that there is a noticeable overlap between classes in both cases, albeit the features of the “front” class exhibit greater separability within each category. As such, DoA distributions from simulated sounds would have to demonstrate similar overlaps. In addition, it is clear that the class distribution from SA looks fundamentally different to the one in SB. These findings serve as the first point of evidence that the sound distributions are different depending on the type of the T-junction.

Next, to have a closer look at how the DoA features look like, Figures 4.2 and 4.3 present DoA features for each category {SA, SB}, location {SA1, SA2, SB1, SB2, SB3}, and class {left, front, right}. The “none” class is excluded from the visualizations, as it is not possible to simulate it with the proposed simulator setup. From the figure, it is evident that there is noticeable variance in DoAs not only between each category but also within each category. For instance, in SB2, DoAs exhibit a distinct intensity peak to the right when a vehicle approaches from the left, and vice versa. In contrast, the distributions in SB1 differ significantly, indicating that other factors in junctions may influence the propagation of sound to the ego-vehicle. These factors may include:

1. Wider roads;
2. Different wall materials, heights;
3. Vegetation;
4. Average distance to the surrounding walls from the approaching vehicle;
5. Presence of other occluders, such as parked cars, other walls;

Taking these factors into account with the BEV encoding could potentially help the subsequent detection model to expect alike DoA distributions for the similar junction settings. Admittedly, the suggested BEV encoding solely captures data regarding the vehicle’s proximity to surrounding walls. Hence, it may only cater to addressing points 1 and 4.

In the end, the experiment shows that there is noticeable variance in DoAs between not only location types, but the distinct locations as well. Based on these findings, it’s implied that the DoA distributions within the simulated settings should exhibit similar properties observed in the real-world data to be representative.

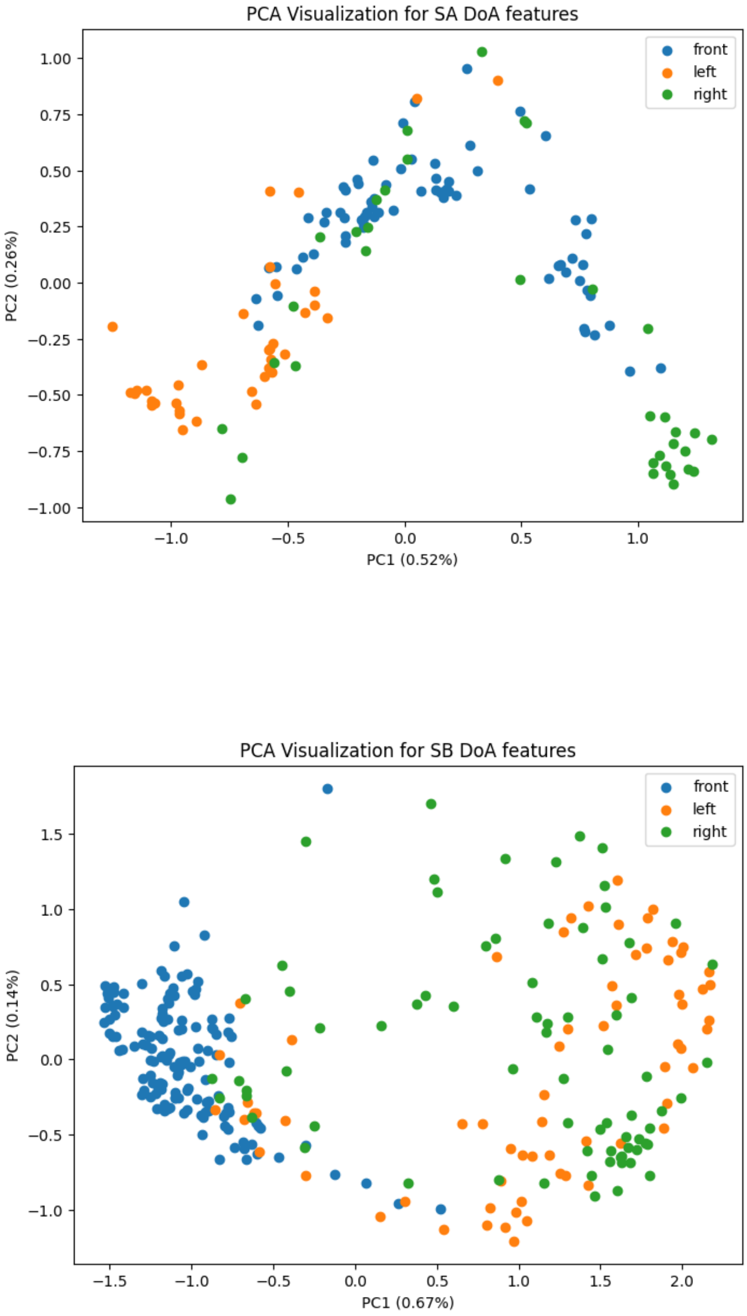


Figure 4.1.: Dimensionality reduction with PCA for all DOA feature vectors of three distinct arrival classes in SA and SB categories.

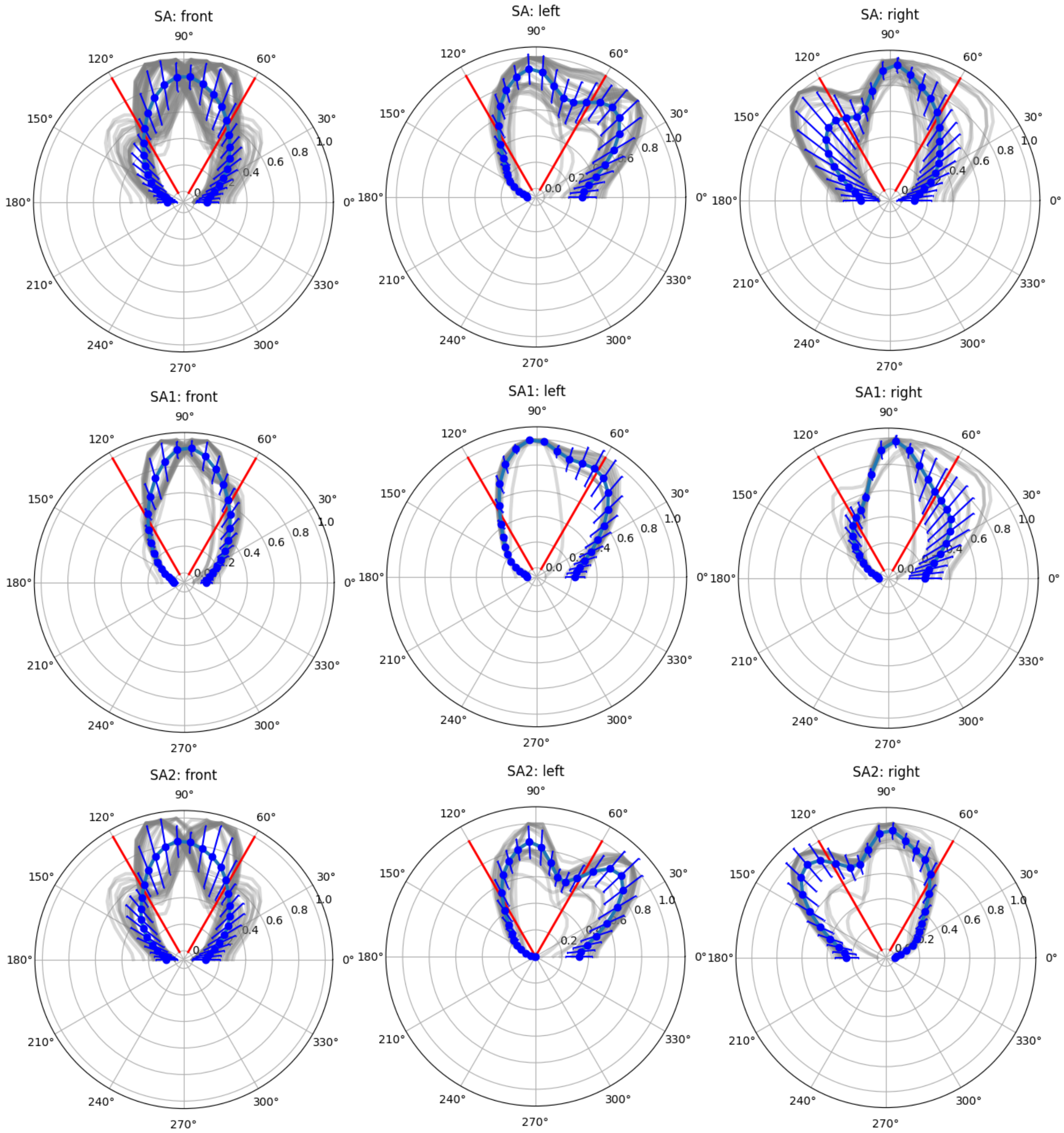
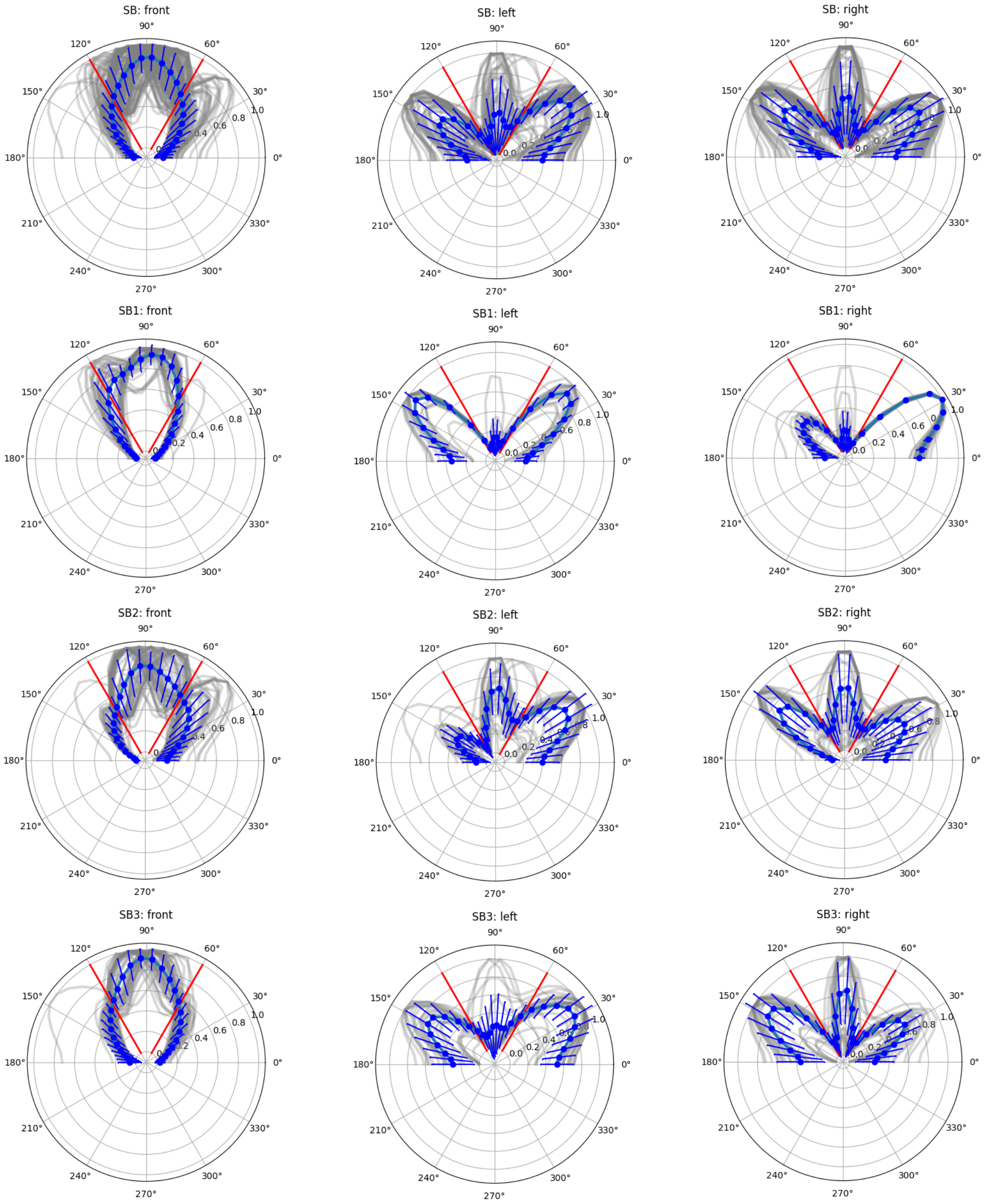


Figure 4.2.: Visualized DoA features for the SA category and locations. The columns represent entries for front, left, and right classes (reading from left to right). The rows represent entries for (1) the whole SA category, (2) SA1, (3) SA2 locations (reading from top to bottom). In each image, red lines represent camera's fov, grey lines DoAs computed for all sounds from a category or location, a blue line shows a mean DoA feature and blue stripes represent standard deviation per azimuth angle.



SB: front

SB: left

SB: right

SB1: front

SB1: left

SB1: right

SB2: front

SB2: left

SB2: right

SB3: front

SB3: left

SB3: right

Figure 4.3.: Visualized DoA features for the SB category and locations. The columns represent entries for front, left, and right classes (reading from left to right). The rows represent entries for (1) the whole SB category, (2) SB1, (3) SB2, (4) SB3 locations (reading from top to bottom). In each image, red lines represent camera's fov, grey lines DoAs computed for all sounds from a category or location, a blue line shows a mean DoA feature and blue stripes represent standard deviation per azimuth angle.

4.2. Experiment 2: Assessing the effectiveness of acoustic simulations in imitating sound propagation observed in the real world

Having outlined the procedure for generating the DoA dataset for simulated locations in Section 3.6, and examined the real-world DoA distributions in T-junctions, the next step is to assess how effectively the simulations imitate real-world observations.

We approach this task by training classification models using DoA from the simulated junctions and testing on the respective real-world junctions. Consequently, the accuracy of the model trained on the simulated data and tested on the respective real-world data serves as a quantitative metric. This metric assesses the effectiveness of utilizing simulations to predict direction of arrival classes in real-world sounds. Hence, it is used as a factor of how representative the simulations are.

In addition, we aim to identify the most important simulator properties that can affect the classification accuracy on the real world data. By outlining these properties, we can pinpoint the specific factors to prioritize when attempting to imitate a real-world junction and compute simulated sounds accurately. The proposed simulator setup has multiple parameters that can affect the resulting sounds (Figure 4.4): (1) width of parts of the junction w, (2) position of the microphone array p_m, (3) positions of a sound source p_s, (4) ISM reflection order, (5) type of the microphone array and the number of microphones. As such, we study how changing a single factor while leaving the other unaffected changes the resulting classification performance.

4.2.1. Experimental setup

The experiment involves training a classifier using simulated data and subsequently testing it on DoAs computed from the real data in multiple rounds. Each round involves altering only one of the previously mentioned simulator parameters at a time. In addition, for a single round, a model only gets trained on DoAs from a single simulated junction (e.g., SimSB1), and tested on the respective real-world junction (e.g., SB1). The trained classifier possesses the following properties:

- **Model type:** SVM with RBF kernel, no regularization. Choice of the classifier is motivated by previous work (Schulz et al., 2020);
- **Training data:** normalized DoAs from sound sources in a single simulated junction, which can be SimSA1, SimSA2, SimSB1, SimSB2, or SimSB3;
- **Test data:** normalized DoAs in the corresponding real-world junction, which can be SA1, SA2, SB1, SB2, or SB3.

The accuracy of the model is utilized as a quantitative indicator of the effectiveness of the simulations. Given the deterministic nature of SVM, to enhance robustness without altering the random seed, the training set is varied ten times using 10-Fold cross-validation. The accuracies from each fold are then averaged. Finally, for a qualitative assessment of the model, DoAs for the simulations with the most optimal properties are visualized and directly compared to the DoAs computed from the real-world sounds.

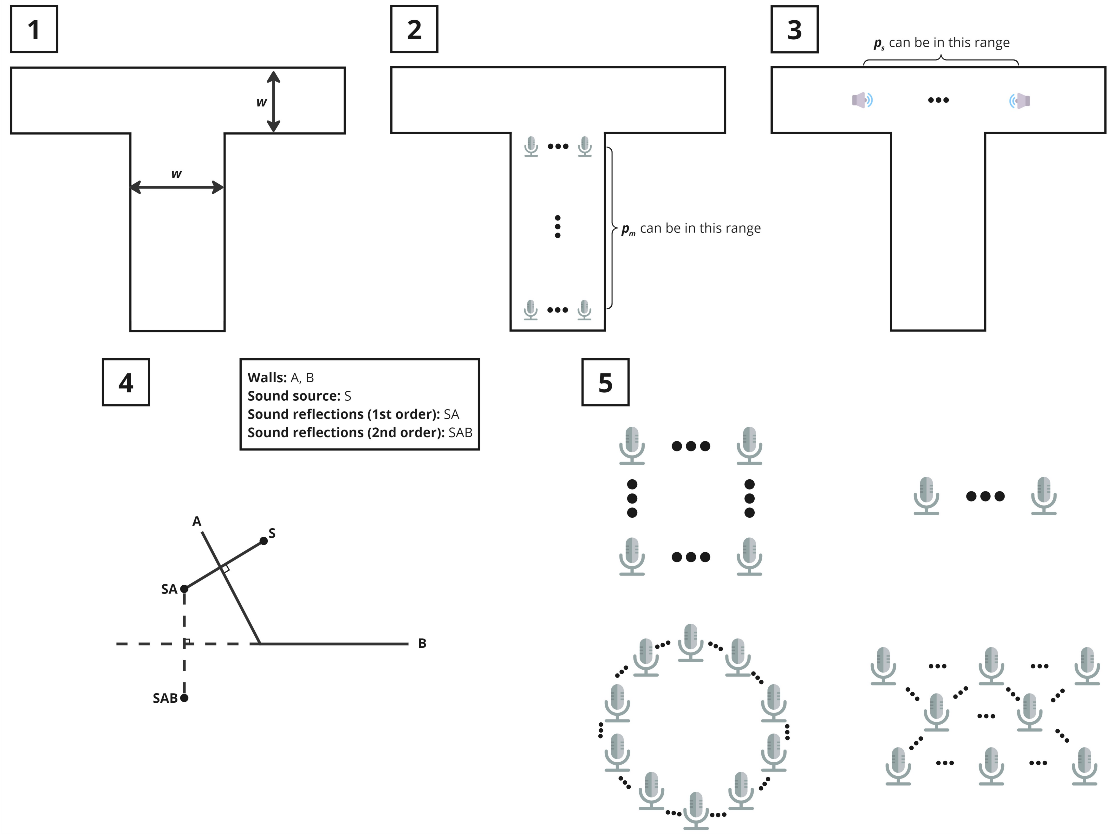

1
w
w
2
p_m can be in this range
3
p_s can be in this range
4
Walls: A, B
Sound source: S
Sound reflections (1st order): SA
Sound reflections (2nd order): SAB
A
S
SA
B
SAB
5

Figure 4.4.: Parameters within the simulator that can influence the sound perceived by the microphone array. Parts of the figure represent the following parameters: (1) width of parts of the junction w, (2) position of the microphone array p_m, (3) positions of sound sources p_s, (4) ISM reflection order, (5) type of the microphone array and the number of microphones.

4.2.2. Results

Having identified the properties of DoA distributions of the real-world dataset, simulations can be used and compared to the real-world responses. Given the multitude of parameters that can impact sound within the chosen simulation software, the subsequent Figure 4.5 and Tables 4.2, 4.1, 4.3, 4.4 present the outcomes of training a model using the simulated data and subsequently testing it on corresponding real-world DoA features.

Overall, it's noticeable that the model's performance exhibits significant variation based on the chosen simulated location. In some cases, it achieves a perfect accuracy of 1.00 on the real-world junctions' test data (as shown in Table 4.1), while in other instances, it performs poorly (as indicated in Table 4.3, SimSA1 column). Additionally, considering three classes with an equal number of samples in each run, a random classifier would typically achieve around 0.33 accuracy. However, most accuracies surpass this random baseline, suggesting that the DoAs derived from the simulated data offer valuable insights.

Next, the selected simulation properties also have an impact on performance, to varying degrees. Observing the standard deviations of performances per location, it becomes evident that, among all the considered properties, (1) position of the microphone array, (2) the sound source position along its part of the road, and (3) the type of microphone array emerge as the most influential factors in generating simulations that closely imitate real-world sound characteristics. However, there is no distinct trend on which properties work consistently well amongst all types of locations. For example, the results concerning the impact of the microphone array position roughly align with the settings in Schulz et al. (2020). Optimal performance occurs when a microphone array is 7-12 meters away from the intersection (as evident by Figure 4.5). This is comparable to the paper's setting, which indicates an average distance of 7-10 meters. However, similar generalizations cannot be extended to other simulation properties. Notably, findings from sound source position measurements are counter-intuitive, as, for some simulated locations, closer-than-actual source positions lead to improved performance. Nevertheless, when examining the DoAs for models trained with the settings resulting in the highest performance, it is apparent that simulating sound with the chosen software can produce sound features closely resembling those computed from real-world settings. This resemblance is particularly notable for category B, when comparing Figures A.2 and 4.3.

Ultimately, this experiment provided insights into the primary simulation properties influencing performance outcomes. Additionally, the reported accuracies demonstrate varying performance, with most exceeding what a random classifier would achieve. The findings suggest that the DoAs derived from simulations enable models to acquire transferable knowledge for real-world performance, emphasizing the utility of acoustic simulations for the given task.

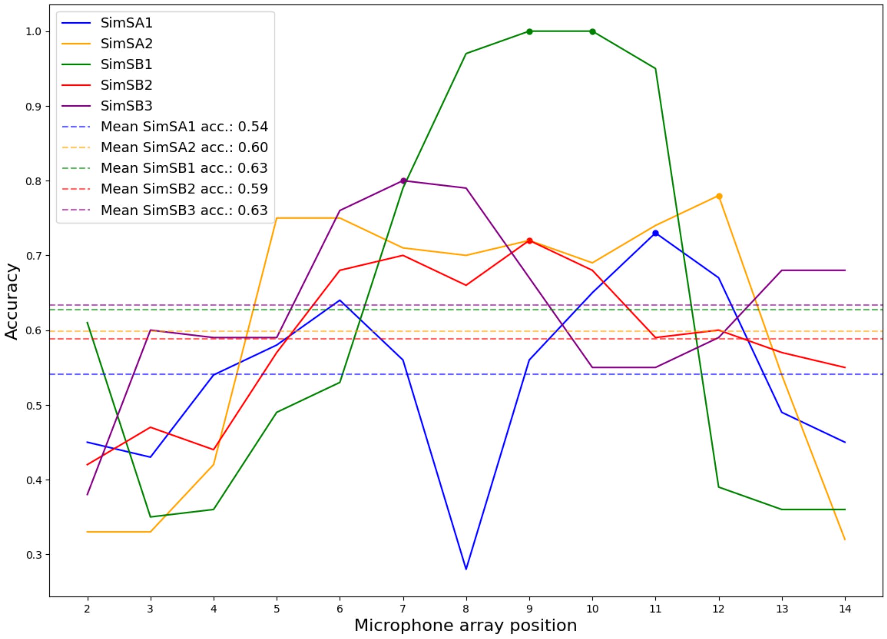


Figure 4.5.: Relationship between microphone array position p_m and the accuracy of the model. Each graph consists of points that represent average accuracies from 10-fold cross-validation when training on DoA features from a single simulated junction and testing on DoA features from the respective real-world junction. The bold colored points represent the highest accuracies within the given train and test settings.

Material	SimSA1	SimSA2	SimSB1	SimSB2	SimSB3
Brickwork	0.56	**0.72**	**1.00**	**0.72**	**0.67**
Rough lime wash	0.57	**0.72**	**1.00**	**0.72**	0.65
Ceramic tiles	0.54	**0.72**	**1.00**	**0.72**	**0.67**
Brick wall rough	0.57	**0.72**	**1.00**	**0.72**	0.66
Wooden lining	**0.58**	**0.72**	0.99	0.71	0.65
Plasterboard	0.56	**0.72**	**1.00**	0.71	**0.67**
Wood 16mm	0.56	**0.72**	**1.00**	0.71	0.65
Mean μ	0.56	0.72	1.00	0.72	0.66
Standard Deviation σ	0.01	0.00	0.00	0.00	0.01

Table 4.1.: Effect of changing the shared wall material in the simulation. Each entry in the table represents an average accuracy from 10-fold cross-validation when training on DoA features from a single simulated junction and testing on DoA features from the respective real-world junction.

Image Source Model order	SimSA1	SimSA2	SimSB1	SimSB2	SimSB3
3	**0.60**	0.72	0.85	0.61	**0.75**
4	0.54	**0.73**	**1.00**	0.71	0.67
5	0.56	0.72	**1.00**	**0.72**	0.67
6	0.56	0.71	0.99	**0.72**	0.67
7	0.55	0.71	**1.00**	**0.72**	0.68
Mean μ	0.56	0.72	0.97	0.70	0.69
Standard Deviation σ	0.02	0.01	0.06	0.04	0.03

Table 4.2.: Effect of changing the ISM reflection order. Each entry in the table represents an average accuracy from 10-fold cross-validation when training on DoA features from a single simulated junction and testing on DoA features from the respective real-world junction.

Sound source position	SimSA1	SimSA2	SimSB1	SimSB2	SimSB3
9	0.40	0.62	0.86	0.65	0.78
8	0.47	0.61	0.56	0.68	**0.83**
7	**0.57**	0.73	0.83	0.67	0.82
6	0.56	0.72	**1.00**	**0.72**	0.67
5	0.10	**0.77**	0.52	0.70	0.73
Mean μ	0.42	0.69	0.75	0.68	0.77
Standard Deviation σ	0.17	0.06	0.18	0.02	0.06

Table 4.3.: Effect of changing the source positions. Sound source position indicates y position of a sound source. Each entry in the table represents an average accuracy from 10-fold cross-validation when training on DoA features from a single simulated junction and testing on DoA features from the respective real-world junction.

Microphone Array	SimSA1	SimSA2	SimSB1	SimSB2	SimSB3
Circular (32 mics, radius=0.25)	0.56	0.72	**1.00**	0.72	0.67
Circular (32 mics, radius=0.23)	0.57	0.72	0.99	0.72	0.67
Circular (32 mics, radius=0.22)	0.58	0.72	0.97	0.72	0.67
Circular (32 mics, radius=0.20)	0.62	0.72	0.97	0.72	0.67
Circular (32 mics, radius=0.15)	0.62	0.76	0.96	0.72	0.66
Circular (56 mics, radius=0.30)	0.39	0.75	**1.00**	0.71	**0.68**
Circular (48 mics, radius=0.25)	0.56	0.72	**1.00**	0.72	0.67
Circular (48 mics, radius=0.35)	0.36	0.73	**1.00**	0.69	**0.68**
Square (16 mics, d=0.1)	**0.63**	0.77	0.96	0.72	0.67
Square (25 mics, d=0.1)	0.61	0.73	0.96	0.72	0.67
Square (25 mics, d=0.05)	0.55	0.81	0.94	0.68	0.60
Square (36 mics, d=0.05)	0.55	**0.82**	0.95	0.70	0.60
Square (36 mics, d=0.03)	0.55	0.78	0.91	0.63	0.55
Linear (12 mics, d=0.08)	0.38	0.68	**1.00**	**0.73**	0.56
Linear (24 mics, d=0.08)	0.36	0.67	0.97	0.67	0.48
Linear (36 mics, d=0.06)	0.34	0.67	0.96	**0.73**	0.52
Linear (24 mics, d=0.04)	0.40	0.67	**1.00**	**0.73**	0.56
Linear (29 mics, d=0.04)	0.36	0.67	**1.00**	0.71	0.57
Linear (28 mics, d=0.04)	0.38	0.67	**1.00**	0.72	0.56
Linear (27 mics, d=0.04)	0.37	0.68	**1.00**	0.72	0.57
Mean μ	0.49	0.72	0.97	0.71	0.61
Standard Deviation σ	0.11	0.05	0.03	0.02	0.06

Table 4.4.: Effect of changing the microphone array. Array entries contain their shapes, number of microphones, array radius/distance between each microphone in metres. Each entry in the table represents an average accuracy from 10-fold cross-validation when training on DoA features from a single simulated junction and testing on DoA features from the respective real-world junction.

4.3. Experiment 3: Training a direction of arival classification model on DoA acoustic features and BEV encodings

The remaining goal of the thesis was to train a classifier that can generalize to sounds from the intersection types not covered by its training dataset. This is why with this last experiment, we wanted to evaluate the effectiveness of using a BEV encoding for training a generalizable classifier. Following the description of the experimental results, the section concludes by conducting a hyperparameter study. This study aims to assess the robustness of the proposed BEV encoding when subjected to various parameter variations.

4.3.1. Experimental setup

The evaluation of the BEV encoding is carried out by comparing performance of two different models from Table 4.5. The evaluation compares the following models: the model conditioned on the BEV encoding, and the model that follows the approach from Schulz et al. (2020), employing an SVM with a linear kernel using only DoAs as inputs. Importantly, it was decided to only train and test the models on the simulated data. Firstly, there is not much variation in the types of junctions in the available real-world dataset from Schulz et al. (2020) – the dataset only covers T-junctions and the two respective types SA and SB. In addition, mixing simulated and real-world data was also out of the consideration. There is an inherent domain gap between the simulated sounds and the real-world sounds because of the simplified simulator setup and the sound propagation model. Therefore, incorporating both types of sounds would introduce additional inherent variability that can potentially influence the experimental results. This can compromise the interpretability of the experimental outcomes.

Model Type	BEV Encoding	Model Properties	Training Data	Test Data
SVM	None	Linear kernel, regularized (C=2)	Normalized DoAs from sound sources in simulated junctions	Normalized DoAs from sound sources within a new simulated junction, excluded from the training dataset
SVM w. BEV	Polar representation with *l=20, p=50, n=20, aggregation function= averaging*	Linear kernel, regularized (C=2)	Vectors that are results of concatenating normalized DoAs from sound sources in simulated junctions and BEV encodings of the respective junctions	Concatenated vectors comprising (1) normalized DoAs from sound sources within a new simulated junction (excluded from the training dataset) and (2) the junction's BEV encoding

Table 4.5.: Descriptions of an unconditioned SVM model and the SVM model conditioned on BEV encodings.

In this experiment, the two types of classifiers get evaluated on a new dataset, an extension of the dataset used in the previous experiment (Section 4.2). This extended dataset includes sounds from five additional intersections: SCross1, SCross2, SAR, SY1, SY2 (Figure 4.6), enlarging the dataset to 928 DoA feature vectors for simulated sounds in total. For each of the junctions, there are 96 DoA feature vectors (32 for each arrival class). The only exception is the SAR junction, in which there cannot be a car arriving from the left, resulting in 64 DoA feature vectors. The classifiers undergo multiple training sessions, excluding one of ten junction sounds from the training data each time, along with the sounds of the entire respective category. For instance, while evaluating the model's performance on SB3, the entire SB category (SB1, SB2, SB3) is omitted from the training dataset to assess performance in previously unencountered junction settings.

Furthermore, to identify consistent trends in model accuracies regardless of the amount of training data, the performances of the models get compared using different proportions of training data (i.e., 10%, 30%, 50%, 70%, 90%, 100%). To ensure robustness, the training data got shuffled ten times. As such, ten distinct runs for each entry in Table 4.7 were conducted, with each entry taking a proportion of a shuffled dataset. This allowed for incorporating randomness into all proportions except for the 100% proportion accuracies, as shuffling and adjusting the proportion does not alter the outcome in that case. In the end, each entry represents an average accuracy from ten distinct runs.

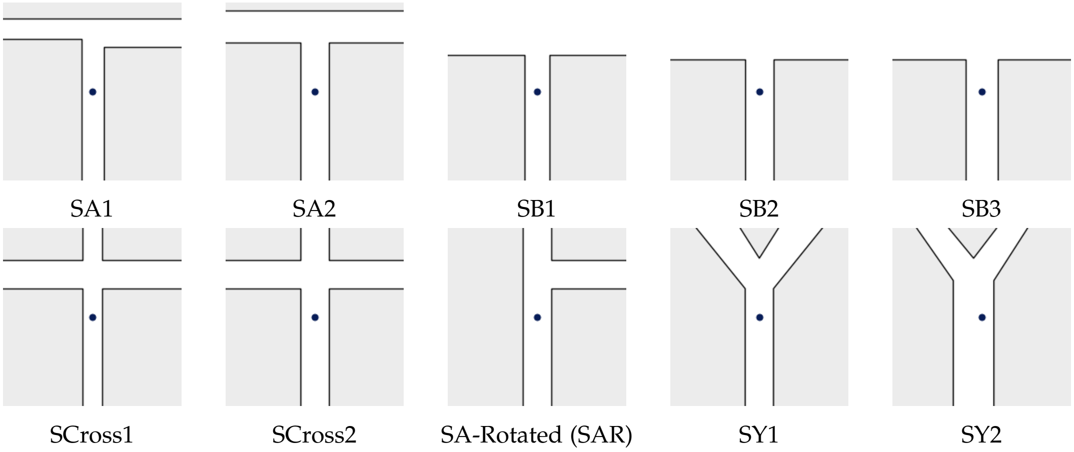


Figure 4.6.: An overview of the top-down junction images for the newly created simulated dataset. The points indicate positions of the microphone arrays. Black lines illustrate the starts of the walls, and grey indicates space beyond the road junction. The junction types (i.e., SA, SB, SCross, SY) are arranged in ascending order based on the width of the road where the microphone array is situated.

4.3.2. Results

Following the training of the model conditioned on the BEV encoding and the unconditioned model, the findings are detailed in Table 4.6. The obtained accuracies indicate that the model trained with BEV encodings and sound DoAs is at least as effective as the model trained without BEV encodings. Notably, in several cases, it demonstrates a significant improvement, as evidenced by the accuracies when testing on DoAs from SA1, SB2, SB3 intersections. On the other side, the conditioned model demonstrates slightly inferior performance within SY1, SAR, SB1 junction settings.

Consequently, when assessing the model performances with the same dataset but varying amounts of training data, a consistent trend emerges. The conditioned model consistently outperforms the unconditioned one when evaluating sounds from SA1, SA2, SB2, and SB3 intersections. In the case of other intersections, the performance of the conditioned model compared to the base model may exhibit slight variations based on the amount of utilized training data. Importantly, the average accuracy remains relatively stable for the unconditioned model when varying the amounts of training data. However, for the model conditioned on BEV encodings, the average accuracy decreases as the amount of training data decreases.

Overall, the findings from the experiment suggest the usefulness of the BEV encoding for training a generalizable model, albeit by a slight margin. As such, more research will be needed involving an assessment on more kinds of junctions. In addition, a subsequent experiment is needed with a stricter control of the variability factors within the simulated junctions (e.g., the microphone array position, the distance of the microphone array to the walls) that form the models' training dataset. This will further allow to test the usefulness of the proposed encoding. Lastly, experiments involving varied data quantities indicate that the conditioned model generally outperforms the other model across most data availability settings. Nonetheless, there is a minimal discrepancy between the two models when trained with very limited amounts of data.

Model	SCross1	SCross2	SY1	SY2	SAR	SA1	SA2	SB1	SB2	SB3	Average
SVM	0.97	1.00	**1.00**	0.79	**1.00**	0.35	0.55	**0.88**	0.71	0.70	0.80
SVM w. BEV	**0.98**	1.00	0.95	0.79	0.98	**0.42**	**0.56**	0.82	**0.92**	**0.94**	**0.84**

Table 4.6.: Accuracies of the models when tested on a new type of intersection. The results are compared between the unconditioned model and the BEV model.

% of training data used	SCross1	SCross2	SY1	SY2	SAR	SA1	SA2	SB1	SB2	SB3	Average
10%	0.96	0.98	0.95	0.64	0.99	0.37	0.45	0.77	0.80	0.84	0.77
	0.93	0.96	0.96	0.74	0.86	0.39	0.49	0.79	0.83	0.82	0.78
30%	0.98	0.99	1.00	0.70	1.00	0.37	0.49	0.81	0.71	0.76	0.78
	0.97	0.99	0.97	0.75	0.90	0.43	0.49	0.82	0.87	0.86	0.81
50%	0.97	1.00	1.00	0.70	1.00	0.37	0.50	0.81	0.70	0.70	0.78
	0.98	1.00	0.98	0.77	0.92	0.44	0.52	0.80	0.91	0.91	0.82
70%	0.97	1.00	1.00	0.75	1.00	0.35	0.53	0.82	0.69	0.70	0.78
	0.98	1.00	0.97	0.79	0.92	0.44	0.54	0.82	0.92	0.93	0.83
90%	0.97	1.00	1.00	0.77	1.00	0.36	0.54	0.84	0.69	0.70	0.79
	0.98	1.00	0.96	0.79	0.99	0.43	0.56	0.81	0.92	0.94	0.84
100%	0.97	1.00	1.00	0.79	1.00	0.35	0.55	0.88	0.71	0.70	0.80
	0.98	1.00	0.95	0.79	0.98	0.42	0.56	0.82	0.92	0.94	0.84

Table 4.7.: Accuracies of the models when tested on a new type of intersection using varying quantities of training data. Each cell with the different training data amounts includes two rows next to it: the first row displays accuracies for the unconditioned model (SVM), while the second row illustrates accuracies for the conditioned model (SVM w. BEV).

4.3.3. Hyperparameter study

As the proposed BEV encoding has several parameters that can affect the quality of the encoding (visualized in Figure 3.5), this section discusses the effect of changing one of the encoding parameters. To reiterate, the proposed encoding works as follows:

1. Shoot **n** rays of length **l**, uniformly separated between each other with an azimuth angle ϕ;
2. Sample **p** points along the ray, by identifying whether a point is inside the junction (0) or outside (1);
3. The result is $n \times p$ matrix.

The resulting matrix can either be flattened into a vector, or p points inside the matrix can be aggregated by using some aggregation function, resulting in a vector of length n. The following aggregation functions get considered:

- First principal component from PCA;
- Averaging p values.

The newly proposed model, conditioned on BEV encodings and acoustic features, underwent multiple retraining cycles. In each cycle, only a single hyperparameter got altered. Tables 4.8, 4.9, 4.10, and 4.11 showcase averages from the accuracies attained when testing the model on data from individual simulated junctions, just as in the last column of Table 4.6. To start off, the model was retrained by changing the number of rays only (Table 4.8). Here, one ray indicates shooting a single ray forward from the microphone position. As for more rays, they denote using multiple uniformly separated rays. Alteration of this parameter revealed minimal variability in the results, especially as the number of rays increased. When observing the influence of picking different aggregation functions in Table 4.9, it becomes evident that using averaging results in the best model performance. Employing PCA surprisingly leads to notably inferior performance compared to even the base SVM model conditioned solely on DoAs, as evident when comparing to entries from Table 4.6. Lastly, the final two tables (4.10, 4.11) demonstrate limited variability in results when either of the parameters gets modified.

Overall, the findings from the hyperparameter study suggest that incorporating the proposed BEV encoding as an additional feature, even with considerable changes to the hyperparameters, results in little variability between the model performances. Importantly, most of the tested models outperform the SVM model that lacks conditioning on BEV encodings at least by a slight margin, as evidenced by the results from Table 4.6.. Notably, the utilization of PCA as an aggregation function demonstrated no utility, indicating that averaging is the preferred aggregation method.

	1 ray	10 rays	20 rays	30 rays	40 rays	50 rays	60 rays
Accuracy	0.79 ± 0.21	0.80 ± 0.21	0.84 ± 0.19	0.80 ± 0.19	0.81 ± 0.18	0.80 ± 0.19	0.80 ± 0.18

Table 4.8.: Accuracies for SVM w. BEV models having different number of rays n. Accuracies show average model accuracy across all simulated junctions with a standard deviation.

	Average	PCA	Full
Accuracy	0.84 ± 0.19	0.69 ± 0.23	0.82 ± 0.22

Table 4.9.: Accuracies for SVM w. BEV models having different point aggregation functions. Accuracies show average model accuracy across all simulated junctions with a standard deviation.

	5 points	10 points	20 points	30 points	40 points	50 points
Accuracy	0.84 ± 0.19	0.81 ± 0.20	0.84 ± 0.19	0.84 ± 0.19	0.84 ± 0.19	0.84 ± 0.19

Table 4.10.: Accuracies for SVM w. BEV models having different number of points p being sampled per ray. Accuracies show average model accuracy across all simulated junctions with a standard deviation.

	ray length 15	ray length 20	ray length 25	ray length 30
Accuracy	0.85 ± 0.17	0.82 ± 0.18	0.84 ± 0.19	0.84 ± 0.19

Table 4.11.: Accuracies for SVM w. BEV models having different ray lengths l. Accuracies show average model accuracy across all simulated junctions with a standard deviation.

5. Conclusions and future work

5.1. Conclusions

In this thesis, multiple research questions were tackled, and by performing the previously described experiments, several observations and conclusions can be made.

The assessment of similarity between simulated sounds and real-world sounds resulted in interesting observations. Notably, no singular configuration of simulation properties was identified to effectively align simulated sounds with the expected sounds at a T-junction when hearing an approaching vehicle. This finding suggests that external engineers or researchers seeking to simulate realistic sound reverberation for any junction face challenges, as a straightforward set of instructions applicable for any type of junction could not be identified. However, it was observed that the utilization of specific simulation parameters can lead to the generation of sounds, from which the computed DoA intensities closely match those calculated from real-world sound data. These findings indeed encourage further exploration of simulations in future studies. Given that the employed simulation setup significantly simplified the real-world scenario, one can anticipate that employing specialized software incorporating a more intricate sound propagation model and additional features could potentially yield improved outcomes in subsequent research.

As sound inherently conveys less spatial information compared to other conventional sensors in autonomous driving, this research aimed to develop a method that would fuse prior information about the surroundings (BEV encoding) with a acoustic features. Namely, the approach aimed to train a classifier capable of operating across environments excluded from the training data. The study introduced a novel BEV encoding technique: transforming the map's top-down view into a polar representation that encodes information about the surrounding walls across all polar directions. The result of the method is a concise fingerprint of the surrounding environment. With the creation of the new encoding, it was combined with DoA features to train a classifier capable of processing sounds from previously unexplored environments. Results from the experiment indicated that the new classifier could either match the base classifier in accuracy when handling sounds from junctions excluded from the training dataset, or outperform it. Granted, more experiments in the future will be needed to confirm this improvement across many settings, and to make the results more reliable and robust. Nevertheless, the conducted experiment suggests the utility of incorporating information about the surrounding driving environment in the form of the BEV encoding in developing a classifier capable of generalizing to new environments.

5.2. Limitations

The methodology employed in this research has several limitations. First of all, the proposed simulator setup relies on external software *pyroomacoustics* that, as of the thesis writing, possessed the following limitations:

- **Inability to define unbounded spaces**. *Pyroomacoustics* is simulation software for indoor acoustics, so it requires a definition of an enclosed space to run acoustic simulations. This is why workarounds for defining junctions with some walls being absent were needed (described in Section 3.3);
- **Inability to define occlusions other than walls**. Another limitation of the software is its absence of support for definition of vegetation or various obstructions beyond walls in the simulated environments. This simplification is notable as sound emitted from a moving vehicle typically encounters multiple obstacles such as other vehicles, vegetation, pedestrians, among others;
- **Usage of acoustic simulations in 2D space instead of 3D**. The proposed simulator setup makes use of 2D simulations instead of 3D. Initially, when experimenting with the package, 3D simulations were used for modelling sound propagation in road junction settings. However, this resulted in unrealistic simulations compared to 2D. Namely, the DoA features computed from sounds simulated in 3D environments did not demonstrate peak energies for the ground truth directions of incoming vehicles. Hence, the resulting DoA vectors were not informative for determining the direction of arrival of an incoming vehicle. As such, the decision was made to utilize 2D acoustic simulations. That is the reason why walls, the microphone array, and sound sources are defined by their 2D coordinates rather than 3D in the proposed simulator setup;
- **Inability to define moving sound sources**. The package does not support definition of moving sound sources. Consequently, modeling the scenario where the ego vehicle approaches the junction's start, along with the obscured sound source, involved maintaining fixed positions for both (as described in Section 3.6);
- **Inability to define background noise**. As previously mentioned, the selected package does not allow definition of background noise in the simulated environments. Consequently, modeling scenarios where there is no approaching vehicle was unfeasible within this framework.

The aforementioned limitations created a substantial domain gap between the simulated sounds and the sounds that propagate in real world. Therefore, it is vital to approach the interpretation of simulation experimental results while considering this domain gap.

Furthermore, the employed methodology possesses another limitation, unrelated to the devised simulator setup. Namely, the BEV encoding experiment (Section 4.3) was conducted solely within the simulated environments to circumvent the aforementioned domain gap. This choice weakens the robustness of the experimental outcomes since their transferability to real-world settings remains uncertain. However, given that simulations demonstrated representativeness in some instances, there certainly exists a potential for the experimental findings to be applicable in real-world scenarios. Nonetheless, validating the transferability requires further experiments and research.

5.3. Future work

This work creates opportunities for exploring several promising research directions:

- **Use newer, more representative simulation software.** As simulation software continues to advance, leveraging updated software for analogous sound simulations within driving environments becomes even more promising. Enhanced features, particularly the capability to define moving sound sources, would inherently enhance the representativeness of simulations in the context of autonomous driving;
- **Verify the utility of the proposed BEV encoding with subsequent experiments**. As the potential utility of the BEV encoding was only assessed with a single experiment, more research is needed to confirm its value. A new experiment would involve (1) a stricter control of the variability factors within the model's dataset (e.g., the microphone array position, the distance of the microphone array to the walls), (2) a bigger coverage of another types of junctions in the dataset. In addition, the experimental results of the already executed experiment can be better analyzed in the future experiment. For example, it is still not clear why the SVM w. BEV model performed much better for SB2, SB3 environments than the unconditioned model (Section 4.3, Table 4.6). As such, in one of the upcoming experiments, the objective would be to comprehend the precise properties of the proposed BEV encoding method that result in either detrimental or improved performance of the classification model;
- **Propose another BEV encoding.** There is certainly a potential for development of a more informative BEV encoding. A more representative encoding would take into account some additional elements like trees, parked cars, or other occlusions that alter the propagation of sound in space;
- **Check the transferability of a new BEV encoding on a real-world dataset.** Since the experiment solely addressed simulated environments, it is crucial to explore whether the efficacy of a new BEV encoding extends to enhancing the robustness of a classifier in real-world scenarios. Given that the utilized real-world dataset exclusively encompasses T-junction data, expanding the dataset to include various junction types becomes imperative to ensure more comprehensive and robust results.

5.4. Ethical considerations

The experiments conducted in this thesis centered on simulations, thereby not presenting immediate ethical concerns regarding the employed methodology. However, the fundamental concept of employing audio as an additional modality for vehicle detection raises valid ethical considerations. Addressing these concerns is imperative to ensure the responsible development and deployment of this technology.

Privacy concerns
Utilizing yet another sensor for recording information while driving introduces significant amounts of new data collection. Consequently, recording and processing audio data may inadvertently capture private conversations or identifiable information. To safeguard individuals' privacy and prevent unauthorized access or misuse of sensitive information, robust privacy measures and stringent data security protocols are essential.

Bias and discrimination
As with any sensor data in autonomous driving, processing information from audio may result in both bias and discrimination. For instance, a hypothetical scenario could arise where the developed model is more prone to detecting motorbikes due to their distinct acoustic features, while being less adept at detecting electric cars. This situation inherently introduces discrimination towards individuals owning different types of road vehicles. Therefore, it is crucial to ensure that the development of future models involves testing for equal performance in detecting all kinds of vehicles.

In addition, another ethical consideration that emerged while writing the thesis pertained to the reproducibility of prior research. The initial assessment of previous works in line-of-sight vehicle detection (as detailed in Appendix B) yielded unexpected results. Both evaluated works had datasets that did not align with the descriptions provided in their respective papers. Additionally, contrary to the claims in Valverde et al. (2021), our reproducibility experiments indicated that the proposed vehicle detection model struggled to generalize to sounds recorded in driving scenarios not covered by the training dataset. These works initially appeared promising due to their dataset specifications, which, upon publication, did not match the provided descriptions. Consequently, these findings prompted a shift in the research focus toward non-line-of-sight acoustic vehicle detection, primarily due to the absence of applicable datasets for line-of-sight acoustic vehicle detection within the field. Effectively, the inability to reproduce the results significantly impacted our research progress, consuming valuable time and resources. This highlights the ethical responsibility in ensuring reproducibility, underscoring its critical significance for the broader scientific community when publishing research.

5.5. Reproducibility

The experimental outcomes described in Chapter 4 can be reproduced by using scripts from the project's Github repository[1].

[1] https://github.com/Borknab/msc-thesis

A. Extra figures

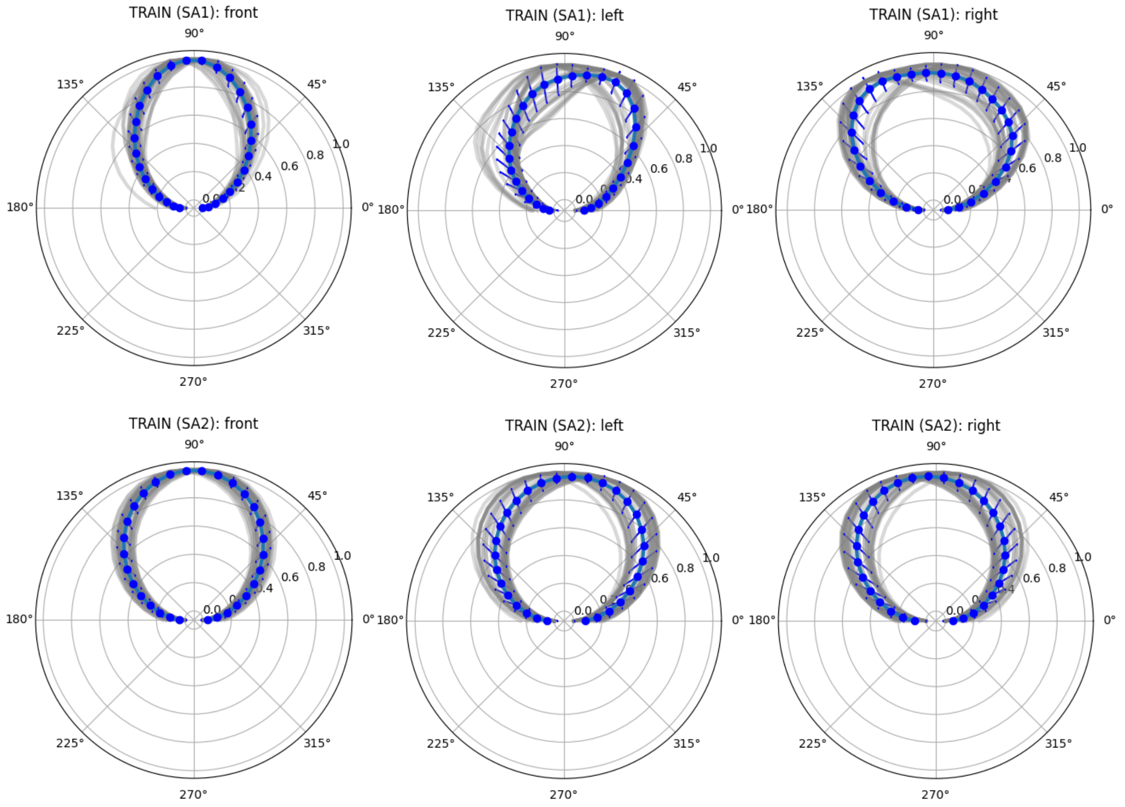


Figure A.1.: Visualized DoA features for simulated SA locations. The columns represent entries for front, left, and right classes (reading from left to right). The rows represent entries for (1) SimSA1, (2) SimSA2 locations (reading from top to bottom). Grey lines represent DoAs computed for all sounds from a location, a blue line shows a mean DoA feature and blue stripes represent standard deviation per azimuth angle.

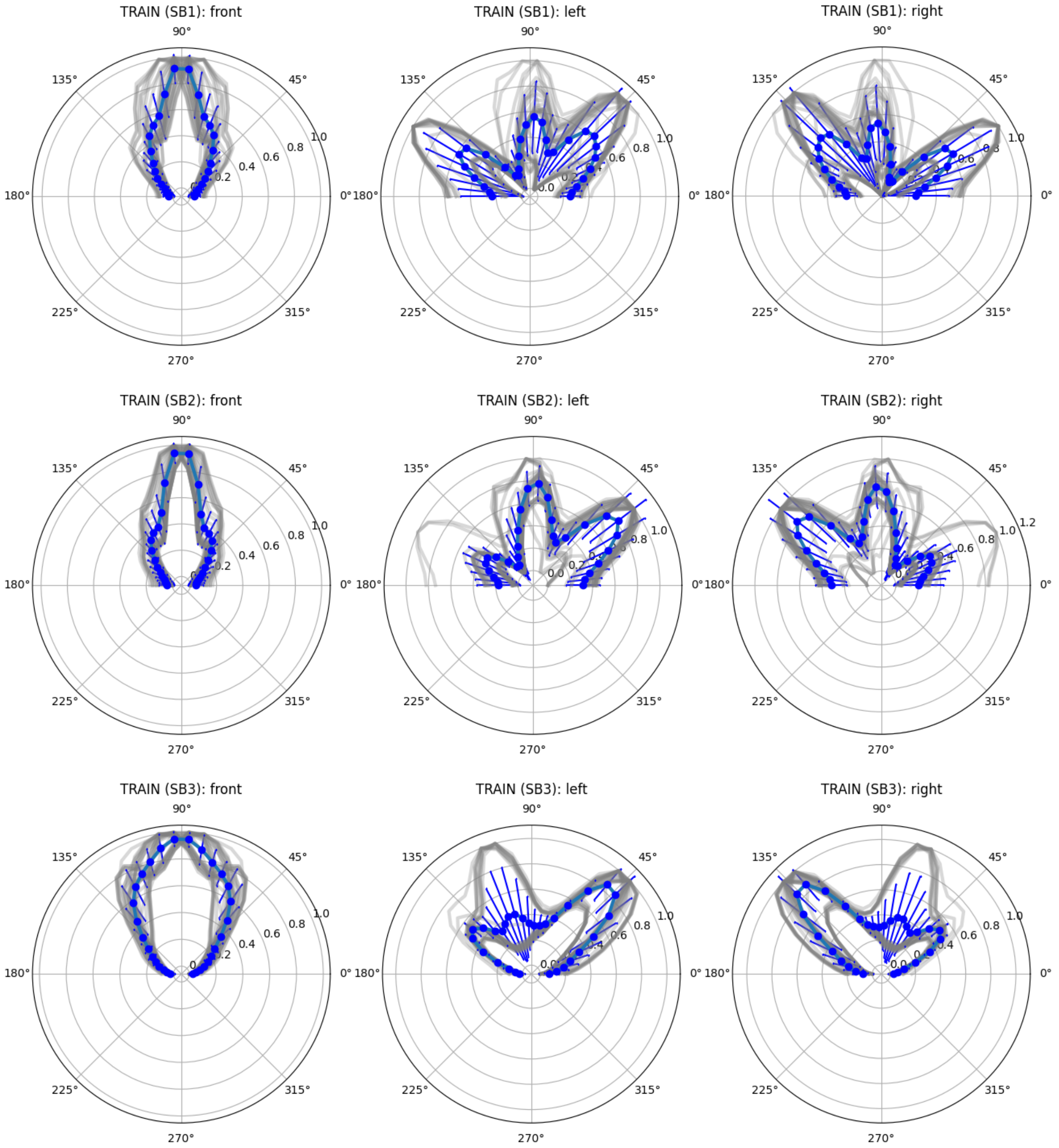


Figure A.2.: Visualized DoA features for simulated SB locations. The columns represent entries for front, left, and right classes (reading from left to right). The rows represent entries for (1) SimSB1, (2) SimSB2, (3) SimSB3 locations (reading from top to bottom). Grey lines represent DoAs computed for all sounds from a location, a blue line shows a mean DoA feature and blue stripes represent standard deviation per azimuth angle.

B. Initial work with line-of-sight acoustic vehicle detection

Initially, the goal of the thesis was to use acoustic information for detecting vehicles that are in direct line of sight. Namely, the research prospect was to use acoustic information and respective features to train a model for the task of object detection. 2D object detection is a task in computer vision that tackles locating instances of objects in images or videos (The MathWorks, 2023c). The outputs of an object detection model are typically bounding boxes defined by their widths, heights, x and y coordinates within the image, as well as respective class confidence score and a class. Having been inspired by the line-of-sight 2D vehicle detection research mentioned in the related work section (Section 2) (Chakravarthy et al., 2023; Gan et al., 2019; Valverde et al., 2021), the underlying vision was to use sound for downstream 3D perception tasks, such as 3D object detection, or semantic map segmentation. Hence, before proceeding with 3D perception tasks, the first experiments of the thesis revolved around reproduction of the results from the 2D detection works, so that any domain gaps and model failure cases could be identified.

Hence, this chapter presents the reproducibility efforts of the works from Valverde et al. (2021) and Chakravarthy et al. (2023). The chapter then concludes the reproducibility experiments and motivates why it was decided to switch the research interest to non-line-of-sight acoustic vehicle detection instead.

B.1. Reproducing the work from Valverde et al. (2021)

The underlying goal from the work done by Valverde et al. (2021) is to tackle 2D object detection by using sound features as an input only. In order to do that, the researchers utilize a knowledge distillation technique. Namely, the authors employ three EfficientDet-D2 modality-specific **teacher** networks (thermal, depth, RGB) that regress the bounding boxes for the nearby vehicles together with class confidence thresholds for a single class ***"car"***. Consequently, the **student** audio network learns to acoustic features to the bounding box predictions from the teacher networks. This architecture implies that the modalities were synchronized between each other, so that audio, as well as thermal, depth, and RGB images were sampled at the same timestamp. The acoustic features employed for the vehicle detection in the student audio network were spectrograms from one second long sound recordings, recorded by the microphone array mounted on the ego vehicle while driving or remaining in a static position. Having trained the knowledge distillation networks, the authors achieved remarkable results, demonstrating that the trained model was able to detect nearby vehicles using sound alone during the inference.

As the work from Valverde et al. (2021) included a code and a dataset release, as well as the weights for the pre-trained model, it was possible to directly assess reproducibility of their work. Consequently, this section provides an overview of the experiments performed to assess the reported results. The section is structured as follows. First, an initial assessment of the model and its outputs gets presented. Then, an important discovered property of the acoustic model – ability to detect parked cars, gets outlined. In order to identify a potential reason for the model's ability to detect parked cars, the model gets retrained using different train/test/-validation splits, and the results from retraining the model get discussed afterwards. Lastly, the issues encountered with the dataset while using it get highlighted, and the section finishes with conclusions from the experiments with the paper.

B.1.1. Assessment of the pretrained model

As the weights of the pretrained model were already provided by the authors, it was possible to directly apply the model to the provided dataset. The presentation of sample outputs is depicted with two subsequent figures. First, Figure B.1 illustrates sample car detections obtained from modality-specific networks. Subsequently, Figure B.2 displays additional vehicle detections generated by the model conditioned solely on audio input. Overall, qualitatively, the audio student model is able to output decent predictions for both static and dynamic driving sequences for some cars. However, several out-of-place predictions with a high confidence score for a car class can be observed.

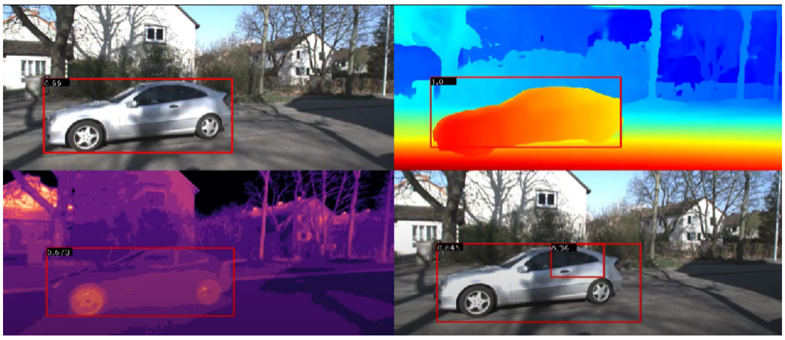


Figure B.1.: Example of inference from 3 teacher networks (RGB on top left, depth on top right, thermal on bottom left) and an audio student network (bottom right).

Model type	mAP@Avg ↑	mAP@0.5 ↑	mAP@0.75 ↑	CDx ↓	CDy ↓
Reported	61.62	84.29	59.66	1.27	0.69
Reproduced	28.69	44.91	25.43	5.35	3.09

Table B.1.: Comparison of the metrics reported in the paper (Table 1) against the reproduced results.

In order to have a quantitative evaluation of the model, it was decided to reproduce Table 1 results from the paper. The results can be seen in Table B.1. As can be observed, the model provided by the authors scores much worse in mean average precision and central distance metrics, which corresponds to the evaluation efforts done by other people on Github[1].

[1] https://github.com/robot-learning-freiburg/MM-DistillNet/issues/11

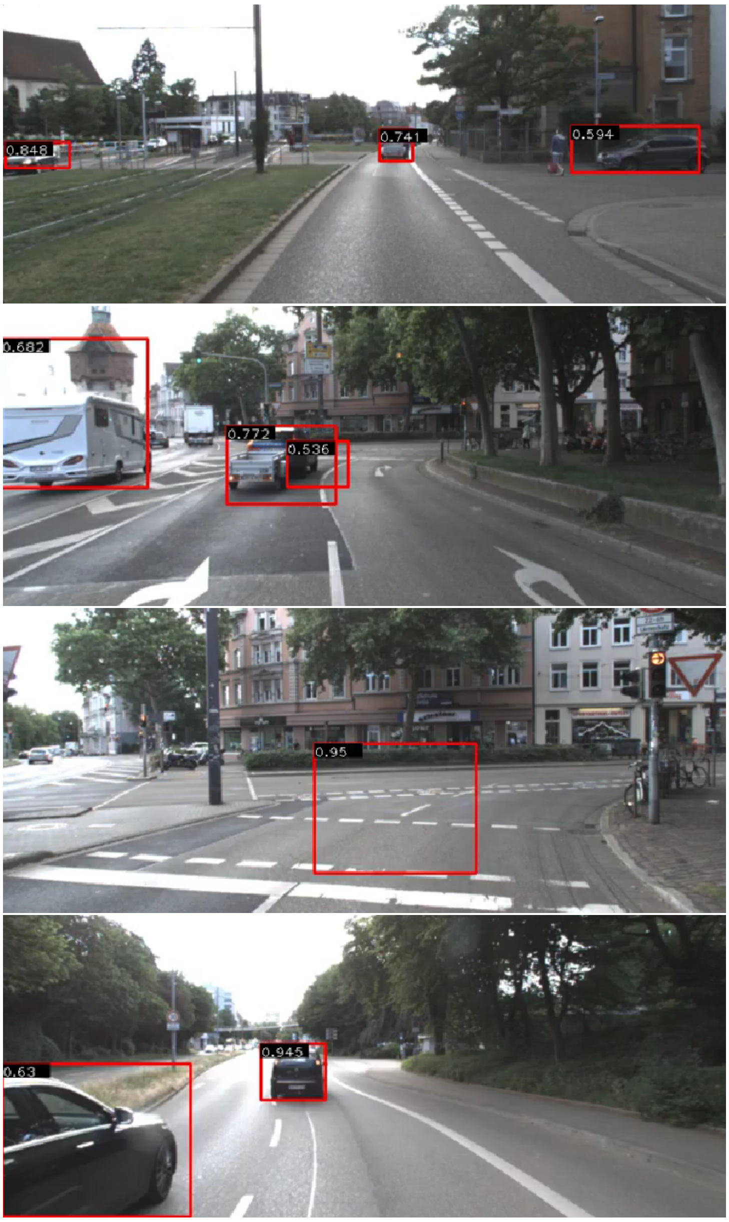


Figure B.2.: Sample car detections from an audio student network that uses sound spectrograms alone during the inference.

Overall, the initial evaluation of the model indicated satisfactory performance for some cars but also revealed misplaced predictions, as evidenced by Figures B.1 ans B.2. Moreover, when the model was being assessed using the same evaluation script as the one used for the paper's Table 1 results, a significant discrepancy emerged. The model's performance metrics were notably worse than those reported in the paper. These findings already indicated issues with the model, and further unexpected insights into its performance are detailed in the next section.

B.1.2. Insights on the model

During the assessment of the model conditioned on the audio input (spectrograms from one second sound recordings), a notable finding emerged: the model was consistently able to detect the presence of parked cars (as shown in Figure B.3). This outcome could be hypothetically attributed to several factors. Firstly, the model could learn to focus on sound reflections from the parked cars. Secondly, the model could be overfitted, which lead to its proficiency in predicting parked cars. The subsequent experiment focuses on the verification of the second hypothesis.

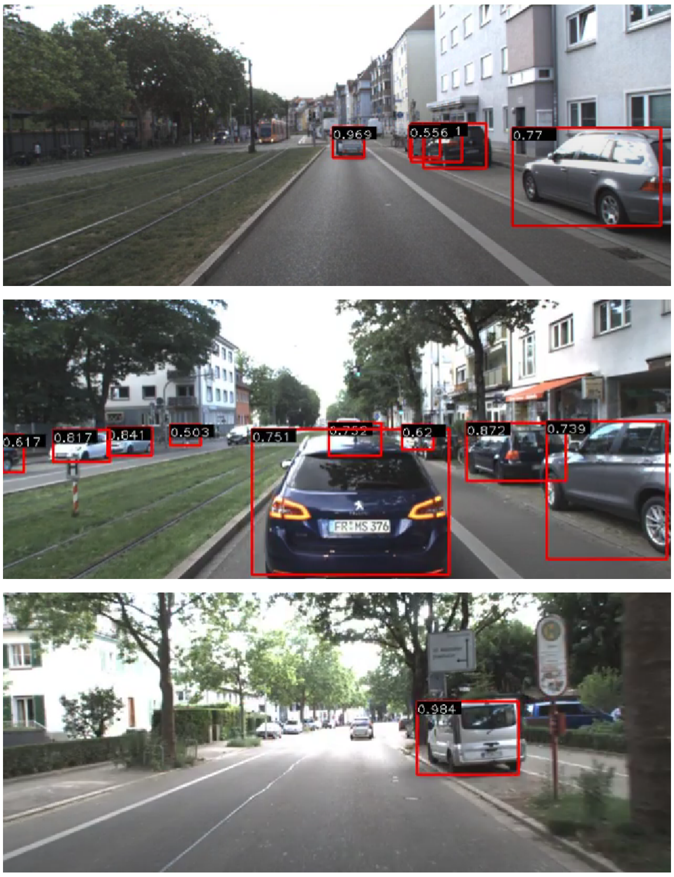


Figure B.3.: Examples of predictions from the audio student network which clearly include parked cars with high levels of confidence.

In order to verify whether the model was overfitting or not, it was decided to start with the inspection of the train/test/validation splits. Overall, the following observations could be made:

- The dataset had 48 drive recordings in total. Each of them consisted of timestamp-synchronized sound, depth, thermal, and RGB images for a distinct location the ego vehicle was located at, or a path it was following while recording the data. Notably, none of the drive recordings got specifically allocated for test or validation splits, meaning that all of them got shared amongst the splits, resulting in no unique driving scenarios for test and validation splits;

- Top 10 locations with the most utilized synchronized modality data were exactly the same for both train, test, and validation splits;
- 14147/18873 (75%) modality entries in the validation set were directly adjacent to one of the frames in the train set. In other words, they were one increment or decrement to the timestamp away, presenting a substantial feature overlap. This problem was already explained in the main part of the thesis (Section 2.2.2) and visualized previously with Figure 2.5;
- 4640/18873 (25%) modality entries in the validation set were directly adjacent to one of the frames in the test set (one increment or decrement to the timestamp away).

These observations prompted further investigation to verify whether the model was indeed overfitting. In order to verify this statement, it was decided to complete the following steps:

1. Retrain the model on half of the data and a quarter of the data respectively (for computational resource availability reasons), while holding out some recording locations uniquely for the validation set. Additionally, to address the aforementioned observation with adjacency of modality entries in the splits, decrease the average overlap (i.e., time distance) between the frames by **two** for the model trained on half of the dataset and by **four** for the other model respectively;
2. Run evaluation script for the resulting modeling models on respective test splits and compare the performance against the model provided by the authors;
3. Evaluate the retrained models and the original model on the shared driving sequences;
4. Evaluate the retrained models and the original model on two driving sequences that were only excluded from the train splits for retrained models.

B.1.3. Retraining the model with new data splits

As mentioned in the previous section, the model was retrained two times – on a half and the quarter of the original training data allocation respectively. Notably, two sequence recordings were completely left out of the train and validation splits both of the times:

- **drive_day_2020_03_18_16_02_15** (static recording, daytime, 95 entries of synchronized data);
- **drive_day_2020_05_21_20_25_14** (dynamic recording, daytime, 1174 entries).

Initially, the models were evaluated against each other using an original test split (18874 entries of synchronized RGB, depth, thermal, and audio data; Table B.2). As can be observed, the model retrained on 1/2 of the dataset actually scored higher than the model with weights provided by the authors. The model trained on 1/4 of the original train set scored much lower than the other models, which may be attributed not only to the decreased size of the dataset, but also the modality-specific recordings (e.g., sound clips, images) being further away to each other in time.

Subsequently, the models were applied to two different recordings that were excluded from the train and test sets (Table B.3), along with two recordings that were included in the train sets of all models (Table B.4). Surprisingly, both of the retrained models failed to generalize to the entries from the driving locations excluded from the training dataset, achieving a mean average precision score of zero. This discovery aligns with the observation made by Chakravarthy et al. (2023), where it was found that a similar method (Gan et al., 2019) to the one employed by Valverde et al. (2021) could not generalize to new driving scenarios, scoring zero in average precision metrics. Consequently, looking at the evaluation results for the two remaining recording sequences that were included in the training dataset (Table B.4), it can be seen that they are more aligned with the general evaluation done in Table B.2.

Model type	mAP@Avg ↑	mAP@0.5 ↑	mAP@0.75 ↑	CDx ↓	CDy ↓
Reported	61.62	84.29	59.66	1.27	0.69
Reproduced	28.69	44.91	25.43	5.35	3.09
1/2 training data	33.39	46.00	31.58	4.77	2.78
1/4 training data	16.29	20.33	15.84	8.57	5.07

Table B.2.: Evaluation of the models trained on different overlaps between the data splits and data amounts.

Model type	mAP@Avg ↑	mAP@0.5 ↑	mAP@0.75 ↑	CDx ↓	CDy ↓
drive_day_2020_03_18_16_02_15 (95 entries, static, day)					
Reproduced	15	33.92	7.93	12.64	5.66
1/2 training data	0.00	0.00	0.00	26.97	12.67
1/4 training data	0.00	0.00	0.00	27.73	13.12
drive_day_2020_05_21_20_25_14 (1174 entries, dynamic, day)					
Reproduced	29.25	50.2	24.13	6.42	3.25
1/2 training data	0.00	0.03	0.00	12.8	7.51
1/4 training data	0.00	0.02	0.00	13.19	7.79

Table B.3.: Evaluation of the models on two different recordings which the retrained models were not trained on.

Model type	mAP@Avg ↑	mAP@0.5 ↑	mAP@0.75 ↑	CDx ↓	CDy ↓
drive_day_2020_03_18_15_52_13 (83 entries, static, day)					
Reproduced	11.12	30.96	7.57	17.25	8.02
1/2 training data	30.74	58.52	20.45	10.63	5.45
1/4 training data	1.32	3.05	0.33	25.67	13.55
drive_day_2020_05_29_17_53_48 (5181 entries, dynamic, day)					
Reproduced	26.47	51.21	18.81	4.04	2.90
1/2 training data	14.48	29.76	9.50	5.85	4.11
1/4 training data	0.17	0.38	0.10	10.33	7.31

Table B.4.: Evaluation of the models on two different recordings which all models were trained on.

B.1.4. Issues with the provided dataset

Several issues emerged while working with the dataset. Firstly, there were noticeable distortions and shifts observed in both thermal and depth images across various driving sequences, as depicted in Figures B.4 and B.5. In addition, the included extrinsics for the sensor mounted on the ego vehicle (i.e., positions and orientations with respect to the the other sensors or the world) were not documented, so it was not clear how to use them. Lastly, as LiDAR point clouds are one of the most common modalities in 3D object detection in the context of autonomous driving (Dong et al., 2023), their inclusion in the dataset would be very beneficial. However, despite the paper indicating the inclusion of LiDAR point clouds in the dataset, the published dataset lacks them.

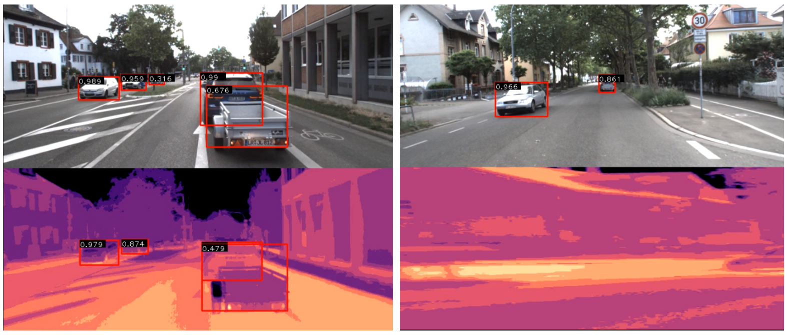


Figure B.4.: An example of 2 RGB images (top) in the dataset and respective correct (bottom left) and corrupted (bottom right) thermal images.

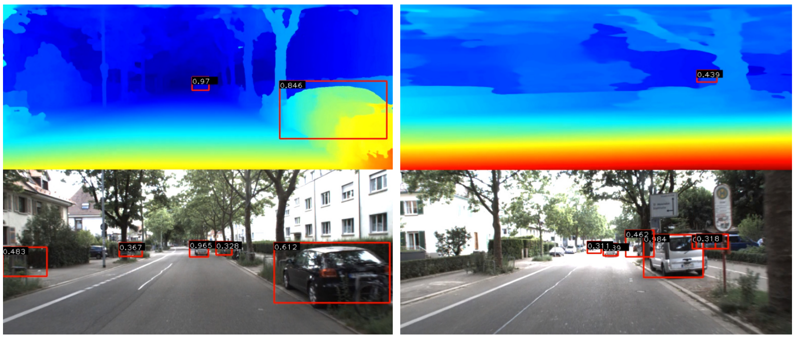


Figure B.5.: An example of 2 RGB images (bottom) in the dataset and respective correct (top left) and corrupted (top right) RGB-D images.

Overall, the aforementioned observations raised some concerns about the quality of the available data and questioned the applicability of the dataset for the tasks of the thesis.

B.1.5. Reflection on the observed results

In the end, the assessment of the proposed approach – knowledge distillation network for the task of detecting vehicles, yielded unexpected results. The proposed model could not generalize beyond the synchronized modality entries from the driving sequences already included in the training dataset. In addition, the issues with the dataset made it impractical to apply it for the tasks of the thesis. Therefore, the choice was made to exclude the authors' approach and abstain from utilizing the dataset for the thesis objectives.

B.2. Reproducing the work from Chakravarthy et al. (2023)

Similar to Valverde et al. (2021), this paper focuses on 2D object detection using acoustic features. This paper was particularly relevant for the thesis as its dataset contained numerous entries of synchronized sound, image, and LiDAR data, essential for training models for downstream 3D perception tasks. In addition, the paper reported impressive results demonstrated the effectiveness of beamforming maps for object detection. Therefore, reproducing the experimental results of the paper was also intriguing.

Unfortunately, at the time of writing the thesis, the authors did not provide a code release that would allow to reproduce the experimental results. In addition, upon investigating the dataset shared on Google Drive[2], it became evident that the published dataset was incomplete. Its size was approximately 190 GB, significantly smaller than the reported 14TB storage utilization. Furthermore, the dataset's contents did not align with the specifications outlined in the paper, and the published dataset lacked proper documentation. In summary, several crucial elements were missing from the published dataset:

- Images from one of the cameras;
- Sound recordings from the microphone array.

Ultimately, the published dataset proved inadequate for the thesis objectives for the aforementioned reasons, and most importantly, for missing acoustic data. Therefore, the dataset could not be utilized for the intended purposes of the thesis.

B.3. Conclusions

In the end, while reviewing both studies, significant insights surfaced. While exploring the research by Valverde et al. (2021), it became apparent that the provided model and architecture could not generalize to new driving scenarios, and issues arose regarding the quality of the published dataset. In addition, the study conducted by Chakravarthy et al. (2023) was of particular interest due to its dataset. However, the published version of the dataset proved incomplete and inadequate for the thesis's research objectives. Given the absence of other studies providing suitable datasets for the intended 3D vehicle perception using acoustic features, a decision was made to redirect the thesis focus towards non-line-of-sight vehicle detection.

[2] https://drive.google.com/drive/folders/1CJHbDfqtglHpCa12HLMzc4xfymDuhZBK

Bibliography

Almutairi, W., & Janicki, R. (2020). On relationships between imbalance and overlapping of datasets. In G. Lee & Y. Jin (Eds.), *Proceedings of 35th international conference on computers and their applications* (Vol. 69, pp. 141–150). EasyChair. Retrieved from https://easychair.org/publications/paper/Xk8r doi: 10.29007/h71z

Baron, V., Bouley, S., Muschinowski, M., Mars, J., & Nicolas, B. (2019). Drone localization and identification using an acoustic array and supervised learning. In J. Dijk (Ed.), *Artificial intelligence and machine learning in defense applications* (Vol. 11169, p. 111690F). SPIE. Retrieved from https://doi.org/10.1117/12.2533039 doi: 10.1117/12.2533039

Bezzam, E., Scheibler, R., Cadoux, C., & Gisselbrecht, T. (2020). A study on more realistic room simulation for far-field keyword spotting. *CoRR*, *abs/2006.02774*. Retrieved from https://arxiv.org/abs/2006.02774

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., ... Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *Cvpr.*

Chakravarthy, P., D'Souza, J., Tseng, E., Bartusek, J., & Heide, F. (2023, 06). Seeing with sound: Long-range acoustic beamforming for multimodal scene understanding. In (p. 982-991). doi: 10.1109/CVPR52729.2023.00101

Damiano, S., & van Waterschoot, T. (2022). Pyroadacoustics: a road acoustics simulator based on variable length delay lines. *Proc. 25th Int. Conf. Digital Audio Effects (DAFx20in22), (Vienna)*, pp. 216-223.

Dibiase, J. H. (2000, August). A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays.

Dong, Y., Kang, C., Zhang, J., Zhu, Z., Wang, Y., Yang, X., ... Zhu, J. (2023, jun). Benchmarking robustness of 3d object detection to common corruptions in autonomous driving. In *2023 ieee/cvf conference on computer vision and pattern recognition (cvpr)* (p. 1022-1032). Los Alamitos, CA, USA: IEEE Computer Society. Retrieved from https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.00105 doi: 10.1109/CVPR52729.2023.00105

Fang, W., Zhang, F., Sheng, V., & Ding, Y. (2018, 01). A method for improving cnn-based image recognition using dcgan. *Computers, Materials & Continua*, *57*, 167-178. doi: 10.32604/cmc.2018.02356

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Synthetic data augmentation using gan for improved liver lesion classification. In *2018 ieee 15th international symposium on biomedical imaging (isbi 2018)* (p. 289-293). doi: 10.1109/ISBI.2018.8363576

Gan, C., Zhao, H., Chen, P., Cox, D., & Torralba, A. (2019). Self-supervised moving vehicle tracking with stereo sound. *IEEE International Conference on Computer Vision*. doi: 10.1109/iccv.2019.00715

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.

Gulino, C., Fu, J., Luo, W., Tucker, G., Bronstein, E., Lu, Y., ... Sapp, B. (2023). *Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research.*

Johnson, D. H., & Dudgeon, D. E. (1993). *Array signal processing: concepts and techniques*. Englewood Cliffs, NJ: P T R Prentice Hall Englewood Cliffs, NJ.

Kim, S., Oh, S.-Y., Kang, J., Ryu, Y., Kim, K., Park, S.-C., & Park, K. (2005). Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion. In *2005 ieee/rsj international conference on intelligent robots and systems* (p. 2173-2178). doi: 10.1109/IROS.2005.1545321

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. *Computer Vision and Pattern Recognition*. doi: 10.1109/cvpr.2019.01298

Li, H., Sima, C., Dai, J., Wang, W., Lu, L., Wang, H., ... Qiao, Y. (2023). Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1-20. doi: 10.1109/TPAMI.2023.3333838

Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D., & Han, S. (2022). Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation.

Marchegiani, L., & Fafoutis, X. (2021). How well can driverless vehicles hear? a gentle introduction to auditory perception for autonomous and smart vehicles. *IEEE Intelligent Transportation Systems Magazine*. doi: 10.1109/mits.2021.3049425

Multiphysics, C. (1998). Introduction to comsol multiphysics ®. *COMSOL Multiphysics, Burlington, MA, vol. 9*.

NVIDIA. (2023). *Nvidia drive sim - built on omniverse.* Retrieved 13-12-2023, from https://developer.nvidia.com/drive/simulation

Picaut, J., & Fortin, N. (2012a, April). I-Simpa, a graphical user interface devoted to host 3D sound propagation numerical codes. In S. F. d'Acoustique (Ed.), *Acoustics 2012.* Nantes, France. Retrieved from https://hal.science/hal-00810893

Picaut, J., & Fortin, N. (2012b, April). SPPS, a particle-tracing numerical code for indoor and outdoor sound propagation prediction. In S. F. d'Acoustique (Ed.), *Acoustics 2012.* Nantes, France. Retrieved from https://hal.science/hal-00810894

Rhinehart, T. A., Chronister, L. M., Devlin, T., & Kitzes, J. (2020). Acoustic localization of terrestrial wildlife: Current practices and future opportunities. *Ecology and Evolution*, *10*(13), 6794-6818. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.6216 doi: https://doi.org/10.1002/ece3.6216

Scheibler, R., Bezzam, E., & Dokmanic, I. (2018, apr). Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP).* IEEE. Retrieved from https://doi.org/10.1109%2Ficassp.2018.8461310 doi: 10.1109/icassp.2018.8461310

Schulz, Y., Mattar, A. K., Hehn, T. M., & Kooij, J. F. P. (2020). Hearing what you cannot see: Acoustic detection around corners. *CoRR*, *abs/2007.15739*. Retrieved from https://arxiv.org/abs/2007.15739

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., ... Anguelov, D. (2020, June). Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr).*

The MathWorks, I. (2023a). *Room impulse response simulation with the image-source method and hrtf interpolation.* Retrieved 06-11-2023, from https://nl.mathworks.com/help/audio/ug/room-impulse-response-simulation-with-image-source-method-and-hrtf-interpolation.html

The MathWorks, I. (2023b). *Source localization using generalized cross correlation.* Retrieved 23-12-2023, from https://nl.mathworks.com/help/phased/ug/source-localization-using-generalized-cross-correlation.html

The MathWorks, I. (2023c). *What is object detection?* Retrieved 29-11-2023, from https://nl.mathworks.com/discovery/object-detection.html

Valverde, F. R., Hurtado, J. V., & Valada, A. (2021). There is more than meets the eye: Self-supervised multi-object detection and tracking with sound by distilling multimodal knowledge. *Computer Vision and Pattern Recognition*. doi: 10.1109/cvpr46437.2021.01144

Voort, V. D., & Aarts, R. M. (2009). Development of dutch sound locators to detect airplanes (1927-1940). Retrieved from https://pub.dega-akustik.de/NAG_DAGA_2009/data/articles/000554.pdf

Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., & Solomon, J. (2021). DETR3D: 3d object detection from multi-view images via 3d-to-2d queries. *CoRR*, *abs/2110.06922*. Retrieved from https://arxiv.org/abs/2110.06922

Yangfan Liu and J. Stuart Bolton and Patricia Davies. (2021, June 15). Acoustic source localization techniques and their applications. *Summer Bridge on Noise Control Engineering*, *51*(2). Retrieved from https://www.nae.edu/255823/Acoustic-Source-Localization-Techniques-and-Their-Applications