From Sound to Structure

Robust Localization of Sources, Sensors, and Surroundings

Tegler, Erik

2025

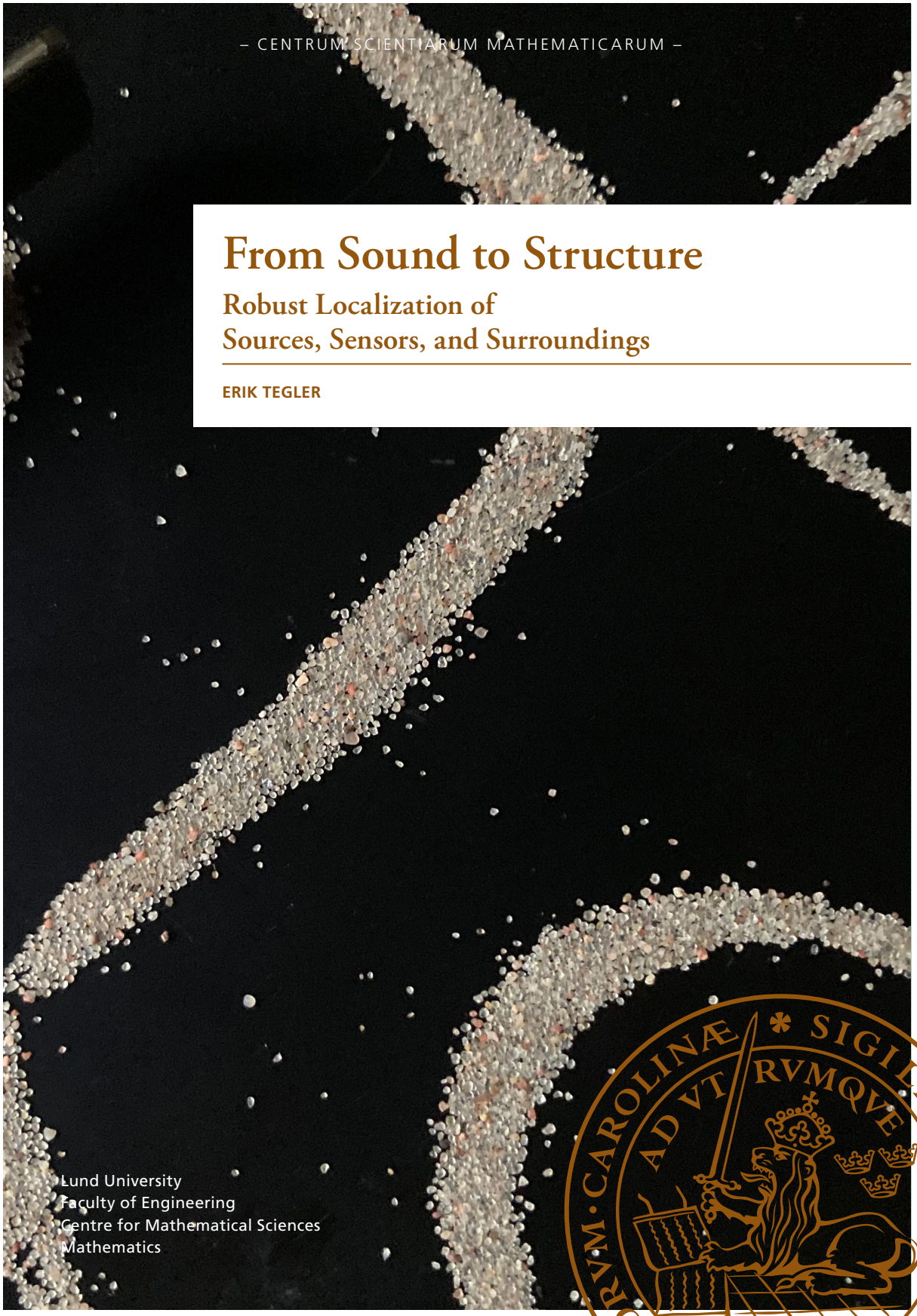
Document Version:
Publisher's PDF, also known as Version of record

– CENTRUM SCIENTIARUM MATHEMATICARUM –

From Sound to Structure

Robust Localization of Sources, Sensors, and Surroundings

ERIK TEGLER



Lund University
Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

From Sound to Structure

Robust Localization of Sources, Sensors, and Surroundings

Erik Tegler

LUND
UNIVERSITY

ACADEMIC THESIS

which, with due permission of the Faculty of Engineering at Lund University, will be publicly defended on 16th of January, 2026, at 13:15 in lecture hall MH:Hörmander, Centre for Mathematical Sciences, Märkesbacken 4, Lund, for the degree of Doctor of Philosophy.

Thesis advisors:
Prof. Karl Åström,
Assoc. Prof. Magnus Oskarsson, Asst. Prof. Viktor Larsson,
Prof. Fredrik Tufvesson, Prof. Bo Bernhardsson,

Faculty opponent:
Prof. Tuomas Virtanen, Tampere University, Finland

Organization	Document name
LUND UNIVERSITY Centre for Mathematical Sciences Box 118 SE–221 00 LUND Sweden	**Doctoral thesis**
	Date of presentation 2026-01-16
Author(s) Erik Tegler	Sponsoring organization ELLIIT

Title and subtitle

From Sound to Structure – Robust Localization of Sources, Sensors, and Surroundings

Abstract

The topic of this thesis is how to infer geometric information using sound data. Achieving this requires solving several subproblems. First, signal processing of the recorded sound is needed to compute measurements of primitive geometric relations. Secondly, robust estimation is needed to go from primitive geometric measurements to more useful higher-level information such as the locations of microphones and sound sources.

In the case of an uncontrolled sound source, one of the main ways of extracting geometric information comes from computing the time between a sound arriving at each of two microphones. This measurement is referred to as the Time-Difference-of-Arrival (TDOA) and it defines a hyperboloid relative to the two microphones, on which the sound source must lie. While classical correlation-based techniques exist for how to compute the TDOA from two recordings, they typically struggle in reverberant environments where the two signals are not just shifted noisy versions of each other. One of the results of this thesis is showing that better time-delay estimation can be performed by using a learning-based approach. The main issue with using a learning-based approach in this domain is a lack of data. However, this thesis demonstrates that it is possible to solve this issue by utilizing simulations of sound propagation to create synthetic data. This data can then be used to train an energy-based model, which demonstrates improved performance on real data compared to classical methods.

After computing primitive geometric relationships from the sensor data, the goal is to convert them into more useful higher-level information such as the locations of microphones and sound sources. The main problem here lies in that a fraction of the measurements are outliers which means that robust estimation methods such as RANSAC (a hypothesis-and-test framework) need to be used. Since the speed of hypothesis creation is key when using RANSAC, this thesis shows how to construct new minimal solvers for several problems. One example is that we show that sensor network self-calibration in the presence of a reverberant plane allows for minimal problems containing fewer microphones than in the echo-free case.

Key words

sound, localization, time-of-arrival, TOA, time-difference-of-arrival, TDOA, self-calibration, robust estimation, RANSAC, minimal solvers, energy-based models, time-delay estimation, GCC-PHAT, reverberations

Classification system and/or index terms (if any)

Supplementary bibliographical information		Language English
ISSN 1404-0034		ISBN 978-91-8104-763-9 (print) 978-91-8104-764-6 (pdf)
Recipient's notes	Number of pages xiv+192	Price
	Security classification	

From Sound to Structure

Robust Localization of Sources, Sensors, and Surroundings

Erik Tegler

LUND
UNIVERSITY

Cover illustrations: Close-up of patterns formed by sand on a Chladni plate.

Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden
www.maths.lu.se

Doctoral Theses in Mathematical Sciences 2025:6
ISSN: 1404-0034
ISBN: 978-91-8104-763-9 (print)
ISBN: 978-91-8104-764-6 (electronic)
LUTFMA-1091-2025

Printed in Sweden by Media-Tryck, Lund University, Lund 2025

Abstract

The topic of this thesis is how to infer geometric information using sound data. Achieving this requires solving several subproblems. First, signal processing of the recorded sound is needed to compute measurements of primitive geometric relations. Secondly, robust estimation is needed to go from primitive geometric measurements to more useful higher-level information such as the locations of microphones and sound sources.

In the case of an uncontrolled sound source, one of the main ways of extracting geometric information comes from computing the time between a sound arriving at each of two microphones. This measurement is referred to as the Time-Difference-of-Arrival (TDOA) and it defines a hyperboloid relative to the two microphones, on which the sound source must lie. While classical correlation-based techniques exist for how to compute the TDOA from two recordings, they typically struggle in reverberant environments where the two signals are not just shifted noisy versions of each other. One of the results of this thesis is showing that better time-delay estimation can be performed by using a learning-based approach. The main issue with using a learning-based approach in this domain is a lack of data. However, this thesis demonstrates that it is possible to solve this issue by utilizing simulations of sound propagation to create synthetic data. This data can then be used to train an energy-based model, which demonstrates improved performance on real data compared to classical methods.

After computing primitive geometric relationships from the sensor data, the goal is to convert them into more useful higher-level information such as the locations of microphones and sound sources. The main problem here lies in that a fraction of the measurements are outliers which means that robust estimation methods such as RANSAC (a hypothesis-and-test framework) need to be used. Since the speed of hypothesis creation is key when using RANSAC, this thesis shows how to construct new minimal solvers for several problems. One example is that we show that sensor network self-calibration in the presence of a reverberant plane allows for minimal problems containing fewer microphones than in the echo-free case.

Popular Summary

Imagine an action you regularly perform. As an example, consider turning around to face a friend who has called for your attention. In order to achieve this, you use the sound you heard to know which direction to turn to. If you instead wanted to pick up an object, you need to use a combination of your sense of sight and touch to first place your hand in the approximately right position and then grab with the right amount of force. In general, processing sensor data into useful information is key, regardless of the task at hand. While many types of sensors exist, each with their respective advantages and disadvantages, the primary focus of this thesis is how to use sound to gain information about our surroundings. More precisely, we focus on finding the location of various sound sources, microphones or walls, see Figure P.1.

How do we use sound recordings to infer where objects are? An illustrative example is a thunder strike. You first see a flash of light, three seconds later you hear the sound of thunder. Since the sound traveled for three seconds and the speed of sound is around 340 m/s, you can conclude that the distance to the strike is about one kilometer. Similarly, if the same sound is picked up by two microphones, the sound will arrive at the closer microphone first. In this way, we can process the recorded sound to obtain information on the relative positions of microphones and sound sources.

After the signal processing is completed, we know simple geometric relationships such as the distances between pairs of objects. These basic measurements can then be refined into more high-level knowledge, such as the 3D position of each object. This could for instance be finding the position of a sound source using the sound recorded from a few microphones of known position. A more complex and less well-known problem is self-calibration. Essentially, even without prior knowledge of the position of sound sources and microphones, we can simultaneously estimate all the locations using only the recorded sound.

These types of problems are challenging because they involve real-world data. If we consider a robot arm automating a task in a factory, it can perform difficult tasks with a high degree of efficiency. However, it can only operate in a very controlled environment. If the conditions change, such as just moving the position of the robot arm slightly, it will completely stop the robot arm from performing its task. For the problems described above, it is common to not only have noisy measurements but also for some to be incorrect altogether. While the problems are still solvable, they require methods with robustness; how to achieve this is the subject of this thesis.

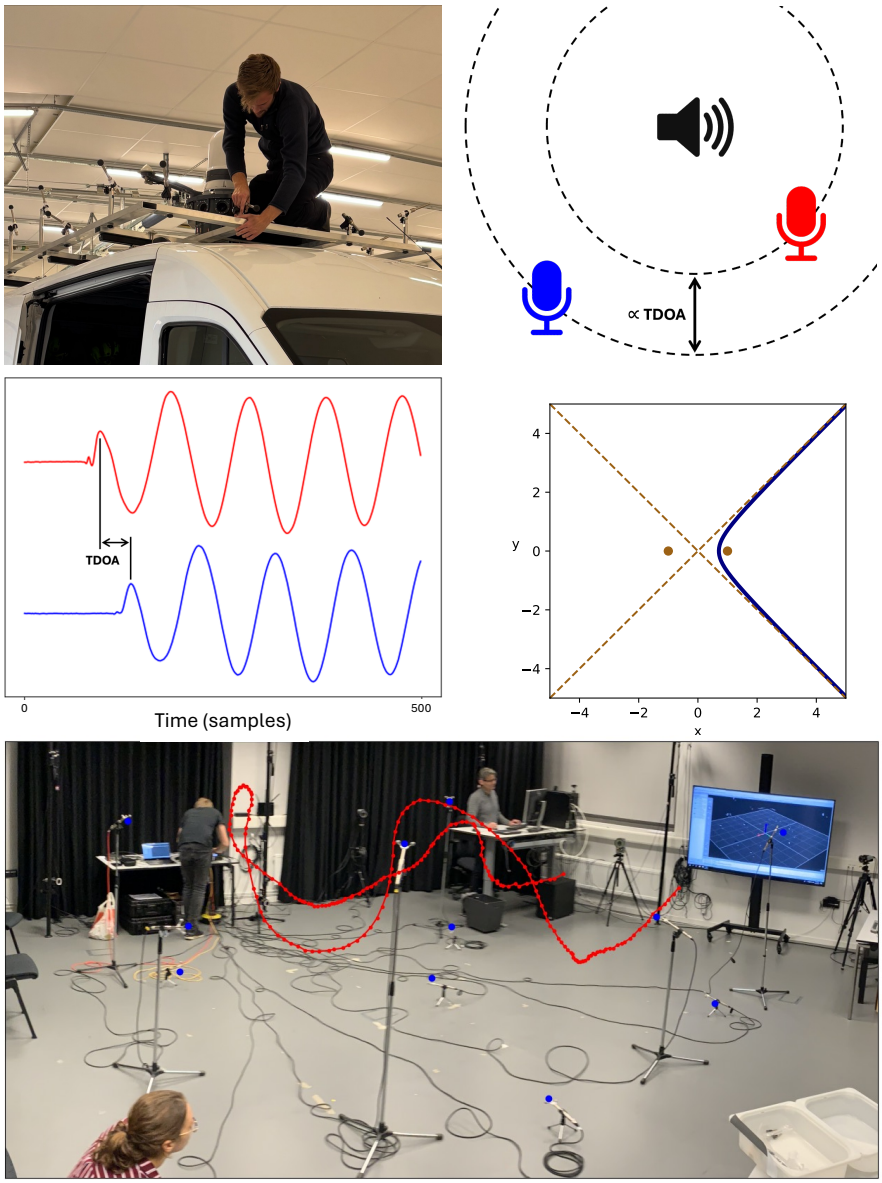


Figure P.1: (**top left**) A set of microphones with unknown positions. (**top right**) Using pairs of microphones, it is possible to measure the marked distance. (**center left**) Measuring is done by finding the corresponding time-delay in the recorded sound of the two microphones. (**center right**) This measurement reduces the possible positions of the sound source relative to the microphones. Possible locations are shown as the blue curve and the two microphones are the two bronze points. (**bottom**) Combining many of these measurements allows for the simultaneous estimation of the position of both microphones and speakers. Here the estimated microphones are marked as blue points and the red trajectory is the estimated path a speaker was moved along.

Populärvetenskaplig Sammanfattning

Föreställ dig en handling som du ofta utför. Som exempel kan vi tänka oss att du vänder dig mot en vän som har ropat på dig. För att veta åt vilket håll du ska vända dig använder du din hörsel. Om du istället vill plocka upp ett föremål behöver du använda en kombination av syn och känsel för att först placera handen ungefär rätt och sedan greppa med lagom kraft. Den gemensamma nämnaren är att du behöver använda dina sinnen för att interagera med din omvärld, det vill säga omvandla sensordata till användbar information. På samma sätt är det värdefullt att utveckla algoritmer som kan tolka sensordata, eftersom det skulle förbättra maskiners förmåga att interagera med omvärlden. Fokus för denna avhandling är hur ljud kan användas för att ta reda på information om vår omgivning. Mer specifikt fokuserar vi på att hitta positionerna för olika ljudkällor, mikrofoner eller väggar, se Figure P.2.

Hur går vi från ljud till att veta var saker befinner sig? Ett illustrativt exempel är ett blixtnedslag. Du ser först ett ljussken, och tre sekunder senare hör du åskan. Eftersom ljudet har färdats i tre sekunder och ljudets hastighet är ungefär 340 m/s kan du dra slutsatsen att avståndet till nedslaget är ungefär en kilometer. Om vi istället har två mikrofoner kan vi på liknande sätt lista ut att mikrofonen som är närmast ljudkällan kommer höra ljudet först. På detta sätt kan vi utföra signalbehandling för att gå från ljud till information om var mikrofoner och ljudkällor befinner sig.

När signalbehandlingen är klar har vi mätningar, till exempel avståndet mellan par av objekt. Dessa enkla mätningar kan sedan användas för att dra mer användbara slutsatser, till exempel den tredimensionella positionen för varje objekt. Beroende på vilken information man har till att börja med blir det olika geometriproblem att lösa. En typ av problem är att vi vet var våra mikrofoner är men vill identifiera positionen av en ljudkälla. Ett mer komplext och mindre välkänt problem är självkalibrering. Det innebär att utan förhandskunskap om var ljudkällor och mikrofoner befinner sig är det möjligt att skatta alla positioner samtidigt, endast baserat på det inspelade ljudet.

Dessa geometriska problem är utmanande eftersom de involverar riktig data. Om vi till exempel tänker oss en robotarm som automatiserar en uppgift i en fabrik, kan den utföra komplexa uppgifter med hög effektivitet. Den fungerar dock bara i en kontrollerad miljö. Om vi till exempel skulle flytta robotarmen en liten bit skulle den inte längre utföra sin uppgift, den saknar robusthet med andra ord. För de problem som beskrivits ovan är det vanligt att inte bara ha brusiga mätningar, utan även att vissa mätningar är direkt felaktiga. Även om problemen fortfarande går att lösa kräver de metoder med robusthet, hur detta kan uppnås är ämnet för denna avhandling.

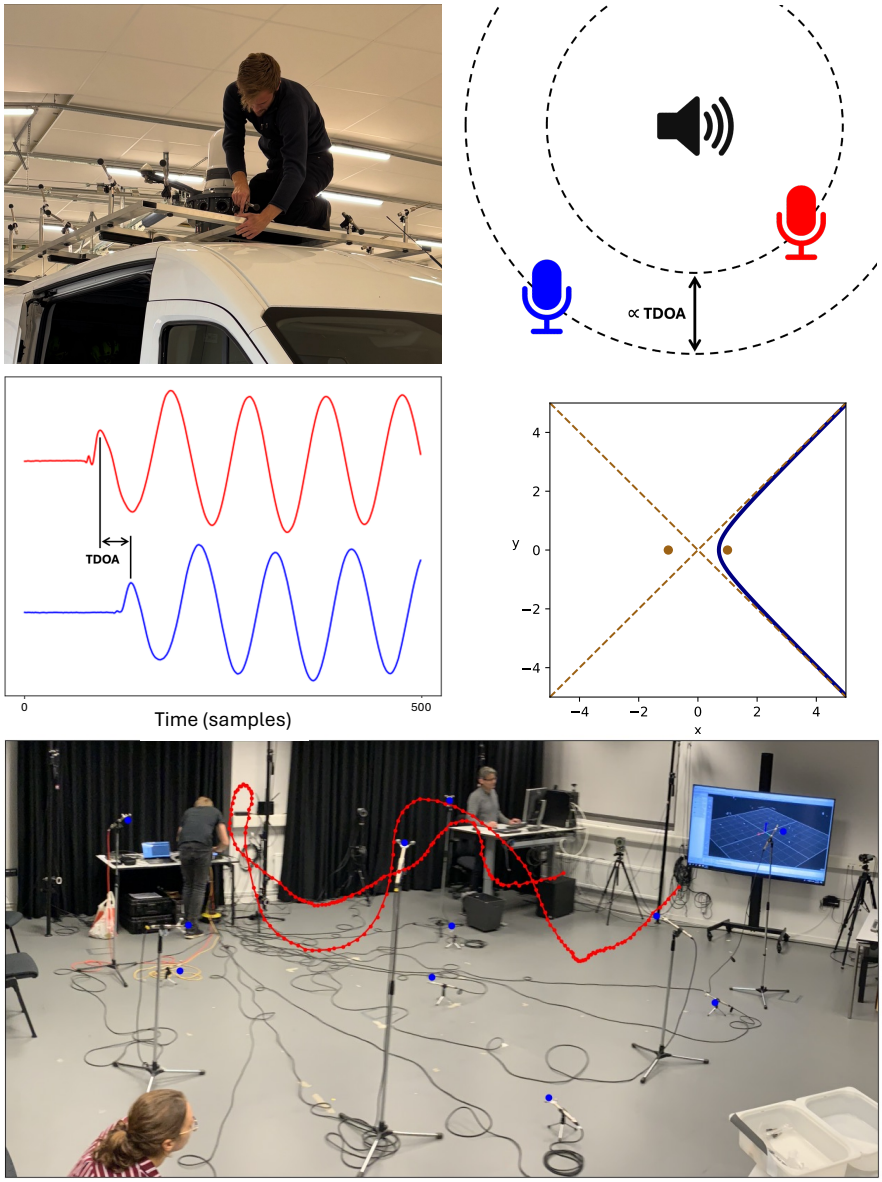


Figure P.2: (**övre vänster**) En samling av mikrofoner med okända positioner. (**övre höger**) Genom att använda par av mikrofoner är det möjligt att mäta det markerade avståndet. (**mitten vänster**) Detta görs genom att uppskatta tidsfördröjningen mellan att ljudet hörs i respektive av de två mikrofonerna. (**mitten höger**) På grund av denna mätning kan vi begränsa var ljudkällan kan befinna sig. Möjliga positioner för ljudkällan är markerat med blått och de två mikrofonerna är de bronsfärgade punkterna. (**nedre**) Genom att kombinera flera av dessa mätningar är det möjligt att samtidigt uppskatta positionerna för mikrofonerna och ljudkällorna. I bilden är de uppskattade mikrofonpositionerna markerade som blåa punkter och banan vi tror ljudkällan har rört sig längs är markerad med rött.

List of Publications

This thesis is based on the following publications, referred to by their Roman numerals. They are reproduced and included in this thesis with the permission of their respective publishers. The author's contributions to each paper is described in Chapter 6.

Main papers

I **SONNET: Enhancing Time Delay Estimation by Leveraging Simulated Audio**
E. Tegler, M. Oskarsson and K. Åström
International Conference on Pattern Recognition (ICPR), 2024.

II **Sensor Node Calibration in Presence of a Dominant Reflective Plane**
E. Tegler, M. Larsson, M. Oskarsson and K. Åström
European Signal Processing Conference (EUSIPCO), 2022.

III **The Multi-view Geometry of Parallel Cylinders**
E. Tegler, J. Engman, D. Gillsjö, G. Flood, V. Larsson, M. Oskarsson and K. Åström
Scandinavian Conference on Image Analysis (SCIA), 2023.

IV **Detection and Localization of Drones and UAVs Using Sound and Vision**
E. Tegler, M. Modig, P. Skarin, K. Åström, M. Oskarsson and G. Flood
Computer Vision and Pattern Recognition Workshops (CVPRW), 2025.

V **Minimal Solvers for Point Cloud Matching with Statistical Deformations**
G. Flood, E. Tegler, D. Gillsjö, A. Heyden and K. Åström
International Conference on Pattern Recognition (ICPR), 2022.

VI **The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization**
I. Yaman, G. Tian, M. Larsson, P. Persson, M. Sandra, A. Dürr, E. Tegler, N. Challa, H. Garde, F. Tufvesson, K. Åström, O. Edfors, S. Malkowsky, L. Liu
International Conference on Robotics and Automation (ICRA), 2024.

Auxiliary papers

i **Trilateration Using Motion Models**
M. Larsson, E. Tegler, K. Åström and M. Oskarsson
International Conference on Information Fusion (FUSION), 2022.

ii **LuViRA Dataset Validation and Discussion: Comparing Vision, Radio, and Audio Sensors for Indoor Localization**
I. Yaman, G. Tian, E. Tegler, J. Gulin, N. Challa, F. Tufvesson, O. Edfors, K. Åström, S. Malkowsky, L. Liu
Journal of Indoor and Seamless Positioning and Navigation, 2024.

Acknowledgments

I would like to take this opportunity to express my gratitude to the many people who have helped me over the past five years. Without them, this thesis would not exist for you to read.

From work, I would like to thank my supervisors for providing me with plenty of guidance: Kalle Åström for always being positive, as well as Magnus Oskarsson and Viktor Larsson for answering many of my countless questions. One of the best things about this workplace has been spending time with friends and colleagues. They have made every fika a pleasure and I want to thank them for the many laughs we have shared. I want to thank Magnus Fries for many games of Go and Felix Augustsson for giving me plenty of new perspectives.

Outside of work, I want to thank my friends for keeping me motivated and for ensuring that no plan or project is ever too normal. I want to thank Linn Rydberg for brightening my days and, I presume, developing the right kind of selective hearing and patience to put up with me during trying times. Finally, I want to thank my family for their stability and for always being there for me.

Funding

The following organizations have supported this work.

- The strategic research project ELLIIT provided the main funding.
- The Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
- Computational resources were provided by the Swedish National Infrastructure for Computing at C3SE and NSC, partially funded by the Swedish Research Council, grant agreement no. 2018- 05973.
- The Swedish Foundation for Strategic Research project, Semantic Mapping and Visual Navigation for Smart Robots (grant no. RIT15-0038).
- The ADACORSA project with funding from ECSEL JU in the H2020 Framework Programme (H2020/2014-2020) and National Authorities, under GA 876019.
- Royal Physiographic Society of Lund.
- Sweden's Innovation Agency VINNOVA and the Swedish Armed Forces within the programme "Civil-Militärt Innovationsprogram", project nr 2024-03191.

Contents

Background and Research Context

Chapter 1

Introduction

Interpreting sensory data is a fundamental problem shared across biological and artificial systems. Humans use information from multiple senses to perform everyday tasks such as maintaining balance or locating objects. Similarly, improving the perception of artificial systems is a key problem in expanding the domain in which these systems can operate, which would increase their usefulness. While there are many sensory modalities and tasks available, the main research question of this thesis is how to infer geometric information using sound.

Localization is the task of finding the position of objects in some reference frame. While vision has traditionally dominated research in spatial perception and scene understanding [1–4], sound offers a complementary and often underexplored modality. Acoustic sensing can provide information in situations where vision is unreliable, such as in darkness or when objects are occluded. While sound is unlikely to replace vision for most tasks, a combination of sensors can leverage the advantages of both modalities. A biological example of this is how humans use sound to decide which direction to look. Unlike sensing methods such as radar [5] or lidar [6], sound can be used passively, i.e. without emitting signals, making it both energy-efficient and unobtrusive. These characteristics make sound a relatively inexpensive and versatile source of spatial information, motivating the study of how to extract geometric knowledge from acoustic data.

The initial challenge in acoustic localization lies in transforming the recorded signals into primitive geometric relations between objects. One example of how this can be done is to compare the arrival times of a sound at two microphones. Using the difference in arrival times together with the speed of sound, we get a constraint on the relative positions of the sound source and the two microphones. Other information can also be extracted, depending on what prior knowledge is available.

The question of using low-level geometric relationships to reconstruct high-level spatial information gives rise to several classes of problems. One example is triangulation, where the positions of sound sources are estimated using known positions of microphones. Another example is self-calibration, where the positions of both sound sources and microphones are estimated simultaneously. Self-calibration is particularly challenging because the geometry must be inferred solely from the data, without prior spatial reference. An example of a problem setup is shown in Figure 1.1.

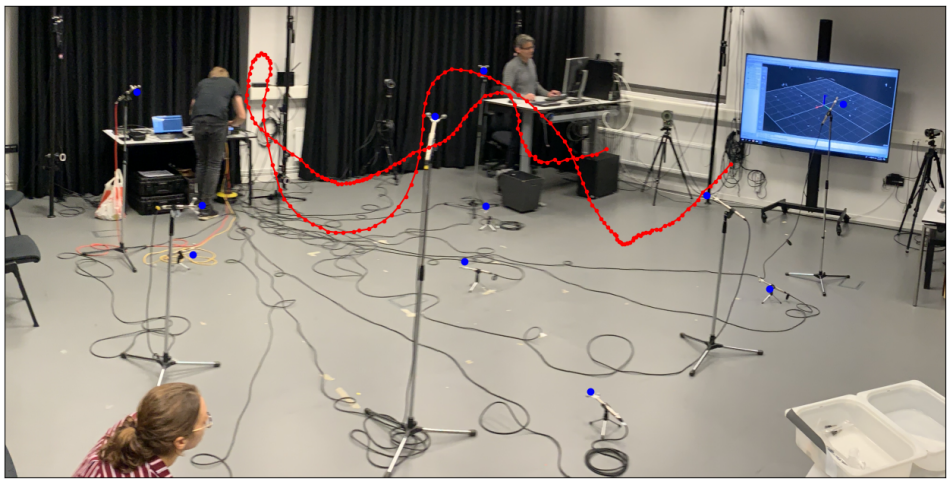


Figure 1.1: The image shows an experiment setup and result for the problem of self-calibration. The blue points mark estimated microphone locations and the red curve shows the estimated path a speaker moved along while emitting sound.

A key difficulty in using the measurements is that they are imperfect. Not only in the sense of noise, but measurements are also outliers some fraction of the time, i.e. entirely incorrect. This is the case in real-world environments, where sound propagation is affected by reverberation, noise, and sensor uncertainty. Because of this, the estimation techniques used need to be robust to outliers, which makes the estimation problem significantly harder.

As explained above, robust localization is essential in order to interact with the physical world and sound is one of multiple sensor modalities useful to that end. However, there are also some applications where acoustic localization is particularly advantageous. Examples of these applications are: active acoustics e.g. using speakers and microphones to affect the acoustics of an auditorium [7], virtual perception [8], and active noise control (also known as noise-cancelling) in a point or volume [9]. However, existing localization methods need more robustness to function in a real-world setting. The aim of this thesis is therefore to develop and improve methods for robust geometric inference from acoustic data, enabling accurate localization and calibration.

While the common thread among the papers in this thesis is that they can be used for acoustic localization, individually they can also be applied to other areas. The most common such area is computer vision, which is exemplified by the fact that two of the papers are framed as computer vision problems. The reason for the overlap with computer vision is that both areas share challenges such as solving geometric problems using measurements that are outlier-prone.

This thesis is made up of a collection of papers, preceded by an introductory part where relevant background information is presented with the aim of being accessible to new PhD students. The introductory part consists of the following chapters. In Chapter 2, sound and how it propagates are described. Chapter 3 contains an introduction to statistics, robustness, and a brief description of machine learning with a focus on energy-based models. Using the background from previous chapters, Chapter 4 presents the geometric estimation problems of interest. In Chapter 5, the theory needed to create minimal solvers is described. This is a key building block when performing robust estimation. Finally, Chapter 6 describes each paper in its scientific context.

Chapter 2

Sound

2.1 The Wave Equation

In this thesis, we infer information about the world from sound. Thus, a good starting point is to gain some understanding of sound and how it propagates. While the underlying physics is not directly the topic of the papers presented in this thesis, it is nevertheless valuable background information to have. Also, sound propagation is a large subject, and as such this is only a brief introduction discussing some aspects. See [10–12] for more information.

Qualitatively, when a note is played, it creates a change in air pressure in one location. This oscillation then travels through the air until it interacts with our eardrums, which is interpreted as a sound. In order to make this description quantitative, we introduce the pressure field $p(\mathbf{x}, t) : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}$, which is the pressure at a position $\mathbf{x}$ at time t. The propagation of sound is then governed by the wave equation

$$\frac{\partial^2 p}{\partial t^2} = c^2 \nabla^2 p, \tag{2.1}$$

where c is the speed of sound in the medium and

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}, \tag{2.2}$$

is the Laplace operator. Further insight can be gained by seeing how the wave equation is derived from the laws of mechanics. For the interested reader, included below is a derivation of the wave equation in 1D based on lecture 47 of the Feynman Lectures on Physics [13].

2.1.1 Derivation of the Wave Equation

First, the problem can be broken down into the three following subproblems:

(I) A change in density will affect the pressure.

(II) The displacement of air will change its density.

(III) Pressure differences result in a net force which causes air motion.

Consider subproblem (I) first. The pressure is connected to the density by some function $f(\rho)$ and before we have perturbations in the pressure field we have some equilibrium pressure p_0 and density ρ_0 such that $p_0 = f(\rho_0)$. We split the pressure and density into an equilibrium part and an excess part

$$p = p_0 + p_e, \quad \rho = \rho_0 + \rho_e. \tag{2.3}$$

Assuming the changes in density and pressure are small, we can use a first-order approximation

$$p = p_0 + p_e = f(\rho_0 + \rho_e) = f(\rho_0) + \rho_e \frac{df}{d\rho}\Big|_{\rho=\rho_0}. \tag{2.4}$$

By introducing the constant

$$\kappa = \frac{df}{d\rho}\Big|_{\rho=\rho_0}, \tag{2.5}$$

the relation between excess pressure and density is

$$p_e = \kappa \rho_e. \tag{I}$$

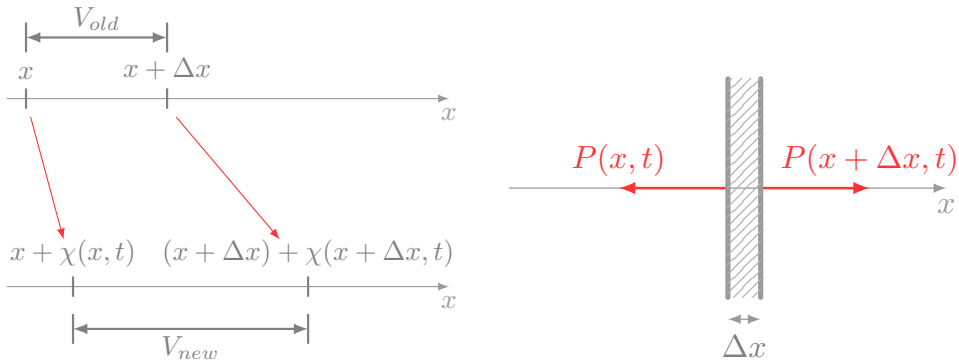
In the case of subproblem (II) we have some displacement $\chi(x,t)$, monotonic in x, which describes how the air that was initially at position x has moved at time t. In order to find the change in density we study the change in volume of the air between x and $x + \Delta x$, see Figure 2.1a. Initially, this air had a volume

$$V_{old} = (x + \Delta x) - x = \Delta x. \tag{2.6}$$

After being displaced, the same air now occupies

$$V_{new} = (x + \Delta x + \chi(x + \Delta x, t)) - (x + \chi(x,t)) \tag{2.7}$$

$$= \Delta x + \chi(x + \Delta x, t) - \chi(x,t). \tag{2.8}$$



(a) **(b)**

Figure 2.1: **(a)** Sketch showing how a displacement field corresponds to a volume a slice of air occupies. **(b)** Sketch showing that the net force on a slice of air is proportional to the derivative of the pressure function.

Since mass is conserved, we know the density $\rho = \rho_0 + \rho_e$ fulfills

$$\rho_0 \Delta x = \rho(\Delta x + \chi(x + \Delta x, t) - \chi(x, t)). \tag{2.9}$$

By dividing both sides by Δx and taking the limit $\Delta x \to 0$ we get

$$\rho_0 = (\rho_0 + \rho_e)\left(1 + \frac{\partial \chi(x,t)}{\partial x}\right), \tag{2.10}$$

which can be rewritten as

$$\rho_e \left(1 + \frac{\partial \chi(x,t)}{\partial x}\right) = -\rho_0 \frac{\partial \chi(x,t)}{\partial x}. \tag{2.11}$$

In the case of sound waves we assume that $\frac{\partial \chi(x,t)}{\partial x} \ll 1$, which is reasonable since the pressure changes from sound are mu§ch smaller than the equilibrium pressure. This means we can approximate

$$\rho_e = -\rho_0 \frac{\partial \chi(x,t)}{\partial x}, \tag{II}$$

which is how we will model air movement's effect on density.

For subproblem (III), again consider a small section of air between x and $x + \Delta x$, see Figure 2.1b. The net force on this air is

$$p_e(x,t) - p_e(x + \Delta x, t), \tag{2.12}$$

note that pressure and force in 1D are equivalent. Combining the net force with Newton's second law of motion yields

$$p_e(x,t) - p_e(x+\Delta x,t) = \rho_0 \Delta x \frac{\partial^2 \chi(x,t)}{\partial t^2}. \tag{2.13}$$

Again, consider the limiting case when $\Delta x \to 0$

$$-\frac{\partial p_e(x,t)}{\partial x} = \rho_0 \frac{\partial^2 \chi(x,t)}{\partial t^2}. \tag{III}$$

We can now combine the three relations found to get the wave equation. First, combining (III) and (I) gives

$$-\kappa \frac{\partial \rho_e(x,t)}{\partial x} = \rho_0 \frac{\partial^2 \chi(x,t)}{\partial t^2}. \tag{2.14}$$

We can then further reduce this expression using (II)

$$\kappa \frac{\partial^2 \chi(x,t)}{\partial x^2} = \frac{\partial^2 \chi(x,t)}{\partial t^2}. \tag{2.15}$$

This can then be turned into a relationship of pressure by differentiating both sides by x and using (I) and (II) as

$$p_e = -\kappa \rho_0 \frac{\partial \chi(x,t)}{\partial x}, \tag{2.16}$$

to get

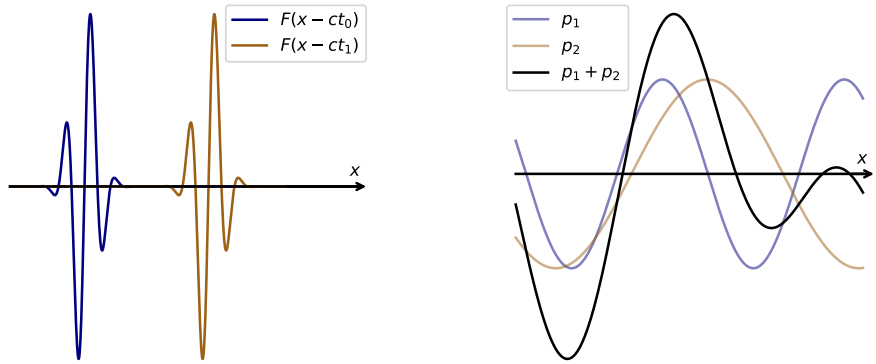
$$\kappa \frac{\partial^2 p_e(x,t)}{\partial x^2} = \frac{\partial^2 p_e(x,t)}{\partial t^2}, \tag{2.17}$$

which is the wave equation in 1D. Since adding a constant to p_e will not change whether it solves the wave equation, we can add back the equilibrium pressure p_0 to the excess pressure p_e. I will therefore hereafter write p and only refer to pressure and not specify if it is excess pressure or not.

The general solution to the wave equation in 1D was derived by Jean le Rond d'Alembert [14]. The solution takes the form

$$p(x,t) = F(x-ct) + G(x+ct), \tag{2.18}$$

where F and G are arbitrary 1D functions. The interpretation of this is that the solution corresponds to a wave F traveling in the positive x-direction and a wave G traveling in the negative x-direction, see Figure 2.2a. Both waves have a speed of c since a unit change in t results in translating the function in x by c. That this is a solution to the wave equation can easily be shown by substitution.



(a) (b)

Figure 2.2: **(a)** Demonstration of how $F(x-ct)$ can be interpreted as a wave traveling in the positive x-direction. Plotted is the function $F(x-ct) = Ce^{(x-ct)^2}\sin(kx-kct)$, for the times $t_1 > t_0$. **(b)** Since derivation is a linear operator, the principle of superposition states that the linear combination of two solutions is also a solution (assuming all boundary conditions hold).

2.2 Propagation from Point Source

Another interesting problem to study is how sound propagates from a point source in $\mathbb{R}^3$. We will model the point source as an oscillating boundary condition on a small sphere of radius ϵ, which starts oscillating at a fixed time $t_0 = \frac{\epsilon}{c}$. The goal is then solving for the $p(x,t)$ outside this boundary. Consider the problem

$$\begin{cases} c^2\nabla^2 p = \dfrac{\partial^2 p}{\partial t^2}, & x \in \Omega, \\ p(x,t) = \dfrac{1}{\epsilon}\sin\left(t - \dfrac{\epsilon}{c}\right), & x \in \partial\Omega, \quad t \geq t_0, \\ p(x,t) = 0, & x \in \Omega, \quad t < t_0, \end{cases} \tag{2.19}$$

with the domain

$$\Omega = \{x \in \mathbb{R}^3 \mid \epsilon < ||x||\}. \tag{2.20}$$

The boundary condition corresponding to the oscillation might look more convoluted than necessary, but the reason for this will become apparent soon. Since the problem is symmetric with respect to rotations we change to spherical coordinates, since the derivatives with

respect to θ and ϕ can be ignored

$$\begin{cases} c^2 \dfrac{\partial^2}{\partial r^2}(pr) = \dfrac{\partial^2}{\partial t^2}(pr), & r > \epsilon, \\ p(\epsilon, t) = \dfrac{1}{\epsilon} \sin\left(t - \dfrac{\epsilon}{c}\right), & t \geq t_0, \\ p(r, t) = 0, & r > \epsilon, \quad t < t_0. \end{cases} \tag{2.21}$$

Set $f(r, t) = p(r, t)r$ to get

$$\begin{cases} c^2 \dfrac{\partial^2 f}{\partial r^2} = \dfrac{\partial^2 f}{\partial t^2}, & r > \epsilon, \\ f(\epsilon, t) = \sin\left(t - \dfrac{\epsilon}{c}\right), & t \geq t_0, \\ f(r, t) = 0, & r > \epsilon, \quad t < t_0, \end{cases} \tag{2.22}$$

which as we have seen before is the wave equation in 1D. This problem can be solved using separation of variables and the solution is given by d'Alembert's formula [14].

$$f(r, t) = F(r - ct) + G(r + ct), \tag{2.23}$$

where $F, G : \mathbb{R} \to \mathbb{R}$. We can find F and G by using the initial condition

$$f(r, t) = 0, \quad r > \epsilon, \quad t < t_0, \tag{2.24}$$

which gives

$$F(a) + G(b) = 0, \quad a > 0, \quad b \in \mathbb{R}, \tag{2.25}$$

and thus

$$\begin{cases} F(a) = C, & a > 0, \\ G(b) = -C, & b \in \mathbb{R}, \end{cases} \tag{2.26}$$

where C is a constant. All that is left is to solve for $F(a)$ when $a \leq 0$, by using the boundary condition, we get

$$f(\epsilon, t) = \sin\left(t - \frac{\epsilon}{c}\right) = F(\epsilon - ct) - C, \quad t \geq t_0, \tag{2.27}$$

so

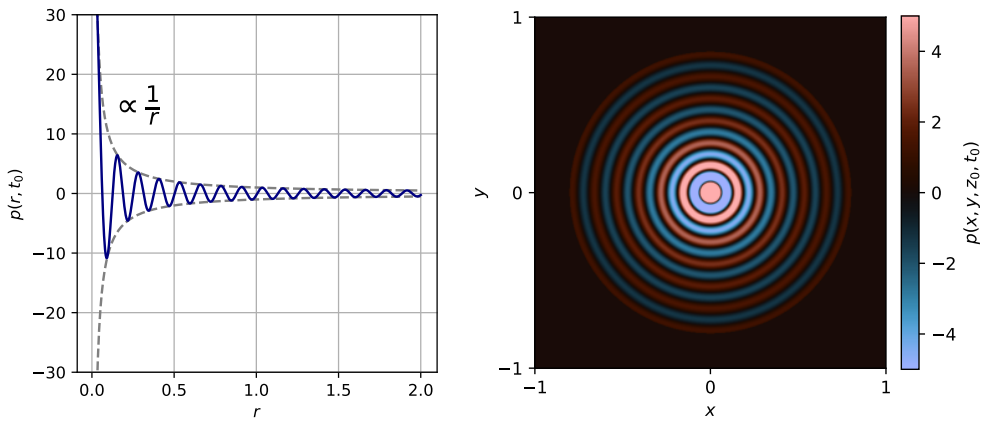
$$F(a) = \sin\left(-\frac{a}{c}\right) + C, \quad a \leq 0. \tag{2.28}$$

Combining everything together we get our final solution

$$p(r, t) = \begin{cases} \dfrac{1}{r} \sin\left(\dfrac{ct - r}{c}\right) & , \quad ct - r \geq 0, \quad t \geq t_0, \quad r > \epsilon, \\ 0 & , \quad \text{else.} \end{cases} \tag{2.29}$$

Notice that the solution does not depend on ϵ. In particular, we can let ϵ get arbitrarily close to 0, in which case we have a point source which oscillates with an arbitrarily high amplitude. It is also possible to do this derivation for the limit case, however, we would then need to work with distributions instead of functions; see [15] for an example of this. The solution is shown in Figure 2.3.

One key takeaway from this derivation is that the amplitude of the wave decreases with distance proportionally to $\frac{1}{r}$. This is usually explained by using an energy conservation argument on larger and larger spheres. Since the energy of a wave scales as the square of the amplitude, and thus decreases proportionally to $\frac{1}{r^2}$, this is known as the inverse-square law [11, 16]. However, as shown, we could also arrive at this result from the wave equation.



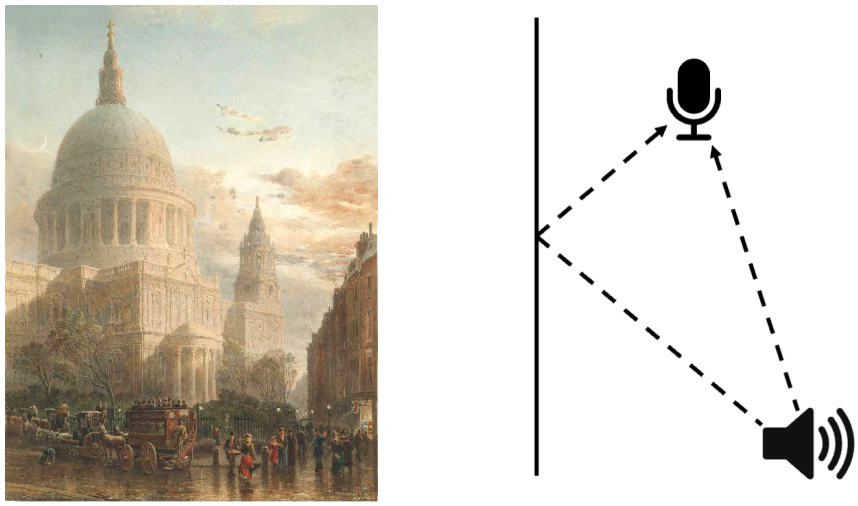
(a) (b)

Figure 2.3: **(a)** Pressure $p(r, t_0)$, shown as a function of distance to the point source. The fixed time t_0 is large enough for the wave to have propagated beyond $r = 2$ shown in the figure. The amplitude decreases in proportion to $\frac{1}{r}$. **(b)** Pressure $p(x, y, z_0, t_0)$ shown in the 2D slice $z_0 = 0$. Here t_0 is a smaller value, so the wave has not propagated to the edge of the image. Note that the color values shown are truncated since very large values are found close to the origin.

2.3 Reverberations

So far we have discussed how sound propagates in an open environment consisting solely of air. However, when objects are present this will affect how the sound propagates. This is called reverberation, and it is the reason why different acoustics are found in a stone church compared to a small closet. Even though the same initial sound was produced in both environments, the pressure field will be different because of the interaction with the

walls. A spectacular example of this can be observed in the whispering gallery of St Paul's Cathedral in London, see Figure 2.4a. Sound waves travel along the circular wall of the dome allowing whispers originating close to the wall to be heard on the other side of the dome [17].



(a) (b)

Figure 2.4: **(a)** Painting of St Paul's Cathedral. The whispering gallery beneath the dome is a notable example of a reverberation effect. **(b)** Sketch of an environment consisting of a wall, speaker, and microphone. Also shown are the two paths which the sound travels along. This sketch corresponds to the setup used in Figure 2.5.

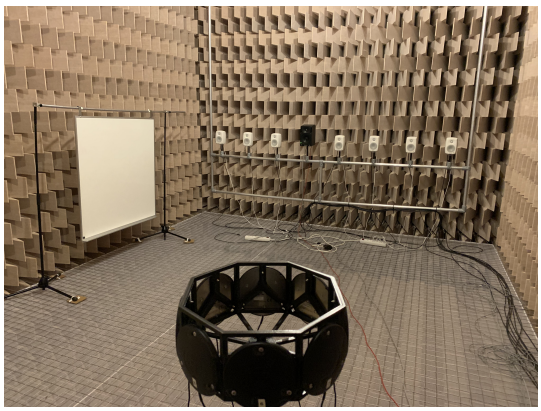
An example of the same phenomenon is echoes. An echo is a repetition of the sound after a short delay and typically occurs when walls or other surfaces are present in the environment. The sound can then be reflected by the walls, creating multiple possible paths from the speaker to the listener, see Figure 2.4b. Since the two paths have different lengths, the sound will arrive at the listener at two different times, creating an echo. By performing measurements with and without a single wall in an anechoic (i.e. soundproof) chamber, it is possible to clearly separate the direct path from the echo, see Figure 2.5.

We can also study reverberation theoretically. One way of doing this is to model the air next to the wall as having zero displacement in the normal direction $\boldsymbol{n}$ of the wall, as

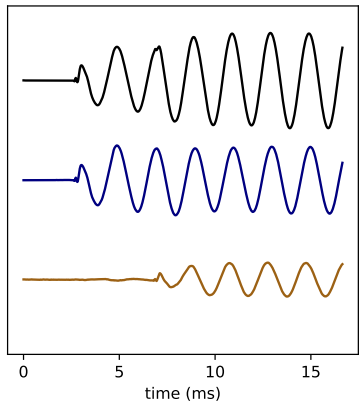
$$\frac{\partial p}{\partial \boldsymbol{n}} = 0. \tag{2.30}$$

An example problem in 3D is when there is a point source located at $(1, 0, 0)$ and the yz-plane is a wall, and solving the problem on the domain

$$\Omega = \{(x, y, z) \in \mathbb{R}^3 \mid x > 0\}, \tag{2.31}$$



(a)



(b)

Figure 2.5: **(a)** Recording setup in an anechoic chamber. The octagon closest to the camera is the microphone and the black box on the opposite wall is the speaker. The whiteboard on the left was present during one recording and then removed from the room for a second recording without moving any other part of the setup. **(b)** Sound recording, i.e. air pressure as a function of time, with the board present (black), without the board present (blue). The difference between the recordings (bronze) is well described as a shifted and scaled version of the recorded sound without the board.

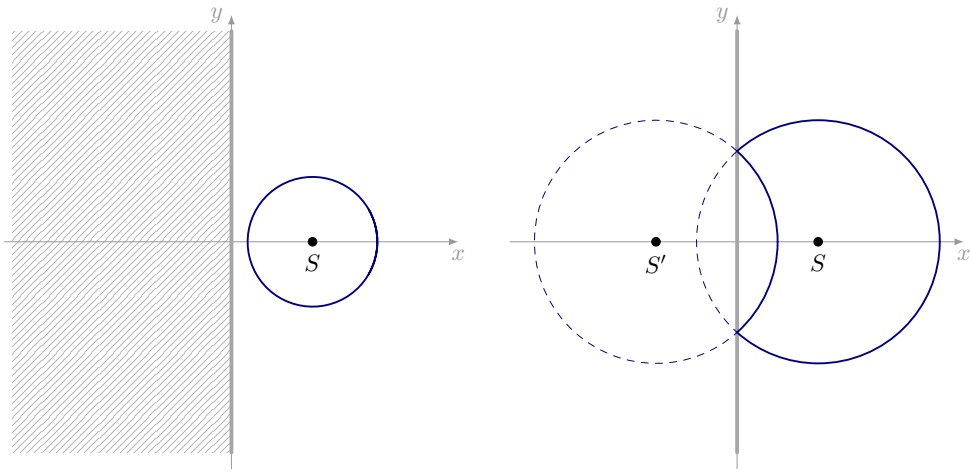
is of interest. Here the problem can be solved by extending the domain and introducing an additional point source at $(-1, 0, 0)$ producing the same sound as the first source. Since the problem is now symmetric with respect to the reflection in the yz-plane, we know that the solution will also be symmetric. This means that the boundary condition $\frac{\partial p}{\partial \boldsymbol{n}} = 0$ in the yz-plane is fulfilled. The problem now becomes solving the wave equation for two point sources. Since the equations are linear, we can simply add the solutions for each of the point sources to get the solution, see Figure 2.2b. The setup and solution are shown in Figure 2.6.

2.3.1 Impulse Response

If we have a single reverberant plane causing the echo, viewing the received sound $p_r(t)$ as a combination of the played sound $p_s(t)$ and a delayed and scaled version of the played sound

$$p_r(t) = a_1 p_s(t - \Delta t_1) + a_2 p_s(t - \Delta t_2), \tag{2.32}$$

is a useful way to view what is happening. However, if we introduce a second reverberant surface we now not only get another term corresponding to echoes from the new surface,



(a) (b)

Figure 2.6: Example of wave propagation in the presence of a wall (y-axis). The sound source S has emitted a pulse, shown in blue. (**a**) The wavefront of the pulse before it arrives at the wall, it propagates radially. (**b**) Sketch shows the same pulse after hitting the wall. The interaction with the wall results in a sound field which is equal to introducing a mirrored sound source S' and pretending that there is no wall.

we also get higher-order echoes where the sound interacts with multiple surfaces before arriving at the microphone

$$p_r(t) = \sum_i a_i p_s(t - \Delta t_i). \tag{2.33}$$

In the general case, the number of paths could be uncountable, and so this summation is typically described as a convolution instead

$$p_r = h * p_s, \tag{2.34}$$

where h is referred to as the room impulse response. Naturally, it depends on the environment as well as the position of the microphone and sound source.

2.4 More Properties of Sound

Many of the sound phenomena we observe can be derived as consequences of the wave equation, and thus the wave equation contains much of the information needed to understand sound. It is nevertheless useful to look further at some of these phenomena to better understand how sound propagates.

2.4.1 Propagation Speed

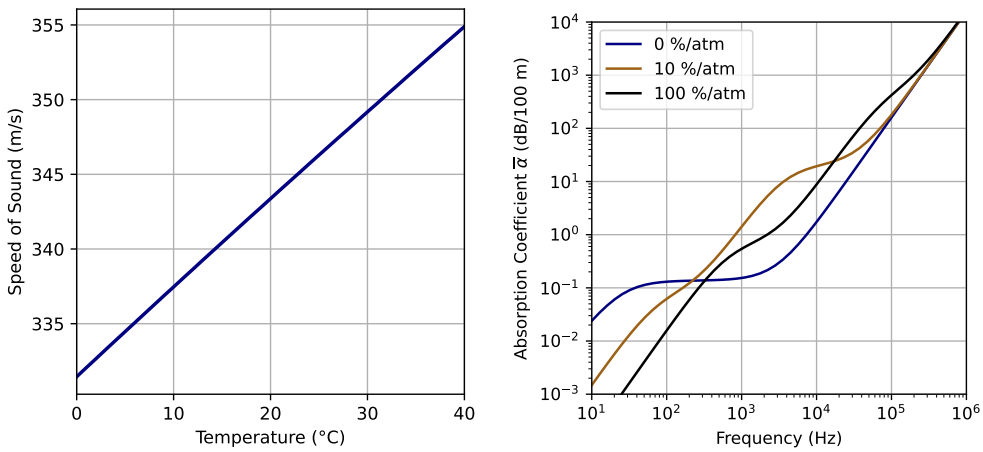
From our derivation earlier we saw that the propagation speed was related to how pressure changes as a function of density

$$c = \sqrt{\kappa} = \sqrt{\frac{df}{d\rho}\Big|_{\rho=\rho_0}}. \tag{2.35}$$

Assuming this function f comes from the ideal gas law, the speed of sound then depends on the temperature of the air as

$$c \propto \sqrt{T}. \tag{2.36}$$

We can compare this to measurements of the speed of sound, see Figure 2.7a. In reality, the speed of sound depends on more factors, e.g. humidity, see [18] for a more detailed exposition of the subject. Later on in this thesis, the travel time of sound is used to measure distance. This means that the uncertainty in the air temperature will give an uncertainty in our distance measurements. Although this is a source of error, it should be noted that sound, in conjunction with other sensors such as vision, could potentially be used to measure temperature.



(a) (b)

Figure 2.7: (**a**) Speed of sound in dry air as a function of temperature, data from [18]. Although the data looks quite close to a line, using a square-root model leads to a better fit of the data. (**b**) Attenuation at 20 °C for three levels of relative humidity, based on model from [19]. From the plot we can conclude that lower frequencies travel further than higher frequencies, which explains why the distant rumble of thunder sounds different from thunder which originated close by.

2.4.2 Diffraction

While both sound and light are waves, we have to be careful about making the same assumption when modeling them. For multiple applications such as ray tracing [20], light is treated as a ray. This means if you put an object between you and a light source, you no longer see any light. If you switch out the light source for a sound source the object no longer stops you from hearing the sound source (even if it is a perfect sound absorber). This is merely a consequence of the wave equation; however, the phenomenon is usually referred to as diffraction. The reason we observe different behavior in sound and light is that diffraction is wavelength-dependent and because light has a much shorter wavelength, the effects of diffraction can in many cases be ignored. When dealing with high-frequency sound this assumption can also be made, this area of study is called geometrical acoustics [21].

2.4.3 Attenuation

While the wave equation is a good model and captures many aspects of how sound propagates, it is nevertheless just a model with assumptions and simplifications made in modeling. This means not all aspects of real sound propagation are captured. One such example is attenuation. If an experiment with a point source is performed, we would see that the amplitude of the oscillations would decrease faster with distance than the theory predicts. One reason for this is that we have not modeled attenuation, which is that some energy gets lost due to the viscosity and heat dissipation in air [19]. This effect obviously depends on many factors, one of which is the frequency of the sound. To get an understanding of the size of this effect, see Figure 2.7b.

Chapter 3

Statistics and Machine Learning

3.1 Estimation of Parameters

This chapter contains an introduction to statistics, since we are interested in performing inference from data. For more complete source material, see [22, 23]. In the typical statistical inference problem we have an unknown distribution $\mathcal{D}$ and data x sampled from that distribution

$$x = (x_1, \ldots, x_n), \quad x_i \in \mathcal{D}. \tag{3.1}$$

The goal is to infer some information about $\mathcal{D}$ from the data. One problem is that $\mathcal{D}$ is unknown and could therefore in theory be almost anything, regardless of the observed data x. This is known as the problem of induction [24]. Because of this, we typically make some assumptions about $\mathcal{D}$, and as such our conclusions will be based on those assumptions being true.

As an example, consider the problem of inferring the mean, θ from the measurements

$$x \in \mathbb{R}^n, \quad x_i \in \mathcal{N}(\theta, 1). \tag{3.2}$$

The values we estimate, θ in this case, are typically referred to as parameters. One approach to solve this problem is to use Bayesian inference. If we have a belief $p(\theta)$, known as a prior, about the probability of different values of θ before getting our data, then based on the data x we should update our belief using Bayes' rule

$$P(\theta|x) \propto P(x|\theta)P(\theta), \tag{3.3}$$

where $P(x|\theta)$ is the probability (or probability density) of our data x given θ. When $P(x|\theta)$ is viewed as a function of θ it is referred to as the likelihood function. Thus, if

we initially believe all values of θ are equally likely, then after observing the data we should believe the most probable value of θ to be the value which maximizes the likelihood $P(x|\theta)$. This method is appropriately named maximum likelihood estimation (MLE).

In our example this would mean choosing our parameter to be

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \prod_i P_\theta(x_i) = \underset{\theta}{\operatorname{argmax}} \prod_i \frac{1}{\sqrt{2\pi}} e^{-(x_i-\theta)^2}. \tag{3.4}$$

Finding θ^* is done by first using that the logarithm is monotonic, so that

$$\underset{\theta}{\operatorname{argmax}} \prod_i \frac{1}{\sqrt{2\pi}} e^{-(x_i-\theta)^2} = \underset{\theta}{\operatorname{argmax}} \sum_i \ln\left(\frac{1}{\sqrt{2\pi}} e^{-(x_i-\theta)^2}\right), \tag{3.5}$$

then the expression can be further reduced to

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_i (x_i - \theta)^2, \tag{3.6}$$

which is known as the least squares method. Both maximum likelihood and least squares are common terms. One way to view how they are connected is that the least squares method is performing MLE in the case of a normal distribution as shown above.

In our example, solving the least squares problem is done by computing the mean of the data. However, it is also possible to find a solution in the larger class of linear problems, which is the class of problems that can be formulated as a matrix equation

$$\underset{\theta}{\operatorname{argmin}} ||A\theta - b||_2, \quad A \in \mathbb{R}^{m\times n}, \quad \theta \in \mathbb{R}^n, \quad b \in \mathbb{R}^m, \tag{3.7}$$

where A and b are the data and θ are the parameters. The solution can be found in closed form, using Moore-Penrose inverse [25]. In the non-linear case, a closed-form solution cannot generally be found, a common approach is to use local optimization instead.

3.2 Local Optimization

Local optimization refers to a process with the goal of finding the value θ which optimizes a given function $L(\theta)$ using local properties such as gradients. The best value of θ is referred to as the global optimum. In the case of trying to find a minimum, the function we are optimizing over is often referred to as a loss function. For example, consider the problem from the previous section, where θ^* is the global optimum of the loss function

$$L(\theta) = \sum_i (x_i - \theta)^2. \tag{3.8}$$

One commonly used process, Gradient Descent, consists of starting at an arbitrary point θ_0. The point is then iteratively updated using

$$\theta_{i+1} = \theta_i - \gamma \nabla L(\theta_i), \tag{3.9}$$

where γ is a small positive number, typically referred to as step size or learning rate. By stepping in the direction of the negative gradient, the idea is to get progressively lower values of $L(\theta_i)$, hopefully ending up close to θ^*. For small changes $\Delta\theta$

$$L(\theta + \Delta\theta) \approx L(\theta) + \Delta\theta \nabla L(\theta), \tag{3.10}$$

and thus the update results in θ_{i+1} such that

$$L(\theta_{i+1}) < L(\theta_i), \tag{3.11}$$

given a small enough step size γ.

One important property when discussing local optimization is if the loss function is convex. A function L is convex if for any two points θ_1, θ_2 we know for all $\lambda \in [0, 1]$, we know $\lambda\theta_1 + (1 - \lambda)\theta_2$ exists and

$$\lambda L(\theta_1) + (1 - \lambda) L(\theta_2) \geq L(\lambda\theta_1 + (1 - \lambda)\theta_2). \tag{3.12}$$

Optimization on convex function is the topic of convex optimization, and it can in some circumstances provide guarantees of convergence to the global optimum [26]. An example of convex problems is the class of linear problems described above.

3.3 Robust Estimation

In Robust estimation, the goal is to perform estimation even when the data distribution contains some fraction of measurements that have large errors. Here, large error does not mean normally distributed with a large standard deviation, rather, it should be thought of as data points not having any connection to what it is measuring. These are referred to as outliers. Instead of following the assumed noise distribution, outliers are drawn from some unknown distribution. As an example, consider the problem of linear regression. We have measurements (x_i, y_i) where

$$x_i \in U(0, 1) \quad \text{and} \quad y_i = kx_i + m + \epsilon, \quad \epsilon \in \mathcal{N}(0, \sigma^2). \tag{3.13}$$

An instance of this problem is shown in Figure 3.1a. In Figure 3.1b, 40% of the samples have been replaced with outliers (outliers are actually $y_i \in U(0, 1)$ in this example, but that is unknown). The goal here is to estimate the parameters $\theta = \{k, m\}$ of the model

$f_\theta(x) = kx + m$. For the outlier-free case, this can be done using linear least squares. However, using linear least squares in the case with outliers does not result in estimated parameters close to the true parameters. The outliers have a large impact on the model parameters. Repeating the experiment many times we can study how often the estimated parameters are close to the underlying parameters, see Figure 3.2a.

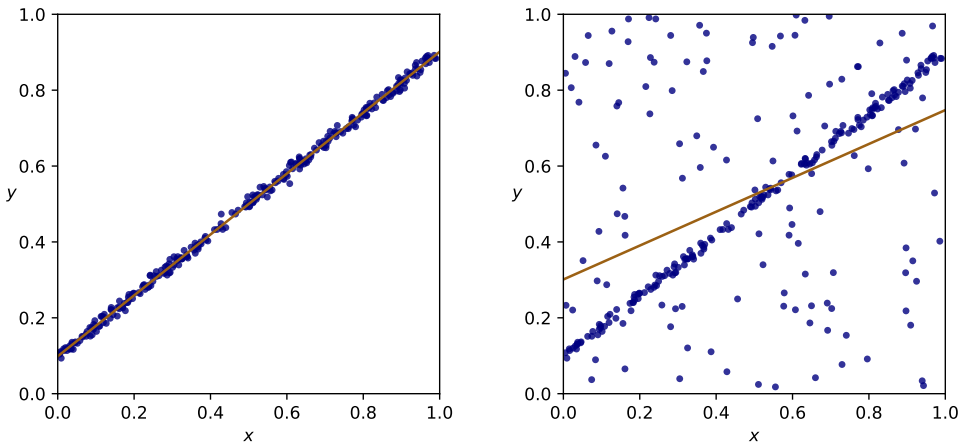


Figure 3.1: Estimation of parameters from line model using the least square method on: **(a)** data free from outliers, **(b)** data containing 40% outliers.

3.3.1 Robust Loss Function

Since we want our estimate to be close to the true parameters of the underlying distribution, we have to make some change to the optimization problem. Currently, our loss function looks like

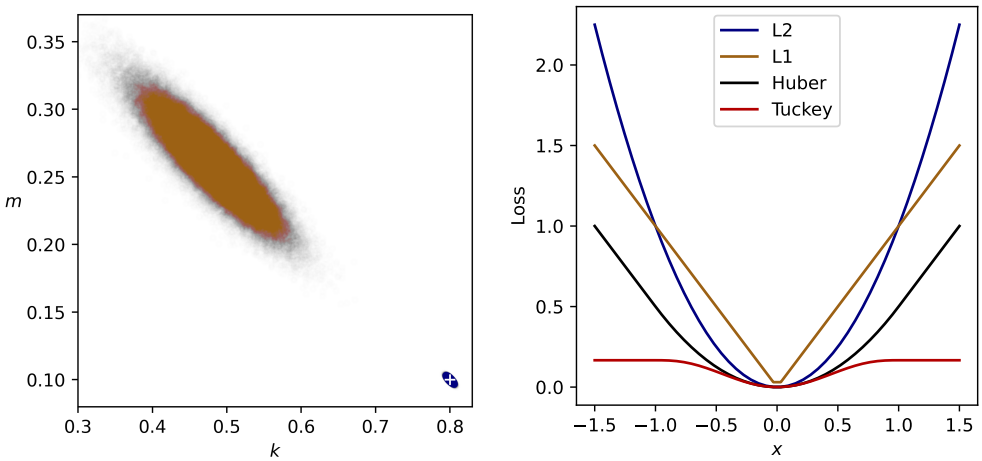
$$L(\theta) = \sum_i l(f_\theta(x_i) - y_i), \tag{3.14}$$

where $l(x) = x^2$ and maps the residual into a loss and is called L2 loss. One way of mitigating the impact of outliers is to change l from L2 loss described above to a different function such as L1 loss

$$l(x) = |x|, \tag{3.15}$$

Huber loss [27]

$$l_c(x) = \begin{cases} \frac{1}{2}x^2, & |x| \leq c, \\ c(|x| - \frac{1}{2}c), & |x| > c, \end{cases} \tag{3.16}$$



(a) (b)

Figure 3.2: (**a**) Point cloud showing the resulting parameters when the line estimation experiment was repeated 10^5 times, without outliers (blue), with outliers (bronze). The white cross corresponds to the true underlying parameters. (**b**) Plot of different loss functions as a function of residual.

or Tukey's biweight loss [28]

$$l_c(x) = \begin{cases} \frac{c^2}{6}\left(1 - \left(1 - \left(\frac{x}{c}\right)^2\right)^3\right), & |x| \leq c, \\ \frac{c^2}{6}, & |x| > c, \end{cases} \tag{3.17}$$

which is a smooth version of truncated L2 loss. The different loss functions are plotted in figure 3.2b. The choice of loss function highlights a trade-off between convexity and influence of outliers, ranging from L2 loss where outliers have high impact, but the optimization problem has good properties, to Tukey's biweight loss where outliers have almost no impact, but the problem is not convex. To be able to find the optimum on non-convex loss functions, we want to use some other method than local optimization. Also, for many of the problems discussed in the next chapter, the problem of interest is not a linear problem and can often also be non-convex.

3.3.2 Random Sample Consensus

A classical robust method for parameter estimation is random sample consensus (RANSAC), introduced by Fischler and Bolles in 1981 [29]. The idea is to use a subset of the data to estimate the parameters. The estimated model can then be evaluated by how well it fits the

rest of the dataset by computing the loss for some chosen robust loss function. Because of this, RANSAC is referred to as a hypothesize-and-test framework. The most common choice of loss function is counting how many points have a residual smaller than some predefined tolerance.

This process is then repeated, choosing new subsets, and storing the model that best fits the data. This model can optionally be improved by local optimization on a robust loss metric, since the parameters are already close to the optimum, non-convexity is not an issue.

One idea behind selecting subsets is to increase the chance that a subset is outlier-free, in which case a non-robust estimator will work. Consider a dataset of size m with an outlier ratio o_r. If a subset of size n is chosen then the probability of the subset containing no outliers is

$$\frac{\binom{m(1-o_r)}{n}}{\binom{m}{n}} \approx (1-o_r)^n, \quad m \gg n. \tag{3.18}$$

The probability of a given subset being outlier free decreases exponentially with the size of the subset. Hence, the subset is often chosen to be as small as possible in order to reduce the number of iterations needed. Later on we will discuss minimal solvers, which are a way of finding a finite set of parameter candidates, which makes them suitable to use in conjunction with RANSAC. They are minimal because they require the usage of the smallest possible subset. The size of this subset is often of interest, which is described for some problems of interest in Chapter 4. How to construct minimal solvers is the subject of Chapter 5.

We can now apply RANSAC to the linear regression example from earlier. First, an inlier threshold is chosen, typically based on some prior knowledge of the noise distribution on the inlier measurements. Two points are selected randomly, since that is the minimal number of points needed to define a line, which is our model. The model is evaluated by counting the number of points with a residual less than the inlier threshold. After iterating the process some number of times the model with the most inlier points is chosen. The process is shown in Figure 3.3.

3.4 Machine Learning

In recent years, machine learning as a field has attracted considerable attention. This is true both in a research context, where the number of researchers active in the field has increased considerably, but also in public awareness with multiple applications such as AlphaGo [30] and Large Language Models [31–33]. While the field as a whole is too broad a topic to discuss here, since machine learning techniques are used in the papers, a brief introduction and discussions of some points of interest is warranted. The main focus of this section will

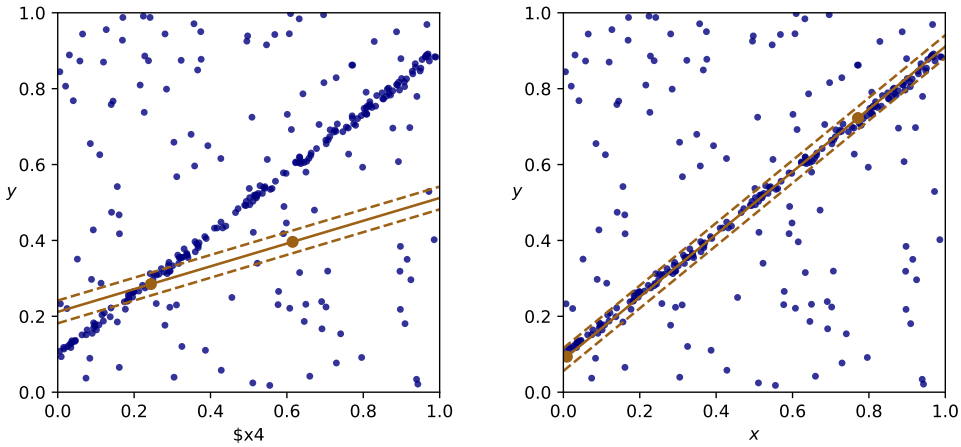


Figure 3.3: Example of using the RANSAC method to estimate a line robustly. **(a)** Two points have been randomly selected defining a model (solid bronze line). After repeating this process 100 times, keeping the line with the most points between the dashed lines, the result is the model shown in **(b)**.

be to introduce energy-based models since that is a topic less commonly covered in other sources. For both a deeper and broader introduction of the field at large, see [34]. One thing to note is that the field has up until this point largely been empirically driven, that is to say, theoretical explanations of why current methods work are generally lacking.

3.4.1 Supervised Learning

One of the most common ways of performing machine learning is supervised learning. Using the same framework as previously in this chapter, the main difference with supervised learning is that we split the data sampled from $\mathcal{D}$ into two pieces (x, y). The estimation problem of interest is to find parameters θ of model f_θ that approximates some aspect of $p(y|x)$ in particular. This is typically done by performing local optimization [35–37] on the function

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[L(f_\theta(x), y)], \tag{3.19}$$

where L is some function.

One of the main ways in which machine learning is different from classical statistics is the number of parameters used in the estimation. This difference stems from the interpretation of the bias-variance trade-off [38], which roughly says that as the complexity of the model

class increases it will be able to fit the training data better and better while the generalization will start to degrade after a certain point. Traditionally, the complexity of the model class has been thought of as correlated to the number of parameters, and thus the reasonable course of action was to limit the number of parameters used. More recent results challenges this connection [39], and results regarding double descent [40], suggest that using more parameters than are needed to exactly fit the data actually results in a model that generalizes even better. This is referred to as the model being in the over-parametrized regime. Since many parameter values fit the training data perfectly this suggests there is some bias in which parameters are selected. It is unknown from where this bias stems from, but one hypothesis is that solutions with better generalization properties take up an increasingly larger part of the space of optimal parameters [41]. They hypothesize that this is a form of simplicity bias, which is why we should expect the model to generalize well, at least according to Occam's razor [42].

Note in particular that we care about how well f_θ performs on the underlying distribution $\mathcal{D}$, not on the specific dataset D we have. Since we have samples from $\mathcal{D}$ we can compute an empirical estimate of the true objective function. However, we should be careful using the same data for choosing θ and estimating the objective, since if we change θ based on the data the empirical estimator is no longer valid.

In practice, we usually split the dataset into three parts: a training set D_{train}, a validation set D_{valid} and a test set D_{test}. The training set is used for directly choosing θ and the validation set is used for selecting hyperparameters, such as learning rate, which indirectly affects which θ is chosen. The test set should then ideally only be used one time, when a final choice of θ is made, to get an unbiased estimation of how well f_θ will work on new samples from $\mathcal{D}$.

3.4.2 Classification and Regression

Problems are often categorized based on which space y is an element in. If $y \in \mathbb{Z}_n$ it is called classification, and regression if y is from a continuous space, usually $\mathbb{R}^n$. In classification problems, the model usually returns a probability distribution over the output space $f_\theta(x) \in \mathcal{P}(\mathbb{Z}_n)$ together with Cross Entropy loss chosen as the loss function

$$L(p, y) = -\ln p(y), \quad p \in \mathcal{P}(\mathbb{Z}_n) \quad y \in \mathbb{Z}_n. \tag{3.20}$$

Minimizing Cross Entropy loss corresponds to MLE. The typical example of a classification problem is classifying images of cats and dogs.

For regression problem, the output of the model is usually a point $f_\theta(x) \in \mathbb{R}^n$ and a common choice of loss function is Mean Squared Error loss

$$L(\hat{y}, y) = ||\hat{y} - y||^2, \quad \hat{y}, y \in \mathbb{R}^n. \tag{3.21}$$

The prototypical example of a regression problem is linear regression as presented earlier. Examples of both problem types are shown in Figure 3.4.

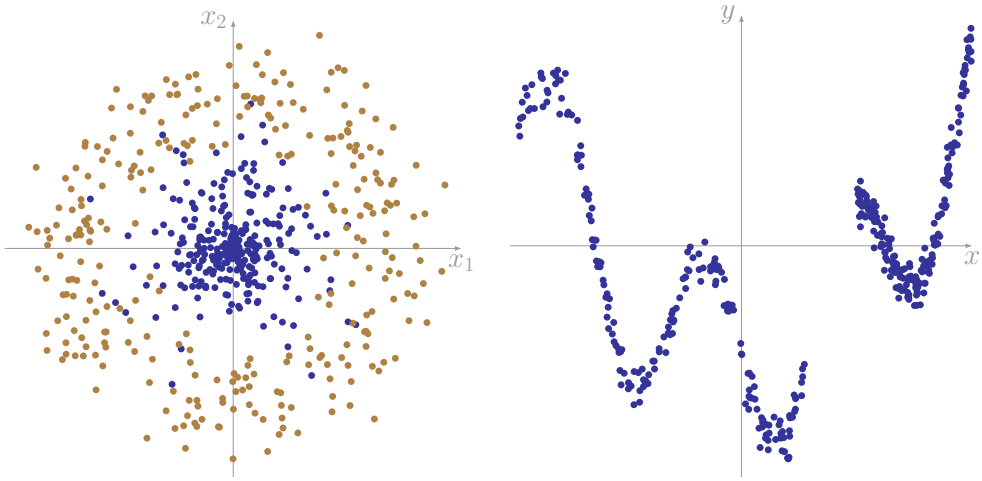


Figure 3.4: **(a)** Example of a classification problem. The aim is to predict the class (shown as color) of a point based on its coordinates (x_1, x_2). **(b)** Example of a regression problem. The aim is to predict y of a point given x. In this case, there is a lack of data in part of the input domain and thus interpolation based on the rest of the data is required.

3.4.3 Energy-Based Models

Consider the regression problem shown in Figure 3.5. For a given input, the underlying data distribution is not necessarily a single value or a unimodal distribution, but instead follows a multimodal distribution. If we, as described above, use a model which only outputs a single value it will not be able to describe the entire distribution.

If we naively apply the regression framework anyway, the model which minimizes regression loss will output the mean of the underlying distribution, which in the multimodal case is unsatisfactory. Because of this, it is desirable if the model can output a probability distribution over continuous spaces. This turns out to be an interesting problem, since it is unclear what the best way is of mapping the model output, that is in $\mathbb{R}^m$, to a probability distribution.

For a start, if we change the output of the network to a probability distribution instead of a point, we have to change the loss function. Similar to the discrete case, we want to

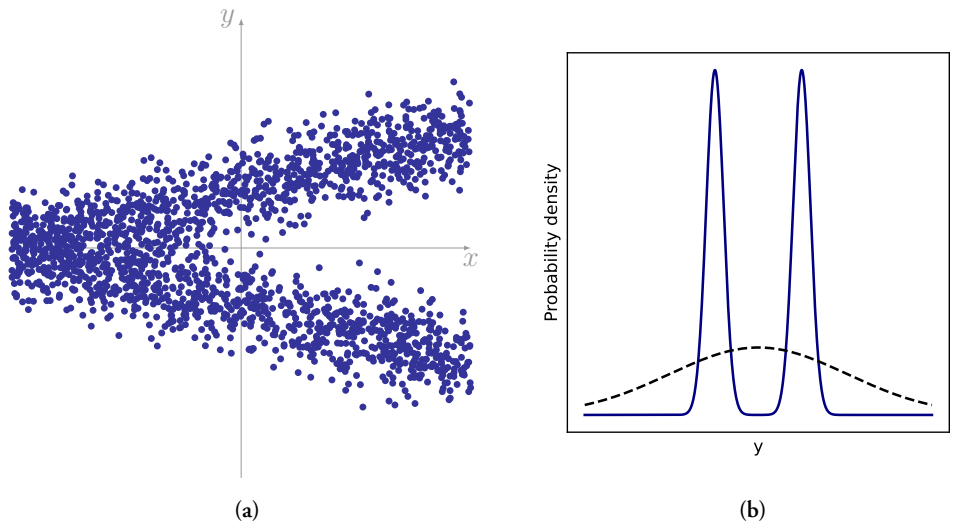


Figure 3.5: **(a)** Plot shows an example of dataset that goes from being unimodal (and Gaussian) on the left to multimodal on the right. If the aim is to minimize the mean squared error the best prediction is $y = 0$, regardless of the input x. However, in the multimodal case we see that this is unsatisfactory, since no probability mass is close to the prediction. **(b)** This shows the underlying probability distribution points (blue) on the right edge of the other plot. The dashed black line shows the Gaussian distribution which maximizes the likelihood, if both mean and standard deviation are parameters. If the goal is to predict a distribution as similar as possible to the underlying distribution, there is clearly room for improvement.

maximize the likelihood of the observed data

$$L(p, y) = -\ln p(y). \tag{3.22}$$

One of the simplest ways of getting a probability distribution as output is to map $\hat{y}$ to $\mathcal{N}(\hat{y}, 1)$. As it turns out, this is equivalent to performing standard regression, because of the correspondence between least squares estimation and MLE with a normal distribution mentioned earlier in the chapter. Naturally, this mapping does not allow us to represent all probability distributions, and it in particular does not solve the problem of representing multimodal distributions.

3.4.4 Regression-via-Classification

Another approach is dividing up the output space into a finite number of bins, allowing the problem to be recast as a classification problem where the output is a distribution over

the bins. This approach is usually referred to as regression-via-classification (RvC). The downside of this approach is that the size of the bins limits the resolution of the probability distributions which is possible to represent. Furthermore, if the dimension of the output is large, this method is unfeasible to use because the number of bins grows exponentially in n, which is known as the curse of dimensionality [43]. However, if n is small, this solution is attractive due to its simplicity.

3.4.5 Energy-Based Models

Since we want to represent a distribution and our model is a function approximator anyway, one idea is to have y in the input and the network outputting the probability density $p(y)$ which is a single number. This is convenient, since during training, the loss is computed by finding the probability density at a single point, which is the answer y. The problem with this approach becomes readily apparent if we consider the constraints p must satisfy to be a probability distribution

$$\begin{cases} p(y) \geq 0, \quad \forall y \in \mathbb{R}^n, \\ \int_{\mathbb{R}^n} p(y) dy = 1. \end{cases} \tag{3.23}$$

That is to say, there is nothing constraining the model to represent a probability distribution. This can however be solved. First, to ensure the output is positive it can be mapped through a function that has a range $x \geq 0$. The exponential function is a common choice. Enforcing the second constraint then amounts to multiplying with the right constant. Together, this means that we go from model output f_θ to probability density p through

$$\begin{cases} p(\hat{y}) = \dfrac{e^{f_\theta(\hat{y})}}{Z}, \\ Z = \int_{\mathbb{R}^n} e^{f_\theta(\hat{y})} d\hat{y}. \end{cases} \tag{3.24}$$

When using this approach, the model is typically referred to as an Energy-based model [44]. The name comes from the similarity to Gibbs distributions from statistical physics [45].

The main problem with using energy-based models is computing the constant Z, since it requires integrating over $\mathbb{R}^n$. Even if computing Z is intractable there are still ways of optimizing the parameters of the model [46]. It is also worth noting that how classification is usually done is an energy-based model on a discrete space. The difference is that computing Z is easy on a discrete space.

3.4.6 Diffusion

An interesting solution to the problem of computing Z is to, instead of modeling the probability density $p(y)$, model $\nabla_y \log p(y)$, which is called the score function. This approach is referred to as score-matching or diffusion modeling [47, 48], and has gained a lot of interest in recent years for generative modeling [49, 50]. These models work using stochastic differential equation theory, which allows sampling to be made from p using the approximated score function, see [51].

The advantage of modeling $\nabla_y \log p(y)$ is that it is independent of Z

$$\nabla_y \log p(y) = \nabla_y \log \frac{e^{g(y)}}{Z} = \nabla_y (g(y) - \log(Z)) = \nabla_y g(y). \tag{3.25}$$

In other words, if a function is rescaled, its corresponding score function does not change. What this means is that if we want to update our belief by adding probability mass to a certain region, the score function outside that region does not change. Therefore, approximating the score function is desirable.

The astute reader might have noticed that we started the discussion about modeling probability distributions with the need to enforce that probability mass is zero and in the end we have a new constraint, the vector field being conservative, which is not enforced in practice today, see [52] for more details about this topic.

Chapter 4

Localization

In this chapter, we first discuss what types of measurements are typically used for acoustic localization and how they are typically found. We then move on to the different types of geometric problems that arise, depending on what prior knowledge is available, and how these problems are solved. For other sources describing this material generally, see [53, 54].

4.1 Geometric Measurements from Sound

4.1.1 Time-of-Arrival

The simplest environment we could consider consists of one speaker, one microphone, and no other walls or objects for the sound field to interact with. If we can measure the time between a sound being emitted by the speaker and the time of its arrival at the microphone, we have a measurement of the distance between the speaker and microphone, given that the speed of sound is known. This is known as a Time-of-Arrival (TOA) measurement. Note that for these types of measurements, TOA can refer to either the measured time or the corresponding distance depending on the context.

An example of this is measuring the distance to a lightning strike. When lightning strikes, the light reaches us approximately instantly. There is then a delay until the thunder arrives. Using the fact that the speed of sound is approximately 340 m/s we know that the lightning struck one kilometer further away for every three seconds of delay, see Figure 4.1.

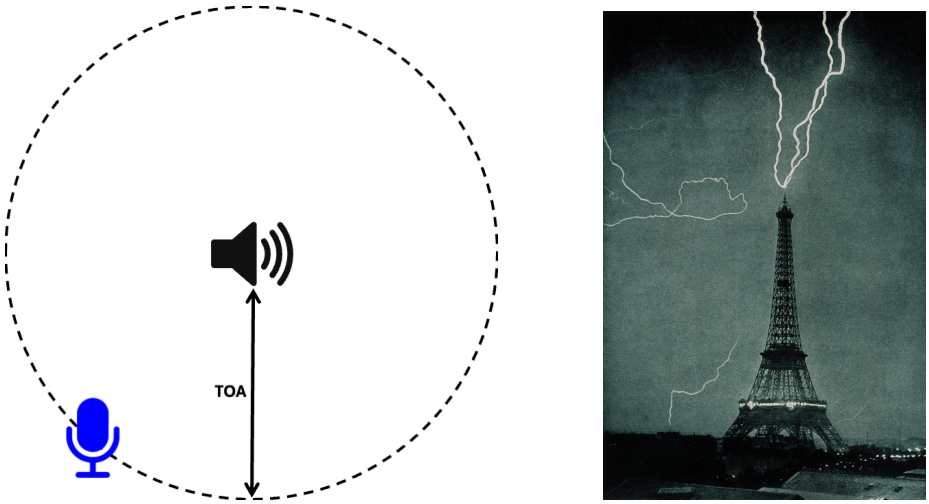


Figure 4.1: **(a)** Example of a speaker and a microphone with the distance measured by the TOA marked. **(b)** By using the time between when this image was taken and the time the thunder arrived, it is possible to measure the distance to the Eiffel Tower.

4.1.2 Time-difference-of-Arrival

One problem with TOA measurements is that they require knowledge of when a sound was emitted, which is not always available. In this case, instead of using only one microphone, we now use two microphones. A sound is emitted at some unknown time, and if we can detect when the sound arrives at each of the microphones, we can compute how much closer the speaker is to one microphone than to the other, see Figure 4.2a. Similar to before, we are computing a time delay, however, here it is the difference between two arrival times and the measurement is therefore called a Time-Difference-of-Arrival (TDOA) measurement.

As can be seen in Figure 4.2b, if the speaker is far away relative to the distance between the microphones, then TDOA corresponds to measuring the direction in which the sound arrives. This measurement is sometimes referred to as a Direction-of-Arrival (DOA) measurement instead.

4.1.3 Other Sources of Information

There are various other ways of extracting information from sound measurements. For instance, the Doppler shift gives information about velocity, or we could use relative amplitude as a way of measuring distance [55]. There is also semantic information available,

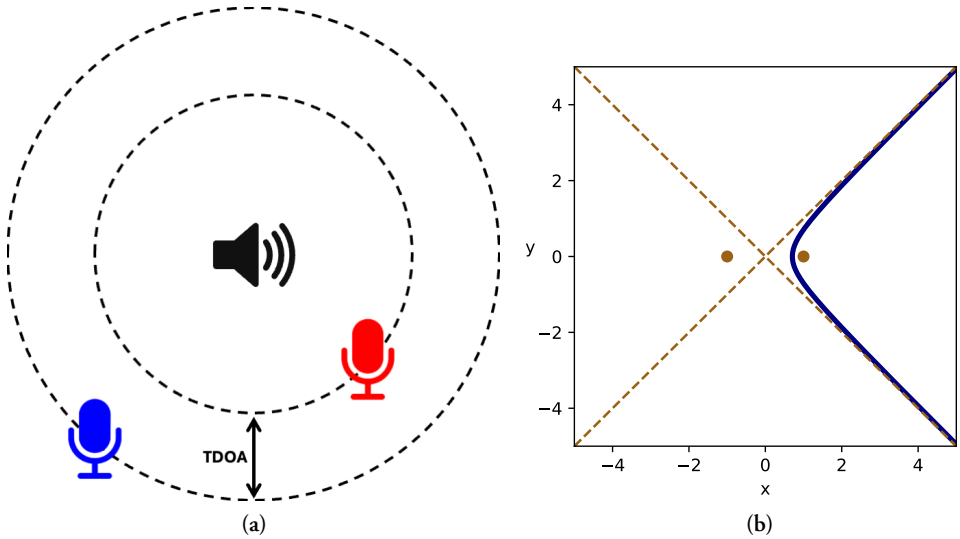


Figure 4.2: **(a)** Two microphones and a speaker shown together with the distance measured by the TDOA. **(b)** The blue curve consists of points which are $\sqrt{2}$ closer to (1,0) than to (-1,0). The dashed lines show that as the speaker moves far away relative to the distance between the microphones, the TDOA corresponds to which direction the sound arrives from.

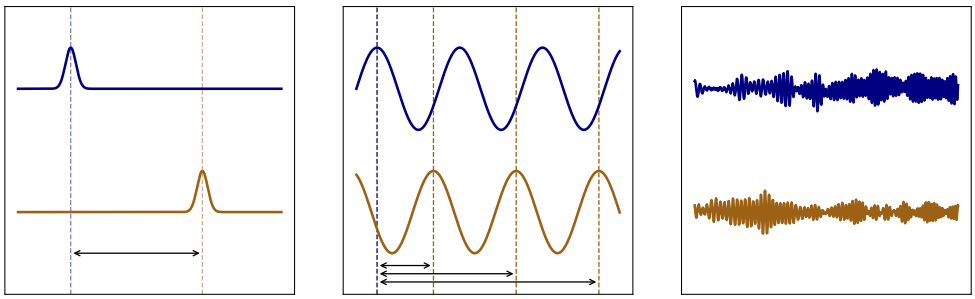
such as what type of animal is emitting the sound or emotion behind a tone of voice. While these sources of information are interesting, they are outside the scope of this thesis.

4.1.4 Time Delay Estimation

In the TDOA setup, we have two microphones and thus two sound recordings. We want to find the optimal delay between them. What we mean by this qualitatively is illustrated in Figure 4.3. To do this, we must first define what being optimal means. Let us say the speaker emits a sound $\tilde{y}(t) : \mathbb{R} \to \mathbb{R}$, then the idealized sound field at the i-th microphone position is $\tilde{x}_i(t) = \tilde{y}(t - cd_i)$, where d_i is the distance between the speaker and microphone i, and c is the speed of sound. For this idealized case, what we mean by optimal delay is the delay $\Delta^* \in \mathbb{R}$ that minimizes

$$\Delta^* = \underset{\Delta}{\operatorname{argmin}} \, ||x_1(t) - x_2(t - \Delta)||_2, \tag{4.1}$$

which is the estimate that maximizes the likelihood of the data assuming white noise, since both signals are shifted versions of each other. This also corresponds to maximizing the correlation between the signals.



(a) (b) (c)

Figure 4.3: Qualitative example of sound signals and delays are shown in the plots. (**a**) Simulated sound pulse, the time delay is easy to estimate by eye. (**b**) Simulated sine wave, multiple time delays fit the data well, and thus the TDOA can not be determined. (**c**) Two recordings of a chirp sound. The delay is not easily estimated by eye, however, it is still possible to estimate using a method such as GCC-PHAT.

When measuring the real sound field, we sample the sound field at some sampling rate f_s. That means the recording is a vector $x_i \in \mathbb{R}^n$, not a function, such that

$$x_i^{(k)} = \tilde{x}_i\left(\frac{k}{f_s}\right) + \epsilon, \quad \epsilon \in \mathcal{N}(0, \sigma^2). \tag{4.2}$$

The microphone also is not a point but has some size, which means that we are measuring some average of the pressure field.

In practice, we use a method known as Generalized Cross-Correlation with Phase Transform (GCC-PHAT), introduced by Knapp and Carter [56], which estimates Δ at the sample level. The method is summarized in Algorithm 1. GCC-PHAT performs a similar task to finding the best Δ but with some adjustments.

An example of a difference is that instead of shifting the signal, the signal is circularly shifted meaning the end of the signal is looped back to the start. This is a consequence of using the Fourier transform. Since the true signal in general is not periodic this is a downside. However, performing the computations in the Fourier domain allows us to speed up the algorithm from a time complexity of $O(n^2)$ to $O(n\log(n))$, which is the time complexity of computing the Fast Fourier Transform [57]. Since the shift is usually a small fraction of the length of the signal, the circular shift has a relatively small impact on the result.

In the case where the sound emitted by the speaker is available, GCC-PHAT can also be used to acquire TOA measurements by using the sound of the speaker along with the recording from one microphone as input.

Algorithm 1: GCC-PHAT

Input: $x_1, x_2 \in \mathbb{R}^n$
$X_1 \leftarrow \mathcal{F}(x_1)$, where $\mathcal{F}$ denotes the discreate Fourier transform.
$X_2 \leftarrow \mathcal{F}(x_2)$
$c \leftarrow \frac{X_1 \odot \bar{X}_2}{|X_1||X_2|}$, where $|\cdot|$ is the element wise norm.
$s \leftarrow \mathcal{F}^{-1}(c)$
Output: s, use argmax s as circular shift between x_1 and x_2.

4.2 Geometric Problem Formulations

Here follows a description of common geometric estimation problems. Since TOA and TDOA measurements can also be measured by other sensor types, e.g. WiFi signal strength [58] or Radio [59], the problems described in this section are applicable to more than sound. Because of this, I use the words sender and receiver instead of sound source and microphone in this section.

Except for the measurement type, there are also variants of all of these problems. This depends on factors such as the dimension of the problem (usually 2D or 3D), or if something is far enough away that a far-field assumption can be made. Unless otherwise stated, assume that the problem is in 3D and that all sound sources and microphones are at comparable distances from each other. However, many of the other variants share similarities with these problems. I have chosen to write $\hat{d}$ to denote the measurement of d. This is relevant when reformulating equations, since constraints involving d hold exactly true, which is not true for $\hat{d}$.

4.2.1 Trilateration

When performing trilateration, the goal is to find the position s of a single sender using TOA measurements $\hat{d}$ to a set of receivers with known positions r. We can formulate the optimization problem for this as

$$\underset{s}{\operatorname{argmin}} \sum_i l(||s - r_i|| - \hat{d}_i), \tag{4.3}$$

where l is some robust loss function. To figure out the minimal number of receivers needed to define a finite number of possible candidates for s, we can study the number of degrees of freedom of the problem. In this case, there are three unknown coordinates in s. Because each receiver gives one constraint, we need at least three receivers to have more constraints than degrees of freedom. Finding s from the measurements is often visualized as finding the

intersection of three spheres. How to solve this algebraic for the 2D case is also provided as an example in the next chapter. For the non-minimal case, the problem can be reformulated into an eigenvalue problem [60].

4.2.2 Multilateration

Multilateration is similar to trilateration except that, instead of having TOA measurements to the set of receivers, we have TDOA measurements $\hat{z}$ between pairs of receivers. The optimization problem is formulated as

$$\underset{s}{\operatorname{argmin}} \sum_{i,j} l(||s - r_i|| - ||s - r_j|| - \hat{z}_{ij}). \tag{4.4}$$

One important aspect to consider is that not all measurements are independent in this case, since for three receivers r_i, r_j, r_k we have the constraint

$$z_{ij} + z_{jk} + z_{ki} = 0. \tag{4.5}$$

This means that each additional receiver beyond the first adds only one new constraint, and because of this the minimal case for multilateration is four receivers and using a set of three measurements such that all receivers are represented, e.g. $\{\hat{z}_{12}, \hat{z}_{13}, \hat{z}_{14}\}$.

Instead of solving multilateration from this set of equations, multilateration is often solved in two stages. First, note that the vector d, with $d_i = ||s - r_i||$, fulfills

$$d_i - d_j = z_{ij}, \quad \forall i, j. \tag{4.6}$$

However, d is not uniquely defined from z, since any vector q with

$$q_i = d_i + o, \quad o \in \mathbb{R}, \tag{4.7}$$

also satisfies

$$q_i - q_j = z_{ij}, \quad \forall i, j. \tag{4.8}$$

As for terminology, we refer to z as the TDOA-matrix and q as the TDOA-vector. We know q fulfills the constraints

$$||s - r_i|| + o = q_i, \quad \forall i, \tag{4.9}$$

which we refer to as the offset formulation, since it looks like the trilateration problem except for the added unknown o. One way of performing multilateration is therefore solving for q from z, and as a second step solve for s and o from q. Both steps typically need to be solved using robust estimation.

It should be noted that breaking down a robust estimation problem into multiple steps is in many cases a significant advantage, referred to as stratification in Paper II. This is because solving subproblems is typically faster and more numerically stable, but also because after solving one subproblem, it provides outlier filtering so that the input to the next subproblem contains fewer outliers. Using multilateration as an example, if z contains a certain ratio of outliers, then the estimated q contains a lower ratio of outliers.

When solving for q from z, any translation of q creates an equally valid solution to the problem. More generally expressed, there exists a family of transformations which transform solutions into new equally valid solutions. This is referred to as gauge freedom. It is important to be aware that there are multiple correct solutions, one example of why is that it affects how we compare the similarity of two solutions. Additional constraints are often added to remove the gauge freedom.

4.2.3 Self-Calibration

Self-calibration, sometimes referred to as sensor network self-calibration or structure from sound, refers to the process of finding both the unknown sender positions s and receiver positions r at the same time. In the case of using TOA measurements we naturally call this TOA self-calibration and the problem is formulated as

$$\underset{s,r}{\operatorname{argmin}} \sum_{i,j} l(||s_i - r_j|| - \hat{d}_{ij}), \tag{4.10}$$

whereas if we have TDOA measurements the problem is called TDOA self-calibration and is formulated as

$$\underset{s,r}{\operatorname{argmin}} \sum_{i,j,k} l(||s_i - r_j|| - ||s_i - r_k|| - \hat{z}_{ijk}). \tag{4.11}$$

Since the problem only depends on the distances between our unknowns $||s_i - r_j||$, the gauge freedom is any transformation which preserves relative distances. This is the set of Euclidean transformations, i.e. translations, rotations, and reflections

$$\tilde{s} = R(s^* + t), \quad \tilde{r} = R(r^* + t), \quad t \in \mathbb{R}^3, \quad R \in O(3), \tag{4.12}$$

since

$$||\tilde{s}_i - \tilde{r}_j|| = ||R(s_i^* + t) - R(r_j^* + t)|| = ||s_i^* - r_j^*||. \tag{4.13}$$

The fact that solutions can differ by a Euclidean transformation is also common when performing structure from motion in computer vision and is often described by saying that the choice of coordinate system is arbitrary, see Figure 4.4a. Finding this Euclidean

transformation is often of interest when performing map merging, i.e. combining solutions from different self-calibration or structure from motion problems, such as in Paper V.

Hereafter, the main focus will be TDOA self-calibration, which is what we refer to when writing only self-calibration. Similar to the multilateration case, the offset reformulation is often used. Using the same logic, we obtain one unknown offset o_i for each sender s_i. The resulting equations used to solve for s, r and o are

$$||s_i - r_j|| + o_i = q_{ij}, \quad \forall i, j. \tag{4.14}$$

Figuring out what the minimal case is for self-calibration is more complicated than the previously discussed cases and was described in [61]. Each sender introduces four unknown variables, three in s_i and one in o_i, while each receiver adds three unknowns. Each pair of sender and receiver adds one constraint. The gauge freedom of the problem also has to be factored in, which means that there effectively are additional constraints equal to the number of degrees of freedom in a Euclidean transformation, which is six. Thus, for the problem to be solvable there needs to be more constraints than unknowns, which means that if we have n senders and m receivers

$$nm + 6 \geq 4n - 3m. \tag{4.15}$$

It is easy to check that $n > 3$ and $m > 4$. After this we can enumerate the minimal cases to be

$$(m, n) = \{(5, 9), (6, 6), (7, 5), (10, 4)\}, \tag{4.16}$$

which is also visualized in Figure 4.4b.

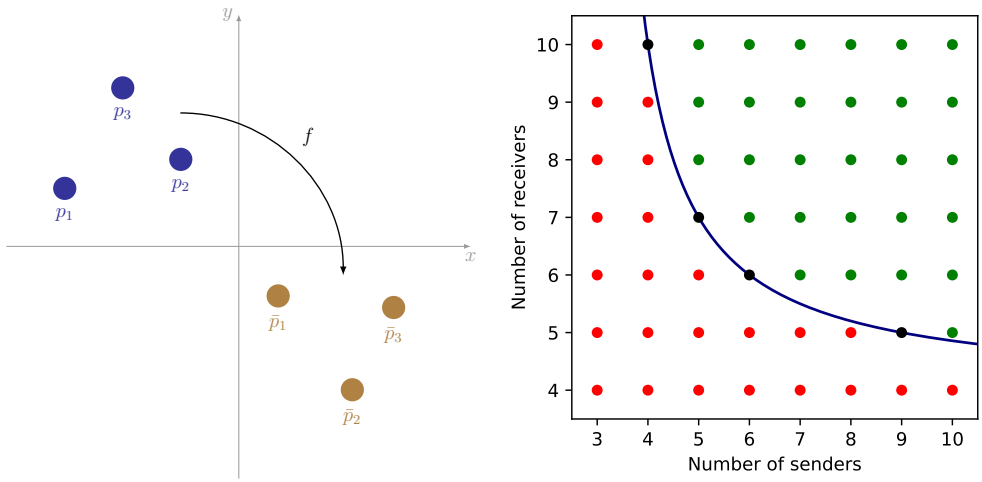
While there is enough information that the problems are solvable, current methods are unable to solve these minimal cases. However, it is possible to solve for the offsets first, as presented in [61]. Using their method, it is possible to solve the cases

$$(m, n) = \{(6, 8), (7, 6), (9, 5)\}. \tag{4.17}$$

The first two cases are solved using tools from algebraic geometry, which were improved upon by [62], while the $(m, n) = (9, 5)$ case can be solved using linear methods. How to solve this is shown in the next section for the interested reader. After solving for the first set of offsets, it is possible to use that solution to expand to more senders, receivers, and offsets, as shown in [63]. Expanding the solution is also what allows us to evaluate the solution in order to make robust estimation possible. What the best way of expanding versus evaluating the solution is currently relatively open problems.

4.2.4 Offset Solver

Included below is how to solve for offsets in the $(m, n) = (9, 5)$, as described in [61]. I have included this to give an understanding of what these types of handcrafted solvers look



(a) (b)

Figure 4.4: **(a)** The plot shows two sets of nodes such that $f(p_i) = \bar{p}_i$, where f is a Euclidean transformation. If all constraints of a problem depend on distances between nodes, then both sets of nodes are equally valid solutions. This is an example of a gauge freedom. **(b)** Visualization showing how many senders and receivers are needed to make self-calibration possible. The plotted curve corresponds to when there are equally many constraints and unknowns. The problem is therefore solvable on or above the curve. The black points mark the minimal cases.

like. To solve for offsets using linear methods, we start with our original equations squared

$$||r_i - s_j||^2 = (q_{ij} - o_j)^2, \quad r_i, s_j \in \mathbb{R}^3, \quad i \in 1 \dots m, \quad j \in 1 \dots n. \tag{4.18}$$

Pick one value for j (here we choose $j = 1$) and form

$$||r_i - s_j||^2 - ||r_i - s_1||^2 = (q_{ij} - o_j)^2 - (q_{i1} - o_1)^2, \quad i \in 1 \dots m, \quad j \in 2 \dots n. \tag{4.19}$$

Expanding both sides gives

$$s_j^T s_j - s_1^T s_1 - 2r_i^T(s_j - s_1) = q_{ij}^2 - q_{i1}^2 - 2q_{ij}o_j + 2q_{i1}o_1 + o_j^2 - o_1^2, \tag{4.20}$$

and rearranging the terms gives

$$s_j^T s_j - s_1^T s_1 - 2r_i^T(s_j - s_1) - o_j^2 + o_1^2 = q_{ij}^2 - q_{i1}^2 - 2q_{ij}o_j + 2q_{i1}o_1. \tag{4.21}$$

The equations can be gathered into a matrix equation where the element on row a and column b correspond to the equation with $i = a$ and $j = b + 1$ above. The left-hand side

can then be factored into AB with

$$A = \begin{bmatrix} 1 & -2r_1^T \\ \vdots & \vdots \\ 1 & -2r_m^T \end{bmatrix}, \tag{4.22}$$

and

$$B = \begin{bmatrix} s_2^T s_2 - o_2^2 - s_1^T s_1 + o_1^2 & \dots & s_n^T s_n - o_n^2 - s_1^T s_1 + o_1^2 \\ s_2 - s_1 & \dots & s_n - s_1 \end{bmatrix}. \tag{4.23}$$

The rank of AB is at most 4 since A has size $m \times 4$ and B has size $4 \times (n-1)$. Assuming B has full rank, for any vector w_1 in the range of A, there exists a vector $w_2 \in \mathbb{R}^{n-1}$ such that

$$ABw_2 = w_1. \tag{4.24}$$

In particular, there exists a $w \in \mathbb{R}^{n-1}$ such that

$$ABw = \mathbf{1}_m, \tag{4.25}$$

where $\mathbf{1}_m$ is a column matrix of length m with every element equal to one.

Coming back to the right-hand side of the matrix equation, it can also be factorized into a product of two matrices CD with

$$C = \begin{bmatrix} Q^{\odot_2} - Q_1^{\odot_2} \mathbf{1}_{n-1}^T & -2Q & 2Q_1 \end{bmatrix}, \tag{4.26}$$

where

$$Q = \begin{bmatrix} q_{12} & q_{13} & \cdots & q_{1n} \\ q_{22} & q_{23} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m2} & q_{m3} & \cdots & q_{mn} \end{bmatrix}, \quad Q_1 = \begin{bmatrix} q_{11} \\ q_{21} \\ \vdots \\ q_{m1} \end{bmatrix}, \tag{4.27}$$

and $[\,\cdot\,]^{\odot_2}$ denotes squaring element-wise, and

$$D = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ o_2 & 0 & \dots & 0 \\ 0 & o_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & o_n \\ o_1 & o_1 & \dots & o_1 \end{bmatrix}. \tag{4.28}$$

We now have C, which contains no unknowns, and we know

$$\mathbf{1}_m = ABw = CDw. \tag{4.29}$$

By introducing the variable

$$u = Dw, \tag{4.30}$$

we can solve for u in $\mathbf{1}_m = Cu$, since the size of C is $m \times (2n-1)$, which in the $(9,5)$ case is a 9×9 matrix. The offsets can then be extracted from u as

$$o_1 = \frac{u_9}{\sum_{i=1}^{4} u_i}, \tag{4.31}$$

and

$$o_j = \frac{u_{j+3}}{u_{j-1}}, \tag{4.32}$$

using $u = Dw$.

Chapter 5

Minimal Solvers

Consider a simple problem of finding x satisfying

$$x^2 + ax + b = 0. \tag{5.1}$$

This can be solved by rearranging the terms and arriving at the following closed-form expression

$$x = -\frac{a}{2} \pm \sqrt{\frac{a^2}{4} - b}. \tag{5.2}$$

Now, regardless of the values of the coefficients a and b, we can quickly find the roots using the formula we have precomputed. As it turns out, something similar is also possible for systems of polynomial equations. This precomputed template for solving a system of polynomial equations is referred to as a minimal solver. The strength of these methods is that they give us a fast way to compute solutions to non-convex problems, without having to rely on a good initial guess. By combining minimal solvers and RANSAC, we can create a method which is very robust to outliers.

5.0.1 Companion Matrix

At first glance, one should be skeptical of the existence of minimal solvers. This is because of Abel's impossibility theorem which states that no closed-form formula can exist for finding the roots of a general quintic polynomial [64]. Instead, we need to look for something other than a closed-form solution. To start with, consider finding the roots of a univariate polynomial

$$p(x) = x^{n+1} + a_n x^n + \cdots + a_1 x + a_0. \tag{5.3}$$

We form the equation system

$$x \begin{bmatrix} x^n \\ x^{n-1} \\ x^{n-2} \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} -a_n & -a_{n-1} & \dots & -a_1 & -a_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} x^n \\ x^{n-1} \\ x^{n-2} \\ \vdots \\ 1 \end{bmatrix}, \tag{5.4}$$

where the equality of the first row comes from p, and the rest are trivial equalities. The matrix

$$C_p = \begin{bmatrix} -a_n & -a_{n-1} & \dots & -a_1 & -a_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \tag{5.5}$$

is called the companion matrix (depending on the source, the companion matrix might be defined with possible permutations of rows and columns and also possibly transposed) [65]. The characteristic polynomial of C_p is p which means that the eigenvalues of C_p coincide with the roots of p.

Good numerical methods exist for computing eigenvalues of a matrix, e.g. the QR algorithm which was introduced in [66, 67] and its more recent variants that incorporates shifts [68]. This gives us a good way of computing roots of a univariate polynomial of degree five or higher. Notice that this does not contradict Abel's impossibility theorem, since the QR algorithm is an iterative method for approximating eigenvalues, not a closed form solution. In fact, the correspondence between eigenvalues and roots means no closed form solution can exist for finding eigenvalues either.

Constructing a matrix such that its eigenvalues contain the solution of interest turns out to generalize to the multivariate case. The reason we could construct the companion matrix was because x^{n+1} could be written as a combination of lower-degree terms using the polynomial p. If a system of polynomial equations can be encoded into a matrix equation of the form

$$\alpha X_b = A X_b, \tag{5.6}$$

where α is a polynomial, A is a matrix of coefficients, and X_b is a monomial vector (defined below), then we could again find the possible values of α by finding the eigenvalues of A. By choosing α to be one of our variables, this method would give us the solutions for that variable. This method is called **the action matrix method**. To find α and A from a system of polynomial equations, we will need some theory from algebraic geometry.

5.1 Algebraic Geometry Terminology

In this section we will briefly introduce some concepts from algebraic geometry based on Cox et al. [69]. This material specifically is also presented in the following sources [53, 70].

Given a finite set of variables $X = \{x_1, x_2, \ldots, x_n\}$, we can define the following terms.

Definition 5.1. A **monomial** w is a finite product of variables

$$w = \prod_i x_i^{d_i}, \tag{5.7}$$

where $d_i \in \mathbb{N}$ denotes the degree of x_i.

Definition 5.2. A **polynomial** p is a linear combination of monomials over a field $\mathbb{K}$.

$$p = \sum_i c_i w_i, \tag{5.8}$$

where w_i is a monomial and $c_i \in \mathbb{K}$ is called a coefficient.

We write $\mathbb{K}[X]$ to denote the ring of all polynomials with coefficients in $\mathbb{K}$ and variables in X. The field $\mathbb{K}$ is typically chosen to either be $\mathbb{C}$ or $\mathbb{Z}_p$, where p is prime.

For a set of polynomials $f = \{f_1, f_2, ...f_m\}$ we make the following definitions:

Definition 5.3. The Ideal I generated by f, denoted $\langle f \rangle$, is

$$I = \langle f \rangle = \left\{ \sum_{i=1}^{m} h_i f_i \;\middle|\; h_i \in \mathbb{K}[X] \right\}. \tag{5.9}$$

Definition 5.4. The **(affine) variety** $V(f)$ is

$$V(f) = \left\{ x \in \mathbb{K}^n \;\middle|\; f_i(x) = 0,\ \forall f_i \in f \right\}. \tag{5.10}$$

In other words, the roots of a system of polynomial equations correspond to an affine variety. Examples of varieties are shown in Figure 5.1. In the case when the variety consists of a finite number of points, it is called zero-dimensional.

In the multivariate case, it is not obvious when a monomial is larger than another. We agree that x^2 is larger than x, but is x^2 larger than xy? Because of this, a monomial ordering is introduced to fix this ambiguity.

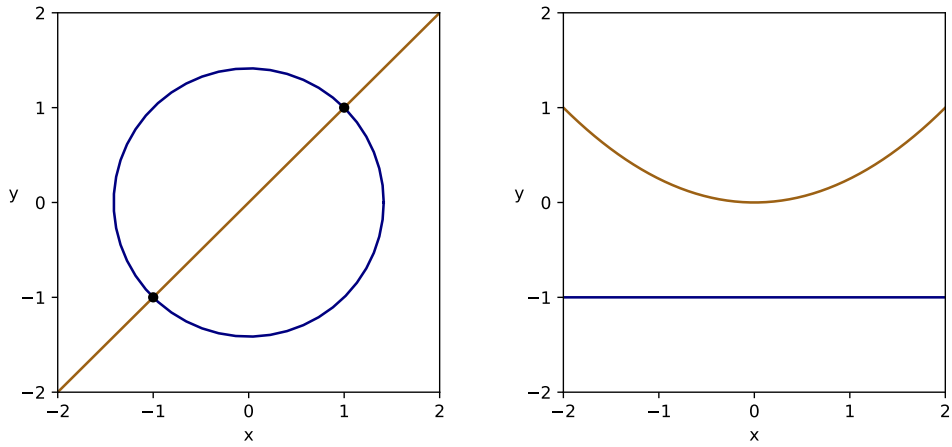


Figure 5.1: **(a)** The plot shows the varieties corresponding to $f_1 = x^2+y^2-\sqrt{2}$ (blue), $f_2 = x-y$ (bronze), and their intersection (black). **(b)** The plot shows the varieties corresponding $f_1 = y+1$ (blue) and $f_2 = x^2 - 4y$ (bronze). While the intersection looks empty from the plot, remember that the field is $\mathbb{C}$, not $\mathbb{R}$ as shown in the plot. This means that the intersection is $\{(x,y) \in \mathbb{C}^2 \,|\, y = -1, x = \pm 2i\}$.

Definition 5.5. A **monomial ordering** $<$ is a binary relation between monomials satisfying

1. $<$ is a total ordering of the monomials.
2. If f, g, h are monomials, then $f < g \Rightarrow fh < gh$.
3. It is a well-ordering of the monomials, i.e. every non-empty set of monomials contains a smallest monomial under $<$.

Multiple monomial orderings are regularly used, two examples are lexicographical ordering and graded lex ordering. In both of these orderings we first fix an ordering among the variables, such as $z < y < x$ or $x_3 < x_2 < x_1$. In lexicographical ordering, the degree of the largest variables is first compared and ties are then broken by comparing smaller and smaller variables. Using graded lex ordering, the total degree (i.e. sum of the degrees of the variables) of the monomial is first compared, lexicographical ordering is then used to break ties. An example of how some monomials are ordered is

$$\text{Lexicographical} \quad y < y^3 < x < xy < x^2, \tag{5.11}$$

$$\text{Graded Lex} \quad y < x < xy < x^2 < y^3. \tag{5.12}$$

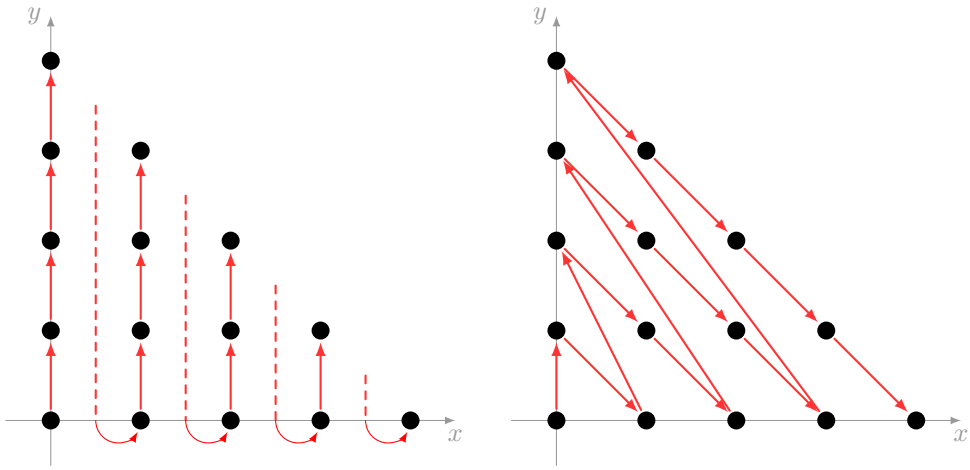


Figure 5.2: The point (a,b) corresponds to the monomial $x^a y^b$. The arrow from monomial m points to the smallest monomial that is larger than m under monomial ordering: **(a)** Lexicographical, **(b)** Graded Lex.

The two orderings are illustrated in Figure 5.2.

Definition 5.6. Given a monomial ordering $<$ and a non-zero polynomial f, define $\mathrm{LT}(f)$ to be the leading term of the polynomial, which is the term in f containing the greatest monomial under $<$. The leading term consists of a coefficient and a monomial which we refer to as the leading coefficient $\mathrm{LC}(f)$ and leading monomial $\mathrm{LM}(f)$.

This means $\mathrm{LT}(f)$ depends on what monomial ordering is used, e.g.

$$\mathrm{LT}(2x + 3y^2 + 4) = 2x, \tag{5.13}$$

using lexicographical ordering, but

$$\mathrm{LT}(2x + 3y^2 + 4) = 3y^2, \tag{5.14}$$

using graded lex ordering. By using a monomial ordering, we also have a partial ordering on $\mathbb{K}[X]$ by defining

$$f < g \iff \mathrm{LM}(f) < \mathrm{LM}(g), \quad f, g \in \mathbb{K}[X]. \tag{5.15}$$

For the rest of this thesis, whenever we talk about reduction, we will assume that we have decided on which monomial ordering to use.

5.2 Building Solvers

5.2.1 Division Algorithm, Gröbner Basis, and Quotient Rings

Given a set of polynomials $f_1, \ldots, f_m \in \mathbb{K}[X]$, we want to be able to rewrite a polynomial $f \in \mathbb{K}[X]$ as a smaller polynomial using the equations $f_i = 0$. It is therefore natural to study the set of polynomials which can be reached by using combinations of our known equations

$$\left\{ f + q_1 f_1 + \cdots + q_m f_m \;\middle|\; q_1, \ldots, q_m \in \mathbb{K}[X] \right\}. \tag{5.16}$$

This set is called an equivalence class and is denoted $[f]$. If we look at the set of all equivalence classes we get a quotient ring. In particular, if $f, g \in [f]$, then

$$f(x) = g(x), \quad \forall x \in V(\langle f_1, \ldots f_m \rangle). \tag{5.17}$$

Definition 5.7. Given a ring $\mathbb{K}[X]$ and an ideal I, the **quotient ring** $\mathbb{K}[X]/I$ is the ring with elements consisting of the equivalence classes induced by

$$a \sim b \iff a - b \in I, \tag{5.18}$$

such that $[\,\cdot\,]$ is a ring homomorphism, i.e. satisfying

$$a \oplus b = c \Rightarrow [a] \oplus [b] = [c], \quad \forall a, b \in \mathbb{K}[X], \tag{5.19}$$
$$a \otimes b = d \Rightarrow [a] \otimes [b] = [d], \quad \forall a, b \in \mathbb{K}[X], \tag{5.20}$$
$$[1] \otimes [a] = [a], \quad \forall a \in \mathbb{K}[X]. \tag{5.21}$$

Example 5.8. Consider the ring $\mathbb{Z}$ and the ideal of integers divisible by 3

$$I_3 = \{3n \mid n \in \mathbb{Z}\}. \tag{5.22}$$

The quotient ring $\mathbb{Z}/I_3$ then consists of three equivalence classes, $\{[0], [1], [2]\}$.

To understand a given quotient ring $\mathbb{K}[X]/I$ we want to answer two questions about it:

1. What equivalence classes are there?
2. How to find the equivalence class of a given polynomial?

To answer these questions we need to talk about what a Gröbner basis is.

Definition 5.9. Given an ideal I, a set $\{g_1, \ldots, g_k\} \neq \{0\}$ is a **Gröbner basis** if and only if for any $f \in I$ there exists an i such that $\mathrm{LT}(f)$ is divisible by $\mathrm{LT}(g_i)$ and $\langle g_1, \ldots, g_k \rangle = I$.

Definition 5.10. Given a Gröbner basis $\{g_1, \ldots, g_k\} \neq \{0\}$, the **standard monomials** are the monomials which are not divisible by any $\mathrm{LT}(g_i)$.

Given an ideal I, we can find a Gröbner basis, e.g. using Buchberger's algorithm [71, 72]. One reason we care about having a Gröbner basis is that we can combine it with the division algorithm, see Algorithm 2.

Proposition 1. Given a Gröbner basis G, any polynomial $f \in \mathbb{K}[X]$ has a unique reduction which is a linear combination of standard monomials. This unique reduction can be found as the remainder by using the division algorithm with G.

Algorithm 2: Division Algorithm

Input: $f, f_1, \ldots f_m \in \mathbb{K}[X]$
Output: $r, q_1, \ldots q_m \in \mathbb{K}[X]$

$r, q_1, \ldots q_m \leftarrow 0$
while $f \neq 0$ **do**
  $i \leftarrow 1$
  progress $\leftarrow$ False
  while $i \leq m$ *AND* progress = False **do**
    if $\mathrm{LT}(f_i)$ divides $\mathrm{LT}(f)$ **then**
      $q_i \leftarrow q_i + \frac{\mathrm{LT}(f)}{\mathrm{LT}(f_i)}$
      $f \leftarrow f - \frac{\mathrm{LT}(f)}{\mathrm{LT}(f_i)}$
      progress $\leftarrow$ True
    else
      $i \leftarrow i + 1$
  if progress = False **then**
    $r \leftarrow r + \mathrm{LT}(f)$
    $f \leftarrow f - \mathrm{LT}(f)$

We can use this to answer both of our questions. First, the equivalence classes of $\mathbb{K}[X]$ form a vector space $\mathbb{K}^d$, where d is the number of standard monomials. The standard monomials form a linear basis of this space. Secondly, we can find the equivalence class of a given polynomial by reducing it using the division algorithm.

We now have all the pieces needed to use the action matrix method, given that there are a finite number of standard monomials. Select X_b to be a vector consisting of the standard monomials, then regardless of the choice of α we can reduce every monomial in αX_b into a linear combination of standard monomials. Hence, we can form the equation

$$\alpha X_b = A X_b, \tag{5.23}$$

and solve for the value of α by computing the eigenvalues of A. The maximum number of solutions is the number of standard monomials. One way of solving for the values of all the variables is to choose α as the monomial consisting of a single variable, then repeating this method for each variable. We can then test all combinations of variable values to find all the solutions. Usually, there are easier ways of finding the rest of the variables such as using back-substitution into an earlier equation or finding the eigenvectors of A.

5.2.2 Trilateration Example

We will now show how to use the action matrix method to solve trilateration in 2D. The position of the sender we want to find is (x, y), and the given information is the position of two receivers located at $(x_1, y_1) = (3, 2)$ and $(x_2, y_2) = (-5, 0)$, along with their respective distances from the sender $d_1 = 5$ and $d_2 = 13$. This gives us the equation system

$$\begin{cases} \sqrt{(x-3)^2 + (y-2)^2} = 5, \\ \sqrt{(x+5)^2 + (y-0)^2} = 13. \end{cases} \tag{5.24}$$

To start with, we have to turn this into a polynomial equation system for our method to work. We solve this by squaring both sides, while noting that this might introduce spurious solutions. We also rearrange the terms such that we have zero as the right-hand side

$$\begin{cases} (x-3)^2 + (y-2)^2 - 5^2 = 0, \\ (x+5)^2 + y^2 - 13^2 = 0. \end{cases} \tag{5.25}$$

Using Buchberger's algorithm with graded lex ordering, we find a Gröbner basis for the ideal generated by the polynomials

$$G = \{4x + y - 33, 17y^2 - 106y + 105\}. \tag{5.26}$$

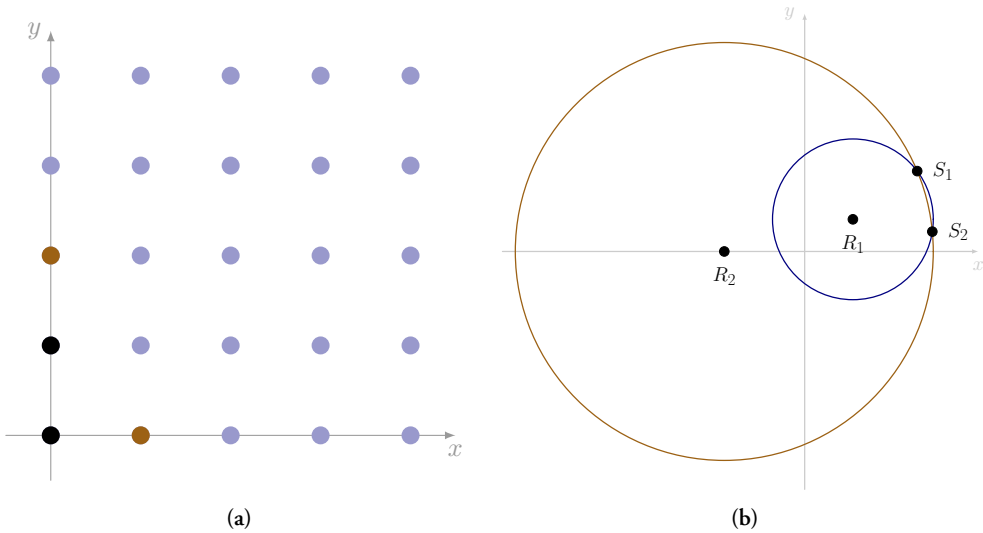
The set of leading monomials in our Gröbner basis is $\{x, y^2\}$, which means that the standard monomials are $\{1, y\}$, see Figure 5.3a. Hence, we know we have at most two solutions. Choosing y as the action polynomial, we then need to reduce the polynomials in αX_b using the division algorithm with the Gröbner basis

$$y \begin{bmatrix} y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{106}{17} & \frac{-105}{17} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ 1 \end{bmatrix}. \tag{5.27}$$

Finally, we solve for y by computing the eigenvalues of A. For each solution to y we then back-substitute into an earlier equation, e.g. $4x + y - 33 = 0$. The candidate solutions are then filtered by verifying that they solve all of our original equations. This gives us the solutions

$$(x, y) \in \left\{ (7, 5), \left(\frac{135}{17}, \frac{21}{17} \right) \right\}. \tag{5.28}$$

The solutions are also shown in Figure 5.3b.



(a) (b)

Figure 5.3: **(a)** The point at (α, β) should be interpreted as the set of polynomials with $x^\alpha y^\beta$ as the leading monomial. The Gröbner basis $\{4x + y - 33, 17y^2 - 106y + 105\}$ is marked as the two bronze points. Any polynomial corresponding to a non-black point can be reduced using the Gröbner basis into one of the black points, which are the standard monomials. **(b)** The figure shows the variety corresponding to the first (blue) and second (bronze) equation. The variety corresponding to the equation system consists of the two points $\{S_1, S_2\}$, which are the possible positions of the sender. Also marked in the figure are the known positions of the receivers $\{R_1, R_2\}$.

5.2.3 Elimination Template

As described in Chapter 3, a typical setting where solving polynomial equation systems is relevant is in a hypothesize-and-test framework such as RANSAC. In this case, we want to solve many instances of the same system of polynomial equations. These instances are all made up of the same monomials terms but possibly with different coefficients. It is possible to solve each of the problem instances separately using the method from the previous section. However, since there is a common structure among the problem instances, the solution can be found quicker by using something called an elimination template.

To understand the idea behind an elimination template, consider our previous solution approach. The main problem was to reduce αX_b to a linear combination of X_b using our given equations f_i. We did this by first finding a Gröbner basis and then using it and the

division algorithm to find $h_i \in \mathbb{K}[X]$ such that,

$$\alpha X_b - \sum_i c_i X_{bi} = \sum_i h_i g_i, \tag{5.29}$$

where $c_i \in \mathbb{K}$. Each of our Gröbner basis elements is a combination of our original equations

$$g_i = \sum_j q_{ij} f_j, \tag{5.30}$$

where $q_{ij} \in \mathbb{K}[X]$. Using this, we can rewrite the reduction to be in terms of the original equations

$$\alpha X_b - \sum_i c_i X_{bi} = \sum_i h_i (\sum_j q_{ij} f_j,) = \sum_j w_j f_j, \tag{5.31}$$

with

$$w_j = \sum_i h_i q_{ij}. \tag{5.32}$$

The idea behind an elimination template is that we first study an example problem instance and compute h_i for it. We then use that for almost all problem instances only the coefficients of h_i will change, i.e. they will be made up of the same monomials (see the end of this section for a discussion of this assumption). If we know the monomials in h_i and want to find the coefficients for a new problem instance we set up the matrix equation $CX = 0$ where X contains all the monomials present in $h_i f_i$ for some i, and the matrix C, called the elimination template, contains the coefficients such that each row of CX corresponds to one monomial in w_i multiplied by f_i. Since we are free to choose the permutation of our monomials we choose to order them as

$$X = \begin{bmatrix} X_e \\ X_r \\ X_b \end{bmatrix}, \tag{5.33}$$

where X_b is again the standard monomials, X_r are the monomials we want to reduce, i.e. αX_b, except the monomials already present in X_b, and X_e are the rest of the monomials present in $w_i f_i$, for some i. Correspondingly, we now have

$$C = \begin{bmatrix} C_e & C_r & C_b \end{bmatrix}, \tag{5.34}$$

which is made up of the coefficients from our equations f_i. By performing Gaussian elimination on C we can reduce it such that the bottom rows of it become

$$\begin{bmatrix} 0 & I & -A \end{bmatrix}, \tag{5.35}$$

where A corresponds to the action value matrix for this problem instance [70]. The key advantage of using this method to find A is that performing the Gaussian elimination is

faster than the method described in the previous section. The size of C will affect how fast a solution can be found and the numerical stability of the solver, and there exist methods to sometimes, such as when the solutions are symmetric, reduce the size of C, see [73–75].

As for our assumption that w_i will almost always contain the exact same monomials for different problem instances, this will be the case as long as in all the steps, i.e. when computing a Gröbner basis and reducing αX_b, the same terms cancel for both problem instances. If the original equations have coefficients in $\mathbb{C}$, one can show that this condition is almost always true. This is because two terms canceling is the same thing as the coefficients being a root of some polynomial. This almost never happens in the sense that the set of roots to a non-zero polynomial has lower dimension than the number of variables. In practice, we always have some finite precision when representing numbers, so there is some probability that our assumption is not satisfied. However, since we are using this method inside a RANSAC framework this is not much of a problem, since it can be mitigated by running more iterations.

5.2.4 Saturation

For some problems, we know ahead of time that only some solutions are of interest. Often this occurs as a consequence of how the modeling of the problem is done. In this situation, we would like to remove unwanted solutions from the variety. By reducing the number of solutions, we typically get a solver that is both faster and more numerically stable. For some problems, the variety might not be zero-dimensional initially, so removing solutions is necessary before the method described in this chapter can be applied, see Figure 5.4. Removing solutions where the polynomial g is zero can be done using Rabinowitsch trick [76] and considering the larger ideal

$$J = \langle f_1, \dots, f_m, 1 - gy \rangle, \tag{5.36}$$

where we introduce y as a new variable in addition to the original variables $x_1, \dots, x_n$. To see that this works, we check what points are in the corresponding variety $V(J)$. Consider the three cases:

- If $f(x) \neq 0$ then $x \notin V(J)$
- If $f(x) = 0$ and $g = 0$, then $x \notin V(J)$ since $1 - g(x)y = 1 - 0y \neq 0$
- If $f(x) = 0$ and $g(x) \neq 0$, choosing $y = \frac{1}{g(x)}$ ensures that $1 - g(x)y = 0$, so $x \in V(J)$

This means that the projection

$$\{x \,|\, \exists y,\, (x, y) \in V(J)\} = V(\langle f_1, \dots, f_m \rangle) \setminus V(\langle g \rangle), \tag{5.37}$$

contains the solution set we want. Since y is an auxiliary variable, we would prefer not to have to solve for it along with x. There are ways of achieving this, see [69, 77].

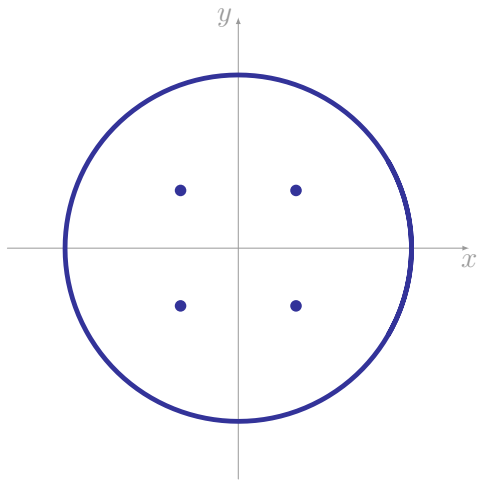


Figure 5.4: The variety corresponding to the ideal generated by $f_1 = (x-1)(x+1)(x^2+y^2-9)$ and $f_2 = (y-1)(y+1)(x^2+y^2-9)$. To find the four discrete points, the techniques described in this chapter cannot be applied since the variety is not zero-dimensional. The solution is to first remove the points along the circle (x^2+y^2-9) using saturation.

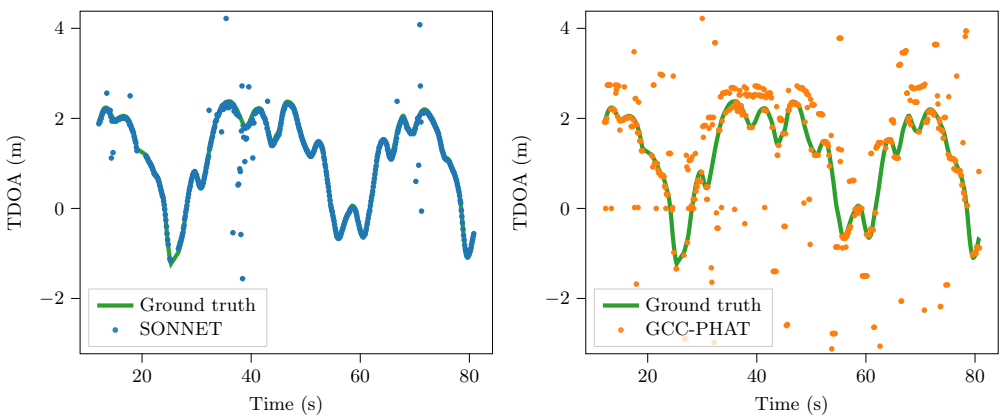
5.2.5 Automatic Solver Generation

While one can perform the computation required by the above method by hand, there exists good software for working with polynomial ideals. Two examples of such software are Macaulay2 [78] and the Python package SymPy [79]. There also exist tools [80, 81], known as automatic solver generators, to allow the user to specify a set of equations and more directly get a solver from these. In this thesis, [81] has mainly been used which uses MATLAB [82] and Macaulay2.

Chapter 6

Summary and Contributions

I **SONNET: Enhancing Time Delay Estimation by Leveraging Simulated Audio**
E. Tegler, M. Oskarsson and K. Åström
International Conference on Pattern Recognition (ICPR), 2024.



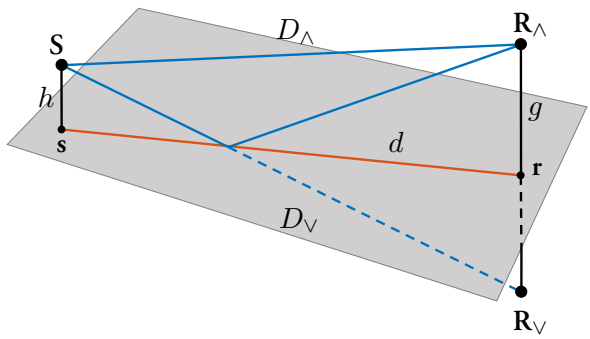
Scientific contribution: The problem considered in this paper is the signal processing task of time delay estimation. While there exist classical methods for reverberation-free environment with Gaussian noise, these methods typically struggle in real-world scenarios when their theoretical assumptions are invalid. Furthermore, it is unclear how to extend classical methods to improve performance in environments with reverberation. One possible reason for this is that classical methods consider all possible signals, while only a small subset are naturally occurring signals. This is one of multiple reason why learning-based

systems could outperform classical methods at time delay estimation. The contribution of this paper consists of demonstrating this, as well as giving access to the proposed system. Since learning-based systems need data, the other contribution of the paper is showing that this problem can be solved by using simulated data. This approach is also a more generally applicable to some problems where we can generate new problem instances but lack solution methods.

Author's contribution: For this paper ET came up with the idea, performed the experiments and did the majority of the writing. MO and KÅ provided support by supervising and helping with writing.

II Sensor Node Calibration in Presence of a Dominant Reflective Plane

E. Tegler, M. Larsson, M. Oskarsson and K. Åström
European Signal Processing Conference (EUSIPCO), 2022.



Scientific contribution: While reverberation increases the difficulty of the signal processing task, it may also increase the information content in the recorded signal. This paper studies this in the simplest form of a reverberant environment: a single plane. The sound produced by a speaker can then be described as having two paths to each microphone: a line-of-sight path and a reflected path. Given the TDOA measurements between all pairs of paths (to different microphones), the paper shows that a stratified approach can be used in order to solve for: offsets, distances from reflective plane, projections in the reflective plane. One key advantage of using this method is that the required number of microphones and sound events decreases to (3,4) from (6,8), which is required in the non-reverberant

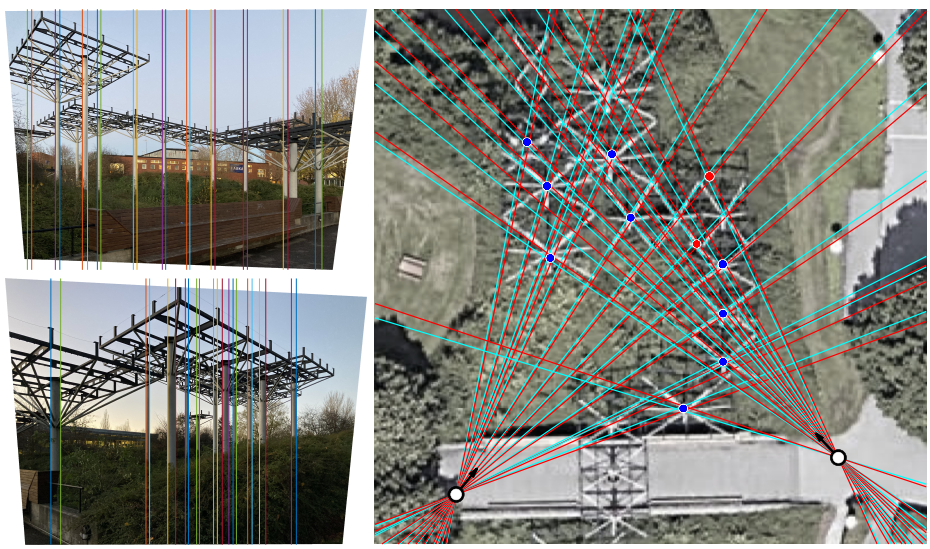
case. Since the polynomial system is smaller, the solver also runs faster and exhibits better numerical stability. Because the approach is stratified, it can also solve the problem of self-calibration given TOA measurements for all paths, by skipping the initial offset estimation. In the TOA case, the improvement lies in decreasing the required number of microphones and sound events to (3,4) from (4,6). The proposed method also estimates the position of the plane with respect to the microphones. This is of interest because it allows geometric estimation of the environment.

Author's contribution: The original idea was developed by KÅ and MO. The experiments and writing were done jointly by all the authors.

III The Multi-view Geometry of Parallel Cylinders

E. Tegler, J. Engman, D. Gillsjö, G. Flood, V. Larsson, M. Oskarsson and K. Åström

Scandinavian Conference on Image Analysis (SCIA), 2023.



Scientific contribution: In vision the basic component used for geometric estimation is usually a keypoint, i.e. a salient point in an image. One problem with using keypoints is that they are typically hard to match between images when the change in viewing direction is large. One approach to address this problem is to use semantic information instead, so

rather than using the coordinates of a keypoint one uses information such as what types of objects one can see. The main idea is that we might be able to identify the semantic class of an object from most directions. Another advantage of semantic positioning is that we can select features to be more stable over time. This paper aims to combine the advantages of both paradigms. Instead of relying on keypoints, we study larger primitives: cylinders. This means that we get more geometric information than pure semantic positioning, allowing for more accurate position estimation. The paper also demonstrates localization under very large viewpoint changes. The paper specifically contributes solutions to the minimal problem of simultaneously solving for the structure (the position and the size of the cylinders) and the camera poses in the case of parallel cylinders. This paper is also connected to audio localization. Imagine multiple microphone arrays calibrated to register the same sound pressure as the same amplitude. For each sound played, both an amplitude and a direction can be registered for each microphone array. By associating the width of a circle with the amplitude of sound, we get the same geometric problem as presented in the paper. So the minimal solvers developed in the paper is directly applicable to this sound application.

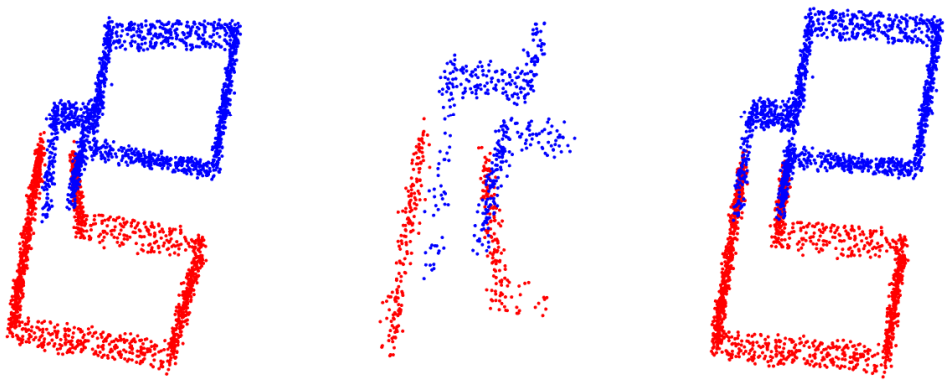
Author's contribution: The main ideas for the paper were developed by ET and KÅ, based on previous work of MO. Implementation and experiments was primarily performed by JE and ET and MO. The writing of the paper was done jointly by all the authors.

IV **Detection and Localization of Drones and UAVs Using Sound and Vision**
E. Tegler, M. Modig, P. Skarin, K. Åström, M. Oskarsson and G. Flood
Computer Vision and Pattern Recognition Workshops (CVPRW), 2025.

Scientific contribution: This paper should be viewed as an application paper, where the contribution lies in demonstrating possibilities and limitations of current techniques. The problem studied is detecting and estimating the location of drones using an ad-hoc microphone array. As an application, the localization of drones has gained significant attention in recent years. While it is possible to find drones using many types of sensors, and each sensor has its own set of advantages, using passive sensors offers low detectability and in general low power consumption. In this paper we demonstrate that it is possible to detect drones using sound at a distance of several hundred meters, even under windy conditions. This is accomplished by computing TDOA measurements between pairs of microphones combined with grid searching to estimate the best angle of incidence under the assumption of a planar wave. A further contribution also lies in the collection and public release of relevant data for the this application.

Author's contribution: The data collection was done in collaboration by all the authors. ET did the majority of work with data analysis and results. The paper was written by all the authors.

V **Minimal Solvers for Point Cloud Matching with Statistical Deformations**
G. Flood, E. Tegler, D. Gillsjö, A. Heyden and K. Åström
International Conference on Pattern Recognition (ICPR), 2022.

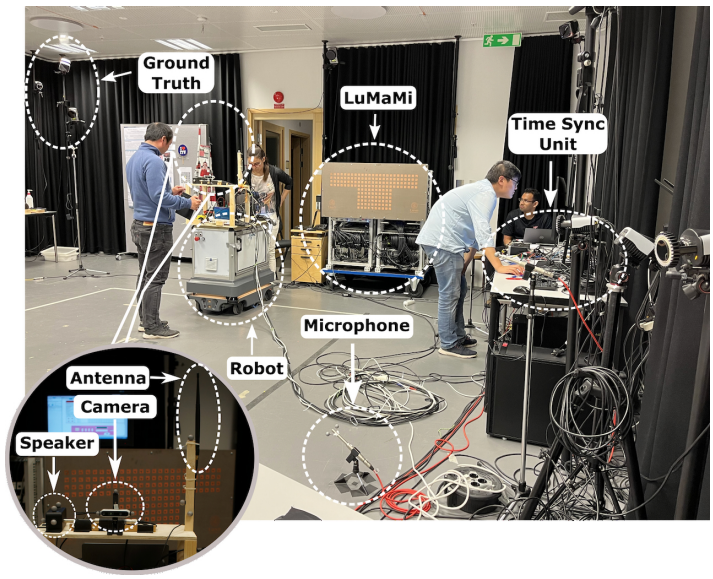


Scientific contribution: While most of this thesis deals with how to create accurate 3D maps from sensor data, this paper instead deals with a post-processing step of map creation. The paper builds on the observation that the same 3D point is sometimes represented multiple times. This can happen in two ways. The first way is that the same real-world point has multiple representations within a single map. This is commonly referred to as the loop closure problem, since it often occurs when the camera return to a location it has previously recorded. In this case, the accuracy of the map can be improved if the information that certain points corresponds to the same location is incorporated. The second way is when there exist multiple maps of the same environment, hence it is natural that some points are represented in multiple maps. In this case, the matched points provide an opportunity to merge the maps into one large map. A problem in both cases is that by adding the constraints that some points should end up in the same location, the residuals minimized during map creation will increase. The paper solves this problem by searching in directions which will increase the residuals as little as possible. These directions are found by computing the eigenvectors corresponding to the smallest eigenvalues of the Hessian of the optimization problem. The paper develops methods to do this search efficiently.

Author's contribution: ET did the work with developing and writing the section about merging duplicated points within a point cloud. GF took a leading role in both writing and research. The idea to use minimal solvers to refine matches came from KÅ, based on discussion with the other coauthors.

VI The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization

I. Yaman, G. Tian, M. Larsson, P. Persson, M. Sandra, A. Dürr, E. Tegler, N. Challa, H. Garde, F. Tufvesson, K. Åström, O. Edfors, S. Malkowsky, L. Liu
International Conference on Robotics and Automation (ICRA), 2024.



Scientific contribution: As this is a dataset paper the main contribution is the collection and publication of data. In particular this dataset simultaneously collects data from multiple sensor modalities: vision, radio and audio. The idea behind the simultaneous collection was both to allow for comparison between different positioning methods, but also to provide a dataset enabling sensor fusion between the different modalities.

Author's contribution: The main work was done by IY, this also includes organizing the collaboration between the large number of coauthors. The parts of the experiment which dealt with audio was done by ML with support from ET, and KÅ. ET did the main body

of work with post-processing of the audio, such as: time synching, mapping into common coordinate system and organizing the data.

References

[1] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[2] David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. Prentice hall professional technical reference, 2002.

[3] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[4] Erwin Kruppa. *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung*. Hölder, 1913.

[5] Merrill Ivan Skolnik. *Introduction to radar systems*, volume 3. McGraw-hill New York, 1980.

[6] Ninad Mehendale and Srushti Neoge. Review on lidar technology. *Available at SSRN 3604309*, 2020.

[7] Mark A Poletti. Active acoustic systems for the control of room acoustics. *Noise & Vibration Worldwide*, 44(4):10–26, 2013.

[8] Jean-Marc Jot, Rémi Audfray, Mark Hertensteiner, and Brian Schmidt. Rendering spatial sound for interoperable experiences in the audio metaverse. In *2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, pages 1–15. IEEE, 2021.

[9] Jihui Zhang, Thushara D Abhayapala, Wen Zhang, and Prasanga N Samarasinghe. Active noise control over space: A subspace method for performance analysis. *Applied Sciences*, 9(6):1250, 2019.

[10] Steven L Garrett. *Understanding acoustics: an experimentalist's view of sound and vibration*. Springer Nature, 2020.

[11] H.D. Young and R.A. Freedman. *University Physics*. Pearson, 2019. ISBN 9780135216118. URL `https://books.google.se/books?id=PULlxAEACAAJ`.

[12] Malcolm J Crocker. Theory of sound—predictions and measurement. *Handbook of noise and vibration control*, pages 17–42, 2007.

[13] Richard Phillips Feynman. The feynman lectures on physics. *(No Title)*, 1:46, 1963.

[14] Jean le Rond d'Alembert. Recherches sur la courbe que forme une corde tendue mise en vibration. 1747.

[15] Luca Nanni. Solving wave equations in the space of schwartz distributions: the beauty of generalised functions in physics. *Indian Journal of Pure and Applied Mathematics*, pages 1–11, 2024.

[16] Fraidoon Mazda. *Telecommunications engineer's reference book*. Butterworth-Heinemann, 2014.

[17] JW Strutt and Baron RayLEIGH. The problem of the whispering gallery. *Philosophical magazine*, 20(5):1001–1004, 1910.

[18] Dennis A Bohn. Environmental effects on the speed of sound. *J. Audio Eng. Soc*, 36 (4):223–231, 1988.

[19] Henry E Bass, Louis C Sutherland, Allen J Zuckerwar, David T Blackstock, and DM Hester. Atmospheric absorption of sound: Further developments. 1995.

[20] Andrew S Glassner. *An introduction to ray tracing*. Morgan Kaufmann, 1989.

[21] Lauri Savioja and U Peter Svensson. Overview of geometrical room acoustic modeling techniques. *The Journal of the Acoustical Society of America*, 138(2):708–730, 2015.

[22] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[23] Frederik Michel Dekking. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media, 2005.

[24] David Hume. *A treatise of human nature*. Oxford University Press, 2000.

[25] Roger Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge University Press, 1955.

[26] Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

[27] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[28] Albert E Beaton and John W Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

[29] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[30] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[32] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[33] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[34] Simon JD Prince. *Understanding deep learning*. MIT press, 2023.

[35] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[37] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

[39] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[40] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[41] Ping-yeh Chiang, Renkun Ni, David Yu Miller, Arpit Bansal, Jonas Geiping, Micah Goldblum, and Tom Goldstein. Loss landscapes are all you need: Neural network generalization can be explained without the implicit bias of gradient descent. In *The Eleventh International Conference on Learning Representations*, 2022.

[42] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

[43] Richard E. Bellman. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. URL `https://books.google.se/books?id=rZW4ugAACAAJ`.

[44] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[45] Enrique Osvaldo Civitarese and Manuel Gadella Urquiza. *Methods in Statistical Mechanics*. Springer, 2020.

[46] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[47] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[48] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[49] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[51] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[52] An B Vuong, Michael T McCann, Javier E Santos, and Yen Ting Lin. Are we really learning the score function? reinterpreting diffusion models through wasserstein gradient flow matching. *arXiv preprint arXiv:2509.00336*, 2025.

[53] Martin Larsson. *Localization using distance geometry: Minimal solvers and robust methods for sensor network self-calibration*. Phd thesis, 2022.

[54] Gabrielle Flood. *Mapping and Merging Using Sound and Vision: Automatic Calibration and Map Fusion with Statistical Deformations*. Phd thesis, Lund University, 2021.

[55] Gabrielle Flood, Anders Heyden, and Kalle Åström. Stochastic analysis of time-difference and doppler estimates for audio signals. In *International Conference on Pattern Recognition Applications and Methods*, pages 116–138. Springer, 2018.

[56] Charles H. Knapp and G.Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24(4):320–327, 1976.

[57] Pierre Duhamel and Martin Vetterli. Fast fourier transforms: a tutorial review and a state of the art. *Signal processing*, 19(4):259–299, 1990.

[58] Chris Rizos, Andrew G Dempster, Binghao Li, and James Salter. Indoor positioning techniques based on wireless lan. 2007.

[59] Xuhong Li, Kenneth Batstone, Kalle Åström, Magnus Oskarsson, Carl Gustafson, and Fredrik Tufvesson. Robust phase-based positioning using massive mimo with limited bandwidth. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7. IEEE, 2017.

[60] Martin Larsson, Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Optimal trilateration is an eigenvalue problem. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5586–5590. IEEE, 2019.

[61] Yubin Kuang and Kalle Åström. Stratified sensor network self-calibration from tdoa measurements. In *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5. IEEE, 2013.

[62] Martin Larsson, Gabrielle Flood, Magnus Oskarsson, and Kalle Åström. Fast and robust stratified self-calibration using time-difference-of-arrival measurements. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4640–4644. IEEE, 2021.

[63] Martin Larsson, Gabrielle Flood, Magnus Oskarsson, and Kalle Åström. Upgrade methods for stratified sensor network self-calibration. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4851–4855. IEEE, 2020.

[64] Niels Henrik Abel. *Mémoire sur les équations algébriques, où on demontre l'impossibilité de la résolution de l'équation générale du cinquième dégré*. 1824.

[65] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[66] John GF Francis. The qr transformation a unitary analogue to the lr transformation—part 1. *The Computer Journal*, 4(3):265–271, 1961.

[67] Vera N Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Mathematical Physics*, 1(3): 637–657, 1962.

[68] Gene H Golub and Charles F Van Loan. Matrix computations, forth edition, 2013.

[69] David A Cox, John B Little, and Donal O'Shea. Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra. *Undergraduate Texts in Mathematics*, 1997.

[70] Viktor Larsson. *Computational Methods for Computer Vision: Minimal Solvers and Convex Relaxations*. Phd thesis, Lund University, 2018.

[71] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. Phd thesis, University of Innsbruck, 1965.

[72] David A Cox, John Little, and Donal O'shea. *Using algebraic geometry*. Springer Science & Business Media, second edition edition, 2005.

[73] Viktor Larsson and Kalle Åström. Uncovering symmetries in polynomial systems. In *European Conference on Computer Vision*, pages 252–267. Springer, 2016.

[74] Erik Ask, Yubin Kuang, and Kalle Åström. Exploiting p-fold symmetries for faster polynomial equation solving. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3232–3235. IEEE, 2012.

[75] Yubin Kuang, Yinqiang Zheng, and Kalle Astrom. Partial symmetry in polynomial systems and its applications in computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 438–445, 2014.

[76] JL Rabinowitsch. Zum hilbertschen nullstellensatz. *Mathematische Annalen*, 102(1): 520–520, 1930.

[77] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Polynomial solvers for saturated ideals. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2288–2297. IEEE, 2017.

[78] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at `http://www2.macaulay2.com`.

[79] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL `https://doi.org/10.7717/peerj-cs.103`.

[80] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *European Conference on Computer Vision (ECCV)*, pages 302–315. Springer, 2008.

[81] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 820–829. IEEE, 2017.

[82] The MathWorks Inc. Matlab version: 9.12.0 (r2022a), 2022. URL `https://www.mathworks.com`.

Scientific Publications

Printed by Media-Tryck, Lund 2025 NORDIC SWAN ECOLABEL 3041 0903

Doctoral Theses in Mathematical Sciences 2025:6
ISBN 978-91-8104-763-9
ISSN 1404-0034