

Blind as a bat: spatial perception without sight

Présentée le 26 novembre 2021

Faculté informatique et communications
Laboratoire de communications audiovisuelles
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Frederike DÜMBGEN

Acceptée sur proposition du jury

Prof. A. Argyraki, présidente du jury
Prof. M. Vetterli, Dr A. J. Scholefield, directeurs de thèse
Prof. D. Scaramuzza, rapporteur
Prof. L. Carlone, rapporteur
Prof. A. Martinoli, rapporteur

Für meine Familie, meine besseren fünf Sechstel.

Acknowledgments

On ne voit bien qu'avec le cœur. L'essentiel est invisible pour les yeux.
— *Antoine de Saint-Exupéry*

This PhD marks an important milestone for me, and I want to take this moment to thank everyone who made this journey so special. Martin, you promised me that I would be able to pursue my own ideas if I were to come to your lab, and it could have not been more true. Thank you for your immense support and trust, for relentlessly helping us find what is the best solution (or problem) for us, while never forgetting our well-being (“how is the violin?”). Thank you for letting me be stubborn and learn from my own mistakes, and guiding me back to the correct path whenever needed. There was never a meeting after which I did not feel more inspired than before.

Adam, you have been a constant mentor, a great collaborator and a source of inspiration. Thank you for staying up late with me on multiple occasions, for supporting me in meeting tight deadlines, and for hands-on help with experiments, put simply: thank you for always, really always, lending a helping hand, even when juggling a baby and a start-up at the same time. You always kept the right directions and taught me what’s important in research. To the thesis jury, thank you for spending your valuable time to read these pages and challenge the work, which really helped improving and finalizing it. Katerina, thank you for being a great support and making everything go smoothly.

Karen, you helped immensely with this thesis, be it by carefully reading every single chapter of it, but also by regularly checking in on me. Most importantly, thank you and Bernd for making sure I’d still have good times (and food!) during the process. Adrien, you not only were extremely important for the drone project as a skilled engineer, you also for sure made the work much more fun. Whether it was by discussing music, travels or the latest tools, it was always hard to not get sidetracked with you. Thank you for teaching me so much about hardware, and for letting me teach you git in exchange. Mihailo, thank you for always being interested in my work and well being, even after you left the lab, giving great advice and always finding that one thing in my results that needs further investigation. Robin, if my Master’s project with you had not been this fun, I would not have joined the lab, so thank you! Nicoletta, thank you for keeping the lab together, for organizing wonderful outings that I will always remember, and for masterfully dealing with all the paperwork that we create, saving us so much time and energy.

Thank you to all my bright collaborators. Michalina, working with you was extremely enrich-

Acknowledgments

ing. Thank you for always patiently sharing your knowledge, for being a great collaborator on code, and for your help during our last joint mile. Thank you to Marta, Gilles, Miranda and Golnoosh for creating and letting me join their team and teaching me early on in the PhD how to conduct research, write a journal paper and how to deal with that third reviewer. Miranda, it was always great to have you as a comrade working on localization and discuss new ideas. Cynthia, thank you for teaching me how to write code and data documentation more properly, I always think of you now whenever I write too much code and too little documentation. Finally, thank you to my students who trusted me to guide them through their Bachelor and Master's semester projects, some of which ended up concretely contributing to this thesis (and others which allowed me to explore orthogonal ideas): Isaac, for the first prototypes of the audio deck, Pedro, for the successful learning-based solution for indoor localization and Emilia for the drone simulation framework. Katie, Lutz and Renate, thank you for proof-reading the abstracts of the thesis.

Thank you to all members of my extended lab families that I did not directly collaborate with, but that I enjoyed all other aspects of campus life with: Adrian, Benjamin, Dalia, Eric, Julien, Matthieu, Paolo and Sepand from LCAV, Edo, Marjan, RK and Ruofan from IVRL, Isinsu, Jan, Martin, Sena and Victor from CV lab and Enrica, Enrico, Fabian, Valentin and Vivek from LIS. Being surrounded by so many skilled researchers and kind people during these five years has been a great gift.

Many quotes in this thesis are contributed by friends and colleagues and are a symbol for how important all of you are for me. Majed, you were my first collaborator and you have stayed such a great colleague and friend ever since. Thank you for our countless lake walks, hackathons and joint research and course projects, where you shared your knowledge, enthusiasm and humour. Vidit, Fayez and Pei-i (“the flatmates”), thank you for generously sharing great food and music, sports and games, and always an open ear. Ali, Ariane, Sonja — my own flatmates — thank you for always reminding me that there is more to life than a PhD when I came home. Katie, the time we shared during our Master's prepared me for this journey, in terms of nerd skills, but more importantly, in terms of life skills. Thank you! Cyril, Elliott, Baptiste, Guillaume, Sarah, Etienne and Vincent, thank you for being such fun friends during studies and beyond, I hope we will still have regular dinners when we are gray and old. Finally, I had the great luck to stay close — geographically and emotionally — to many cherished friends from my hometown. My dear friends from Berne, thank you for staying in touch and making me feel like I am always welcome home.

Finally, I can not emphasize enough the immense support I receive from my family — Kim, Lena, Moritz, Renate and Lutz, this thesis is dedicated to you for being the most wonderful companions. Sven, you have been challenging this last point since the moment I met you, and I am so grateful that you are now part of my life. You are all great champions at mastering life, and I will always be striving to follow your lead. Thank you for your unconditional love and for always helping me recharge my batteries, wherever we are in the world.

Lausanne, November 16, 2021

Frederike Dürnberg

Abstract

Among our five senses, we rely mostly on audition and vision to perceive an environment. Our ears are able to detect stimuli from all directions, especially from obstructed and far-away objects. Even in smoke, harsh weather conditions, or at night — situations where our eyes struggle to operate — they are an essential source of information. On the other hand, our eyes add rich and instantaneous information to the auditory signals. These two senses are thus complementary; taken together, they constitute a powerful system for localization and navigation.

In this thesis, non-visual (“blind”) modalities are studied to solve spatial perception problems, particularly those involving localization and mapping in robotics. Although the potential of blind modalities has long been recognized for niche applications, including sonar for underwater navigation and radio-frequency signals for indoor localization, less progress has been made on blind methods than visual methods, and many interesting problems remain unsolved.

In the application-oriented Part I of the thesis, an indoor localization solution based on WiFi and Bluetooth measurements is proposed that, unlike competing approaches, does not require offline calibration. Next, the localization of a moving device is analyzed, based on single, sequential distance measurements, which constitutes a fundamental but unsolved variation of trilateration. This problem is solved using a closed-form algorithm that includes recovery guarantees. Finally, a drone is equipped with microphones and a buzzer, thus emulating a bat — an expert of blind navigation. Algorithms are presented to detect walls using echoes as well as external sound sources. Throughout these chapters, the optimal exploitation of a device’s motion, which is commonly ignored in sensing algorithms, is studied for improving spatial perception.

Although blind modalities carry less information than images, they can be used to extract fundamental features, such as distances or angles between objects of interest. In Part II, we therefore treat the fundamental question of localization from distance and angle measurements. When performing localization from distance measurements, one can resort to distance geometry, a mature field with well-established results and methods. Many fewer results exist for angle measurements. Thus, novel localization algorithms are proposed to bridge this gap, using different angle measurements to supplement or replace distances. These algorithms are useful beyond localization, as they contribute to the much broader field of low-dimensional embedding of entities, such as images, words, or abstract concepts.

Overall, novel algorithms, systems, and theories are proposed for spatial perception from a

Abstract

variety of non-visual signals. Along the way, a wide range of important problems from localization is approached from a signal-processing perspective. This enables the formulation of optimal and guaranteed algorithms in some cases, and efficient approximate solutions otherwise. Guaranteeing optimality has recently emerged as a pressing problem for achieving robust real-world operation in robotics; this thesis contributes to this ambitious goal.

Keywords: spatial perception, localization, mapping, robotics, signal processing, distance geometry, radio frequency, sound

Zusammenfassung

Von unseren fünf Sinnen verlassen wir uns hauptsächlich auf das Hören und Sehen, um eine Umgebung wahrzunehmen. Unsere Ohren sind in der Lage, Reize aus allen Richtungen wahrzunehmen, insbesondere von verdeckten und weit entfernten Objekten. Selbst bei Rauch, rauen Wetterbedingungen oder nachts – Situationen, in denen unsere Augen Schwierigkeiten haben zu funktionieren – sind sie eine wesentliche Informationsquelle. Auf der anderen Seite fügen unsere Augen den Hörsignalen reichhaltige und sofortige Informationen über unsere Umgebung hinzu. Die beiden Sinne ergänzen sich also und bilden zusammen ein leistungsfähiges System zur Lokalisierung und Navigation.

Wir untersuchen nicht-visuelle (“blinde”) Modalitäten zur Lösung von Kernproblemen der räumlichen Wahrnehmung, insbesondere Lokalisierung und Kartierung für die Robotik. Obwohl das Potenzial von blinden Modalitäten seit langem für Nischenanwendungen anerkannt ist, darunter Sonar für die Unterwassernavigation und Hochfrequenzsignale für die Indoor-Lokalisierung, wurden für diese Modalitäten, verglichen mit visuellen Methoden, in den letzten Jahren deutlich weniger Fortschritte gemacht, und viele interessante Probleme bleiben ungelöst.

Im ersten, anwendungsorientierten Teil der Arbeit schlagen wir zunächst eine Indoor-Lokalisierungslösung basierend auf WiFi- und Bluetooth-Messungen vor, die im Gegensatz zu konkurrierenden Ansätzen keine Offline-Kalibrierung erfordert. Als nächstes untersuchen wir die Lokalisierung eines sich bewegenden Geräts basierend auf jeweils einer Entfernungsmessung, die eine grundlegende, aber ungelöste Variation der Trilateration darstellt. Wir lösen das Problem mit einem optimalen Algorithmus inklusive Wiederherstellungsgarantien. Schließlich statten wir eine Drohne mit Mikrofonen und einem Summer aus und emulieren damit eine Fledermaus — die Expertin für blinde Navigation. Wir präsentieren Algorithmen zur Erkennung von Wänden mithilfe von Echos sowie zur Lokalisierung externer Schallquellen. Der rote Faden dieses ersten Teils der Arbeit ist die optimale Nutzung der Bewegung eines Geräts für die räumliche Wahrnehmung.

Der zweite Teil der Arbeit behandelt die Lokalisierung aufgrund von Entfernungs- und Winkelmessungen auf einer abstrakteren Ebene. Obwohl blinde Modalitäten weniger Informationen als Bilder enthalten, können sie verwendet werden, um grundlegende Merkmale wie Abstände oder Winkel zwischen interessierenden Objekten zu extrahieren. Tatsächlich verwenden alle im ersten Teil der Arbeit diskutierten Algorithmen Distanzen oder Winkel, oder liefern Schätzungen derselben. Bei der Lokalisierung mit Distanzen kann man auf die Distanzgeometrie zurückgreifen, ein ausgereiftes Gebiet mit etablierten Ergebnissen

Zusammenfassung

und Methoden. Für Winkelmessungen liegen wesentlich weniger Ergebnisse vor. Um diese Lücke zu schließen, stellen wir neuartige Algorithmen zur Lokalisierung mit verschiedenen Arten von Winkelmessungen bereit, welche Entfernungsmessungen ergänzen oder ersetzen. Diese Algorithmen sind über die Lokalisierung hinaus nützlich: Sie tragen zum breiteren Feld der niederdimensionalen Einbettung von Entitäten wie Bildern, Wörtern oder abstrakten Konzepten bei.

Zusammenfassend präsentieren wir neuartige Algorithmen, Systeme und Theorien zur räumlichen Wahrnehmung durch eine Vielzahl nicht-visueller Signale. Dabei behandeln wir ein breites Spektrum wichtiger Problemstellungen aus der Lokalisierung, die wir aus der Perspektive der Signalverarbeitung angehen. Dies ermöglicht es uns, in einigen Fällen optimale und garantierte Algorithmen, und andernfalls effiziente Näherungslösungen zu formulieren. Optimalität und Garantien in die Robotik zu bringen, wurde kürzlich als eines der drängenden Probleme für dessen robusten Betrieb in der realen Welt beschrieben, und mit dieser Arbeit tragen wir dazu bei, diesem ehrgeizigen Ziel näher zu kommen.

Stichwörter: räumliche Wahrnehmung, Lokalisierung, Kartierung, Robotik, Signalverarbeitung, Distanzgeometrie, Radiofrequenzen, Ton

Contents

Acknowledgments	1
Abstract (English/Deutsch)	3
List of Acronyms	11
1. Overview of this thesis	13
1.1. Blind spatial perception for moving devices	15
1.1.1. Overview of the first part of the thesis	17
1.1.2. Audio, the undervalued modality for spatial perception	19
1.1.3. List of contributions of the first part of the thesis	21
1.2. Embedding based on angle and distance measurements	23
1.2.1. Overview of the second part of the thesis	25
1.2.2. List of contributions of the second part of the thesis	26
I. Blind spatial perception for moving devices	27
2. Background of the first part	29
2.1. Localization: notation and definitions	29
2.2. Landscape of solutions	31
2.3. Probabilistic robotics in a nutshell	32
2.3.1. Generative perspective	32
2.3.2. Discriminative perspective	34
2.3.3. Measurement modalities	35
2.4. Outlook	37
3. Multi-modal indoor localization	39
3.1. Problem setup	41
3.2. Related work	42
3.3. Conditional random fields	44
3.3.1. Modeling feature functions	47
3.3.2. Location inference	49
3.4. Processing and calibration	50
3.4.1. Outlier rejection	50
3.4.2. Online calibration	51
3.4.3. Feature weights	51
3.5. Results	52
3.5.1. Environment of experiments	52

3.5.2.	Overview of used technologies	53
3.5.3.	Data accuracy and calibration	54
3.5.4.	Evaluation of available modalities	55
3.5.5.	Comparison with other methods	56
3.6.	Conclusion and ongoing work	58
3.6.1.	Auto-encoder solution	60
3.6.2.	Non-visual teach and repeat	61
3.6.3.	Conditional neural fields	62
4.	Dynamical range-based localization	63
4.1.	Problem setup	64
4.2.	Related work	65
4.3.	Relax and Recover	67
4.3.1.	Relaxation	67
4.3.2.	Recovery	69
4.3.3.	Statistical analysis	71
4.4.	Results	72
4.4.1.	Noise study in simulation	73
4.4.2.	Results on lawn-mower dataset	74
4.5.	Conclusion and ongoing work	77
4.5.1.	Extension to splines	77
4.5.2.	Splines as the natural solution to regularized optimization	79
4.5.3.	Generalization to different models	80
4.A.	Calculation of covariance matrix	80
5.	Dynamical sound-based wall localization	81
5.1.	Problem setup	83
5.2.	Background	84
5.2.1.	Room impulse response reconstruction	84
5.2.2.	Interference function	85
5.3.	Related work	86
5.4.	Wall localization based on the interference function	88
5.4.1.	Compact form of interference function	89
5.4.2.	Problem setup revisited	90
5.4.3.	Probabilistic period estimation	91
5.4.4.	Frequency slices (solution to Subproblem 1)	94
5.4.5.	Distance slices (solution to Subproblem 2)	96
5.4.6.	Notes on practical implementation	97
5.4.7.	Recovery limits	98
5.5.	Simulation results	99
5.5.1.	Frequency slice algorithm	99
5.5.2.	Distance slice algorithm	101
5.6.	Experimental framework	102
5.6.1.	Hardware design	103
5.6.2.	<i>Crazyflie</i> firmware	104
5.6.3.	Communication	105
5.6.4.	Audio preprocessing	105
5.6.5.	Buzzer signal design	106
5.6.6.	Ground station	108

5.7. Experimental results	108
5.7.1. Interference matrix	108
5.7.2. Frequency slice algorithm	111
5.7.3. Distance slice algorithm	114
5.8. Preliminary results on the <i>e-puck2</i> robot	116
5.9. Conclusion and ongoing work	116
5.A. Window study	118
5.B. Interference matrix cleaning	119
6. Dynamical sound source localization	121
6.1. Problem setup	123
6.2. Related work	124
6.3. Background	126
6.3.1. Delay-and-Sum (DAS)	127
6.3.2. Minimum Variance Distortionless Response (MVDR)	128
6.3.3. Steered-response DOA	128
6.3.4. Recovery limits	129
6.3.5. Estimating the covariance matrix	132
6.4. Incorporation of movement	132
6.4.1. Time alignment method	133
6.4.2. Angle alignment method	134
6.5. Performance evaluation	134
6.6. Simulation results	137
6.7. Conclusion and ongoing work	139
II. Embedding based on angle and distance measurements	141
7. Background of the second part	143
7.1. Distance-based embedding	143
7.1.1. Problem statement	143
7.1.2. Lost information	145
7.1.3. Realizability	145
7.1.4. Recovery algorithms	146
7.1.5. Unlabeled sensing	147
7.1.6. Number of measurements	147
7.2. Introducing angles: notation and definitions	148
7.3. Distance and angle-based embedding: preliminary results	149
7.4. Alternative embeddings and metrics	151
7.5. Outlook	152
8. Embedding based on coordinate differences	153
8.1. Problem setup	154
8.2. Solution in one dimension: Coordinate Difference Matrices	156
8.2.1. Related work	156
8.2.2. Coordinate Difference Matrices	156
8.2.3. Properties of CDMs	157
8.2.4. Reconstructing coordinates based on CDMs	158
8.2.5. Comparison with EDMs	160

8.3. Solution in multiple dimensions: optimal <i>vs.</i> splitting	161
8.3.1. Vector form in one and multiple dimensions	161
8.3.2. Optimal solution	162
8.3.3. Splitting algorithm	162
8.3.4. Statistical Analysis	162
8.3.5. The cost of splitting	163
8.3.6. Algorithmic considerations	165
8.3.7. Uniqueness and number of solutions	168
8.4. Applications	168
8.4.1. Multi-modal sensor array localization	169
8.4.2. Sensor array calibration	171
8.5. Conclusion and ongoing work	173
9. Embedding based on inner angles	175
9.1. Problem setup	176
9.2. Related work	176
9.3. Degrees of freedom of realizable angles	178
9.4. Recovering points from angles	178
9.5. Characterization of realizable angles	179
9.5.1. Linear constraints	179
9.5.2. Automatic creation of linear constraints	180
9.5.3. Non-linear constraints	181
9.5.4. Guarantees	183
9.6. Denoising and recovery algorithm	183
9.7. Simulation results	184
9.7.1. Realizability study	185
9.7.2. Performance evaluation	185
9.8. Conclusion and ongoing work	186
9.A. Minimal set of linear constraints	186
9.B. Sketch of proof	187
10. Conclusion and future work	189
Bibliography	193

List of Acronyms

AOA	angle of arrival
AR	augmented reality
ARR	auto-reload register
BLE	Bluetooth low-energy
CCR	compare and capture register
CDM	coordinate difference matrix
CNF	conditional neural field
CRF	conditional random field
CRTP	Crazyflie realtime protocol
CSI	channel state information
DAS	delay and sum
DFT	discrete Fourier transform
DMA	direct memory access
DOA	direction of arrival
DTFT	discrete-time Fourier transform
EDM	Euclidean distance matrix
EEPROM	electrically erasable programmable read-only memory
EKF	extended Kalman filter
EVD	Eigenvalue decomposition
FFT	fast Fourier transform
FMCW	frequency-modulated continuous wave
HMM	hidden Markov model
HRTF	head-related transfer function
ILD	inter-aural level difference
IMU	inertial measurement unit
ITD	inter-aural time difference
LCMV	linearly constrained minimum variance
LM	Levenberg-Marquardt
LOS	line of sight
MDS	multi-dimensional scaling
MEMM	maximum entropy Markov model
MEMS	micro-electro-mechanical system
MLE	maximum likelihood estimator
MSE	mean squared error

List of Acronyms

MVDR	minimum variance distortionless response
NLOS	non-line-of-sight
NMR	nuclear magnetic resonance
PCB	printed circuit board
PSC	prescaler
PSD	positive semi-definite
PWM	pulse-width modulation
RF	radio-frequency
RIR	room impulse response
RLS	range-based least squares
RMSE	root mean squared error
ROS	robot operating system
RSSI	received signal strength indicator
RTT	round trip time
SAIL	synthetic aperture imaging lidar
SAR	synthetic aperture radar
SAS	synthetic aperture sonar
SDP	semi-definite program
SfM	structure from motion
SLAM	simultaneous localization and mapping
SNR	signal-to-noise ratio
SPI	serial peripheral interface
SRLS	squared-range based least squares
SSL	sound source localization
SVD	singular value decomposition
TDOA	time-difference of arrival
TOA	time of arrival
UWB	ultra wide band
WLS	weighted least squares

1 | Overview of this thesis

Part is realized as software, part is peopleware, part is hardware.
— *Margaret Hamilton*

In the current era of artificial intelligence, we are facing many difficult problems: how can an agent make morally sound decisions, how can it understand other intelligent agents (*e.g.* humans) and interpret their orders and intentions, or how can it choose from an infinity of possible actions at this very moment, in a way that it will achieve a certain goal in the far future. These questions and many others need to be answered for making a truly intelligent agent.

Although these are undoubtedly core components, there is more to an intelligent agent than such decision-making capabilities. In fact, many decision-making techniques rely on the assumption that the intelligent agent, *e.g.* the robot, can perceive the environment sufficiently well. This concept has been given the name “spatial perception”, and is omnipresent in robotics. It includes, for instance, creating three-dimensional representations of a scene, detecting objects of interest, and pose estimation for robotic manipulation. In this thesis, we focus on a core component of spatial perception: localization. Localization is crucial for successful operation of an intelligent agent, as the following examples highlight.

Many algorithms have been designed to calculate the sequence of commands to get a walking robot from point A to point B. Considering the complex dynamics of a walking robot, this is an extremely challenging task. However, above all, successful navigation requires the agent to localize itself as well as the end point, otherwise it does not know which direction to take or whether the goal is reached.

Other algorithms tell us how robots within a swarm can move to reach a common target. This requires potentially three types of localization: first, each swarm member needs to locate its neighboring members to avoid collision, the outer members need to detect and avoid obstacles, and finally, some members need a sense of the common target.

Finally, spatial perception is also important in applications that do not necessarily exhibit any movement. In the famous cocktail-party problem, an agent has to correctly decipher what another agent is saying, successfully cancelling out interfering noise sources. The agent can focus its attention in the direction of the sound source and attenuate all other

1. Overview of this thesis

Table 1.1.: Overview of the structure of the thesis.

	Chapter/Section	Anchors	Measurements	Platform
Part I	Chapter 3	known uncalibrated	multi-modal	mobile phone
	Chapter 4	known calibrated	distances	lawnmower
	Chapter 5	walls	distance and angle	drone
	Chapter 6	sound source	angle	drone (simulation)
Part II	Section 7.1	none	distances	–
	Section 7.3	none	distances and inner angles	–
	Chapter 8	none	distances and absolute angles	–
	Chapter 9	none	inner angles	–

sounds; but in order to do this, it needs to know where the desired sound is coming from — another example of spatial perception.

Humans and animals are astonishingly good at all three example problems, in particular thanks to their incredible sensing capabilities: complex processes are constantly running in our brains, combining information from all available senses, thus creating a coherent model of the world. When one modality fails, either temporarily or permanently, we are quick at adapting, trying to replace this modality with other sources of information, or filling the blanks through priors and beliefs.

Despite great progress in sensor technologies and algorithms, robots are far from being as good at spatial perception as animals or humans are. This is due to hardware limitations, but also due to a lack of algorithms and theory that are adapted to real-world settings.

In this thesis, we provide both theory and algorithms that aim to equip robots with tools to perform robust spatial perception when no or little visual information is available. We provide algorithms that, amongst other use cases, help in all aforementioned applications, and we provide missing theoretical results to understand such problems better from a mathematical perspective.

In the first part of the thesis, we perform spatial perception on a moving device, using mostly “blind” measurements. We study this problem in four different settings in terms of available sensing modalities, knowledge of the environment, and available computational resources. In the overview in Table 1.1, we give a summary of these aspects. For the environment, we focus on so-called “anchors”, static points or planes in the environment from which we get measurements. We work with WiFi, Bluetooth, Ultra-Wideband (UWB), auditive and — very

rarely — visual signals, from which we can infer mostly distances and/or angles. The first part of the thesis is close to actual applications, as each developed algorithm was evaluated on a real platform (see last column of Table 1.1): on smartphones, a lawnmower robot, and a micro-drone, respectively.

In the second part of the thesis, we move away from applications and treat the general theory of point embedding from distance and angle measurements. This theory could be directly applied to anchor-free localization, using different combinations of distance and angle measurements, as visualized in the last four rows of Table 1.1. The utility of this part however reaches beyond localization. Recovering point coordinates in any dimension, given similarity measurements such as angle and distance measurements, is a fundamental task with many applications; for instance it plays an important role in many data analysis and machine learning applications.

1.1. Blind spatial perception for moving devices

In a nutshell, the first part of this thesis is about spatial perception using “blind” modalities. It is easy to get a sense of the difficulty of the task by closing your eyes. While just before closing them, you had a very detailed idea of your environment, you suddenly lose almost all information. The wealth of visual information is mostly replaced by some auditory stimulations and tactile sensations, as well as proprioception: a feeling of where your different body parts are and how they move. However, as nature impressively demonstrates; not all is lost. A familiar specialist of blind navigation is the bat, which uses ultrasound to detect obstacles and prey. Quite impressively, some visually impaired people also learn how to navigate using clicking sounds only; more about this in Chapter 5. Even without these skills, everybody can move around for a certain amount of time, deprived of any sensory information, using proprioception only. This is achieved through a technique called “dead reckoning”: integrating your relative movement estimates in order to track your path. As we know from the popular kids experiment of turning on the spot with your eyes closed, after some time you lose the sense of orientation as the relative movement errors accumulate. We usually compensate for the drift inherent to dead reckoning by using visual information. In blind methods, we can compensate for it using other modalities; in this thesis, we use in particular sound, Bluetooth, WiFi, and ultra-wideband signals.

Why blind? It might seem like “switching off” visual information is making the spatial perception problem unnecessarily hard. However, having visual information is not always possible, necessary, or optimal. In fact, an optimal map contains the right granularity of detail for the task. Having too much detail can lead to slow decision making and increased energy consumption, while having too little detail can lead to inefficient navigation, to getting lost, and in the worse case to damage and malfunctioning. Below are some examples of applications where blind navigation might be preferred over visual navigation.

- Indoor localization: localizing moving people indoors, using *e.g.* their Smartphone data, should be performed “passively”, *i.e.* the target should not be required to constantly scan

1. Overview of this thesis

their environment. Ideally, the phone can be kept in the pocket and still be localized.

- **Microrobotics:** When dealing with ever-smaller drones or other microrobots, weight and size are crucial factors for successful deployment of sensors. Although cameras are becoming smaller and smaller, they are still bigger than some of the “blind sensors” available such as microphones, ultra-wideband antennas, etc. In particular, when the goal is omnidirectional sensing, multiple or more complex cameras need to be employed. Furthermore, it is not always realistic to equip a robot with the necessary processing power and bandwidth required to make use of visual information.
- **Hostile conditions:** When people or robots are employed for search and rescue in extreme situations, such as fires, in caves with bad visibility or in strong storms, visibility conditions can be prohibitively bad for visual localization solutions. Even some common environments exhibit too little features for reliable operation of visual feature-based methods: wide grass fields, empty rooms, or high altitudes are environments characterized by a lack of visual features. Laser-based solutions such as lidar could be used as a replacement but they are computationally heavier and more costly, and may fail when exposed to *e.g.* sunlight and transparent or reflective surfaces. Blind signals such as audio or radio-frequency signals are not affected by bad visibility or by an environment lacking unique visual features, and can be a great complement in these cases.

This list is of course not comprehensive and other use cases might call for blind solutions, for instance applications with privacy concerns.

The solutions proposed in this part of the thesis cover all three application areas. In Chapter 3 we propose an indoor localization solution using mostly blind signals and very scarce visual information. The solution proposed in Chapter 4 is appropriate for lawnmowers, as shown in its successful application for the latter. Finally, the solutions proposed in Chapter 5 and Chapter 6 are tailored to (but not limited to) micro-drones which would struggle with handling rich visual information, both in terms of the physical weight imposed by the sensor and the heavy computational payload of visual information.

Why moving? Finally, let us highlight why we are particularly interested in moving devices. The obvious reason is that many key problems in robotics involve movement, be it to map out or inspect an unknown environment or to perform a higher-level task such as collecting or delivering goods. A more subtle reason is that movement can help the localization performance. In classical localization algorithms, we use motion in the so-called “prediction” step: Given our current position estimate and a movement measurement, we predict the next position. This prediction is refined once we get new measurements (more on this in Section 2.3). We successfully integrate motion estimates in a similar way in Chapter 3. In the subsequent parts of the thesis, we explore more novel ways of integrating motion, in particular for solving problems that are not solvable in a static setting:

- **Chapter 4:** We exploit motion and recover a trajectory estimate rather than position estimates, thus solving a problem that is otherwise not analytically tractable.

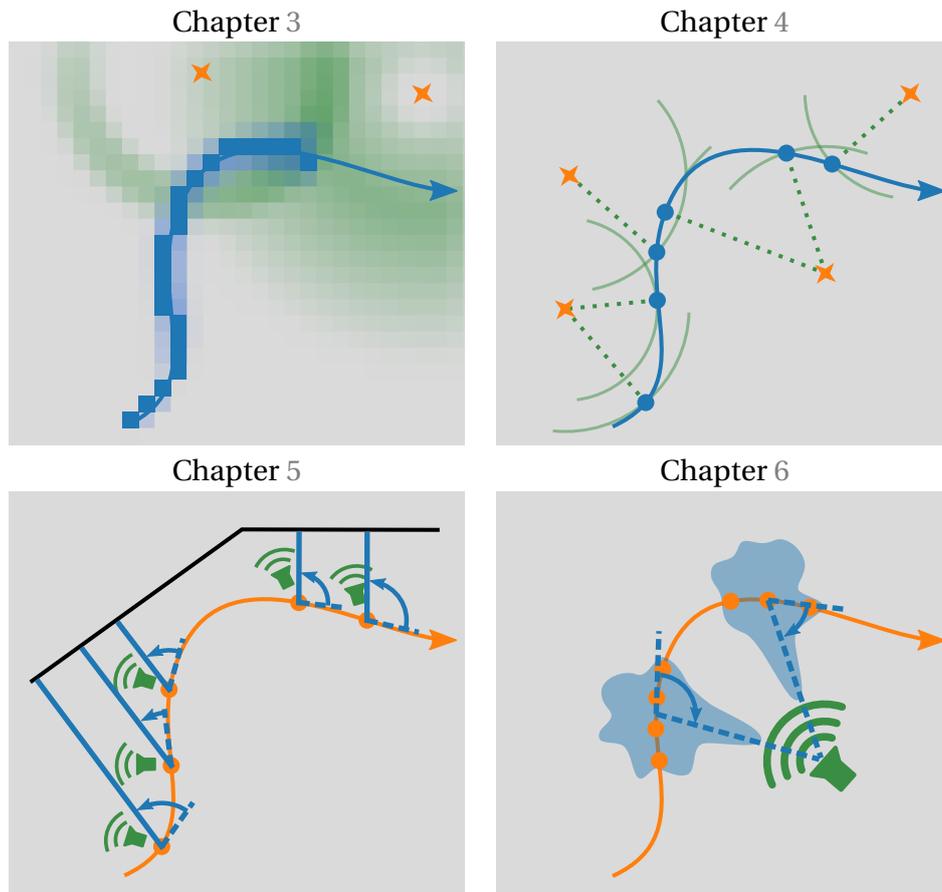


Figure 1.1.: Overview of the first part of this thesis. We consider a moving device which performs spatial perception using mostly blind modalities. We mark in green the measurements, in orange what is assumed to be (approximately) known, and in blue what is estimated. In the first two chapters, Chapter 3 and 4, we localize the device from multimodal measurements and range-only measurements, respectively. We use a discretized state representation in Chapter 3, and a continuous representation in Chapter 4. In Chapter 5, a bat-like drone estimates the location of the closest wall from audio measurements. Finally, in Chapter 6, a moving device estimates the direction of a static sound source.

- Chapter 5: We use the motion of the drone to measure echo interference patterns spatially, allowing us to detect and localize walls.
- Chapter 6: We use the motion of the drone to virtually multiplex the microphone array, thus achieving more effective beamforming and higher accuracy in source localization.

1.1.1. Overview of the first part of the thesis

Figure 1.1 visualizes the different settings studied in the first part of the thesis. In the following, we explain their connections and background. A brief overview of the contributions of the thesis is given in Section 1.1.3.

1. Overview of this thesis

We start the thesis by treating an important application of “blind” localization: indoor localization. Using modalities that are available on a smartphone, in particular Bluetooth, WiFi and the inertial measurement unit (IMU), we design a localization algorithm which combines all measurements. The algorithm was successfully tested in a university hall and deployed in our industrial partner’s software. This chapter inspired the two following research questions, which we address in the remainder of the first part of the thesis:

1. We get IMU measurements at a very high rate compared to the more sparse distance measurements from WiFi and Bluetooth. Furthermore, the quality of distance measurements decays a lot over distance, in particular in non-line-of-sight (NLOS) conditions. As a consequence, we often only have few reliable distance measurements at a time, and we need to bridge this idle time by integrating IMU measurements in the meantime. This setup inspires an interesting inverse problem, which although quite elementary, is not exactly solved to date: Given a moving device which can only measure one distance at a time, can we uniquely recover its trajectory?
2. The modalities used are very noisy: Bluetooth measurements only give reliable estimates of distance when the anchor is within a few meters proximity; and both Bluetooth and WiFi measurements are sensitive to NLOS conditions. Sound, on the other hand, can also be measured on every smartphone (that’s what phones were designed for in the first place) and has both a wide range and great NLOS behavior. Could sound thus be used as a complementary modality for localization?

In Chapter 4, we investigate the first question, proposing a range-only localization algorithm which is guaranteed to find a unique solution under conditions that we characterize exactly. This is visualized in the top-right of Figure 1.1. Chapters 5 and 6, shown in the bottom row, investigate the sound modality for spatial perception. First, we investigate how to localize walls by sampling the interference pattern created by the echos from walls in Chapter 5. Then, we investigate how to localize a static sound source from a moving device in Chapter 6. In both chapters, our emphasis is on creating solutions that are adapted for deployment on a micro-drone — a platform characterized by low computational power and strict weight requirements. Together, these two methods help build foundations for using sound in localization of moving devices.

We can also view the four chapters as treating localization with different kinds of “anchors”, which we define as fixed landmarks of which we can get measurements. In the first two chapters, we treat measurements from anchors emitting radio waves (Bluetooth, WiFi and UWB), where we have different levels of information about the anchors. In Chapter 3, the anchors have known positions but are not calibrated — we use sparse visual information to calibrate them online. In the future work section we pinpoint ways to relax the required prior knowledge about anchor positions through learning-based solutions. In Chapter 4, the anchors are known and calibrated, and we can directly retrieve distance measurements. In Chapter 5 we replace point anchors by walls, to which we can measure the distance and angles using sound. Finally, in Chapter 6, the role of the anchor is taken by a sound source. We do not know its absolute position, but the drone can estimate its angle in local coordinates.

Table 1.2.: Overview of research areas related to spatial perception, using either visual or auditory information

<i>Problem</i>	Visual	Audio
<i>Imaging</i>	RGB / multi-spectral cameras	Acoustic cameras [192]
<i>Full representation</i>	Plenoptic function [148]	Plenacoustic function [2]
<i>Event-based sensing</i>	Event-based vision [72]	Event-based audio [41, 223]
<i>Sparse imaging</i>	Locating stars (radiowaves) [158]	Locating speakers [157]
<i>Remote sensing</i>	Seeing through haze [57] and around a corner [62, 217]	trivial*
<i>Sensor-based motion</i>	Visual servo [76, 134, 222]	Aural servo [132]
<i>SLAM</i>	Visual SLAM [35]	EchoSLAM [109, 110]
<i>Interference sensing</i>	Lippmann imaging [10]	Chapter 5
<i>Motion-aided sensing</i>	Burst processing [216]	Chapters 6 and 5

* Haze and other particles are obviously no problem for sound. “Listening around a corner” is trivial, creating an accurate spatial acoustic image from behind a corner is significantly harder.

1.1.2. Audio, the undervalued modality for spatial perception

We finish this introduction by taking a small detour to further motivate our interest in the audio modality. Despite audio being clearly a useful tool for spatial perception, as demonstrated by many examples from the animal kingdom, the available algorithms for audio signals are limited compared to those for visual signals. We provide an overview of problems and their respective solutions using visual or blind modalities in Table 1.2. The work in this thesis aims to fill certain gaps in audio-based algorithms compared to their visual-based counterparts. Before explaining the contributions in more detail (mostly the last two lines of Table 1.2), we briefly discuss some of the other areas and prevalent solution approaches. We also use this opportunity to highlight some of the work done by the author that are not included in the thesis for the sake of a consistent story.

The most common application of visual signals is **imaging**: capturing the appearance of a scene with a pixel array (usually RGB, but multispectral versions exist), similar to the way we see the world. The equivalent for audio signals is implemented in acoustic cameras. Here, microphone arrays are used to decompose incident sound into its different directions and spectral components, creating multi-spectral polar images of the sound intensity.

These images capture the auditory or visual scene from exactly one given viewpoint. Extensions of this method to capture a more **complete representation** of a scene have been studied for both modalities. The plenoptic function, for instance, characterizes light rays in space, and devices such as the Lytro lightfield camera were built to measure this information.

1. Overview of this thesis

One interpretation of lightfield images is that they capture the same scene from a variety of closely spaced view points; like from a tiny camera array.¹ To the best of our knowledge, the plenoptic function in sound, the “plenacoustic function”, has been mostly studied from a theoretical perspective to this date [2], and has not yet found commercial applications.

In certain settings, these frame-based imaging methods are suboptimal; for instance, when we are interested in recording the dynamic components of a scene at very high rate and resolution. In this case, the memory and processing overhead from dense camera frames, consisting of a mostly static background, is a nuisance. Here, **event-based** cameras offer a much more targeted solution, providing the coordinates of changed pixels over time rather than the full frames [72]. This is now a mature technology with first successful commercial applications. Event-based audio has only recently started to be studied [223]: rather than sampling microphones continuously, we trigger an event as soon as the sound level exceeds a certain threshold. With more progress on optimally leveraging this kind of information, for instance the development of appropriate neural networks [41], we could for instance lower energy consumption during idle times of smart speakers.

Because of its high wavelengths, sound is not susceptible to scattering from small particles in the air, and it can bend around obstacles, making it robust to **NLOS** conditions. Vision has much smaller wavelengths and is thus more susceptible to scattering. Therefore, methods exploiting near-infrared information, which has slightly higher wavelengths and can also be captured by conventional cameras, can sharpen images in hazy conditions, as shown in our work [57]. Quite impressively, advances have been made recently in visual NLOS sensing: single-photon-sensitive cameras have been employed to “see around corners” in [62, 217] but such approaches remain experimental, for the time being requiring sophisticated hardware. Sound, on the other hand, can naturally “see around corners”.

To map an environment, visual simultaneous localization and mapping (**SLAM**) is most commonly implemented, using either cameras or laser sensors [35]. Its counterpart for audio signals has also been studied in recent works [109, 110]. These proposed methods promise reliable online mapping of room geometry, but rely on sophisticated sensors and actuators. Other works even “guess” geometrical appearance and geometrical details from sound [44], using learning-based solutions. As we will see in this thesis, interference sensing is a great alternative when high-end hardware, or sufficient training data, are not available. Furthermore, the results from our audio-based sensing algorithms could be readily integrated in the above **SLAM** frameworks.

The main target of this thesis are the two last application domains.

Interference sensing Exploiting interference has a long history in the visual world. Classical tools such as the Michelson interferometer have driven important discoveries in science since its invention in the late 19th century — most recently a version of it enabled the first-ever detection of gravitational waves [1]. Interference is also the enabling factor of

¹Our work on view synthesis [59] performed at Disney Research, uses for instance a one-dimensional 16-camera array and deep learning to synthesize the lightfield along a line, thus emulating a camera sweep. Similar methods are being employed in the three-dimensional replay of highlights of sports events.

holography, which can be used to generate 3D images, and it is exploited in the recently developed ptychographic x-ray tomography [94]. Finally, interference can also be used to create hyper-spectral cameras; a famous early example of this is Lippmann photography [10]. In acoustics, sound waves also interfere with each other, a phenomenon widely used in noise-cancelling headphones, for example. Interference also happens when we record a sound source next to a wall, between the direct path and the reflect path. Because of the higher wavelength of auditive compared to visual signals, the interference pattern is in the range of centimeters rather than the micrometers. As a consequence, a robot can measure this interference not only along frequency, but also along the spatial dimension, which we exploit in Chapter 5 to localize a wall.

Motion-aided sensing Most applications considered in Table 1.2 involve *static* cameras or microphones. In our treatments, we study *moving* microphones instead. Movement is of course tightly coupled to localization in SLAM, but there, it is typically used in the back-end, *i.e.* for higher-level localization algorithms, and not in the front-end, *i.e.* to extract meaning from raw measurements. However, most microphones and cameras used nowadays — for example the ones in smartphones — are mobile during each capture. The fact that the phone moves during each hand-held image capture is used in modern computational photography algorithms to perform denoising, super-resolution, and high-dynamic-range imaging [144, 216].² In audio-based sensing, to the best of our knowledge, no such algorithms exist. This is the topic of Chapter 6. We investigate how the movement of a microphone array can be used to emulate a virtual microphone array with many more microphones, increasing the performance in determining the direction of sound sources.

1.1.3. List of contributions of the first part of the thesis

In this section, we give a brief summary of the contributions of the first part of the thesis.

Chapter 3: Multi-modal indoor localization In this chapter, we propose a flexible framework incorporating different types of measurement modalities in a discrete-space probabilistic localization framework suitable for indoor localization.

Summary of contributions in Chapter 3

- Extension of conditional random fields to integrate model-based measurements.
- Method for optional online calibration using sparse visual measurements.
- Demonstration on real datasets of indoor localization.
- Preliminary results of learning-based solutions relaxing the knowledge about anchors.

²In our patent [142], we also explore how to use such micro-movements to recover a depth image.

Chapter 4: Dynamical range-based localization In this chapter, we study trajectory recovery using one distance measurement at a time.

Summary of contributions in Chapter 4

- Linear algorithm for global trajectory recovery.
- Sufficient conditions for a unique solution.
- Demonstration on real dataset of lawnmower.
- Outline of spline-based algorithm for piecewise recovery.

Chapter 5: Dynamical sound-based wall localization Inspired by bats, we create a drone that can localize walls using sound. We develop algorithms adapted to this setup based on interference sensing.

Summary of contributions in Chapter 5

- Reproducible audio deck for Crazyflie drone.
- ROS framework to collect and process audio measurements.
- Wall distance and angle localization algorithm.
- Recovery guarantees based on sampling theory.
- Performance evaluation in simulation and on real drone.

Chapter 6: Dynamical sound source localization In this work-in-progress chapter, we are interested in the recovery of the direction of sound sources on a moving device.

Summary of contributions in Chapter 6

- Study of fundamental resolution limits in classical beamforming.
- Formulation of sound source localization algorithm exploiting motion.
- Performance results in simulation and comparison with static and filtering methods.

1.2. Embedding based on angle and distance measurements

Up to this point, we know that this thesis will cover a myriad of applications where a moving device performs spatial perception from different “blind” measurement types. In this second part of the thesis, we take a step up in abstraction and ask questions of the following form:

(general embedding problem) Given similarity measurements between a number of entities, recover a low-dimensional embedding of the entities such that similar ones are close and dissimilar ones are far.

Before dissecting this general problem statement, we explain it with an intuitive example. Assume that you are at a birthday party of a distant friend. While waiting for your turn at the dessert buffet, you look around the room and try to figure out who is who. There must be some family, mostly friends, and maybe some uninvited strangers who came for the magnificent chocolate muffins? There is a girl who has the same accent as your friend and is roughly the same age — maybe she is a school friend from a similar area? There is a man who seems to be from the older generation and who has brought an amazing cake — maybe it’s the dad? And then there is this guy who looks a little nervous and has chocolate on his fingers — maybe a chocolate predator? What you performed in your head is exactly an instance of this general embedding problem. You observe similarity features and you try to put people into clusters based on these features.

Let us now discuss the problem in more technical terms. By similarity measurements, we mean a measure of “closeness” of a given set of entities. The **kind of measure** can depend on the **nature of the entities**. In the field of medical imaging for instance, the entities can be images of a retina, of which we seek an embedding such that the images of the healthy patients are clustered [85, 202]. Here, similarity could be a perceptual metric between two images such as the structural similarity index measure (SSIM) [212]. In text-based information retrieval, entities could be words, whose embedding should be close when words are semantically related (*e.g.* the words “tiger” would be closer to “lion” than for instance “chair”) [206]. In this case, the similarity could be measured as the number of word co-occurrences in training documents. Compared to these two examples, localization is more trivial: in this case, entities are points that we seek to localize, and the embedding is simply an estimate of the point location itself.

The measure can **involve different numbers** of entities: for instance, distance measurements typically involve two entities and angle measurements involve three entities. We can also compute the correlation between two pairs of entities, in which case 4 entities are involved.

Measurements can be **labeled or unlabeled**, meaning that we may or may not know which points are involved for each measurement. If they are unlabeled, we first need to perform what is known as the labelling problem in signal processing, or as data association in robotics: determining which measurement corresponds to which entities.

The **number of points** to embed can also vary a lot depending on the application: the applications seen in the first part of this thesis, for instance, all involve less than 10-15 points, but applications in information retrieval from large text bases can involve thousands of points.

1. Overview of this thesis

Note that we are localizing all points jointly, so we relax the idea of anchors used in the previous part. As we will see later, anchors can be easily incorporated through Procrustes analysis [186].

Finally, **finding the embedding** seems like a fancy wording for “localizing” points, *i.e.* finding their coordinates in Euclidean space. The terminology “low-dimensional embedding” is however more general than that, as it also encompasses manifolds or other non-Euclidean spaces. We cover some alternatives in Section 7.4. The emphasis on lower dimensions is obvious for the application in localization, which happens in two or three dimensions. In general, low-dimensional embeddings are desired because they allow for straight-forward visualizations and discovering important structure of the data.

Posing the problem in this form rather than the more specific forms encountered in the first part of the thesis creates a number of interesting opportunities:

- Relaxing the anchors requirement allows the presented theory to be applied for a larger set of problems in robotics where no such infrastructure or information is available. Such applications include robotic swarms, but also SLAM with unknown landmarks or ad-hoc sensor networks employed for instance for environmental monitoring.
- Even when anchor information is available, it can be beneficial to treat each anchor as just another unknown point. In particular, this is useful when the anchors have unreliable calibration, leading for instance to bias on measurements, or when their position is only roughly known. Treating anchors as points with unknown location increases tolerance for such errors.
- As we will see later, many interesting theoretical results exist for problems of this form. Furthermore, efficient algorithms have been developed for such problems, some of which have closed-form solutions and recovery guarantees. These results are very useful for robotics or localization problems.
- As mentioned above, using this more general problem statement opens up a whole area of applications outside of robotics. Finding lower-dimensional embeddings from similarity measurements is interesting for data analysis in general, with applications ranging from biology, over psychometrics, and to text-based information retrieval.

In this thesis, we focus on specific instances of the general embedding problem. In particular, for our similarity measurements, we consider distances, inner angles, and absolute angles. Although these measurements are arguably the most common ones for localization, more unconventional measurements, in particular non-metric measurements, can be an interesting alternative. Some alternatives are briefly discussed in Section 7.4.

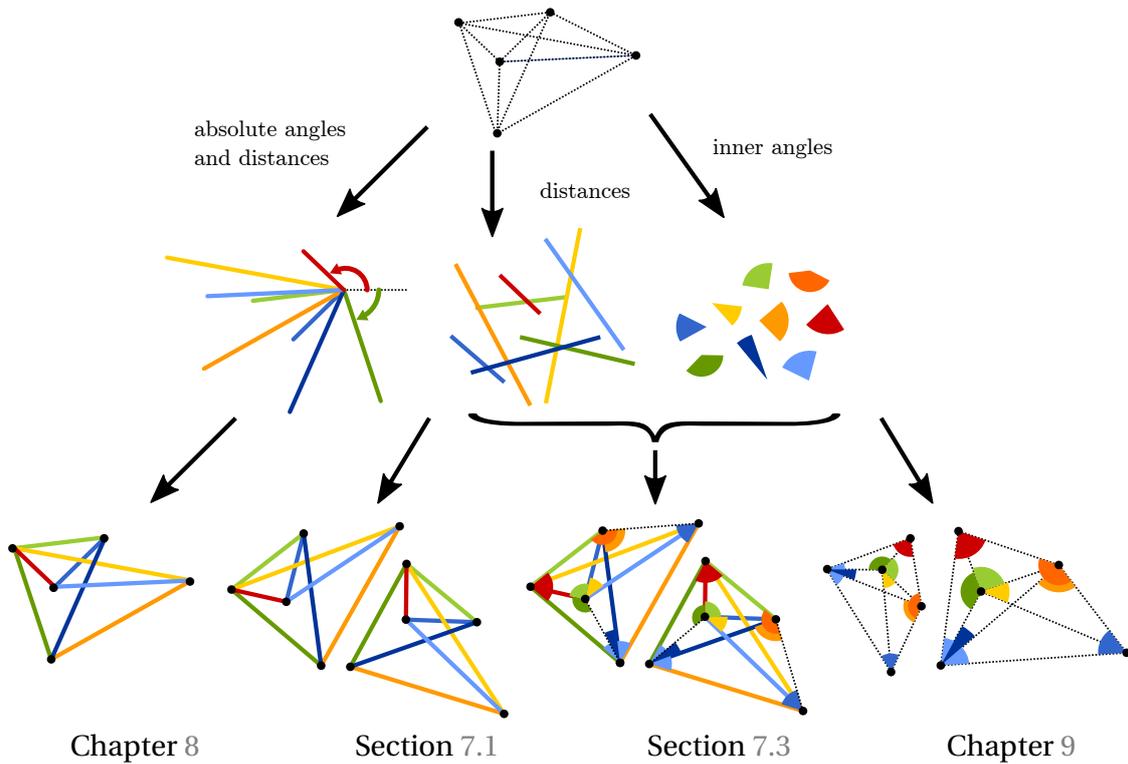


Figure 1.2.: Overview of the second part of the thesis. There is a set of entities with an unknown low-dimensional embedding (top row). We seek to recover this embedding, but we are given only certain similarity measurements. Using these measurements, we can recover an estimate of the low-dimensional embedding (last row). We order the studied cases by decreasing information from left to right. In the last row, we show different embeddings which yield the same measurements. Note that in all cases, we lose the absolute translation of the original embedding (not shown). With distance and inner-angle measurements, we also lose the orientation, and for only angle measurements, we lose orientation and scale.

1.2.1. Overview of the second part of the thesis

Figure 1.2 shows an overview of the problems discussed in the second part of the thesis. We start by discussing distance-based embedding, which has by far obtained the most attention in the literature, in the introductory Chapter 7. We summarize the results from distance-based embedding in Section 7.1, formalize the different angle measurements in this thesis, and present a method to include inner-angle measurements in Section 7.3. After these introductory topics, we move over to our main contributions.

First, in Chapter 8, we combine distances with absolute angles. Intuitively, this problem is simpler than distance-only or angle-only localization: angles and distances can be combined to vector measurements, and point recovery is less ambiguous than for the other cases. Mathematically, the problem is however more interesting than it seems at first sight. Concretely, we can split the problem into one-dimensional ones, where we can draw an interesting parallel to distance-only localization. Furthermore, the solution can be generalized to any problem where we measure projections to arbitrary directions, examples of which are geometrical microphone array calibration and structure from motion.

1. Overview of this thesis

Finally, we consider localization from inner-angle measurements only. Using again parallels with distance geometry, we first establish realizability conditions on angles. These conditions can be exploited for various tasks; we showcase their utility by creating a constrained optimization problem for denoising measurements. Using the denoised angles, we can recover a unique point set, using a simple build-up algorithm. Finally, we compare, in simulation, the angle-only embedding method with a classical distance-only embedding, highlighting in which noise regimes one should be preferred over the other.

1.2.2. List of contributions of the second part of the thesis

Chapter 7: Background of the second part This is an introductory chapter, but it does cover one algorithmic contribution, the “constrained edge-kernel method”. This method improves over the original one [131] by ensuring realizability of the recovered edge vectors, thus increasing localization accuracy.

Chapter 8: Embedding based on coordinate differences In this chapter, we treat localization from distance and absolute angle measurements. Phrasing the problem as recovery from coordinate differences, we make the following contributions:

Summary of contributions in Chapter 8

- Presentation and study of coordinate difference matrices, including their exact characterization and recovery algorithm in one dimension.
- Generalization to multiple dimensions and conditions for equivalence between global solution and more efficient frame-wise (“splitting”) solution.
- Demonstration of performance on sensor network localization and geometric microphone calibration.

Chapter 9: Embedding based on inner angles In the final chapter, we study localization from inner-angle measurements only. Inspired by distance geometry, we split the problem in two parts: first, we find the set of realizable angles closest to given, noisy angles. Then, we recover the points based on the realizable angles.

Summary of contributions in Chapter 9

- Necessary conditions in the form of linear and non-linear constraints for realizable angles, including method to automatically construct the linear constraints.
- Conjecture on sufficient conditions, verified in simulation, and sketch of proof.
- Localization results in simulation and comparison with distance-only method.

Part I.

**Blind spatial perception for moving
devices**

2 | Background of the first part

In this introductory chapter, we lay out the foundations for spatial perception, in particular localization and mapping, treated in this first part of the thesis.

2.1. Localization: notation and definitions

Out of the many facets of spatial perception, we are mostly interested in localization in this thesis. We study the localization of a moving device in the first two chapters, and the localization of external objects (also called mapping) in the last two chapters.

Basics Localization describes the estimation of the pose of a robot, given certain measurements. The pose of a device, consisting of its position and orientation, is described by a function of time t : $(\mathbf{r}(t), \boldsymbol{\alpha}(t)) \in (\mathbb{R}^D \times SO(D))$, with $SO(D)$ the special orthogonal group, consisting of rotations in D dimensions. When convenient, we represent angles by the corresponding rotation matrix $\mathbf{R}(t) \in SO(D)$, which is a D by D matrix. Extensive literature exists on correctly handling rotations in state estimation and in particular localization [14, 25]. In the applications considered in this thesis, the rotational part of the pose can be restricted to the plane, so we only consider rotations in $D = 2$, where they are described by one angle only, which we denote by $\alpha(t)$. Furthermore, we introduce $\mathbf{r}_n := \mathbf{r}(t_n)$, the position sampled at time t_n , as well as $\mathbf{r}_{1:n} := \{\mathbf{r}_1, \dots, \mathbf{r}_n\}$ to denote a series of positions. We use the same indexing convention for all variables.

Measurements We distinguish between two types of measurements. Map-based, or exteroceptive, measurements, denoted by $\mathbf{z}(t)$, include measurements taken with respect to external and fixed points, such as access points or walls (see next paragraph). On the other hand, we have “internal”, or proprioceptive, measurements $\mathbf{u}(t)$: they describe the robot’s movement using some sensors such as step counters or IMUs. From these measurements, we can derive the travelled distance $l(t) \in \mathbb{R}$ and rotation $\boldsymbol{\beta}(t) \in SO(D)$, with respect to some reference point. In this thesis, we consider mostly distance and angle measurements to map elements, and we measure relative movement using IMU. When convenient, we include the internal measurements in the map-based measurements, in which case $\mathbf{z}(t)$ includes all available measurements.

2. Background of the first part

Notation for this part	
$D \in \{2, 3\}$	dimension of space
$\mathbf{r}(t) \in \mathbb{R}^D$	device trajectory
$\boldsymbol{\alpha}(t) \in SO(D)$	device orientation
$\mathbf{x}(t)$	a continuous function...
$\mathbf{x}_n := \mathbf{x}(t_n)$... its sample at time t_n ...
$\mathbf{x}_{1:n} := \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$... and a set of samples
$\mathbf{z}(t) \in \mathbb{R}^N$	map-based measurements
$\mathbf{u}(t) \in \mathbb{R}^D \times SO(D)$	movement measurements
\mathcal{M}	map
$\mathbf{a}_m \in \mathbb{R}^D$	m -th anchor position
Z, Y	random variables for measurements and state
z, y	realizations of Z and Y

Table 2.1.: Different “map” representations used throughout this thesis, defined as collections of map elements and their respective parameters.

	Map elements	Parameters
<i>Chapter 3</i>	Bluetooth beacons, WiFi access points, visual anchors / floor plan	coordinates, gain, bias / state connectivity
<i>Chapter 4</i>	UWB access points	coordinates
<i>Chapter 5</i>	wall(s)	wall normal(s), reflection factor(s)
<i>Chapter 6</i>	sound source(s)	coordinate(s), signal(s)

Map representations In general, we can describe the map of an environment as a collection \mathcal{M} of M objects. In feature-based maps, each element of the map contains some properties and coordinates. This kind of representation, consisting of only as many map elements as interesting features in the environment, is also called a landmark-based representation of space. In location-based maps, on the other hand, we typically discretize the space and assign properties to each location, such as occupancy or traversability. This is also called a dense representation of space. We use mostly landmark-based representations in this thesis: in particular, the map consists of the locations of anchor points (WiFi, Bluetooth, UWB, visual markers), the angles of sound sources, or the location and reflection factors of walls, as summarized in Table 2.1. In Chapter 3, we also consider a dense representation, namely floor plans, in indoor localization settings.

For the probabilistic treatments, we introduce the random variable Y to describe the full state that we wish to infer. The state, one realization of Y , is denoted by y . Note that this state can include, depending on the application, either only the location, the pose, the map, or any combination of these quantities. Similarly, the random variable describing extrinsic measurements is given by Z , and intrinsic measurements by U . When appropriate, we combine all measurement types in the variable Z . For ease of notation, we write the probability of Y taking value y as $p(y) := p(Y = y)$, and ensure that it is clear from the context which random variable is involved.

2.2. Landscape of solutions

Different applications impose different constraints on available measurements, allowed computational load and required accuracy/latency. Therefore, solutions to the localization and mapping problems come in many different flavors. For example, when it comes to recovering a trajectory $r(t)$, one can choose among the following approaches.

discrete vs. continuous recovery We can either try to recover a description of the continuous trajectory, $r(t)$, or a list of trajectory points $r_n, n = 1 \dots N$. The more natural approach is discrete recovery, since we get measurements at discrete times. Originally, trajectories are however continuous, and since they describe the movement of an object with an inherent inertia, they exhibit some degree of smoothness. As we show in Chapter 4, we can exploit smoothness and recover a continuous trajectory, in particular when recovering single positions is an underdetermined problem.

probabilistic vs. deterministic recovery In the real world, measurements are noisy and systems are often underdetermined. For robust solutions, it is therefore more sensible to recover a probabilistic form, $p(r(t)), p(r_n)$ rather than $r(t), r_n$. This is the predominant approach in the first part of this part of the thesis. Even for our deterministic recovery algorithm in Chapter 4, we provide the (a-posteriori) probabilistic description.

filtering vs. batch estimation As the device moves and more measurements come in, we can either choose to only consider the most recent measurements for localization, or we can try to find the optimal solution given all measurements. In what the authors of [35] call the age of “classical localization”, the former approach, which is also called filtering, was more common: finding for example the most likely pose given the previous pose and most recent measurement. As robots have increasingly more computational power, and thanks to great algorithmic improvements, batch-based estimation has received more attention recently [102, 103], which takes into account all measurements and optimizes over all poses jointly. Both filtering and batch-based approaches are covered in more detail in the next section. In Chapters 3 and 4, we consider batch-based solutions, while the estimates from Chapters 5 and 6 could be integrated into either framework. When applied to SLAM, the solutions considered in the second part of this thesis could also be considered as batch methods, as they seek solutions consistent with all distance or angle measurements seen so far.

generative vs. discriminative models This distinction is rather technical and will be covered in more detail in Section 2.3. Put briefly, generative models use forward models to predict measurements from the current state, which are then compared to the actual measurements to infer the likelihood of the states. Discriminative models do not require a forward model since they exploit more general cost functions between the measurements

2. Background of the first part

and the states. Discriminative models are used more commonly in natural language processing and computer vision than in robotics; we investigate their potential for localization in Chapter 3. In general, in this thesis, we assume measurement models to be known (or we seek to approximate them) and we are thus more inclined towards the generative perspective.

In the following, we describe generative and discriminative models in more detail, since they are important building blocks of the material covered in this thesis.

2.3. Probabilistic robotics in a nutshell

2.3.1. Generative perspective

The goal in Bayesian localization is to find the probability over the poses y over time, given measurements z and u ; in other words, we are looking for $p(y_{1:n}|z_{1:n}, u_{1:n}, y_0)$, also called the posterior over the state space. Under certain assumptions such as the Markov property and conditional independence (see [201] for an in-depth treatment), this probability distribution can be factored as:

$$p(y_{1:n}|y_0, z_{1:n}, u_{1:n}) = \eta p(y_0) \prod_n p(y_n|y_{n-1}, u_n) p(z_n|y_n), \quad (2.1)$$

where η is a normalization factor to satisfy the axiom of total probability. Inspecting (2.1), we can identify the three main ingredients of typical Bayesian algorithms. The **prior**, $p(y_0)$, is the initial estimate of the state distribution. The movement model $p(y_n|y_{n-1}, u_n)$ tells us how to **predict** the next state given the current state and a measurement of the movement u_n .¹ Finally $p(z_n|y_n)$ is the measurement model, it tells us how to **correct** our estimate, given the latest measurements. Combined, (2.1) tells us how to calculate the **posterior**.

Evaluating (2.1) is also known as the *full SLAM* solution and it leads to batch-based methods using all available measurements. Unless probabilities are assumed Gaussian and measurements are assumed linear [102, 103], it can quickly become prohibitively expensive to perform in real time. A popular simplified version is *online SLAM*, or Bayesian filtering, where we only update the most recent state estimate, given the latest measurements [201, p.310]. A great overview of the connections between full and online SLAM is given in [14]. In this case, we use the recursive update rule:

$$p(y_n|y_0, z_{1:n}, u_{1:n}) = \eta p(z_n|y_n) \int p(y_{n-1}|y_0, z_{1:n-1}, u_{1:n-1}) p(y_n|u_n, y_{n-1}) dy_{n-1}. \quad (2.2)$$

Equation (2.2) tells us how to compute the posterior $p(y_n|y_0, z_{1:n}, u_{1:n})$, given the previous posterior $p(y_{n-1}|y_0, z_{1:n-1}, u_{1:n-1})$, the map-based measurements z_n and motion estimates u_n . The measurements are incorporated through the forward model $p(z_n|y_n)$ and the mo-

¹This measurement u_n can either take the form of movement estimates, e.g. from odometry or IMU readings; or it can be induced from movement commands.

tion rule $p(y_n|u_n, y_{n-1})$. In practice, (2.2) is often solved in two steps: first, we calculate the integral (prediction), and then we update the distribution corresponding to the measurements (update). Both cases typically require the forward and motion models, which we formalize as

$$z_n = g(y_n) + \epsilon_g, \quad y_n = h(y_{n-1}, u_n) + \epsilon_h, \quad (2.3)$$

where $g(\cdot)$ is the measurement forward model and $h(\cdot)$ is the motion model, and ϵ_g, ϵ_h are additive noise. Since these models allow to generate measurements given the current state, the reviewed algorithms are also called “generative” models.

Different representations of $p(y_n)$ yield different well-known filtering schemes. With the help of the toy example depicted in Figure 2.1 a), where the state consists only of the device position, we describe the possible approaches. First, assuming Gaussian distributions and linear measurement and movement models g and h , we obtain the famous Kalman filter for the online SLAM approach, shown in Figure 2.1 b). If the models are non-linear, the same method can be applied after linearization through Taylor expansion, yielding the extended Kalman filter (EKF). For the full SLAM approach, efficient methods relying on the Gaussian assumption and linearization have been proposed as well [102, 103].

The Gaussian and linear assumptions are somewhat restrictive (more about this in 2.3.3). Therefore, a whole family of methods exist which approximate the distribution in a non-parametric way; using either histograms or a finite set of samples. In the latter, a finite set of particles are drawn from the prior distribution and evolved according to the prediction and update rules. This approach is shown in Figure 2.1 c). Histogram filters typically discretize the state space on a grid and assign to each grid region a single probability, Figure 2.1 d). This yields a piecewise constant approximation of the original distribution. The commonly used Hidden Markov Model (HMM) can be used on this category of methods, although in its first-order form it assumes constant transition probability. To account for different transition probabilities, the model can be extended to a second-order HMM, where each state consists of two poses.

Considerably less explored are methods which — rather than finding “snapshots” of the probability distributions at given times — try to infer a parametric description of the continuous trajectory $r(t)$, as depicted in Figure 2.1 e). Initially, solutions based on spline parametrizations were explored [69] but Gaussian processes have recently gained more popularity thanks to their flexibility and inherent estimation of uncertainty [203].

Finally, the field has recently seen learning-based methods obtain very interesting results. Conceptually speaking, we can, for instance, train a neural network to output a state estimate given the previous estimate and new measurements, as shown in Figure 2.1 f). The type of representation of $r(t)$ can vary depending on the chosen architecture type: in the simplest case, one can infer deterministic estimates r_n from measurements, as we discuss in Section 3.6.1. This is only one approach of a fast-growing field of learning-based solutions, which tackle different aspects of localization and navigation.

A comprehensive treatment of this research is out of the scope of this thesis, instead we give a brief overview. For this, it is useful to split the spatial perception problem into a front end — converting sensory information into usable forms — and a back end — solving for the

2. Background of the first part

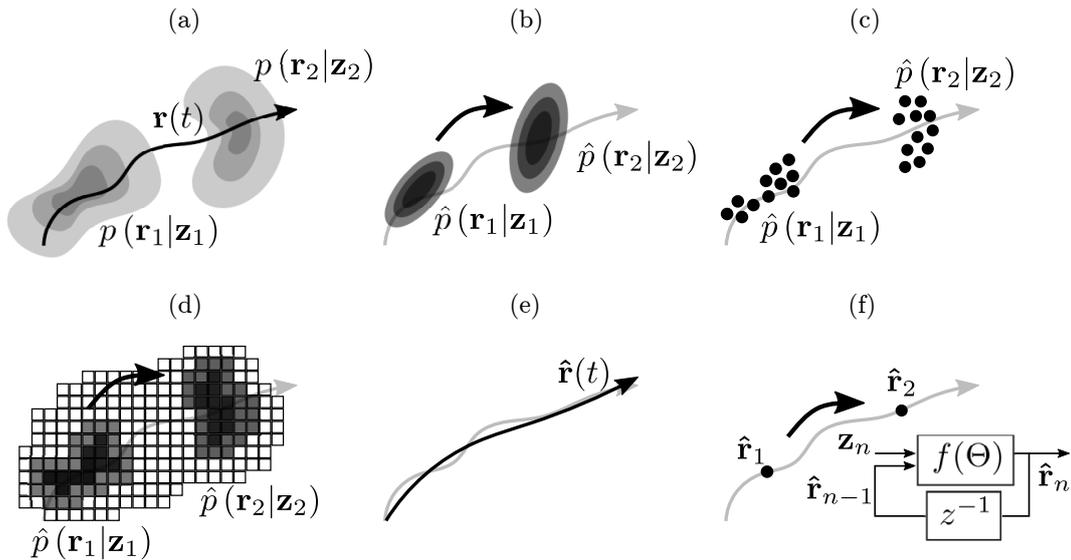


Figure 2.1.: Overview of different representations of the trajectory distribution. a) shows the example posterior state distributions at two subsequent times. They can be approximated using b) Gaussians in a Kalman-filter approach, c) finite points in a particle-filter approach, and d) a finite grid. Finally, e) depicts a parametric approach and f) shows an example of a learning-based approach

state given these measurements [35]. For the front end, notable methods include feature learning from lidar [226] or radar [34] measurements, extraction of gate poses for drone racing [104], or compact scene representations from vision [24]. Also worth mentioning are semantic methods, assigning labels to objects in the environment [181]. For methods including the back end, end-to-end navigation solutions have been proposed, going directly from sensory data to motion commands. Such solutions have recently enabled autonomous drone acrobatics [105] and vision-based flight [185].

2.3.2. Discriminative perspective

In Section 2.3.1 we discussed methods to find the posterior distribution $p(y|z)$ by expressing it as a combination of the joint distribution $p(y, z)$, and the generative model $p(z|y)$ (note that for convenience we now include proprioceptive and exteroceptive measurements in z).

In some cases, we do not know the generative models for $p(z|y)$ or we do not want to estimate it. In fact, if we are only interested in finding the most likely state, it might seem like a detour to find the generative model for each candidate state, and then comparing the output with the given measurements. Instead, we could simply learn the decision boundaries between measurements — allowing us to relate measurements to the most likely states — rather than being able to generate these measurements ourselves.

Discriminative models have been designed with exactly this purpose in mind. Rather than modelling $p(y|z)$ through the joint distribution, we try to estimate it directly, given some training data. It is common to do this through the Maximum Entropy principle [98]; which

tells us that the best distribution estimate is the one which, while satisfying some constraints learned from training data, has the highest possible entropy. This leads to Maximum Entropy Markov Models (MEMMs) [138]. We can formulate the constraints in terms of feature functions; where each feature function captures some characteristic of the training data. Calling the feature function $f_a(y, z)$, the desired maximum entropy distribution is given by [138]:

$$p(y|z) = \eta \exp \left(\sum_a \lambda_a f_a(y, z) \right). \quad (2.4)$$

One might argue that the generative models are of greater importance in robotics than discriminative models, because it is not always trivial to collect training data², but it is possible to model the measurements. The extension of MEMMs however, called conditional random fields (CRFs) [114], have some interesting properties that standard Bayesian approaches do not have. CRFs are a generalization of MEMMs which, rather than considering one state transition and one measurement at a time, consider all measurements, and compute the joint probability over all labels. Mathematically,

$$p(y_{1:n}|z_{1:n}) = \eta \exp \left(\sum_{e,k} \lambda_k f_k(e, y|_e, z_{1:n}) + \sum_{v,k} \mu_k g_k(v, y|_v, z_{1:n}) \right), \quad (2.5)$$

where v and e are vertices and edges of the underlying graph, with nodes indexed by the possible states, and edges between states that have non-zero transition probabilities. Each feature function thus operates on all measurements and can incorporate more complex dependencies than the generative functions in Bayesian models, which only operate on one state, or the feature functions in MEMMs.

2.3.3. Measurement modalities

In generative models, a crucial component is the forward model for both the measurements and the movement updates, formalized in (2.3). In conditional models, such forward models are not required, but they can be helpful in creating meaningful feature functions f_k and g_k in (2.4), as shown in Chapter 3.

In this section, we want to give some intuition on the consequences of linear, underdetermined and non-linear models. We focus on the measurement model $g(y)$, fixing the state to contain a one-dimensional position only: $y = r$, but the considerations also apply for the movement model and more complex states. We use the visualization in Figure 2.2, whose style we borrowed from [201]. Each sketch shows the transition from a prior state distribution $p(r)$ (x-axis, on the bottom) through a given function $g(r)$, to the measurement distribution $p(z|r)$ (y-axis, on the left). Vice-versa, we can map the measurement distribution through the inverse measurement function to yield back the posterior state distribution

²For instance, researchers might not have access to the robotic vehicles. Even if they do, if a robot is supposed to operate in extreme search and rescue missions, it could be very expensive or dangerous to collect realistic data.

2. Background of the first part

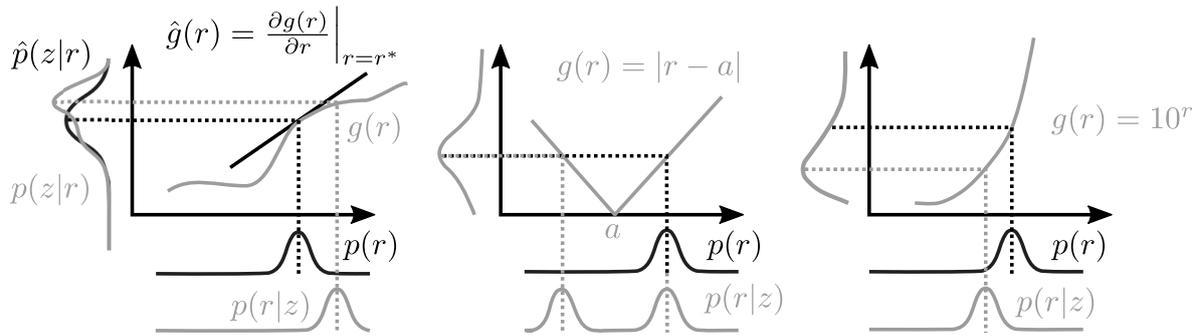


Figure 2.2.: Overview of different measurement modalities: in practice, linear models are often assumed, obtained through linearization around the operating point (left). In this thesis, we do not use this linearization in the study of underdetermined models (middle) and other non-linear functions (right).

$p(r|z)$. The plots have been generated by hand and certain features are exaggerated to highlight important points.

linear models: Linear measurement models maintain Gaussianity. This characteristic is used in the well-known Kalman filter, which simply evolves the mean and covariance of the state by incorporating linear measurement and movement models as measurements come in. The same principle can be used for nonlinear models, in which case we simply linearize around the current operating position using Taylor expansion. As the example in the left-most plot of Figure 2.2 shows, this can yield a very rough approximation of the actual measurement distribution.

underdetermined models: Two of the most common measurement modalities are angles and distances. Taken individually, they are both non-linear, and in particular, they can be underdetermined³. As an example, the second plot of Figure 2.2 shows a distance measurement in one dimension; the absolute value. Mapping back measurements of this form to the state probability distribution results in ambiguities, symmetric around the anchor position a . This result is intuitive: given a distance measurement, we only know that we are somewhere on a sphere, circle or given offset around an anchor, depending on if we look at the problem in three, two or one dimensions.

non-linear models: The previously treated underdetermined models are part of a larger class of models that do not maintain Gaussianity. In fact, all non-linear models do not maintain Gaussianity. As an example, we plot the exponential model in Figure 2.2. Exponential models are relevant when we measure the received signal strength of a signal. Assuming there is Gaussian noise on the received signal strength, the measured distances are log-normally distributed. One important characteristic of the log-normal distribution is that its mode does not coincide with its maximum. Therefore, if we choose the maximum probability distance given the measurement distribution and apply the inverse transform, we do not

³Distances are always underdetermined, as shown in the example here. Angles are only underdetermined if we assume that the directionality is lost; which is the case for instance for the front-back ambiguity in audio (or in general for any microphone array with two microphones only). Otherwise, angle measurements fall in the category of non-linear but determined models.

end up with the maximum probability received signal strength.

Given that even common measurement models are non-linear, and sometimes even under-determined, emphasizes the need for algorithms circumventing linearization, which are the topic of this thesis.

2.4. Outlook

We conclude this section by looking at the bigger picture of the work in this thesis with respect to the field of localization. One common thread throughout this thesis is that we do not impose Gaussian assumptions on the state distribution. Looking at the used measurements — predominantly distances and angles — we now understand why: such measurements are underdetermined and nonlinear (see Section 2.3.3), and do thus do not maintain Gaussianity. Therefore, we use a grid-based state distribution in Chapter 3 and a parametric description in Chapter 4. The same holds for the measurement distributions derived in Chapters 5 and 6: due to the underlying physics, these distributions are highly non-Gaussian, and treating them differently would be a harsh simplification. Instead we obtain discrete versions of the probability distributions (histograms) through Bayesian inference (Chapter 5) and beamforming (Chapter 6), respectively.

A second overarching idea of this thesis is not to use classical linearization and iterative solutions. Instead, we use the Viterbi algorithm in Chapter 3 for discrete localization, and we augment the problem dimension to yield a linear system that can be solved optimally in Chapter 4. Similarly, in the second part of the thesis, we seek global solutions to the localization problem that are inspired by distance geometry, which offers optimal and closed-form solutions. As this chapter demonstrates, there is no one-size-fit-all solution for the localization problem, but with the theory and methods presented in this thesis we hope to contribute novel insights to a wide range of applications.

3 | Multi-modal indoor localization

Es git nichts Gutes, ausser man tut es.
There's nothing good unless you do it.

— German saying, by Erich Kästner

Almost everyone carries a smartphone on them nowadays¹; so being able to localize it at any time and in any environment has important societal and economical consequences. However, despite more than a decade of progress on this front, no technology has achieved widespread use for indoor localization to this day. This can be largely attributed to the fact that existing systems suffer from one or more of the following drawbacks: they

- require additional expensive dedicated infrastructure;
- require significant training/calibration;
- are not robust to changes of the environment;
- require the user to actively scan the environment; or
- are not smartphone-compatible.

¹According to a report published in August 2021, more than 6 billion people use smartphones worldwide, for a total number of 7.9 billion people [150].

This chapter is based on the publication:

F. Dümbgen, C. Oeschger, M. Kolundzija, A. Scholefield, E. Girardin, J. Leuenberger, and S. Ayer. "Multi-Modal Probabilistic Indoor Localization on a Smartphone". In: International Conference on Indoor Positioning and Indoor Navigation. 2019, pp. 1–8

Detailed contributions: AS, MK and FD designed research, FD designed and implemented algorithm in python, designed and evaluated experiments. AS, MK and FD wrote the paper, CO and EG implemented data acquisition algorithms on smartphones and collected preliminary datasets, converted algorithm to C. SA and JL provided the usecase, Augmented Reality software and advice.

The dataset used in this chapter is available at
<https://github.com/duembgen/indoor-localization-dataset>.

3. Multi-modal indoor localization

For example, ultra-wideband (UWB) signals, such as the commercially available *Pozyx*, can provide accurate localization over a wide range of distances with a quick and stable system setup. This modality has been widely applied in robotics [116, 167, 168] but in general, such solutions require dedicated infrastructure and are not smartphone-compatible.²

Other solutions require only a smartphone and no infrastructure. For instance, cameras combined with IMU can be used to scan the environment and, using methods from SLAM, infer an accurate 3D map and the user’s location with centimeter-level precision [214]. However, it is not always convenient for the user to actively scan the surroundings; an ideal indoor localization system would be passive — meaning that it requires no active participation of the user. Most visual systems also require the environment to possess uniquely identifiable visual features, which is not always the case, in particular in exhibition rooms, lecture halls, and sterile environments such as hospitals. Finally, visual systems tend to break down when the environment is changed, causing the recorded maps to not match the reality anymore.

In order to create a passive localization system, a third category of approaches use smartphone-compatible measurements, for example radio-frequency (RF) signals such as Bluetooth low energy (BLE) and WiFi. Since WiFi base stations are already prevalent and BLE beacons are inexpensive, these approaches can be deployed with reasonable infrastructure costs. In indoor environments, RF signals are heavily affected by multi-path, shadowing and fading [101]. Furthermore, synchronization issues, unknown latency times and unknown variations in antenna characteristics induce important measurement errors [174]. Therefore, the best-performing algorithms leveraging such signals use a “fingerprinting” phase, in which the signal’s characteristics are extracted at a finite set of positions in the region of interest. These fingerprints are able to capture stationary non-line-of-sight (NLOS) and multipath effects. In the online localization phase, the recorded signal is matched with the fingerprint database, and—after possible combination with IMU data and map constraints—a position estimate is inferred [84, 191, 218].

In this chapter, we present an indoor localization system that is smartphone compatible, easy to install and robust to environmental changes. The proposed system combines BLE beacons, WiFi hotspots, the smartphone’s IMU data, and visual fingerprints recorded by its camera. The visual information is sparse in both time and space. In particular, we assume that certain visual landmarks—such as artworks in a museum, emergency exit signs, navigation maps, etc.—do not change over a foreseeable time and these landmarks can be “scanned” by the user at a handful of time points. At these moments, we obtain centimeter-level localization, which allows us to calibrate offsets in the other measurements.

For the localization algorithm, we use a conditional random field (CRF) to efficiently combine the different measurement modalities. This allows us to account for the expected relative accuracy of the different systems, and to obtain a distribution rather than a single estimate of the user’s position.

²Note that since we have conducted this research, some high-end smartphones of Apple and Samsung have been equipped with UWB capabilities, to be used for key chain finding, for instance. Such phones might become candidates for UWB-based indoor localization, provided that compatible UWB-based anchors will become available.

Notation for this chapter	
M_B/M_W	number of available Bluetooth / WiFi anchors
$\mathcal{I}_B/\mathcal{I}_W$	active Bluetooth / WiFi anchor indices
$d_m, m \in \mathcal{I}_W$	WiFi distance measurement
$P_m, m \in \mathcal{I}_B$	Bluetooth path loss measurement
$\mathbf{r}_v \in \mathbb{R}^D$	position estimate from visual scanning
$\beta \in [0, 2\pi), l \in \mathbb{R}^+$	latest path direction and length, from IMU measurements
<hr/>	
$f_k(\cdot)$	k -th feature function
K	number of feature functions
$\mathbf{r}[i] \in \mathbb{R}^D$	position of i -th grid point
N_g	number of grid points

In sum, the proposed framework provides robust localization that does not break down when measurements or entire modalities are missing or unreliable. Furthermore, the localization is passive for the vast majority of the time. In terms of accuracy, we achieve mean localization errors of around two meters in a challenging indoor environment, without the need for any fingerprinting or prior calibration.

3.1. Problem setup

Our goal is to find the location of a moving device in a given indoor environment. We assume that there are M_W WiFi access points and M_B Bluetooth beacons of known positions, from which we get distance and path loss measurements, respectively. We index the anchors with $m = 1 \dots M_B + M_W$ and introduce the index sets \mathcal{I}_B and \mathcal{I}_W to denote the *active* anchors, meaning anchors from which a measurement is obtained. From a WiFi anchor $m \in \mathcal{I}_W$ we measure distance d_m , and from Bluetooth beacon $m \in \mathcal{I}_B$ we measure path loss P_m . We combine these measurements in vectors $\mathbf{d}_{\mathcal{I}_W} \in \mathbb{R}^{|\mathcal{I}_W|}$ and $\mathbf{P}_{\mathcal{I}_B} \in \mathbb{R}^{|\mathcal{I}_B|}$. We use the IMU to obtain an estimate of the device's travelled path since the last processing time, described by $\beta \in [0, 2\pi)$ for the direction and $l \in \mathbb{R}^+$ for the path length, respectively. Note that we are only interested in the device's orientation in the x-y plane (its rotation around the z-axis), as we are treating objects moving mostly in a horizontal plane, in particular walking pedestrians. Finally, we sometimes obtain a position estimate $\mathbf{r}_v \in \mathbb{R}^D$ from visual scanning of sparsely distributed landmarks.

Above measurements, collected at a given discrete time instant t_n , are combined into a vector-valued observation variable \mathbf{z}_n .³ To summarize, each measurement vector is thus composed of a subset of:

$$\mathbf{z}_n = \left[\mathbf{d}_{\mathcal{I}_W}^\top, \mathbf{P}_{\mathcal{I}_B}^\top, \mathbf{r}_v^\top, \beta_n, l_n \right]^\top. \quad (3.1)$$

³Since \mathbf{z}_n can change in size from one time to another, sometimes a set notation is preferred. For the method described here, the vector notation is more compact, and the fact that \mathbf{z}_n changes in size, although problematic from a purely mathematical point of view, does not pose problems in implementation where we can introduce a special symbol for missing measurements.

3. Multi-modal indoor localization

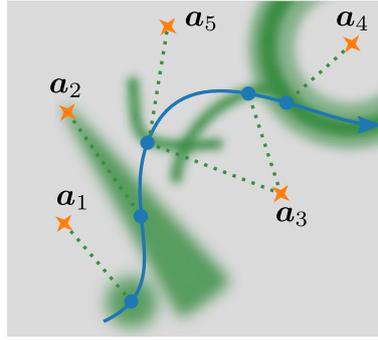


Figure 3.1.: Examples of measurement types that can be considered in the proposed framework. Anchor 1 provides point-wise measurements (e.g. from visual scanning of paintings, as considered in this chapter), anchor 2 provides an angular measurement (e.g. from sound, see Chapters 5 or 6), anchors 3 and 5 provide time-difference-of-arrival, which yields a hyperbolic measurement, anchor 3 provides a distance measurement (e.g. through WiFi round-trip-time, as considered in this chapter) and anchor 4 provides received signal strength (e.g. through Bluetooth, as considered in this chapter).

Finally, we want to incorporate some knowledge about the map: in particular, we want to be able to specify the regions that are actually accessible, and regions that are separated by walls. We provide this information through a selection of locations to be considered in the inference process as well as their connections, described as binary edges, as explained in Section Section 3.3.

Based on above considerations, we can formulate the problem solved in this chapter as follows:

Problem 1. *Given partial and multi-modal measurements z_n at times t_n , $n = 1 \dots N$, determine the sequence of positions $r_{1:N}$, subject to map constraints, that most likely generated the measurements.*

Before proceeding to discuss this problem, we emphasize that the proposed solution is not limited to the measurements types considered in this chapter. In Figure 3.1, anchor a_2 provides an angular measurement, and the time-difference-of-arrival from anchor a_3 and a_5 provide a hyperbolic measurement. As we discuss in 3.6.1, even measurements without known forward model could be integrated into the proposed framework.

3.2. Related work

Measurement modalities While an abundance of measurement modalities have been proposed for indoor localization [152], we focus on modalities that are smartphone compatible. Among these systems, WiFi, Bluetooth, IMU and images are most commonly available and used. We will investigate sound as an alternative in the last two chapters of Part I of this thesis.

For the RF signals, the coarsely quantized and often unreliable Received Signal Strength

Indicator (RSSI) is most commonly used. With the WiFi Round Trip Time (RTT) feature, known as *IEEE 802.11mc FTM*, more accurate distance measurements are now available [97]. With the publication of the new Bluetooth 5.1 standard, it is also expected that angular information will soon become more widespread [198]. Compared to WiFi, Bluetooth uses less power and is cheaper to deploy. However, it has a shorter range of operation (1-5 meters compared to 50-100 meters for WiFi[152]) and is typically only used for proximity detection.

It is widely known that the quality of RF signals degrades in challenging indoor environments. While it is difficult to reduce this effect, estimating the accuracy of each measurement and processing this appropriately can significantly improve performance. By identifying them, we can for instance remove the usually very unreliable NLOS signals. In this regard, Xaio *et al.* [220] extract different RSSI-based features to identify NLOS conditions in Wifi-based localization and Li *et al.* [119] leverage the observation that NLOS signals have a higher variability than line-of-sight (LOS) signals. Bahillo *et al.* [11] use distances inferred from RTT measurements as constraints to improve RSSI-based distance estimates. Li *et al.* [118] propose new features extracted from the channel state information (CSI) which can be used to differentiate LOS from NLOS measurements. While CSI can greatly improve performance, it should be noted that, currently, it is not readily available on smartphones. In addition, most methods distinguishing LOS from NLOS rely on a training phase and are thus not suitable if the aim is to have little setup and calibration time.

Almost all smartphone-based systems leverage IMU data to improve performance. Since IMUs provide an estimate of the device's relative movement, both in terms of travelled distance and movement direction, integration can be applied to obtain position estimates; however, double integration of accelerometer data is notoriously inaccurate and step detection combined with estimated step length and direction is more commonly used [84, 135, 224]. We use a simple step counter in combination with a step length for inference, which is a rough approximation of travelled distance, but, as we show in our evaluation, it is sufficient when complemented with absolute measurements from other modalities.

On the vision side, object recognition and localization from visual features are well-studied topics and a review of these is beyond the scope of this chapter. Therefore, we refer the interested reader to standard textbooks on multi-view geometry and computer vision [89, 130, 170].

Localization algorithms Indoor localization algorithms can be very broadly split into two categories: those that predominantly use geometric information—such as lateration—and those that learn how signals behave in the environment, also called fingerprinting. Since we discuss geometric methods in many parts of this thesis, we focus on fingerprinting in this section.

The concept of fingerprinting has seen widespread use in indoor localization. For example, Guimarães *et al.* [84] use WiFi fingerprint maps for coarse location estimation, and refine the estimate using magnetic fingerprints and IMU measurements. Shu *et al.* [191] use the same measurement modalities, but combine them in a particle filter. Since their system

3. Multi-modal indoor localization

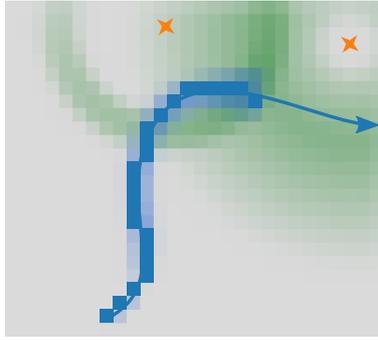


Figure 3.2.: Pictorial example of the discretization of the state space. The measurements from anchors (in red), in this case a range and a power measurement, can be translated to a posterior state distribution that is highly non-Gaussian (in green). The trajectory (in blue) is estimated on the same grid.

uses a bidirectional dynamic time warping method, it operates with a delay. Xiao *et al.* [218], on the other hand, propose a real-time localization system using similar modalities. Their probabilistic framework introduces features for WiFi fingerprints and IMU measurements to yield a probability distribution for the device’s position. The listed approaches require fingerprint databases, which have to be collected and updated regularly in time-consuming offline calibration phases. Since the creation of these databases is the major bottleneck in terms of setup efforts, there have been multiple lines of research proposing to speed up this process, or to crowd-source the fingerprint creation and maintenance over time [171, 172, 219, 224]. Other methods automatically create fingerprint maps using robots [38]. The sensitivity of fingerprint maps to short-time variations in the environment however stays an unavoidable shortcoming of these methods. For instance, it was reported that a moving elevator was the main cause of fingerprint disturbances in the 2017 Microsoft Indoor Localization Competition [129].

The method proposed in this chapter does not rely on fingerprint maps and therefore avoids the lengthy setup time. The mathematical framework is based on the conditional random field formulation used by Xiao *et al.* [218], but we replace the fingerprint measurements by RSSI-based and RTT-based distance measurements. In [218], global orientation offsets are calibrated by inferring the position over multiple time steps, which can introduce unwanted latency in the system. In contrast, we achieve calibration by using sparse visual features and therefore do not introduce delays.

3.3. Conditional random fields

We choose to solve Problem 1 with a conditional random field, which is a discriminative model by design, as discussed in Section 2.3.2. The state over which we perform inference contains the position only: at time t_n , the random variable Y takes as value the current position $y_n = r_n$. We uniformly discretize the state space, and denote the position of grid point $i = 1 \dots N_g$ by $r[i] \in \mathbb{R}^D$. In theory, the points can be in three dimensions, but since we restrict ourselves to one floor, we will use a fixed height. The advantage of such a discretiza-

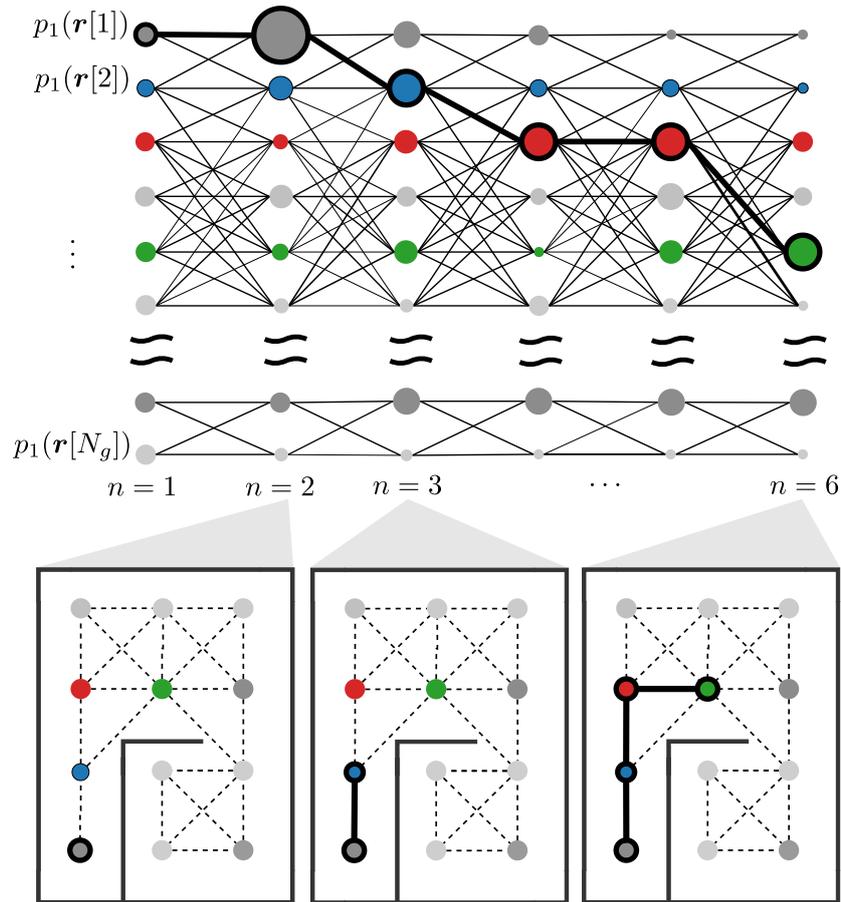


Figure 3.3.: Visualization of the proposed CRF algorithm. At each time index n , the probability of each point (proportional to its size) is computed by considering the connected grid points and the transition probabilities (proportional to the line thickness). After each step, we pick the point with highest probability as our position estimate (circled in black).

3. Multi-modal indoor localization

tion is that it can approximate non-Gaussian state distributions, as visualized in Figure 3.2. The localization accuracy we lose due to this position quantization is an acceptable price to pay since the accuracy we can hope to achieve, given the noisy measurement modalities, is typically up to one to two meters. Between each pair (i, j) of points, we introduce binary-valued edges denoted by $e_{ij} \in \{0, 1\}$, which incorporate map information: the points i and j are only connected if point i is accessible from point j and vice versa.

The goal is to find the sequence of states that maximizes the conditional probability $p(\mathbf{r}_{1:N} | \mathbf{z}_{1:N})$. To improve readability, we omit the conditioning on map constraints such as anchor locations or edges since we do not treat the mapping problem in this chapter.

Because of the sequential nature of the problem, the random field is a chain (called a linear-chain CRF) where each state is connected to its predecessor and successor state in time. In this model, the conditional probability function of states can be represented by a product of potential functions [218]:

$$p(\mathbf{r}_{1:N} | \mathbf{z}_{1:N}) \propto \prod_{n=2}^N \Psi(\mathbf{r}_n, \mathbf{r}_{n-1}, \mathbf{z}_{1:N}). \quad (3.2)$$

The potential functions $\Psi(\mathbf{r}_{n-1}, \mathbf{r}_n, \mathbf{z}_{1:N})$ are, in turn, composed of $K > 0$ feature functions f_k :

$$\Psi(\mathbf{r}_n, \mathbf{r}_{n-1}, \mathbf{z}_{1:N}) = \exp \left(\sum_{k=1}^K \lambda_k f_k(\mathbf{r}_n, \mathbf{r}_{n-1}, \mathbf{z}_{1:N}) \right). \quad (3.3)$$

Here, each feature function characterizes the likelihood of transition from state \mathbf{r}_{n-1} to state \mathbf{r}_n given the observations $\mathbf{z}_{1:N}$. The parameters λ_k are used to tune the relative importance of feature functions and could be learned from data, as described in [218]. Since we incorporate the weight of the different features in the design of their functions through the expected measurement quality, we assign the same $\lambda_k = 1$ to all features. We also use this parameter to inform the algorithm of the absence of certain measurement types by setting it to zero when appropriate.

The location of the device is found sequentially, one step at a time, using the Viterbi algorithm. At any time step n , the algorithm computes, for all possible grid points $\mathbf{r}[i]$, the probability $p_n(\mathbf{r}_{1:n}^{(i)} | \mathbf{z}_{1:n})$ of the most likely sequence of states with $\mathbf{r}_n = \mathbf{r}[i]$, written as $\mathbf{r}_{1:n}^{(i)}$. The main step of the Viterbi algorithm can be written as

$$p_n(\mathbf{r}_{1:n}^{(i)} | \mathbf{z}_{1:n}) = \max_l p_{n-1}(\mathbf{r}_{1:n-1}^{(l)} | \mathbf{z}_{1:n-1}) \Psi(\mathbf{r}[i], \mathbf{r}[l], \mathbf{z}_{1:n}). \quad (3.4)$$

The algorithm is visualized in Figure 3.3.

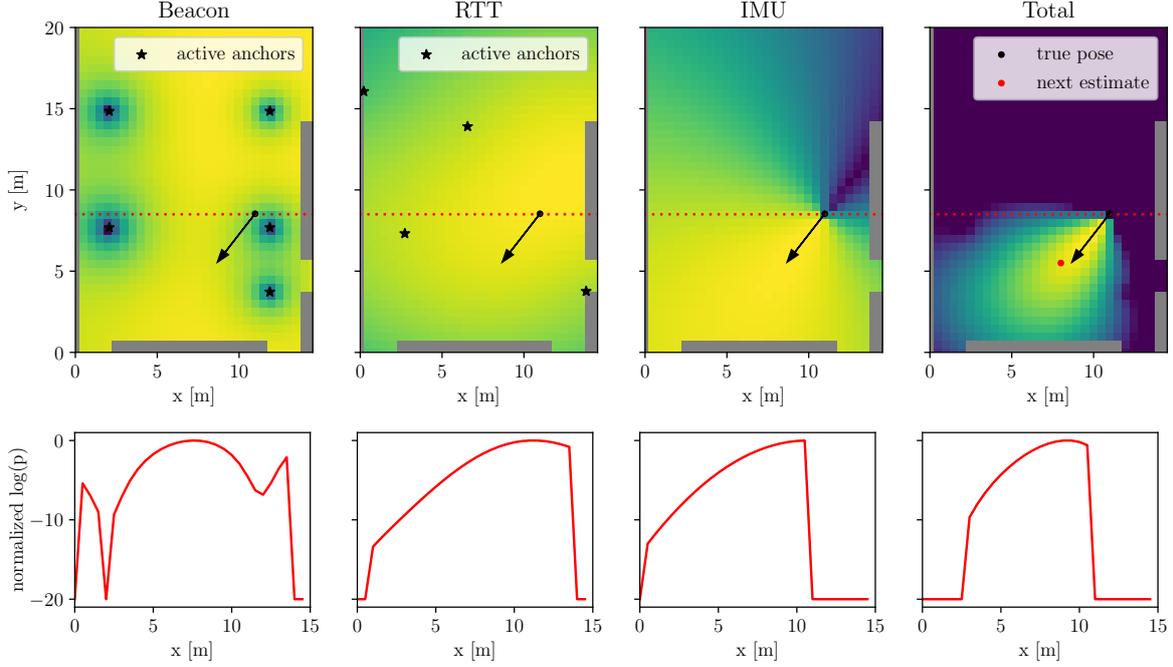


Figure 3.4.: Probability maps inferred at one time instance for the different systems. The standard deviations are 1 dBm for Beacon RSSI, 2 m for IMU distance and 0.1 radians for IMU angle (the sum is shown), and 1 m for RTT distance measurements. The bottom row shows the 1D-cut of the distribution along the red dotted line. Probabilities are normalized for better readability.

3.3.1. Modeling feature functions

The main challenge of the proposed framework is the design of transition probabilities $\Psi(\mathbf{r}_n, \mathbf{r}_{n-1}, \mathbf{z}_{1:N})$. In the following, we introduce features based on generative models for WiFi RTT, Bluetooth RSSI and IMU data, as well as visual measurements.

Note that all features are normalized such that their sum on the considered grid equals one before feeding them into (3.3).

RTT feature function

For the RTT feature function, we assume that distance measurements to anchors follow a Gaussian distribution. Namely, the measured distance d_m to anchor m follows the distribution $\mathcal{N}(d_m^*(\mathbf{r}_n), \sigma_m^2)$, where $d_m^*(\mathbf{r}_n) = \|\mathbf{a}_m - \mathbf{r}_n\|$ is the true distance (conditioned on the device being at \mathbf{r}_n) and σ_m^2 is the measurement variance. We also assume that the measurements to different anchors are independent. This leads to the probabilities

$$p(\mathbf{d}_{\mathcal{I}_W} | \mathbf{r}_n) \propto \prod_{m \in \mathcal{I}_W} \frac{1}{\sigma_m \sqrt{2\pi}} \exp\left(-\frac{(d_m - d_m^*(\mathbf{r}_n))^2}{2\sigma_m^2}\right), \quad (3.5)$$

3. Multi-modal indoor localization

from which we deduce the feature function

$$f_1(\mathbf{r}_n, \mathbf{z}_n) = \sum_{m \in \mathcal{I}_W} \ln \left(\frac{1}{\sigma_m \sqrt{2\pi}} \right) - \frac{(d_m - d_m^*(\mathbf{r}_n))^2}{2\sigma_m^2}. \quad (3.6)$$

RSSI feature function

With measures of received signal strength, such as RSSI, the corresponding posterior distribution is not the same as for RTT. Let P_m be the power of the signal received from anchor m . We assume that, due to noise and interference, P_m follows a Gaussian distribution $\mathcal{N}(P_m^*, \sigma_m^2)$, where P_m^* is the expected (or ground truth) power and σ_m^2 is the power variance, both expressed in dBm.

The expected received signal power, P_m^* , is related to the true distance from the anchor $d_m^*(\mathbf{r}_n)$ via the log-distance path loss model expressed by

$$P_m^* = T_m - 10 n \log_{10} d_m^*(\mathbf{r}_n), \quad (3.7)$$

where T_m is a constant and n is the path loss exponent. Using the same argument as above, we introduce the feature function

$$f_2(\mathbf{r}_n, \mathbf{z}_n) = \sum_{m \in \mathcal{I}_B} \ln \left(\frac{1}{\sigma_m \sqrt{2\pi}} \right) - \frac{(P_m - P_m^*(\mathbf{r}_n))^2}{2\sigma_m^2}. \quad (3.8)$$

IMU feature functions

We calculate a travelled distance l and direction β from IMU measurements at time n as follows. First, we assume that from variations in the vertical acceleration we can identify step counts. We then calculate a step vector, obtained by averaging the product of an estimated step length with the direction for each step. To simplify, we fix the average step length throughout the experiments. Finally, we calculate the distance and orientation estimate as the norm and the angle of the averaged step vector.

Under a Gaussian noise assumption, the IMU features can be written as

$$f_3(\mathbf{r}_n, \mathbf{r}_{n-1}, \mathbf{z}_n) = \begin{cases} \ln \left(\frac{1}{\sigma_l \sqrt{2\pi}} \right) - \frac{(l - \|\mathbf{r}_n - \mathbf{r}_{n-1}\|)^2}{2\sigma_l^2} & \text{if } e_{n,n-1} > 0 \\ 0 & \text{else} \end{cases}, \text{ and} \quad (3.9)$$

$$f_4(\mathbf{r}_n, \mathbf{r}_{n-1}, \mathbf{z}_n) = \begin{cases} \ln \left(\frac{1}{\sigma_\beta \sqrt{2\pi}} \right) - \frac{\Delta(\theta, \angle(\mathbf{r}_n, \mathbf{r}_{n-1}))^2}{2\sigma_\beta^2} & \text{if } e_{n,n-1} > 0 \\ 0 & \text{else} \end{cases}, \quad (3.10)$$

where $\angle(\cdot)$ returns the angle between two states, and σ_β, σ_l are the standard deviation of angle and distance measurements, respectively. The operator Δ returns the angular difference in the interval $[-\pi, \pi]$.

Experimental distributions for the four proposed feature functions, f_1 to f_4 , are shown in Figure 3.4. To ensure a good balance, it is crucial to choose appropriate standard deviations for each of them. The choice of the standard deviations, shown in the Figure, is motivated in Section 3.4.3.

Visual measurements

Our visual system, called *Pixlive*, uses AGAST, BRIEF and ORB [37, 133, 177, 178] to detect and create descriptors of feature points, which are compared to a database of known images. The system selects the best match and finally computes the homography and pose using multi-view geometry techniques [89]. We emphasize that these techniques are standard and not the focus of this chapter.

Since the visual system provides centimeter-level accuracy, we treat measurements from it as “ground truth”. Denoting the position estimate obtained through visual landmark scanning at time n by \mathbf{r}_v , we reset our probability map to a Gaussian distribution around this estimate, using a fixed low variance σ_v ,

$$p(\mathbf{r}_n | \mathbf{z}_n) = \frac{1}{\sigma_v \sqrt{2\pi}} \exp\left(-\frac{\|\mathbf{r}_n - \mathbf{r}_v\|^2}{2\sigma_v^2}\right). \quad (3.11)$$

The variance should be chosen sufficiently small such that only few states around the visual estimate have a non-zero probability. For a 0.5 m-spaced grid we found that $\sigma_v=0.05$ m yielded this desired behavior.

3.3.2. Location inference

By applying (3.4) at time n , we obtain a probability map that contains the probability of the device being at each grid point. To obtain a single position estimate at each time, we pick the point of maximum probability, given by

$$\hat{\mathbf{r}}_n = \arg \max_{\mathbf{r}[i], i=1 \dots N_g} p_n(\mathbf{r}[i] | \mathbf{z}_{1:n}). \quad (3.12)$$

In the pictorial example, shown in Figure 3.3, the obtained position estimates are circled in black.

We reduce the search space of the Viterbi algorithm (3.4) to grid points l for which the probability at the previous time step $p_{n-1}(\mathbf{r}[l] | \mathbf{z}_{1:n-1})$ was significant. With a threshold chosen low enough (in our case, 1e-10), this does not affect the result because these states are not realistic candidates, however it does speed up the position inference significantly.

Finally, we can recursively backtrack the best position estimates as follows. By keeping track of the most likely predecessor state in (3.4), we reconstruct the sequence of states that lead to the current estimate. Experiments show that this approach can help to smooth the

3. Multi-modal indoor localization

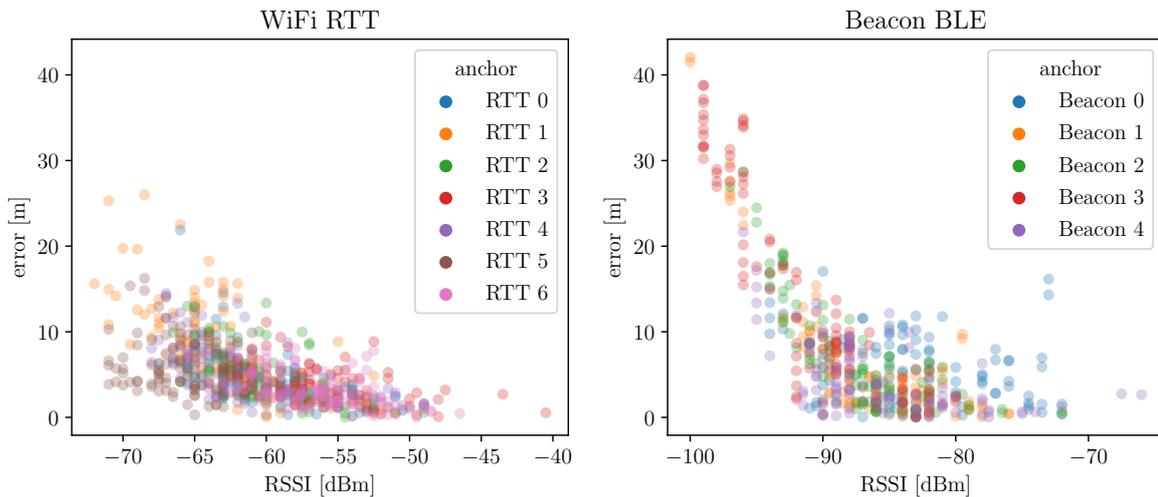


Figure 3.5.: Accuracy of Bluetooth (left) and WiFi (right) anchors for zig zag dataset. Plotted is the absolute error between real and measured distance on y , vs. the received signal strength indicator (RSSI) on x .

trajectory if one allows a small delay; see the *CRF-recursive* solution in Section 3.5. This method can be viewed as the equivalent of performing smoothing rather than filtering in Bayesian localization, yielding the best position estimate considering all measurements obtained so far. In practice, to balance between the higher performance of the smoothing approach and the online capabilities of filtering, one can perform the recursive backtracking over a fixed time window.

3.4. Processing and calibration

3.4.1. Outlier rejection

As outlined in the previous section, we assume distance measurements from RTT anchors to be zero-mean Gaussian and Beacons to be log-normally distributed. In order to get closer to this assumption, we filter the raw measurement before feeding them into our algorithm.

The accuracy of raw BLE Beacon and WiFi RTT measurements are shown in Figure 3.5. The measurements are taken in a challenging environment with multi-path and shadowing effects. By plotting distance error vs. RSSI, we see that low RSSI measurements are correlated with high distance errors. For WiFi signals, which are generally stronger than BLE signals, the threshold at which the signal deteriorates is at around -65 dBm. For the lower-energy Bluetooth signals, it is at -90 dBm, which is close to the receiver sensitivity of standard BLE beacons. We found that rejecting measurements below this threshold was beneficial to the overall localization accuracy.

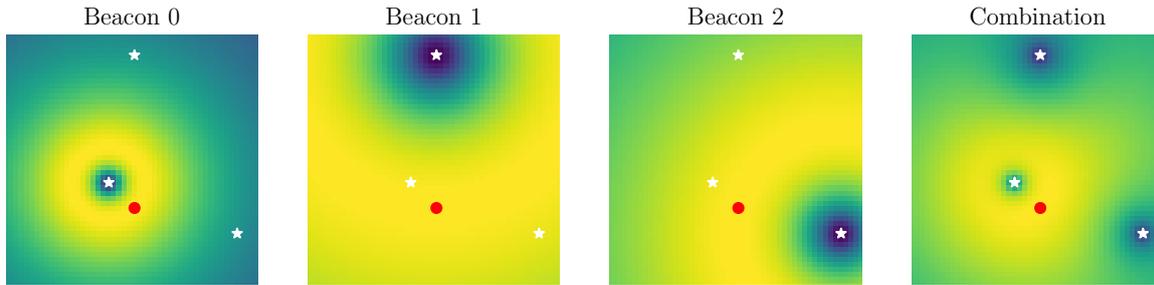


Figure 3.6.: *State distribution after incorporating Bluetooth RSSI measurements, on simulated data. Shown in white and red are anchors and the real device positions, respectively. Note that the distribution flattens out for high distances (for instance for Beacon 1).*

3.4.2. Online calibration

The visual system is not only used to reset the probability map from time to time, it is also exploited for online calibration of the unknown parameters of the different systems: the offset of RTT distance measurements (introduced primarily by multipath and unknown latency issues [174]), the transmit power of the Beacons, and the absolute orientation of the phone (IMU magnetometer tends to be very noisy and gyroscope measurements tend to drift). When a visual measurement is recorded, we record measurements for the following second, assuming the user is standing still during this time, and use the median of the recorded measurements to calibrate each modality. We use the thereby obtained offsets for each WiFi access point, and transmit powers for each Bluetooth beacon, to correct the obtained distance estimates in real-time.

3.4.3. Feature weights

The weights of the different features need to be chosen so that the features complement each other appropriately, and none of them dominate the final probability map. Inspecting Equations (3.6), (3.8) and (3.10), the main parameters to tune are the standard deviations of each modality. One can obtain estimates of standard deviations from each device experimentally. However, we found that fixing one system-dependent standard deviation for each modality yielded sufficiently good results. The chosen standard deviations are shown along with the resulting probability maps for a sample data point in Figure 3.4. Note that the Beacon features get absorbed by the other two features, which is desired since we expect much lower accuracy from Beacons than from RTT or IMU.

As previously mentioned, when it comes to RSSI features, we assume that the received signal strength, expressed in dBm, follows a normal distribution. This noise model choice has the convenient effect that distance measurements from close beacons are considered more accurate than measurements from far beacons. Intuitively speaking, constant variance across different signal strengths will not have the same effect on ranging quality. To give an example, if a 6 dB change amounts to doubling of the distance, the variation in received signal strength that amounts to 6 dB is not the same thing if the true distance is 1 m or 10 m.

3. Multi-modal indoor localization

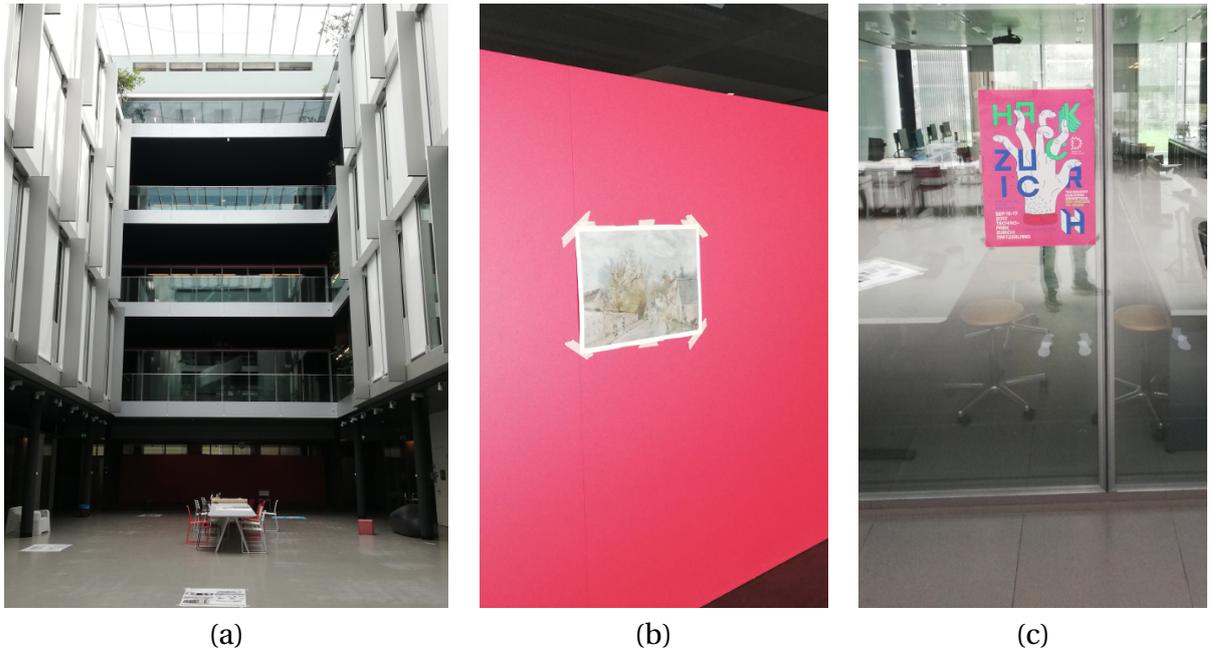


Figure 3.7.: (a) Picture of the BC atrium at EPFL, the location of the experiments. (b-c) Two example images used as visual features for calibration.

The simulated probability distributions plotted in Figure 3.6 obviate this intuition. This is a desired effect since Bluetooth RSSI signals are known to degrade strongly with distance. A high variance at high nominal distances means that distant Bluetooth anchors are effectively ignored in the location inference process.

3.5. Results

We show the effectiveness of the proposed method in two real-world experiments in a university building.

3.5.1. Environment of experiments

The area for experiments is a large, furnished hall containing glass-windowed lecture rooms, depicted in Figure 3.7 (a). Five visual anchors are mounted on walls and glass windows as shown in Figure 3.7 (b) and (c). The different RTT and Bluetooth anchors are distributed over the area as shown in Figure 3.8 (a). We also show the sparse visual anchors (*Pixlive*) and the UWB anchors, used for benchmarking only. We walk two different trajectories, shown in Figure 3.8 (b), named *double loop* and *zig zag*, respectively, because of their characteristic shapes. The pose estimates obtained occasionally from scanning of the visual anchors (at 5 and 6 different points, respectively) are also depicted. During the experiment, there is light traffic of students coming in and out of lecture rooms.

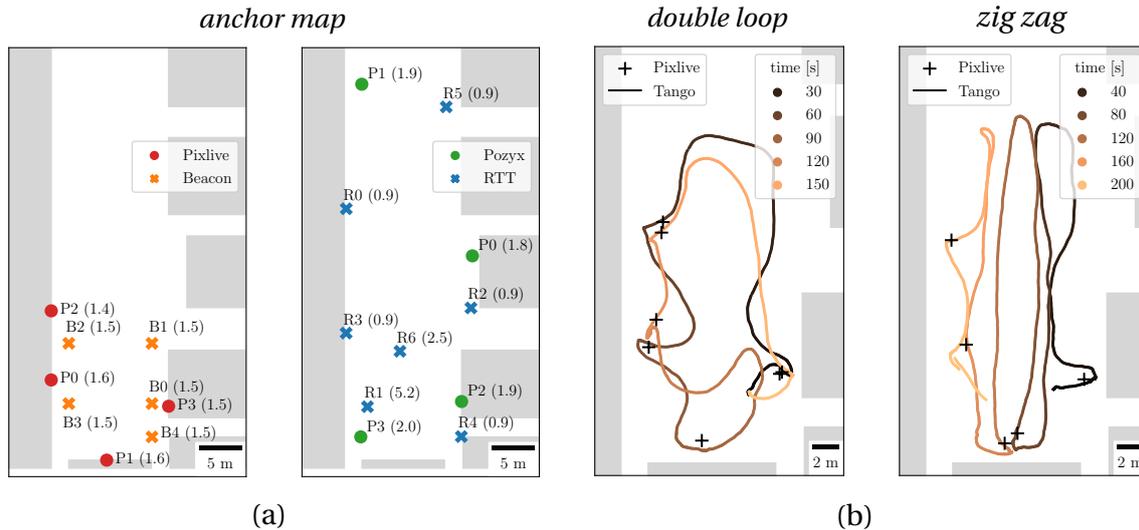


Figure 3.8.: (a) Anchor layout used for experiments. The height of each anchor in meters is included in parentheses. (b) The two studied datasets with ground truth obtained from Tango. The color maps from dark (early) to light (late) positions. Sparse visual position estimates from Pixlive are also shown.

For ground-truth data, we use the Tango visual system. However, the sparsity of robust visual features induced by uniform and repetitive structures and the large glass walls posed problems for this system. Therefore, we added multiple feature-rich posters on the floor to ensure robust localization. We emphasize that these added features are only used to obtain ground-truth data and are not part of our proposed system.

While the phones acquiring Bluetooth, RTT, IMU and visual features for indoor localization are carried by the test subject, they are followed by a second subject carrying a laptop that is used only for the UWB processing. Both the subject and the laptop carrier thus create challenging non-line-of-sight conditions for certain anchors.

3.5.2. Overview of used technologies

The ground truth Tango data is obtained with the augmented reality platform from Google. It is acquired from an indoor localization application running on a *Lenovo Phab 2 Pro* mobile device with a motion tracking camera and RGB-IR camera [214].

WiFi measurements are gathered from RTT-enabled *Fitlet2* access points by *Compulab* using the WiFi Indoor Positioning application [45] from a *Google Pixel* smartphone running *Android 9 Pie*. The measurements from the other systems are acquired with an *LG Nexus 5X* mobile phone. The beacons from *Kontakt.io* are low-cost Bluetooth Low Energy emitting devices for proximity detection. IMU data is obtained from the smartphone's gyroscope and accelerometer. The sensor data is gathered and transmitted by a custom Android application. The visual position estimates are acquired from PixLive and obtained with a custom Android application that uses Vidinoti's SDK [209]. All the systems send their output to a *python* server which stores the acquired data.

3. Multi-modal indoor localization

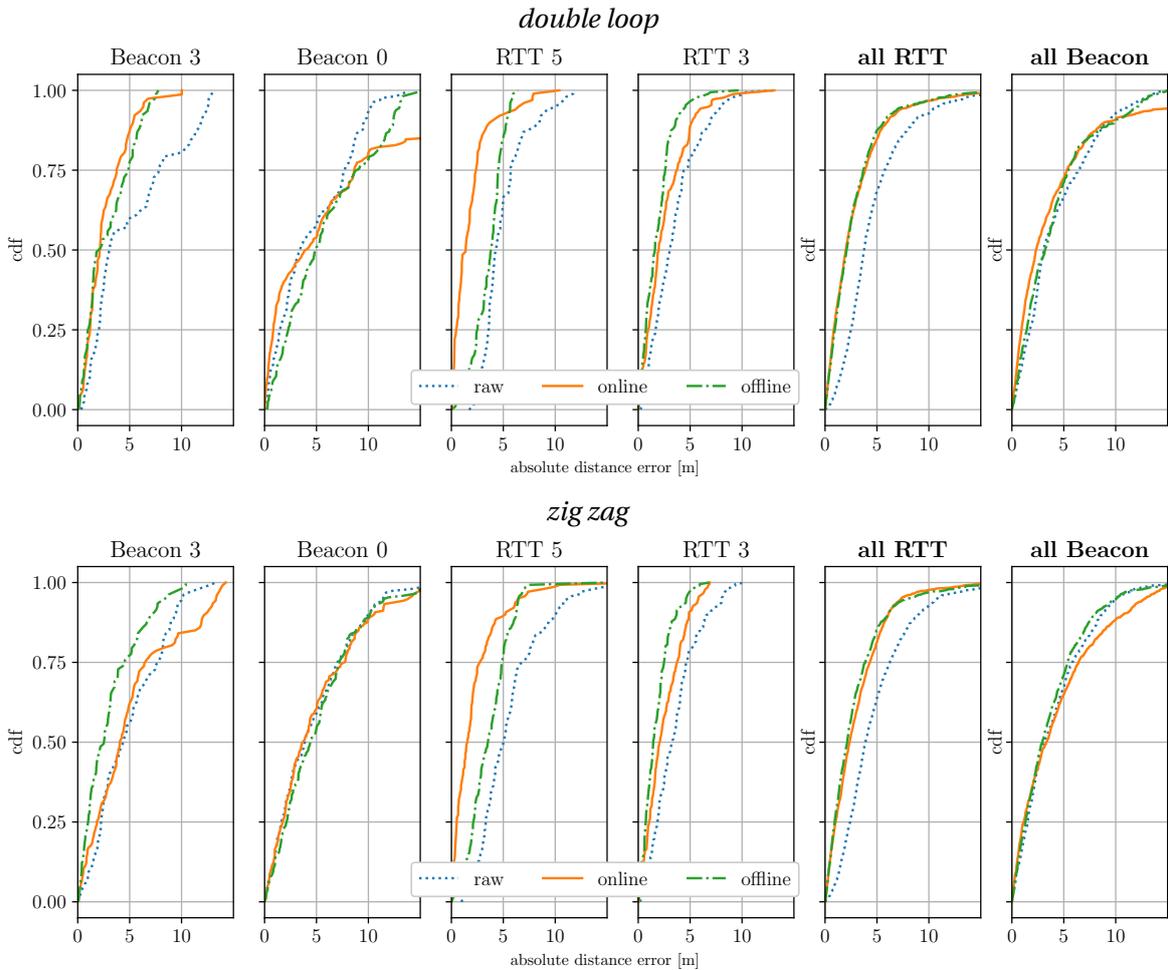


Figure 3.9.: Cumulative distribution functions of absolute distance measurement errors with and without online/offline calibration. On average, online and offline calibration improve the distance accuracy of RTT distance measurements significantly, however online calibration can induce high errors for certain Bluetooth beacons.

The UWB system, used for comparison, is called *Pozyx* [166] and requires four fixed anchors and one tag mounted on an *Arduino UNO*, which is connected with a USB cable to a laptop running the acquisition server.

We fix the two phones and the UWB tag on a custom portable wooden mount, which makes sure their relative positions stay the same throughout the experiment. The Tango device is carried in the other hand so that it can be moved freely to scan the environment. The devices are thus not exactly co-located, from which we expect a small additional positioning error in the range of 5-10 cm.

3.5.3. Data accuracy and calibration

We first evaluate the accuracy of the obtained distance measurements, with and without online calibration. Figure 3.9 shows the obtained distance accuracy for the two studied

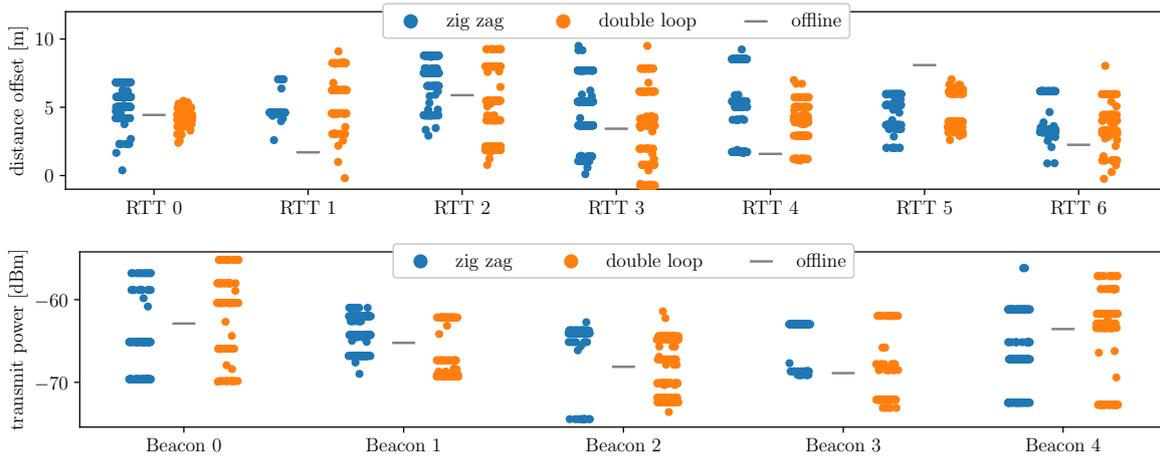


Figure 3.10.: Calibration parameters for WiFi RTT anchors (top) and Bluetooth beacons (bottom). Each colored dot corresponds to one online calibration value, split by dataset, and the gray lines correspond to the values obtained by the offline scheme. Online calibration parameters are recalculated whenever a visual measurement is present. The high variance in calibrated parameters motivates the use of online calibration over offline calibration.

datasets. For comparison, we also plot the result from offline calibration, obtained using measurements at 5 static positions (ca. 60 seconds each) before the experiments started. For each dataset, we show two example Beacon and RTT anchors, respectively, and the cumulative error for all anchors. While calibration is always beneficial for RTT anchors, it can lead to faulty results for the Bluetooth beacons, for instance for *Beacon 0* in the *double loop* dataset. We have found that this is due to the high variance of Bluetooth RSSI, which means that using a small window for calibration can lead to high bias. Figure 3.10 shows that there is indeed a high variance of the calibrated parameters for the two different datasets. However, the localization results show that faulty distances are successfully compensated for by other more accurate modalities. Both offline and online calibration are therefore valid choices, however online calibration has the advantage of not needing any additional setup time and is the preferred solution.

3.5.4. Evaluation of available modalities

For localization, we initialize a grid including connectivity information given the map of the building. A grid size of 0.5 m is found to yield fast enough inference and satisfactory resolution. Since we are limited to one floor, we only use one layer in the z-direction at the approximate height of the devices during experiments (1.2 m).

We first evaluate the performance of the proposed system for different combinations of measurement modalities and calibration schemes. Table 3.1 shows a summary of the obtained localization accuracy, in terms of median and mean localization errors over the whole dataset. We denote by localization error the Euclidean distance between the position estimate and the ground truth obtained from Tango.

3. Multi-modal indoor localization

Table 3.1.: Comparison of proposed method with different calibration schemes and modalities used. The best scores per row and column are highlighted in bold and color, respectively.

calibration		mean error[m]			median error[m]		
		none	offline	online	none	offline	online
double loop	IMU	4.55	4.55	4.55	3.94	3.94	3.94
	RTT	5.85	4.06	3.34	4.99	3.11	2.48
	BLE	7.08	6.65	6.18	5.85	5.68	4.64
	RTT+IMU	4.07	3.21	2.87	3.99	3.03	2.78
	BLE+IMU	4.36	4.21	3.78	3.56	3.85	3.35
	RTT+BLE	5.77	4.06	3.18	4.83	3.20	2.48
	RTT+BLE+IMU	4.28	3.27	2.60	4.09	2.90	2.29
zig zag	IMU	5.32	5.32	5.32	4.99	4.99	4.99
	RTT	5.06	3.36	3.05	5.02	3.07	2.76
	BLE	6.05	5.71	7.93	5.51	5.39	6.40
	RTT+IMU	3.70	2.99	2.62	3.78	3.06	2.49
	BLE+IMU	5.08	5.04	4.46	3.70	4.41	3.26
	RTT+BLE	4.90	3.24	2.93	4.76	2.97	2.68
	RTT+BLE+IMU	3.60	2.93	2.44	3.50	2.98	2.33

It is immediately apparent that adding IMU features increases the accuracy of localization significantly, even though the IMU measurements on their own would yield poor localization results. Adding Beacon measurements on average only slightly improves accuracy. This is expected since Beacon measurements are the least reliable ones. Furthermore, the Beacon feature function is relatively flat compared to the other features by design, so they only have little impact on the global probability distribution.

In terms of calibration schemes, online calibration yields the best results for almost all combinations of systems. Calibration is particularly powerful for RTT measurements, which otherwise can exhibit a high offset: it almost halves the median and mean error for both datasets.

Finally, we emphasize that it is best to combine the three available modalities, leading to the lowest median and mean errors. Considering the significant differences in measurement quality (note the difference in localization performance when using each system individually), our system thus weighs all modalities correctly when combining them. Indeed, the 2D localization plots in Figure 3.11 show that Bluetooth and Wifi RTT measurements alone lead to jumpy estimates, while using IMU only induces high drift. The proposed method favorably combines the used modalities.

3.5.5. Comparison with other methods

In this final section, we provide a quantitative analysis of localization performance. We study our algorithm, which we call CRF, to purely distance-based algorithms. We choose the best-performing combination from Table 3.1 in terms of mean localization error (RTT,

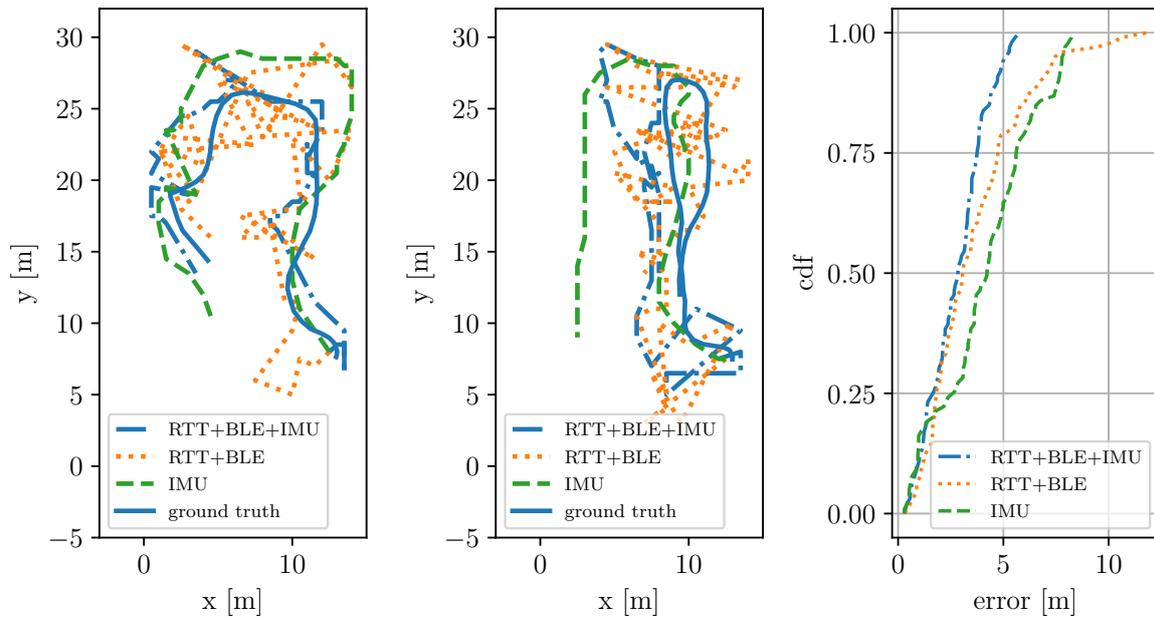


Figure 3.11.: Comparison of localization results depending on the measurement types used, for the first 60 seconds of measurements of the double loop (left) and zig zag (middle) datasets. The total cumulative distribution function of the localization errors is shown on the right.

Beacon and IMU measurements and online calibration). We add the recursive solution discussed in 3.3.2, where we backtrack the trajectory from the final position estimate (called *CRF recursive*). We compare to the grid-search implementation of range-based least squares RLS, denoted by *grid-L2*, and the optimal squared-range-based least squares SRLS solution. Both solutions were proposed by [18] and are discussed in more detail in Section 4.2. We also introduce a variation of RLS using the median rather than mean distance error (denoted by *grid-L1*) and a simple weighted centroid algorithm similar to [5]. In our centroid algorithm, we linearly interpolate the three closest anchor coordinates, using the inverse of the distance measurements as weights.

Figure 3.12 shows 2D plots of the obtained localization for a selection of methods. Thanks to the time consistency imposed by the IMU features in our implementation, the position estimate is smooth and its shape is close to the actual trajectory of the target. For the distance-only methods, the estimate is very volatile and the shape of the trajectory is hard to discern.

Combining all position estimates, we obtain the cumulative distribution function of the position error, shown in Figure 3.13. Compared to the UWB-based solution *Pozyx*, which requires designated hardware, our method yields higher localization errors. However, the position accuracy of the proposed method is the best amongst the shown smartphone-compatible solutions. In particular, high error estimates are significantly reduced in the proposed framework. The difference in performance between the four benchmark methods is quite small: for the grid-based methods and SRLS, this is explained by the relatively high

3. Multi-modal indoor localization

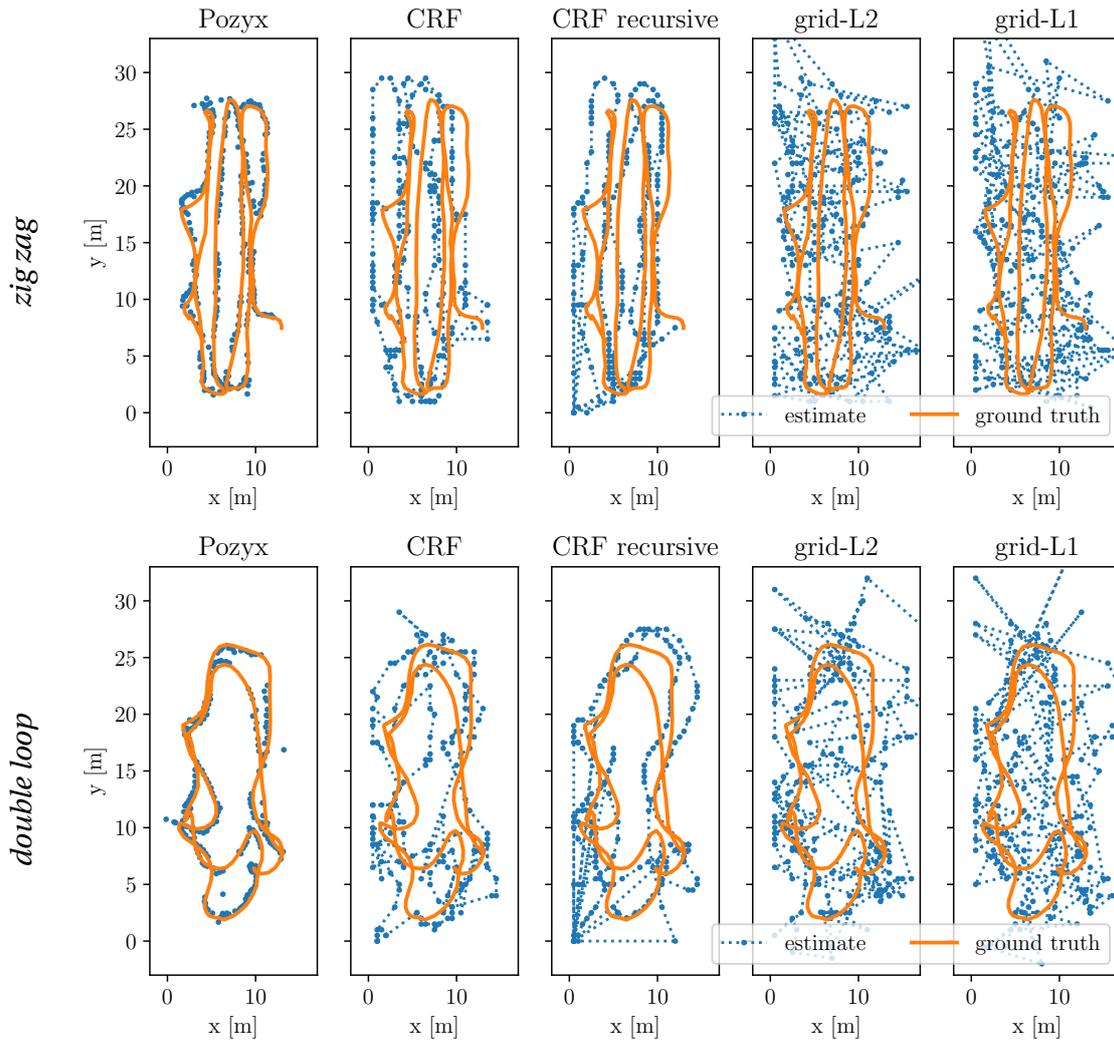


Figure 3.12.: Visualization in 2D of the proposed localization method compared to standard and state-of-the-art methods, on the two studied datasets. Our method (CRF) yields smooth and accurate position estimates, comparable with the UWB-based system Pozyx. The recursive location inference (CRF recursive) smooths the results a posteriori and corrects for the slight drift in the zig zag dataset.

distance accuracy (after calibration), meaning that the median and mean are similar, and also that RLS and SMLS are expected to behave similarly; the only difference being the squaring of distances for SMLS [18]. The reasonable performance of the centroid algorithm is explained by a relatively dense deployment of anchors.

3.6. Conclusion and ongoing work

We have proposed a smartphone-compatible multi-modal indoor localization system that integrates various measurement types using a probabilistic framework. Experimental results show that the system yields localization accuracy of around two meters. The setup can be

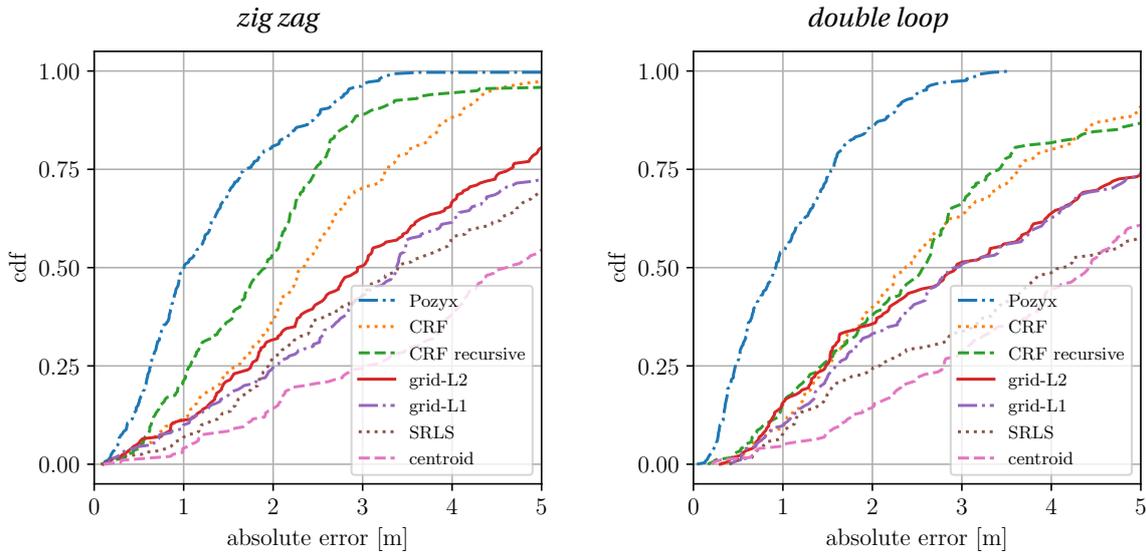


Figure 3.13.: Cumulative distribution function of localization error. We compare the proposed localization methods with other distance-based methods on two different datasets. Overall, our methods (CRF and CRF recursive) are the best-performing method among the shown algorithms, and beaten only by the UWB-based Pozyx system, which is not smartphone compatible. The recursive application of our algorithm improves the accuracy for the zig zag dataset significantly.

quickly installed with no training phase required. It is passive for the vast majority of the time with the user only actively scanning the environment from time to time.

The system has been successfully implemented on a smartphone and was integrated in the industry partner’s software. Since the method is flexible in terms of measurement modalities used, it can be easily adapted to the available equipment. The inference is performed forward only, but the recursive solution could be implemented, using a fixed inference window, if higher accuracy is required.

There are two main points that could be improved in the proposed method. First, the localization accuracy in more adverse environments than the two studied datasets should be investigated. In particular, NLOS effects, *e.g.* anchors whose signals are blocked by walls, are omnipresent in most real-world conditions and might require extra precautions. We expect that the online calibration and outlier rejection schemes can help identify unreliable anchors to some extent, but further studies are required to provide evidence. Along the same lines, different anchor layouts and room geometries could be tested and ablation studies on certain parameters such as the step size used for IMU inference, the grid size of the state space discretization and the buffer size would provide interesting insights on generalizability. Finally, a comparison with batch-based methods or filtering methods, using the same grid-based state space discretization, would be a fairer comparison than the considered baseline methods which do not integrate movement estimates.

Second, a limitation of the proposed method is that the constructed features make use of generative models for all measurements, relying on parameters that, for different applica-

3. Multi-modal indoor localization

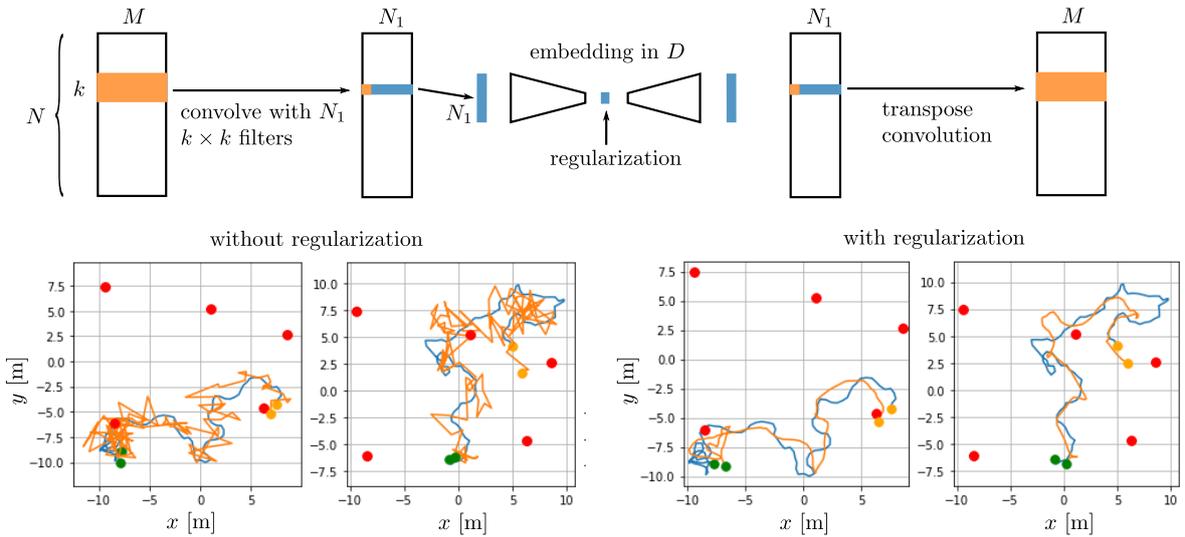


Figure 3.14.: *Visualization of auto-encoder-based approach. Rather than using forward models, we input the M -dimensional measurement vectors at N positions into an auto-encoder framework. By restricting the information bottleneck to D dimensions, we incentivise the network to learn the position of the device (top schematic). The bottom left plots show the positioning results for two example trajectories without regularization. To improve the consistency, we penalize the difference between two subsequent positions during training, which yields significantly smoother estimates (bottom right plot).*

tions, might be hard or impossible to measure. The use cases of our research were mostly museums and retail, where image locations are anyways required for successful operation of the AR software, therefore online calibration comes at no additional cost. For a more generalizable solution, it would be desirable to drop such assumptions. In the following sections, three possible extensions of this work, which relax these assumptions, are discussed.

3.6.1. Auto-encoder solution

This solution was explored in a semester project⁴. The idea is to learn the forward models, including their parameters such as the anchor locations, in an auto-encoder framework, as follows. We train a network end-to-end, where the input vector contains $M = 15$ WiFi and Bluetooth features, consisting of RTT, RSSI and the standard deviation of RSSI measurements for each anchor. By reducing the information bottleneck to dimension $D = 2$, we show that even a simple network (one hidden layer of size 12 with *ReLU* activations) learns a reasonable mapping from features to position, without explicitly imposing it. In Figure 3.14, bottom left, we show simulation results using this simple architecture, where we see that the estimated (orange) trajectory is indeed close to the real (blue) one.

However, the estimate lacks smoothness, which is due to the fact that each inference depends on one measurement vector only. We thus include an additional convolutional layer, which combines multiple features over time with $N_1 = 12$ squared filter of size $k = 11$. An equivalent transpose convolution is added in the decoder stage. Since the network considers

⁴The project was conducted by Pedro Torres Da Cunha and co-supervised by Frederike Dümbsgen and Sepand Kashani.

multiple features over time, we can add a regularization term at the bottleneck during the training phase, penalizing deviations of first-order path differences from relative movement estimates provided by IMU. To anchor the trajectory, we also assume, during training phase that the initial position is known. This network, depicted on top of Figure 3.14, is retrained end-to-end, and the positioning results are both smooth and accurate, as shown in the bottom-right plots of Figure 3.14.

More work is required to turn this prototype into a functional system, such as incorporating missing measurements and extending results to real data, but these first results suggest that this is a promising avenue.

With conditional random fields, we have explored a discriminative model for localization. Since measurements in indoor environments such as WiFi and Bluetooth exhibit high levels of noise, we can hope for at most meter-level precision, which means grid-based representations can be limited to this resolution and are thus cheap to implement. Since we rely on fixed anchor positions and since localization happens in a confined space, a fixed map is appropriate, and information such as wall and door locations can be easily incorporated. Rather than learning features from data, we show how to construct features from generative measurement models, thus somewhat blurring the lines between generative and conditional models. Still, the work visualizes how a discriminative model can do everything a generative model can do. In addition to that, our method could be easily extended with features that incorporate more than the one measurement, or features that do not explicitly rely on the forward measurement model.

We have presented preliminary results of an auto-encoder-based solution, which overcomes the need for prior knowledge of anchor locations and online calibration. Compared to the model-based approach, this approach however suffers from similar shortcomings as the fingerprinting methods: when the environment changes significantly, retraining is required. Furthermore, it is not clear how the method can cope with missing or additional measurements. Below, we present a different method, called “non-visual teach and repeat”, which is more robust to missing or additional features.

3.6.2. Non-visual teach and repeat

The above approach aims at learning a mapping from features to a global position. In a vast body of work [67, 161, 213], it was shown that for navigation in predefined environments, robust and efficient solutions can be obtained by constructing local maps and localizing the robot against these local maps. These methods use visual or lidar features, and go under the umbrella term “Visual teach and repeat”, alluding to the fact that the robot is first driven manually on safe target paths (the “teaching phase”) and then drives the same paths autonomously (“repeat”). Applied to the indoor localization problem, the advantage of such methods as opposed to fingerprinting methods is that they do not require any ground truth positioning. Furthermore, a lot of work has gone into making features robust to changing environments and missing measurements, of which the indoor localization solution would benefit as well. Concretely, WiFi and Bluetooth measurements could be integrated into local

3. Multi-modal indoor localization

feature maps, during various walks of the user through the desired environment. Just like the auto-encoder solution, this method does not require prior knowledge of anchor parameters.

3.6.3. Conditional neural fields

Conditional random fields are discriminative models that do not require generative measurement models by design. Instead of creating feature functions related to the underlying physical processes, the feature functions can be learned. A solution conceptually closer to CRFs than the proposed auto-encoder solution, is the so-called Conditional Neural Field (CNF) [163], where forward models are modelled by neural networks, but the inference process otherwise stays the same as for CRFs. Such methods have been used for monocular depth image prediction, for instance, where unary and pairwise features are learned, modeling depth from superpixels and consistency across superpixels, respectively [122]. In [12], a CNF is proposed to learn structured regression, *i.e.* regression with consistency across outputs over time or space. To the best of our knowledge, such methods have not yet been successfully applied to state estimation for robotics or non-visual measurement functions.

4 | Dynamical range-based localization

If you are not curious about the answer to a question,
the question is not worth your effort.
— Prof. May-Britt Moser, at a presentation at EPFL.

Given sparse range measurements from multiple anchors, a robot can calculate its position through multilateration. While position recovery guarantees exist for traditional lateration in static setups, the problem is less understood when the robot is moving. Using traditional lateration, a static device needs at least $D + 1$ distance measurements to localize itself. In many applications, it is unrealistic to impose these requirements; for example when a device is moving and can only measure one distance at a time. To date, practical systems thus predominately recover trajectories by coupling partial lateration with filtering techniques [201]. While these approaches lead to good performance, they offer little hope of providing fundamental guarantees for the recovery of the robot’s continuous trajectory. When is it possible to uniquely recover the trajectory, even though the device can not localize itself at any individual position?

We answer this question in the setting where a moving robot obtains range measurements from static and known anchors. We do not require the measurements to be perfectly synchronized, nor to be uniformly distributed in time. We make the realistic assumption that the robot can only measure one range at a time and limit ourselves to smooth trajectories—in particular, we focus on bandlimited and polynomial trajectories. Figure 4.1 visualizes the

This chapter is based on the publication:

M. Pacholska, F Dümbsgen, and A. Scholefield. “Relax and Recover : Guaranteed Range-Only Continuous Localization”. In: IEEE Robotics and Automation Letters 5.2 (2020), pp. 2248–2255

Detailed contributions: AS designed research; MP, FD and AS designed the algorithm; FD and MP wrote the code base; FD performed literature review and worked on real data; MP proved algorithm properties, and analyzed simulations; FD performed extension to splines and probabilistic treatment; FD, MP and AS wrote the paper.

The code to reproduce the results of this chapter is available at github.com/lcav/continuous-localization.

4. Dynamical range-based localization

Notation for this chapter

K	trajectory complexity
$\mathbf{c}_k \in \mathbb{R}^D$	k -th trajectory coefficients
$\mathbf{C} \in \mathbb{R}^{K \times D}$	trajectory coefficient matrix
$\mathbf{L} \in \mathbb{R}^{K \times K}$	quadratic form of \mathbf{C} : $\mathbf{L} = \mathbf{C}^\top \mathbf{C}$
$\mathbf{f}(t) \in \mathbb{R}^K$	trajectory basis vector (polynomial or bandlimited)
$\mathbf{A} \in \mathbb{R}^{N \times (KD + K^2)}$	measurement matrix $\mathbf{A} = [\mathbf{T}_A \mathbf{T}_F]$
$\mathbf{U}\Sigma\mathbf{V}$	SVD of \mathbf{T}_F
$\mathbf{b} \in \mathbb{R}^N$	measurement vector

limits of traditional lation, and shows where our contribution comes in. It is straightforward to see that traditional lation cannot provide us with recovery algorithms for the moving device, and surely not with uniqueness guarantees. Trajectory estimation algorithms treating such cases provide either a probabilistic or deterministic description of the continuous trajectory, but no guarantees for perfect recovery.

In this chapter, we first obtain a closed-form solution to the trajectory estimation problem, by relaxing the quadratic constraints. Then, by studying the obtained linear system, we deduce necessary and sufficient conditions for trajectory recovery of the relaxed problem. This also provides a sufficient condition for trajectory recovery of the original (non-relaxed) problem.

4.1. Problem setup

We are seeking to recover the continuous trajectory $\mathbf{r}(t)$ of a moving device (e.g a robot), for t in some given interval, $t \in \mathcal{I} \subset \mathbb{R}$. We assume that the trajectory coordinates belong to some K -dimensional linear space of functions \mathcal{F} :

$$\mathbf{r}(t) = \sum_{k=0}^{K-1} \mathbf{c}_k f_k(t), \quad (4.1)$$

where $\{f_k : k = 0, \dots, K - 1\}$ is a basis for \mathcal{F} , and the vectors $\mathbf{c}_k \in \mathbb{R}^D$ are the multidimensional basis coefficients.

At a set of time instances $\{t_n : n = 0, \dots, N - 1\}$, $t_n \in \mathcal{I}$, we measure the distance from the robot's (unknown) position $\mathbf{r}_n := \mathbf{r}(t_n)$ to one of M fixed anchors. We denote the anchor positions by $\mathbf{a}_m \in \mathbb{R}^D$, $m = 0, \dots, M - 1$, and assume that they are known. The distances are thus $d_n = \|\mathbf{r}_n - \mathbf{a}_{m_n}\|$, where $\|\cdot\|$ is the Euclidean norm and m_n is the index of the anchor used at time n . In practice, we assume that we can measure distances \tilde{d}_n corrupted

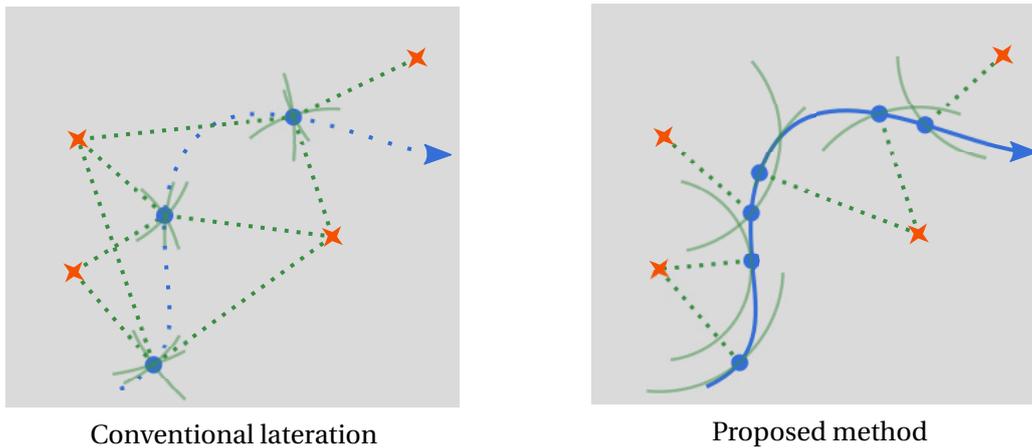


Figure 4.1.: *Two different approaches for recovering a trajectory $r(t)$ (blue) from distance measurements (green) to anchors a_m (orange). In conventional lateration, we recover single points at which we have at least $D + 1$ distance measurements. The proposed method recovers the continuous representation of the trajectory $r(t)$ from non-synchronized measurements.*

by additive zero-mean Gaussian noise: $\tilde{d}_n = d_n + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$.¹

For ease of analysis, we assume that each t_n is different; in fact, this is a strength of our formulation as it means measurements from different anchors are not assumed to be synchronized. However, since the t_n are real numbers, two consecutive t_n can be arbitrarily close.

In this work, we focus on bandlimited functions and polynomials. Both these models can approximate naturally occurring trajectories well. For example, bandlimited trajectories describe the oscillatory motion of a body around a stationary point. Polynomials cover constant-speed motion ($K = 1$), constant acceleration (e.g free fall, $K = 2$) and linearly changing acceleration ($K = 3$). For more complex trajectories, polynomials can be used as the building block for splines, as discussed in Section 4.5.1. The core problem of continuous trajectory recovery from distance measurements is formalized below.

Problem 2. *Given a set of distance measurements d_n , for $n = 0 \dots N - 1$ and corresponding times t_n , find the device's parametric trajectory $r(t)$.*

4.2. Related work

Basic concepts for range-only localization A core concept of many range-based localization algorithms is lateration, or how to estimate an object's location from distances to anchor points of known position. This problem can be tackled from two perspectives: First, the maximum likelihood estimator (MLE) of the device's position at time instant t_n , given

¹Note that, it is easy to extend the noise model to differing variances for each distance. However, we chose to present the simplest case of additive i.i.d. noise as the alternative results in additional notational overhead for concepts that are well known. In the case of correlated measurements, one can write an analogue of Theorem 1; however, the equations become more complex and Algorithm 1 needs to be changed.

4. Dynamical range-based localization

normally-distributed noisy distance measurements, is given by the solution to the following optimization problem:

$$\arg \min_{\hat{\mathbf{r}}_n \in \mathbb{R}^D} \sum_{n=0}^{N-1} \left(\tilde{d}_n - \|\hat{\mathbf{r}}_n - \mathbf{a}_{m_n}\| \right)^2. \quad (4.2)$$

This problem is called the range-based least squares (RLS) [18] problem. It is common to square the two terms inside the brackets, leading to the SMLS problem:

$$\arg \min_{\hat{\mathbf{r}}_n \in \mathbb{R}^D} \sum_{n=0}^{N-1} \left(\tilde{d}_n^2 - \|\mathbf{a}_{m_n}\|^2 + 2\mathbf{a}_{m_n}^\top \hat{\mathbf{r}}_n - \|\hat{\mathbf{r}}_n\|^2 \right)^2. \quad (4.3)$$

While both RLS and SMLS problems are non-convex, they are not equally difficult to solve. In particular, no known algorithm is guaranteed to find the optimal solution of (4.2) but (4.3) can be solved optimally [18]. However, the SMLS solution is not the MLE. To get closer to the MLE, it is common to use a standard non-linear least-squares solver such as Levenberg-Marquardt (LM) [145], which can recover a local minimum of (4.2) [175].

Non-parametric trajectory recovery Using the above methods, a moving object can only be localized at discrete time instances. This imposes a strong requirement on the number of measurements available at each such time instant, and does not ensure consistency between subsequent position estimates.

Numerous algorithms solve these two issues by combining range measurements with movement estimates from IMUs in range-only adaptations of the filtering methods described in Section 2.3 [23, 139, 193, 201, 204]. Note that the obtained accuracy depends strongly on the sampling rate at which position updates can be computed, and problems can arise when IMU measurements are delivered at a much higher frequency than other modalities [68].

Sampling rate issues can be solved by continuous-time non-parametric models. A widely used approach [15, 55, 203] is to impose time consistency using Gaussian processes. Numerous research efforts have been invested to make these computationally expensive methods more efficient, using for example Bayes trees for incremental reordering and just-in-time linearization [102].

Parametric trajectory recovery As opposed to the previously discussed methods, in this chapter, we aim to recover a parametric model of the robot's position. A number of other works have been proposed to this end, predominantly using splines. A comprehensive review of this field is given in [68]. Li *et al.* [117] solve the classical SLAM problem, replacing the position update of the usual state-space equations with a continuous, parametric trajectory. The authors consider polynomial basis functions identical to ours, and update the coefficients over sliding time windows. They use a standard iterative solver which in general converges to a local minimum. Other methods [56, 68, 146] solve the same trajectory estimation problem, parametrizing the trajectory with B-spline basis functions. As B-splines have local support, they automatically offer more flexibility in fitting complex trajectories

without recursively updating the coefficients. However, these papers lack in optimality guarantees since Gauss-Newton solvers are used. As opposed to methods solving the more general SLAM framework with arbitrary measurement modalities, we show that, by focusing on range measurements, a closed-form solution and recovery guarantees can be deduced.

Recently, trajectory estimation has been integrated in the traditional EDM framework with so-called *Kinetic EDMs* [200], where all points are considered to move on trajectories. In a similar spirit, this chapter extends the traditional lateration framework to a single device moving on a trajectory and measuring ranges from fixed anchors.

4.3. Relax and Recover

4.3.1. Relaxation

In this section, we give an outline of our recovery algorithm. We first reformulate (4.3) to include the trajectory model (4.9). Then, we relax the problem by reformulating it into a linear system of equations that can be solved with any linear solver. Finally, we provide intuition of why $O(K)$ measurements are enough to recover C .

For the space of bandlimited functions, we define the basis functions for odd K as

$$f_k(t) = \begin{cases} 2 \cos(2\pi kt/\tau) & \text{for } k \text{ odd,} \\ 2 \sin(2\pi kt/\tau) & \text{for } k \text{ even, } k > 0, \\ 1 & \text{for } k = 0, \end{cases}$$

where τ is the fixed period of the trajectory. For the space of polynomials we simply use the monomial basis $f_k(t) = t^k$.

We can now reformulate (4.3) in terms of the coefficients c_k . By setting $\mathbf{f}_n := [f_0(t_n) \dots f_{K-1}(t_n)]^\top$ and $\mathbf{C} = [\mathbf{c}_0 \dots \mathbf{c}_{K-1}] \in \mathbb{R}^{D \times K}$, we can express the sampled positions in matrix form: $\mathbf{r}_n = \mathbf{C} \mathbf{f}_n$. The distances thus become $d_n = \|\mathbf{C} \mathbf{f}_n - \mathbf{a}_{m_n}\|$ and we can reformulate (4.3) as

$$\begin{aligned} \arg \min_{\hat{\mathbf{C}} \in \mathbb{R}^{D \times K}} \sum_{n=0}^{N-1} \left(\tilde{d}_n^2 - \|\mathbf{a}_{m_n}\|^2 + 2\mathbf{a}_{m_n}^\top \hat{\mathbf{C}} \mathbf{f}_n - \mathbf{f}_n^\top \hat{\mathbf{L}} \mathbf{f}_n \right)^2, \\ \text{s.t. } \hat{\mathbf{L}} = \hat{\mathbf{C}}^\top \hat{\mathbf{C}}. \end{aligned} \quad (4.4)$$

Here, \mathbf{L} is introduced to separate terms linear in \mathbf{C} from those quadratic in \mathbf{C} . We will refer to (4.4) as the *SRLS trajectory recovery problem*. It is common to use semidefinite relaxations to make problems like (4.4) convex [128]. We take a simpler approach and discard the constraint entirely, producing what we refer to as the *relaxed SRLS trajectory*

4. Dynamical range-based localization

recovery problem.²

Solving this relaxed problem is actually equivalent to solving a system of linear equations. To see this, let us introduce the vectorized forms $\text{vec}(\mathbf{C})$ and $\text{vec}(\mathbf{L})$. Since $\mathbf{a}_{m_n}^\top \mathbf{C} \mathbf{f}_n$ is a scalar, it is equal to its trace and thus

$$\mathbf{a}_{m_n}^\top (\mathbf{C} \mathbf{f}_n) = \text{tr}(\mathbf{a}_{m_n} \mathbf{f}_n^\top \mathbf{C}^\top) = \text{vec}(\mathbf{a}_{m_n} \mathbf{f}_n^\top)^\top \text{vec}(\mathbf{C}),$$

where the first equality comes from the commutativity of the trace, $\text{tr}(\mathbf{A}^\top \mathbf{B}) = \text{tr}(\mathbf{A} \mathbf{B}^\top)$, and the second from the fact that $\text{tr}(\mathbf{A}^\top \mathbf{B}) = \text{vec}(\mathbf{A})^\top \text{vec}(\mathbf{B})$. Similarly,

$$\mathbf{f}_n^\top \mathbf{L} \mathbf{f}_n = \text{vec}(\mathbf{f}_n \mathbf{f}_n^\top)^\top \text{vec}(\mathbf{L}).$$

Using this notation, the cost in (4.4) is minimized by the \mathbf{C} and \mathbf{L} satisfying the following system of equations:

$$\tilde{b}_n = \left[\text{vec}(\mathbf{a}_{m_n} \mathbf{f}_n^\top)^\top \quad \text{vec}(\mathbf{f}_n \mathbf{f}_n^\top)^\top \right] \begin{bmatrix} \text{vec}(\mathbf{C}) \\ \frac{1}{2} \text{vec}(\mathbf{L}) \end{bmatrix},$$

where $\tilde{b}_n := \frac{1}{2} (\|\mathbf{a}_{m_n}\|^2 - \tilde{d}_n^2)$. By concatenating the above equations, we obtain

$$\tilde{\mathbf{b}} = \left[\mathbf{T}_A \quad \mathbf{T}_F \right] \begin{bmatrix} \text{vec}(\mathbf{C}) \\ \frac{1}{2} \text{vec}(\mathbf{L}) \end{bmatrix} \Bigg\}_{DK+K^2}, \quad (4.5)$$

where the rows of $\mathbf{T}_A \in \mathbb{R}^{N \times DK}$ are $\text{vec}(\mathbf{a}_{m_n} \mathbf{f}_n^\top)^\top$ and the rows of $\mathbf{T}_F \in \mathbb{R}^{N \times K^2}$ are $\text{vec}(\mathbf{f}_n \mathbf{f}_n^\top)^\top$.

At first sight, it appears that, by additionally solving for \mathbf{L} , the relaxation has introduced K^2 new variables (or $K(K+1)/2$ if we enforce \mathbf{L} 's symmetry). However, as we will formally prove in Section 4.3.2, the required effective increase is only of order K .

To see this, consider applying a Singular Value Decomposition (SVD) to \mathbf{T}_F : $\mathbf{T}_F = \mathbf{U} \mathbf{\Sigma} \mathbf{V}$. Here, $\mathbf{\Sigma}$ is an $r \times r$ diagonal matrix, with r the rank of \mathbf{T}_F . Then, (4.5) becomes

$$\mathbf{b} = \left[\mathbf{T}_A \quad \mathbf{U} \mathbf{\Sigma} \right] \begin{bmatrix} \text{vec}(\mathbf{C}) \\ \frac{1}{2} \mathbf{V}^\top \text{vec}(\mathbf{L}) \end{bmatrix} \Bigg\}_{DK+r}. \quad (4.6)$$

Therefore, if we have at least $DK + r$ measurements such that $\left[\mathbf{T}_A \quad \mathbf{U} \mathbf{\Sigma} \right]$ is full-rank, we can recover \mathbf{C} with any linear solver. This gives rise to a simple recovery algorithm, which is summarized in Algorithm 1. Note that, before solving the linear system, we check invertibility

²The same relaxation is also discussed in [18] for the point-wise SRLS solution, under the name of ‘‘unconstrained squared-range-based least squares’’. However, the results from their algorithm are inaccurate. It seems that combining measurements from a series of positions, as done in this thesis, is crucial for a good functioning of the method.

Algorithm 1: Trajectory Reconstruction

Data: Anchor coordinates \mathbf{a}_m , distance measurements \tilde{d}_n , times and anchor indices t_n, m_n

Result: Trajectory coefficients \hat{C} , empty if not unique

$\mathbf{f}_n \leftarrow [f_0(t_n) \ \dots \ f_{K-1}(t_n)]$;

set up $\mathbf{T}_A, \mathbf{T}_F, \tilde{\mathbf{b}}$ as in (4.5);

$\hat{C} \leftarrow []$;

if conditions (4.7) and (4.8) are satisfied then

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \text{SVD}(\mathbf{T}_F)$;

$\mathbf{A} \leftarrow \text{concatenate}(\mathbf{T}_A, \mathbf{U}\mathbf{\Sigma})$;

$\hat{\mathbf{x}} \leftarrow \text{linsolve}(\mathbf{A}, \tilde{\mathbf{b}})$;

$\hat{C} \leftarrow \text{reshape}(\hat{\mathbf{x}}[1:DK], (D, K))$;

end

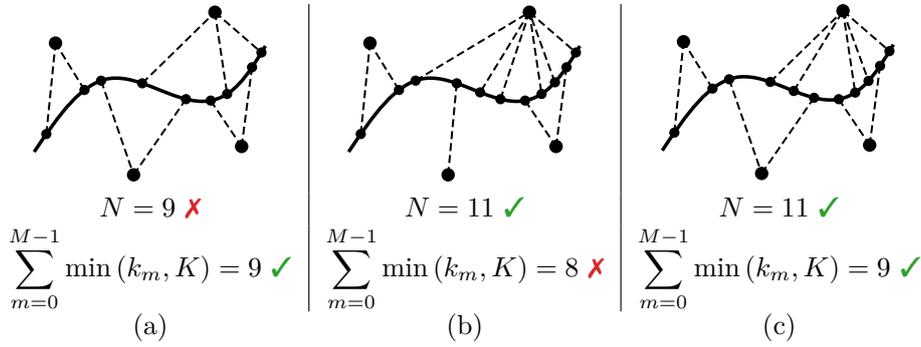


Figure 4.2.: Examples of sufficient and insufficient measurements, with model degree $K = 3$ and embedding dimension $D = 2$; i.e., $K(D+2) - 1 = 11$ and $K(D+1) = 9$. (a) Condition (4.8) is satisfied, but Condition (4.7) is not. (b) Condition (4.7) is satisfied, but Condition (4.8) is not because too many measurements involve the same anchor. (c) Both conditions are satisfied and recovery is guaranteed.

using the conditions derived in the next section.

In the polynomial case, it is easy to see that $r < 2K$. In particular, in this case, each row of \mathbf{T}_F contains the vectorized version of samples of the outer product of a polynomial of degree $K - 1$ with itself. The columns of \mathbf{T}_F are thus samples of polynomials of degrees up to $2K - 2$, and so they span (at most) a $2K - 1$ dimensional space. Similar reasoning can be applied for bandlimited functions by treating them as elements of a polynomial ring, see Observation 1 from [155]. Therefore, the rank of the whole matrix is $O(K)$ and the computational complexity of Algorithm 1 is $O(N^2K + K^4)$.

4.3.2. Recovery

In this section, we prove conditions for when the linear system corresponding to the relaxed SRLS trajectory recovery problem can be solved for a unique C . In the noiseless case, these conditions are sufficient for perfect trajectory recovery. To see this, note that, in the

4. Dynamical range-based localization

noiseless case, any minimizer of (4.4) is also a minimizer of (4.6); therefore, if the recovered \hat{C} from (4.6) is unique, it is also the unique minimizer of (4.4) (assuming (4.4) is solvable). In addition, in the noiseless case, the unique solution of the SMLS problem is also the unique solution of the RLS problem.

When there is noise on the distances, we do not obtain perfect reconstruction but the conditions for the invertibility of the linear system are still valid. This is because the matrix that is inverted does not depend on measured distances.

The main theoretical contribution of this chapter is the following theorem.

Theorem 1. *Let \hat{C} satisfy the relaxed SMLS trajectory recovery problem. Given N measurements from non-degenerate anchors at random times t_0, \dots, t_{N-1} sampled from a continuous distribution on \mathcal{I}^N , \hat{C} is unique and can be recovered with Algorithm 1 if*

$$N \geq K(D + 2) - 1, \quad (4.7)$$

and

$$\sum_{m=0}^{M-1} \min(k_m, K) \geq K(D + 1), \quad (4.8)$$

where k_m is the number of measurements in which the m -th anchor is used. Furthermore, in the noiseless case, $\hat{C} = C$ and perfect trajectory recovery is achieved. Finally, if (4.8) is not satisfied, then \hat{C} is not unique.

Here, we call a set of anchors *non-degenerate* if no $D + 1$ anchors lie on the same affine subspace. This assumption is only slightly stronger than the common requirement that not all anchors lie on the same affine subspace. Randomly placed anchors will satisfy this condition almost surely.

The assumption that times follow a continuous distribution is in place to ensure that the times are not adversarial, in particular that the functions f_k are unlikely to be zero. Note that the times t_n do not have to be independent.

The proof of Theorem 1 can be found in our paper [155]. Here, we try to gain some more intuition of the theorem's meaning. Condition (4.7) formalizes that $O(KD)$ measurements are sufficient to localize. Even without the relaxation, one would expect to need $K(D + 1)$ measurements for recovery. Indeed, to recover a trajectory of complexity K , one can independently localize K points along it, and to localize a single point, we need $D + 1$ distance measurements.

Condition (4.8) describes how measurements cannot be arbitrarily distributed between anchors. In particular, if an anchor provides more than K measurements, only the first K have an effect on uniqueness, see the examples in Figure 4.2. Moreover, unique recovery is not possible with measurements from less than $D + 1$ anchors..

A natural question to ask is, how likely it is to obtain a measurement set sufficient to recover

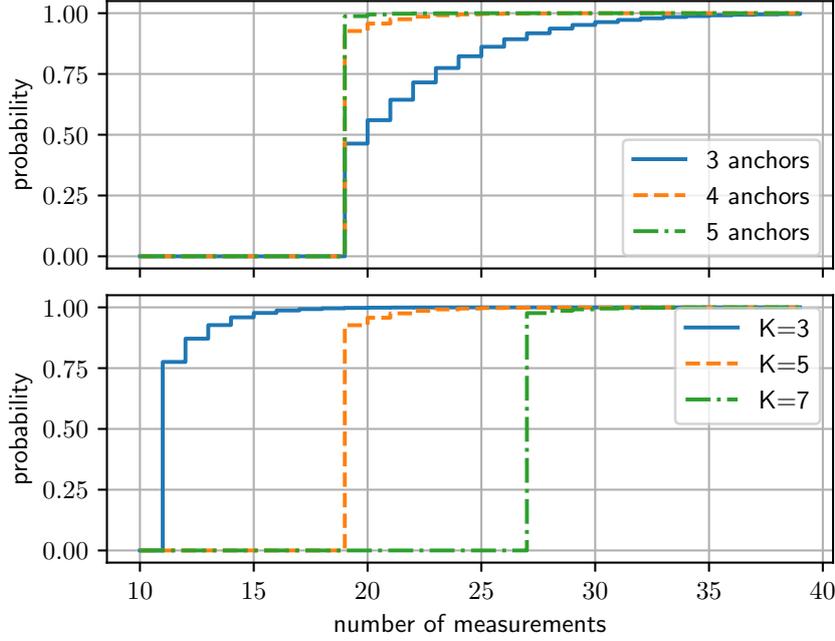


Figure 4.3.: Probability of recovering C , with dimension $D = 2$. Upper plot: the trajectory degree is set to $K = 5$ and the number of anchors vary. We observe that the probability of recovering C grows with the number of anchors. This is because, the more anchors we use the higher the chance that the measurements spread uniformly and satisfy (4.8). Lower plot: the number of anchors is set to $M = 4$ and the trajectory degree varies. For Algorithm 1, (4.7) states that the minimum number of required measurements is 11, 19 and 27 for trajectory complexities of 3, 5 and 7, respectively.

C . Unfortunately, the probability of the set of measurements satisfying Theorem 1 does not seem to have a closed form formula. Fortunately, it depends only on the partition of measurements between the M anchors, so it can be easily calculated numerically by counting partitions that satisfy (4.8), see Figure 4.3. Clearly, the probability of recovering C is non decreasing, because adding a new measurement can only increase the rank of $\begin{bmatrix} T_A & T_F \end{bmatrix}$. In practice, we observe that the probability is already large for $K(D + 2) - 1$ measurements, and grows with the number of measurements to 1.

Note that, in practice, the matrix might be ill-conditioned when the anchors are almost co-linear, or many measurements to the same anchor are taken at almost the same time.

4.3.3. Statistical analysis

We study the reconstruction error and covariance for the proposed method. This allows us not only to understand the behavior of the proposed algorithm, but the covariance can come in handy when the algorithm is integrated in a probabilistic filtering scheme (see Section 2.3).

We denote by $e_n := r_n - \hat{r}_n$ the reconstruction error. For convenience, we use the following,

4. Dynamical range-based localization

form for the trajectory parametrization:

$$\mathbf{r}_n = \underbrace{\mathbf{I} \otimes \mathbf{f}_n^\top}_{\mathbf{F}_n \in \mathbb{R}^{D \times DK}} \underbrace{\text{vec}(\mathbf{C})}_{\mathbf{c} \in \mathbb{R}^{DK}} = \mathbf{F}_n \mathbf{c}, \quad (4.9)$$

with \otimes the Kronecker product. Similarly, we can write $\hat{\mathbf{r}}_n = \mathbf{F}_n \hat{\mathbf{c}}$ for the estimated position. Note that we assume that the trajectory model is correct for this analysis. Finally, we introduce the matrix $\mathbf{B} := \mathbf{N} [\mathbf{T}_A | \mathbf{U}\boldsymbol{\Sigma}]^\dagger \in \mathbb{R}^{DK \times N}$, with $\mathbf{N} = [\mathbf{I}_{DK} | \mathbf{0}_{DK \times r}]$ such that $\hat{\mathbf{c}} = \mathbf{B}\tilde{\mathbf{b}}$ and $\mathbf{c} = \mathbf{B}\mathbf{b}$, with \mathbf{b} the noiseless version of $\tilde{\mathbf{b}}$.

With this, we are equipped to calculate the mean and covariance of the reconstruction error e_n . First, we rewrite e_n :

$$e_n = \mathbf{F}_n \mathbf{B} (\tilde{\mathbf{b}} - \mathbf{b}) = -\frac{1}{2} \mathbf{F}_n \mathbf{B} (\mathbf{d}^2 - \tilde{\mathbf{d}}^2) = -\frac{1}{2} \mathbf{F}_n \mathbf{B} (2\mathbf{d} \circ \boldsymbol{\epsilon} + \boldsymbol{\epsilon}^2), \quad (4.10)$$

where all square operations are element-wise, and \circ is the element-wise product. Recall that $\boldsymbol{\epsilon}$ is the vector of i.i.d. Gaussian measurement noise of variance σ . The mean of e_n is then given by:

$$\boldsymbol{\mu} = \mathbb{E}\{e_n\} = -\frac{1}{2} \mathbf{F}_n \mathbf{B} \mathbb{E}\{2\mathbf{d} \circ \boldsymbol{\epsilon} + \boldsymbol{\epsilon}^2\} = -\frac{1}{2} \mathbf{F}_n \mathbf{B} \boldsymbol{\sigma}^2, \quad (4.11)$$

where $\boldsymbol{\sigma}$ is the vector of all σ . Therefore, the covariance matrix is given by

$$\begin{aligned} \boldsymbol{\Sigma} &= \mathbb{E}\{(e_n - \boldsymbol{\mu})(e_n - \boldsymbol{\mu})^\top\} \\ &= \frac{1}{4} \mathbf{F}_n \mathbf{B} \mathbb{E}\{(2\boldsymbol{\epsilon} \circ \mathbf{d} + \boldsymbol{\epsilon}^2 - \boldsymbol{\sigma}^2)(2\boldsymbol{\epsilon} \circ \mathbf{d} + \boldsymbol{\epsilon}^2 - \boldsymbol{\sigma}^2)^\top\} \mathbf{B}^\top \mathbf{F}_n^\top \\ &\stackrel{(a)}{=} \mathbf{F}_n \mathbf{B} \text{diag}(\mathbf{d}^2 \circ \boldsymbol{\sigma}^2 + \boldsymbol{\sigma}^4/2) \mathbf{B}^\top \mathbf{F}_n^\top, \end{aligned} \quad (4.12)$$

where for clarity, we move the steps omitted in (a) to Appendix 4.A.

A few remarks are due regarding the error statistics. First, we see that our estimator is biased. This is a consequence of squaring the distances, and is the price to pay in exchange for a more efficient recovery algorithm. Since the error is proportional to $\boldsymbol{\sigma}^2$, this effect is expected to be detrimental for higher errors.

Secondly, we note that the estimator variance grows with the distance. This is in accordance with our intuition that, everything else fixed, estimation is more difficult as distances to anchors grow. A more in-depth study of the shape of the covariance depending on the anchor locations and chosen basis functions is an interesting question for future research.

4.4. Results

In this section, we compare the performance of our method with different localization algorithms on simulated and real data. As mentioned before, we assume additive Gaussian noise on distance measurements $\tilde{d}_n = d_n + \epsilon_n$, where ϵ_n are i.i.d. random variables, $\epsilon_n \sim$

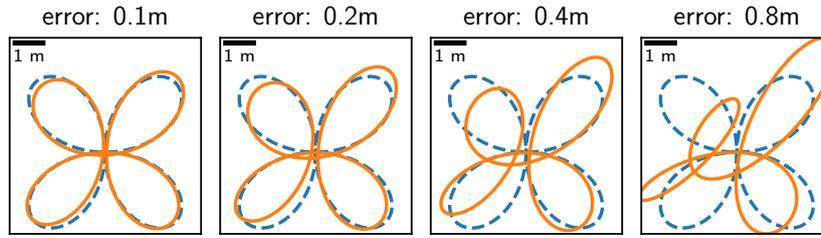


Figure 4.4.: Visualisation of the distance between trajectories. Original bandlimited trajectory of order $K = 7$ (dashed blue) and a randomly perturbed trajectory (solid orange). The Frobenius distance between the trajectory coefficients is displayed above each subplot.

$\mathcal{N}(0, \sigma)$. We approach the limit of necessary measurements, dictated by Theorem 1, both in simulation and on real data, investigating how the ideal performance degrades as we add noise.

In the noisy case, an immediate problem comes from the fact that we use squared distance measurements for trajectory recovery and the distribution of *squared* distances is both additive and multiplicative:

$$\tilde{d}_n^2 = d_n^2 + 2d_n\epsilon_n + \epsilon_n^2,$$

while our recovery method implicitly assumes only centered additive noise. Indeed, in Algorithm 1 we solve a linear system of equations, and we obtain a solution that would be a MLE if the Gaussian noise was added to the squared distances. To alleviate this problem, we propose a Weighted Least Squares (WLS) approach described below.

If the noise is small, the ϵ_n^2 term is negligible. The remaining noise has distribution $2d_n\epsilon_n \sim \mathcal{N}(0, 2d_n\sigma)$. If we knew d_n , we could use WLS, with weights $1/d_n$, to bring the system back to an i.i.d. noise model. Since we do not know the distances, we use the *measured* distances \tilde{d}_n , and obtain a noisy version of the n -th row of (4.5):

$$\frac{d_n}{\tilde{d}_n}\epsilon_n + \frac{1}{\tilde{d}_n} \begin{bmatrix} \mathbf{T}_A & \mathbf{T}_F \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{C}) \\ \frac{1}{2} \text{vec}(\mathbf{L}) \end{bmatrix} = \frac{\tilde{b}_n}{\tilde{d}_n}.$$

Again, assuming that noise is small, d_n/\tilde{d}_n is close to 1, and the noise is approximately i.i.d. In practice, to avoid dividing by extremely small numbers, we can add some small regularisation γ to the distance and divide by $\tilde{d}_n + \gamma$.

4.4.1. Noise study in simulation

In this section, we report the root squared error E of reconstructing \mathbf{C} :

$$E(\mathbf{C}, \hat{\mathbf{C}}) = \|\mathbf{C} - \hat{\mathbf{C}}\|_F, \quad (4.13)$$

where $\|\cdot\|_F$ is the Frobenius norm. For the chosen basis of bandlimited functions, this norm is equivalent to the power of the signal: using power as opposed to energy makes errors comparable between trajectories with different periods. For more intuition, see Figure 4.4.

4. Dynamical range-based localization

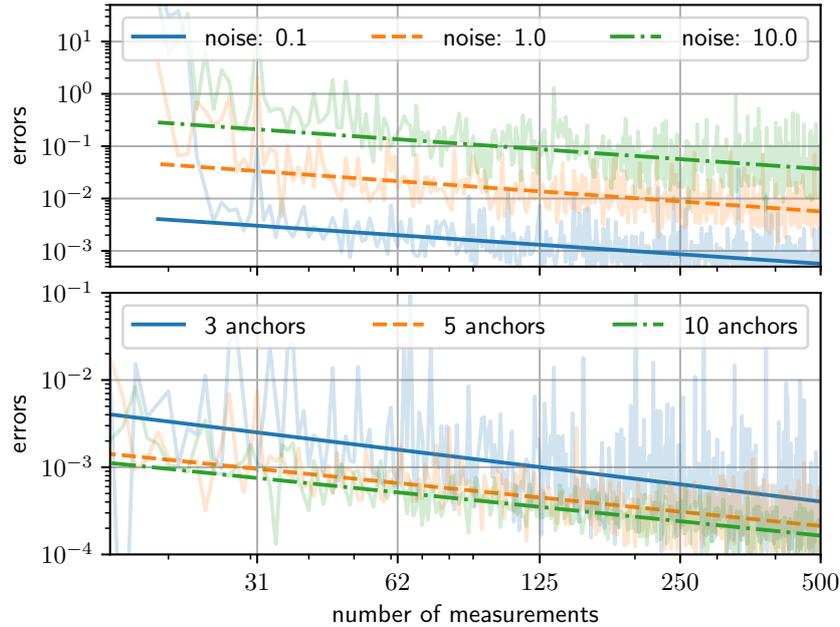


Figure 4.5.: Reconstruction from noisy measurements using the weighted system of equations. For clarity, slopes (dark lines) were fitted to the averages over 1000 simulations (light lines). The simulated distances were between 0.1 m and 10 m. Upper plot: the trajectory degree was set to $K = 5$ and $M = 4$ anchors were used; the magnitude of the noise is changing. In this setup, the minimum number of measurements required is 19. We can see that the algorithm is robust to noise starting from about $3\times$ oversampling. Lower plot: the trajectory degree was set to $K = 3$ and noise magnitude was set to $\sigma = 1$ m; the number of anchors is changing. The reconstruction error does not depend heavily on the number of anchors, but it has much higher variance for $D + 1 = 3$ anchors.

In the simulations, we take samples t_n uniformly in the interval $[0, \tau]$, and at each time we choose anchors uniformly at random. Technically speaking, this violates the assumption of the random measurement times in Theorem 1. However, the assumption mostly serves for mathematical rigour and all typical sampling schemes (random, uniform, etc.) work in practice.

When simulating different values for N , we discard some measurements uniformly at random. We fix $\tau = 2$. Figure 4.5 shows the coefficients reconstruction error obtained using the weighted reconstruction. We can see that the error decreases with oversampling, and the fitted slope is roughly -0.6 . This means that for $10\times$ oversampling we get more than a $5\times$ reconstruction improvement. The regular (non-weighted) solver performed similarly, with a smaller improvement from oversampling.

4.4.2. Results on lawn-mower dataset

We test our trajectory estimation algorithm on a real-world dataset provided by Djugash *et al.* [52]. The dataset consists of an autonomous lawnmower moving on a grass field, using ultra-wideband (UWB) signals to 4 stationary anchors for range measurements, and densely sampled kinematic GPS for ground truth. The distance measurements have an average

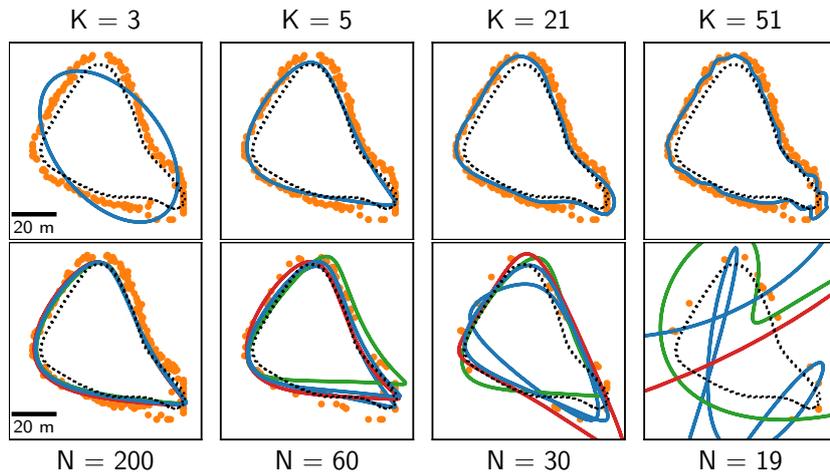


Figure 4.6.: *Reconstruction accuracy of lawnmower trajectory using period $\tau = 54$ s. In dotted black is the ground truth trajectory (from GPS), in solid lines are our reconstructions, and the orange dots show the RLS estimates. The top row shows different complexities K using all available distance measurements. In the bottom row, we fix K to 5 and drop measurements uniformly at random. The different colors correspond to different sets of measurements.*

standard deviation of ca. 0.5 m, with a tendency to overestimate [52].

We first evaluate the *Plaza2* dataset. The trajectory completed by the robot does not perfectly fit our models, but we will see that it can be approximated by the bandlimited model. We can estimate its period τ by visual inspection. Using all 499 range measurements, we use Algorithm 1 to estimate the coefficients for different complexities K , and report the obtained trajectories in the first row of Figure 4.6. We see that the trajectory is well approximated with degrees of $K = 5$ or higher.

Next, we fix K to the smallest sensible value for the given trajectory ($K = 5$) and test the performance of our algorithm when dropping measurements uniformly at random, down to the minimum number required (19 measurements). The second row in Figure 4.6 shows that the obtained reconstruction quality remains satisfactory down to 30 measurements only, and is not too sensitive to the specific distance measurements selected. As seen already in simulation, the variance of the reconstructions is higher for fewer measurements.

For comparison, we also plot localization results with the point-wise RLS method [18], using the latest distance measurements from $D + 1$ anchors. The method uses brute force on a uniform grid initialized in the bounding box of the anchors with a grid size of 0.5 m. The trajectory only starts to be recognizable from $N = 200$ measurements upwards, and the individual estimates are more noisy than ours.

Finally, we compare the provided algorithms quantitatively with standard lateration methods and solvers. Table 4.1 shows the trajectory recovery accuracy for the two UWB-based datasets from [52]. As we have seen before, the trajectory in *Plaza2* is approximately bandlimited. The trajectory in *Plaza1*, on the other hand, covers the lawn in a zig-zag manner. In practice, for such a trajectory, an iterative application of our algorithm would be appropriate. For the purpose of this chapter, we simply report the average reconstruction accuracy over

4. Dynamical range-based localization

Table 4.1.: *MSE of recovered trajectories for two real-world UWB-based localization datasets [52].*

dataset # measurements N model complexity K	bandlimited trajectory (<i>Plaza2</i>)								
	100			300			499		
	5	11	19	5	11	19	5	11	19
<i>model mismatch</i>	3.7	1.8	1.5	3.8	2.0	1.7	3.8	2.1	1.8
SRLS	137.5	141.0	114.5	22.2	22.2	21.8	13.2	13.2	13.2
SRLS <i>fitted</i>	77.9	100.5	4378.0	18.1	17.6	17.8	15.6	14.1	13.9
RLS	113.6	134.3	106.1	16.5	17.2	16.2	9.7	9.7	9.7
RLS <i>fitted</i>	62.1	91.5	5971.6	13.2	12.8	12.9	10.5	9.2	9.4
LM <i>ellipse/line</i>	12.7	147.6	339.8	11.3	25.9	13.2	11.4	11.5	12.0
LM <i>ours weighted</i>	12.7	13.0	20.5	11.3	11.9	12.2	11.4	11.5	12.0
<i>ours</i>	13.9	14.1	69.5	13.2	11.5	11.6	13.2	11.4	11.3
<i>ours weighted</i>	10.3	10.6	68.2	8.8	7.3	7.3	8.7	7.2	6.9

dataset # measurements N model complexity K	polynomial trajectory (<i>Plaza1</i>)			
	10	20	30	50
	2	2	2	2
<i>model mismatch</i>	0.3	0.3	0.4	0.7
SRLS	33.8	15.1	14.2	7.8
SRLS <i>fitted</i>	50.4	119.0	15.6	8.7
RLS	32.5	14.1	12.7	6.3
RLS <i>fitted</i>	48.5	198.8	12.9	6.7
LM <i>ellipse/line</i>	253.4	24.6	16.3	6.6
LM <i>ours weighted</i>	18.7	7.2	7.5	6.6
<i>ours</i>	51.7	4.9	4.8	4.1
<i>ours weighted</i>	5639.1	4.9	4.8	4.3

20 different linear parts of the trajectory. We compute the error by sampling the parametric model at the times at which we have ground truth measurements, compute the mean squared error (MSE) between the predicted and ground truth points, and average over 20 different realizations.

We evaluate the proposed solution with and without the weighting introduced at the beginning of this Section. For the RLS and SRLS methods, we compare both the raw laterated points and a parametric trajectory fitted to these points (RLS/SRLS fitted). The LM method reported for comparison solves (4.2) using the `scipy.optimize` [210] implementation of Levenberg-Marquardt [145] optimization. We compare two different initializations: a simple ellipse / line in the correct order of magnitude (LM *ellipse/line*) and the result of our weighted algorithm as initialization (LM *ours weighted*). The row *model mismatch* shows the error of the model fitted directly to the ground truth position measurements, and gives an upper bound on how well we can hope to do.

We can draw a number of interesting conclusions from the quantitative evaluation. First of all, we note that weighting distance measurements for real data improves the reconstruction accuracy significantly, especially for the bandlimited trajectory (left side of Table 4.1). We believe that this is due to a positive bias in the distance measurements; indeed, if the

bias is removed by an oracle, the weighted and non-weighted algorithms perform almost identically.

We further observe that the reconstruction behaves poorly when the number of measurements is close to the limit given in Theorem 1. For the bandlimited trajectory of degree $K = 19$ this limit is 75 measurements, and $N = 100$ noisy measurements are too few to accurately estimate the trajectory. For the polynomial trajectories, we observe the same behavior for $N = 10$ and $K = 2$, where $N = 10$ is close to the minimum 7.

Finally, we note that the accuracy of the LM methods depends highly on the initialization, which is a well-known limitation [175]. This shows in a big difference between the results for LM *ellipse/line* and LM *ours weighted*, which is particularly pronounced for high model orders and few measurements. Quite surprisingly, even though LM optimizes the RLS cost function and thus aims to recover the MLE estimate for zero-mean Gaussian noise on measurements, it does not compare favorably with our proposed algorithm. We believe that both the fact that the solution is only suboptimal and the bias in the distance measurements play against this method. However, we suggest that if LM is the preferred solution in a complete system for different reasons, our proposed method is an attractive candidate for an accurate initialization.

4.5. Conclusion and ongoing work

We have presented an analytical solution for trajectory recovery. Using a smooth parametrization, we can relax the problem into a linear form and solve it optimally, under certain conditions which we characterize exactly. The method performs well in simulation and on a real dataset of UWB measurements collected on a lawnmower.

In order to make the developed theory applicable to a wide range of applications, we can, on one hand, study different continuous trajectory formulations. Splines, for instance, are a natural extension of the proposed theory since the building blocks are polynomials, as discussed in the section. The fact that splines arise naturally as a solution to certain regularizations is another observation which could be explored in the context of localization Section 4.5.2. Finally, outline different trajectory and measurement models in Section 4.5.3

4.5.1. Extension to splines

One obvious limitation of the proposed solution is the necessity to choose a single parametric form for the device's movement. In the real world, movements can be complex, and vary for instance with the vehicle of choice: highly agile vehicles like drones might need higher trajectory orders than for instance cars or lawnmowers. Even for a given vehicle, the appropriate parametrization can vary over time and depend on the environment and task at hand. One important extension of the proposed solution, making the parametric form more flexible, thus consists of piecewise recovery. A natural choice for piecewise trajec-

4. Dynamical range-based localization

ries are splines, which are also explored in some of the pioneering works for batch-based solutions [68]. This extension is discussed in what follows.

Trajectory formulation using splines We denote the spline order by P , and the number of knots by H . We also introduce the (temporal) spline knots, $s_k, k = 0, \dots, H - 1$, which we assume to be fixed a priori, but not necessarily uniform. For a trajectory in D dimensions, we create D splines over the same knot sequence, one for each dimension. Denoting the vector of coefficients at each knot by $c_k \in \mathbb{R}^D$, we stack the coefficients as columns of the matrix of unknowns $C \in \mathbb{R}^{D \times H}$. Using the matrix notation as elaborated in [169], we can then express the trajectory as

$$\mathbf{r}(t) = \mathbf{C}\mathbf{f}(t), \quad (4.14)$$

where $\mathbf{f}(t) \in \mathbb{R}^H$ is the vector of spline basis functions evaluated at time t . As opposed to the previous formulation in (4.9), the vector of basis functions is sparse. In particular, the basis functions of causal splines of order P have a support of P knot intervals. This means that, for all measurements from \mathcal{T}_k , only the entries $\mathbf{f}_{k-P+1}(t)$ to $\mathbf{f}_k(t)$ may be non-zero. To make this clearer, we stack all N measurement locations in a matrix as follows:

$$\begin{bmatrix} \mathbf{r}(t_1)^\top \\ \vdots \\ \mathbf{r}(t_N)^\top \end{bmatrix} = \begin{bmatrix} \mathbf{f}(t_1)^\top \\ \vdots \\ \mathbf{f}(t_N)^\top \end{bmatrix} \mathbf{C} := \mathbf{F}\mathbf{C} \quad (4.15)$$

where we have introduced $\mathbf{F} \in \mathbb{R}^{N \times H}$, the matrix of basis function evaluations.

The matrix \mathbf{F} has a block-sparse structure, as visualized in Figure 4.7. To understand this structure, it is useful to introduce two different sets; also depicted in the figure. We denote the set of measurements which are recorded between the two spline knots k and $k + 1$ are denoted by \mathcal{T}_k :

$$\mathcal{T}_k = \{t_n | t_n \in [s_k, s_{k+1}], n = 0 \dots N - 1\}, |\mathcal{T}_k| = T_k. \quad (4.16)$$

Similarly we define the set of measurements relevant for knot s_k by

$$\mathcal{S}_k = \{t_n | t_n \in [s_k, s_{k+P-1}], n = 0 \dots N - 1\}, |\mathcal{S}_k| = S_k. \quad (4.17)$$

Observations Because the splines-based parametrization yields the same form as (4.9), we can use the same sequence of operations to reduce the trajectory recovery problem to a linear system of equations. However, because of the different form of the basis function, in particular due to their sparseness, the rank of the resulting matrix behaves differently. We have made the following observations through extensive simulations:

- The maximum rank of the matrix \mathbf{T}_b is exactly $2P - 1 + (n - 1)P$ where n is the number of intervals in which we have measurements.
- The above rank is achieved for example when we have at least P measurements in each interval, and at least $2P - 1$ in one interval.

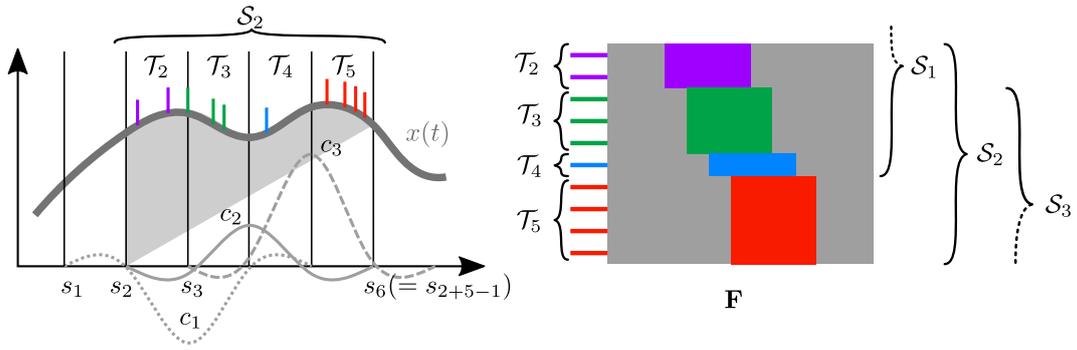


Figure 4.7.: *Left: visualization of the splines parametrization employed for the trajectory estimation. Depicted in thick gray is the device trajectory $x(t)$, which is the weighted sum of the spline basis functions, shown below. Measurement times are indicated with the colored bars. Right: the resulting block-sparse matrix F . For each time window \mathcal{T}_i , the non-zero elements correspond to the spline coefficients contributing to the positions within that window.*

- The rank of the matrix T_a is the rank of F , which is its support (number of nonzero columns) times the dimension D . This means that the maximum rank of the matrix is PD for the first interval, plus D for each newly added interval, so $PD + (n - 1)D$ in total.
- The exact rank r of T_b is

$$r = \sum_i \min(n_i, k_i P - k_i + 1) \quad (4.18)$$

where $k_i = \min(l_i, 2)$, n_i is the number of measurements in the i th block and l_i is the number of new knots these measurements expose.

Combined, the above conditions allow us to choose an appropriate parametrization given the specific application: if we have a high measurement rate, for instance, we can also choose finely spaced spline knots, while ensuring a full-rank measurement matrix. There is a trade-off to consider here: finer spline knots yield higher approximation power, but also fewer measurements per interval, thus higher sensitivity to noise. This trade-off and other questions such as the appropriate spline order and the mathematical proofs of the above observations, are left open for future work.

4.5.2. Splines as the natural solution to regularized optimization

In the above discussion, we have imposed the splines parametrization. An interesting direction of research would pose the trajectory recovery problem as a general, regularized nonlinear inverse problem, where splines arise “naturally” as a solution. A plethora of interesting findings exist for the linear case: in particular, representer theorems tell us that when regularizing the original linear inverse problem with a certain regularizer, the solution will take a certain form depending on the regularizer. For the common Tikhonov regularization, for instance, the solution can be shown to be a piecewise-cubic function

4. Dynamical range-based localization

with knots at the sampling locations, while generalized Total-Variation regularization leads to piecewise-linear functions with a lesser number of adaptive knots [86]. When we replace the linear measurement operator with a non-linear function, the analysis of solutions, in particular uniqueness and continuous dependence on data, become more complex [147, Chapter 11]. Circumventing this complexity by using the relaxation presented in this chapter could help to bridge the two research areas, bringing interesting results, such as the natural appearance of spline solutions, from linear problems to nonlinear problems.

4.5.3. Generalization to different models

The approach we take in this chapter is not restricted to the particular choice of parametrization, nor to the range-only measurement type. It would for instance be natural to explore different measurement modalities, such as angles or range-differences [196], which can also be relaxed to a linear form. As a matter of fact, any measurement modality can be relaxed to a linear form, as long as we add enough additional variables (in our case, the matrix L we introduce in (4.4)). This idea is closely linked to Koopman theory, discussed in the control of dynamical systems [31]. When such a linearization is beneficial, and whether recovery guarantees can be established, are questions for future work.

Regarding different parametrizations, a trajectory type of particular interest is the Gaussian process, as it is more flexible than the parametric models discussed here, and as it inherently models uncertainty. Recent works such as [203] and [55] present convenient starting points for adapting this trajectory model.

Appendix

4.A. Calculation of covariance matrix

We introduce $\mathbf{y} := 2\mathbf{d} \circ \boldsymbol{\epsilon} + \boldsymbol{\epsilon}^2 - \boldsymbol{\sigma}^2$ and $\mathbf{Y} := \mathbb{E}\{\mathbf{y}\mathbf{y}^\top\}$. Then we have:

$$\begin{aligned}
 Y_{ij} &= \mathbb{E}\{y_i y_j\} = \mathbb{E}\{(2\epsilon_i d_i + \epsilon_i^2 - \sigma^2)(2\epsilon_j d_j + \epsilon_j^2 - \sigma^2)\} \\
 &= \mathbb{E}\{4d_i d_j \epsilon_i \epsilon_j + 2d_i \epsilon_i \epsilon_j^2 - 2d_i \epsilon_i \sigma^2 + 2d_j \epsilon_j \epsilon_i^2 \\
 &\quad + \epsilon_i^2 \epsilon_j^2 - \epsilon_i^2 \sigma^2 - 2d_j \epsilon_j \sigma^2 - \epsilon_j^2 \sigma^2 + \sigma^4\} \\
 &\stackrel{(a)}{=} 4\delta_{ij} d_i d_j \sigma^2 + (\sigma^4 + 2\delta_{ij} \sigma^4) - 2\sigma^4 + \sigma^4 \\
 &= \begin{cases} 4d_i^2 \sigma^2 + 2\sigma^4 & \text{if } i = j \\ 0 & \text{else,} \end{cases}
 \end{aligned} \tag{4.19}$$

where in (a) we have used that $\mathbb{E}\{\epsilon_i^2\} = \mathbb{E}\{\epsilon_j^2\} = \sigma^2$ and by independence: $\mathbb{E}\{(\epsilon_i \epsilon_j)^2\} = \mathbb{E}\{\epsilon_i^2\} \mathbb{E}\{\epsilon_j^2\} = \sigma^4$ if $i \neq j$ and $\mathbb{E}\{\epsilon_i^4\} = 3\sigma^4$. Using again independence and the fact that $\mathbb{E}\{\epsilon_j\} = 0$, we have $\mathbb{E}\{2d_j \epsilon_j \epsilon_i^2\} = \mathbb{E}\{2d_j \epsilon_j \sigma^2\} = 0$, and vice-versa for i .

5 | Dynamical sound-based wall localization

रसरी आवत जात ते सिल पर परत निसान

Repeated motion of the soft rope leaves a mark on the hard stone of a well.

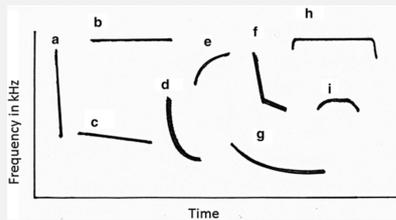
— Part of a Hindi saying by poet Kabir

Bats have fascinated scientists for centuries with their ability to “see” in the dark thanks to ultrasound. First studies conducted by Lazzaro Spallanzini and Louis Jurine in the 1790s, consisting of blindfolding or ear-plugging bats in a variety of settings, suggested that these nocturnal animals use sound more than vision to navigate, but the chirps emitted by bats could not be detected yet [65]. Maybe surprisingly, these results were challenged and people instead believed for a long time that bats used sensory feedback from tactile receptors on the wings for obstacle detection [71]. It was not until more than 150 years later that Donald Griffin detected ultrasonic pulses emitted by caged bats at Harvard university, after which he showed in a series of experiments that these signals are indeed used for echolocation [80]. This ability equips bats with a unique modality when visible information fails, which is essential for this nocturnal species. Among other use cases, bats were recently found to use ultrasound for detecting water surfaces for drinking, as covered in the box on the next page. Other animals such as mice and dolphins [92], have learned to use sound for localizing objects when visible information is not reliable. Even visually impaired humans can learn to locate, and to some extent even classify, objects, by using clicking sounds [6].

This chapter contains work conducted in collaboration with Adrien Hoffet, Mihailo Kolundžija, Isaac Bernardino Dinis, Adam Scholefield and Martin Vetterli. Detailed contributions: AS, MV and FD designed research, MK helped with algorithm design, IBD and AH designed the first version of the audio deck, AH upgraded the audio deck to the final version, FD and AH programmed the firmware, FD and AH conducted experiments. FD designed the experiments, implemented the ROS2 processing pipeline, analyzed data, performed simulations, and wrote this chapter. On the date of publication of this thesis, the material in this chapter is under preparation for a journal paper.

All code to reproduce the results from this chapter, including links to the datasets, is available at <https://github.com/lcav/audioROS>.

Echolocation for bats: more than time of flight



Variety of chirps used by bats for prey localization [65].



Bat approaching water to drink [78].

It is now a well-known fact that bats localize prey and obstacles using ultrasound. A variety of ultrasonic chirp shapes have been developed by different species depending on the task at hand (see top figure). For most tasks, bats measure time of flight to infer distance. However, recent experiments suggest that bats use a different technique to classify and localize flat water surfaces: they seem to detect a specific acoustic response linked to water. To show that no other sensory cues are used by the bats for its detection, Greif and Siemers [78] tricked bats into trying to drink from smooth metallic plates in complete darkness, where they showed the same water approaching and drinking behavior as in the wild (see bottom figure). The fact that even juvenile bats that were never confronted with other bats committed the same mistake, suggests that this is an innate ability put forward by evolution.

As the same authors suggest, this is an interesting finding for autonomous robots using sonar: ultrasound is very directional but bats can detect surfaces from the little energy returned from smooth surfaces outside their field of “view”. Using the same argument, the work in this chapter could thus be relevant for the more directional ultrasound sensing rather than audible sound.

Given these astonishing examples from biology, it might come as a surprise that, to this day, sound is only used very rarely in robotic perception, except for niche applications such as underwater robotics. On land and in air, spatial perception in robotics is largely dominated by visual modalities [35], while sound, if anything, is mostly used for communicating with users. In this and the next chapter we hope to contribute to filling this gap. We present new algorithms for using audible sound on mobile platforms, which allows to localize walls (this chapter) or external sound sources (next chapter).

Algorithms using sound for localization are particularly interesting because of the omnipresence of microphones. In order to communicate with people around them, we can expect many mobile robots to be equipped with microphones and loudspeakers. Using this hardware for obstacle detection too would avoid the cost and weight of additional sensors. Similarly, every smartphone has at least one microphone and a loudspeaker, which could be used as an additional modality in indoor localization systems, or for mapping out rooms. Even when not already present in existing hardware, microphones can be easily added to devices since they have become a cheap off-the-shelf technology.

Besides the wide availability, sound as a localization modality is interesting because it has the potential to perform well when other modalities fail. For instance, laser-based scanning solutions are known to struggle with glass walls [66, 108], from which sound reflects very

Notation for this chapter	
N	number of measurements
m	number of microphones
$d \in \mathbb{R}, \theta \in [0, 2\pi]$	distance and angle of closest wall in drone's reference frame
ρ	wall absorption coefficient
$\tau_s, \rho_s \in \mathbb{R}$	delay and attenuation of image source s
$\mathbf{s}, \mathbf{m}^{(i)} \in \mathbb{R}^D$	global source and i th microphone location
$s(t), x(t)$	source and measured signals
$h(t)$	room impulse response
$\zeta(f, \Delta)$	interference function
$\theta^{(i)} \in [0, 2\pi]$	angle, ...
$r_0^{(i)} \in \mathbb{R}^+$	length of direct path, ...
$r_1^{(i)} \in \mathbb{R}^+$	reflective path, ...
$\Delta^{(i)} \in \mathbb{R}^+$... and path difference to from source to microphone i

well. Camera-based localization solutions struggle when visibility is reduced by fog, haze or bad lighting conditions; again, sound is well suited for such scenarios. Moreover, sound is more omni-directional than other modalities, and speakers and microphones naturally emit or record sound in and from all directions, respectively.

Just like animals have evolved to use ultrasound to “see” and blind people can learn to use sound to detect obstacles, wouldn't it make sense for intelligent agents to evolve to use sound for localization and navigation, when other modalities are not available? In this chapter, we create a prototype of a drone that can, just like a bat, localize a wall using sound. We develop algorithms that are more suitable for robotics than classical echo-localization approaches, which typically can not be directly ported to platforms with limited computational power and low-quality microphones and speakers. We provide recovery limits for the proposed algorithms and we study their behavior in simulation. Then, we deploy them on a real drone. To this end, we create a custom audio deck for the popular *Crazyflie* drone [75]. We make both the firmware and hardware design for the deck publicly available to facilitate research on audio algorithms for drones by other people.

5.1. Problem setup

The setup we study consists of a small speaker, mounted on the drone, and a number of microphones. We denote by $s(t)$ the sound emitted by the on-board speaker, and by $x^{(i)}(t)$ the recorded sound at microphone i , where $i = 1 \dots m$ and m is the number of microphones. We assume that the drone's pitch and roll angle are close to zero during the measurement process, which is a reasonable assumption for a hovering drone. Therefore, we can parametrize the distance and angle of a given wall by its distance d to the drone's center, and the angle θ of its normal in the drone's local frame. We also assume that the

5. Dynamical sound-based wall localization

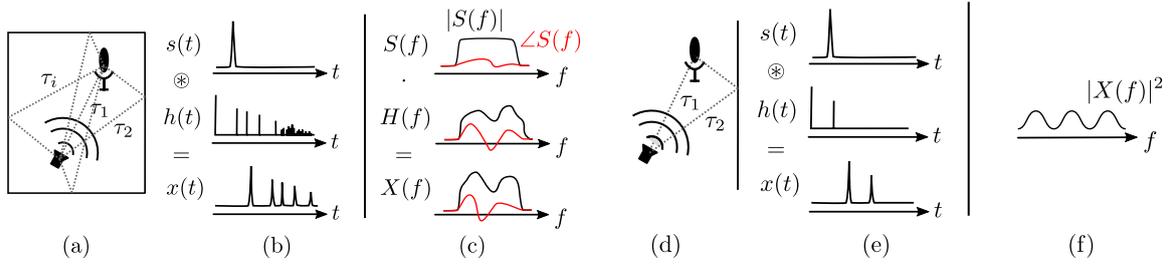


Figure 5.1.: Comparison of standard RIR measurement (a)-(c) vs. interference detection (d)-(f). Sketches of example setups are shown in (a) and (d), the corresponding responses $x(t)$ to input signals $s(t)$, given the respective impulse responses $h(t)$ are shown in (b) and (e) in time domain. The signals used for recovery of the wall location(s), in frequency domain, are shown in (c) and (f), respectively.

microphones and speaker are fixed at known positions on the drone. Then, we can formulate the inverse problem we aim to solve in this chapter:

Problem 3. Given the audio measurements from m microphones, $y^{(i)}(t)$, $i = 1 \dots m$ after emitting a source signal $s(t)$ and the known relative microphone and source locations, recover the distance d and angle θ of the closest wall.

5.2. Background

In the following, we first review methods for solving the more difficult problem of recovering the full room geometry from audio measurements in Section 5.2.1. Then, in Section 5.2.2, we adapt the model to our use case, where we are interested in one wall only.

5.2.1. Room impulse response reconstruction

At the basis of many classical echo-localization techniques [26, 110, 111, 182] is the estimation of the room impulse response. Denoted here by $h(t; \mathbf{m}, \mathbf{s}, \boldsymbol{\beta})$ the room impulse response from the microphone location \mathbf{m} to the speaker location \mathbf{s} models the effect of the sound propagating in space and reflecting from all surfaces. It depends on physical parameters of the room, which we accumulate in the vector $\boldsymbol{\beta}$. A sketch of a typical room impulse response is shown in Figure 5.1 (b). The response consists typically of well-pronounced peaks in the beginning, corresponding to the direct path and the early reflections of the sound, and a more stochastic regime, corresponding to the late reverberation of the room.

Using the room impulse response, we can compute the audio response at any point in the room, which makes it an interesting tool for audio simulation purposes. In echo-localization, we are not interested in the full response, but in a few of the first-order echos, which can then be used to reconstruct the room geometry, source and microphone locations[54, 111]. In the following, we explain how this is typically done.

We introduce the joint gain $g(t; \angle(\mathbf{m}, \mathbf{s}))$ between the microphone and speaker, incorporating their direction and frequency dependence (implicit in the variable t). While this gain could be seen as a part of the room impulse response, we separate it because it does not depend on room parameters but on microphone and speaker characteristics. It could therefore be calibrated independently. Under these assumptions, the audio response recorded of a microphone at position \mathbf{m} is given by

$$x(t; \mathbf{m}, \mathbf{s}, \beta) = h(t; \mathbf{m}, \mathbf{s}, \beta) * g(t; \angle(\mathbf{m}, \mathbf{s})) * s(t), \quad (5.1)$$

where $*$ denotes the convolution in time. In frequency domain, the model becomes

$$X(f; \mathbf{m}, \mathbf{s}, \beta) = H(f; \mathbf{m}, \mathbf{s}, \beta) G(f; \angle(\mathbf{m}, \mathbf{s})) S(f). \quad (5.2)$$

We use the image-source model for the room impulse response [4], which is given by

$$h(t; \mathbf{m}, \mathbf{s}, \beta) = \sum_i \rho_i(\mathbf{m}, \mathbf{s}, \beta) \delta(t - \tau_i(\mathbf{m}, \mathbf{s}, \beta)), \quad (5.3)$$

where τ_i and ρ_i denote the time delay and attenuation of the signal from the image source i . The image source i corresponds to the reflection of the sound source from one or multiple walls, where the number of bounces considered corresponds to the order of the source.

Classic echo-localization proceeds roughly as follows: The microphone emits a wideband sound, such as white noise or a frequency sweep. With appropriate hardware, the frequency response of both microphone and speaker can be considered flat ($G(f) = 1$), and so the room impulse response can be estimated from

$$h(t; \mathbf{m}, \mathbf{s}, \beta) = \mathcal{F}^{-1} \left(\frac{X(f; \mathbf{m}, \mathbf{s}, \beta)}{S(f)} \right), \quad (5.4)$$

where \mathcal{F}^{-1} is the inverse Fourier transform. The peaks of $h(\cdot)$ correspond to the delays from the different image sources. These delays can be readily transformed to distance estimates using the propagation speed of sound, and the room geometry and microphone/speaker location can be reconstructed [54].

5.2.2. Interference function

In many use cases, and in particular in the one of the bat drone considered in this chapter, we are interested in only a subset of the room impulse response. For instance, when limitations imposed by available hardware do not allow for the full room geometry reconstruction in one go, we might be interested in the location of the closest wall only. In this case, denoting by c the speed of sound and by ρ the wall absorption coefficient [184], the impulse response can be simplified to

$$h(t; \mathbf{m}, \mathbf{s}, \beta) = h(t; r_0, r_1, \rho) = \rho_0(r_0) \delta(t - r_0/c) + \rho_1(r_1, \rho) \delta(t - r_1/c), \quad (5.5)$$

5. Dynamical sound-based wall localization

where we have introduced the functions $\rho_0(r_0)$ and $\rho_1(r_1, \rho)$ for the attenuation along the direct path of length r_0 , and along the reflected path of length r_1 , respectively. If we ignore scattering effects and model the attenuation of the signal by free-space propagation, they are given by

$$\rho_0(r_0) = \frac{1}{4\pi r_0}, \text{ and } \rho_1(r_1, \rho) = \frac{1 - \rho}{4\pi r_1}. \quad (5.6)$$

Taking the magnitude squared of the frequency response, we get

$$\begin{aligned} |X(f; \mathbf{m}, \mathbf{s}, \boldsymbol{\beta})|^2 &= |X(f; r_0, r_1, \rho)|^2 = |H(f; r_0, r_1, \rho)G(f)S(f)|^2 \\ &= |G(f)S(f)|^2 \left| \frac{1}{4\pi r_0} \exp(-j2\pi f r_0/c) + \frac{1 - \rho}{4\pi r_1} \exp(-j2\pi f r_1/c) \right|^2 \\ &= \frac{|G(f)S(f)|^2}{(4\pi)^2} \left(\frac{1}{r_0^2} + \frac{(1 - \rho)^2}{r_1^2} + 2 \frac{1 - \rho}{r_0 r_1} \cos 2\pi f (r_1 - r_0)/c \right). \end{aligned} \quad (5.7)$$

As we will see later, it is much easier to do inference from measurements of the form (5.7) rather than (5.4), as it requires fewer measurements, no synchronization and no high-end hardware. Although considering only one wall is a coarse simplification compared to the full-room model used in RIR measurements, we will see that for the current use case, it is sufficient for the task of detecting the closest wall.

5.3. Related work

Audio signals are one of many modalities investigated for wall localization and room mapping. Lidar scanners and cameras, for instance, are often the go-to solution in robotics [35], however, they struggle with non-diffuse surfaces such as glass, metal, mirrors, or shiny and translucent surfaces. Recent methods for indoor mapping have thus focused on improving the performance in the presence of such materials, focusing for instance on angles from which they seem diffuse [66] or detecting inconsistent depth measurements corresponding to reflected points or points behind a transparent surface [108]. Other methods for reliable surface detection rely on millimetre wave (mmWave) radar, which has a long history in aerospace [29] and a recent but successful history in automobile applications [87]. As opposed to lidar or visual modalities, mmWave can see through smoke, and it was shown in [126] to not only succeed in localizing any surfaces reliably, but also classify them based on their different reflective properties.

Moving over to acoustic waves, ultrasound is more commonly employed for localization than sound in the audible range, supposedly because it is less susceptible to noise, and because it creates less disturbances for people around. Ultrasonic range finders are common on many robotic platforms [43, 167]. Because of their directionality, robots typically employ a multitude of ultrasound sensors for good spatial coverage. Some recent methods have focused on frequency ranges slightly outside of the audible spectrum but still recordable by commodity microphones, such as [189].

In this chapter, we focus on sound in the audible range. Besides it being somewhat under-explored compared to other modalities, meaning that some interesting research questions remain unanswered (see next paragraph), studying the aptitude of audible sound for localization has other advantages. As mentioned earlier, microphones are ubiquitous, they can have very small form factor compared to other sensors, and one can get good quality microphones for a low price. One could argue that as other sensors such as lidar and radar are becoming smaller and cheaper, this advantage will soon become irrelevant¹. However, if not a standalone wall detector, sound could always be exploited to complement other modalities. For instance, because of its long wavelengths, sound traverses smoke, haze and fog more easily, and the transparent or reflective surfaces causing headaches to lidar and visual systems, are optimal candidates for sound-based echo-location. The biggest advantage of audible sound with respect to ultrasound or radar, which share the former advantages, is that it is more omnidirectional. One can easily produce a sound wave which travels in all directions and thus has the potential to simultaneously localize all nearby obstacles.

Because of its omnidirectionality, sound can in theory reveal the shape of a room “in a finger-snap” [54]. A variety of solutions have been designed to solve this via RIR reconstruction, as discussed in Section 5.2.1. The methods differ in that they use different microphone-speaker configurations such as one fixed speaker and multiple measurement microphones [54], one fixed speaker and a moving smartphone with microphones [189], or co-located speakers and microphones [109]. Since they rely on recovering as many echoes as possible, these methods typically use high-end hardware for sufficient power and measurement quality. Since the RIR reveals distances only and no phase information, associating each echo with a distinct wall is not straight-forward and [54] use a combinatorial algorithm, while [189] present a more efficient and accurate approach based on semidefinite programming. Because of the underdetermined nature of distances, not any microphone-speaker configuration leads to the unique recovery of room geometry; uniqueness conditions are derived for instance in [109].

Solutions with limited hardware capabilities typically recover the shape of the room wall by wall. Using co-located speakers and microphones on a mobile robot, the authors of [109, chapter 4] use a particle filter approach, extracting always the first peak only from the RIR, and using this along with movement estimates to build a model of the room. A similar method is used by [182] and combined with beamforming to simultaneously recover the angle of the wall, which facilitates the labeling problem. Methods implemented on smartphones, where both speaker and microphones share a common clock, can use the radar-rooted FMCW algorithm [228, 229], which recovers the distance and velocity of reflectors in the field of view.

Both FMCW and RIR methods are difficult to implement for sound on light-weight robotic platforms. In FMCW-based methods, the incoming signal is correlated with the sent signal, and the frequency of the resulting signal reveals the distance of the closest reflector. This requires accurate synchronization and sufficient knowledge of the sent signal (in particular its phase), which are both assumptions that can be difficult to satisfy on a robotic platform.

¹For instance the tiny velabit lidar sensor: <https://velodynelidar.com/products/velabit/>

5. Dynamical sound-based wall localization

When the buzzer is controlled by a different unit than the microphones, synchronization can be cumbersome to achieve, and when the timing of the buzzer's signal cannot be accurately controlled, the exact output signal may not be known.

For RIR reconstruction, both synchronization and a wideband signal are required for sufficiently good recovery, again assumptions difficult to satisfy for systems running on lightweight platforms and in real time. Recently, a first study has demonstrated the potential of focusing on the interference created by the closest wall only [36]. Using an external measurement microphone, the authors detect interference between the direct and reflected rotor noise from the wall. Using a simplified form of the model shown in (5.7), the distance can be derived. As the measurement microphone approaches the rotors, the method however breaks down.

Rather than trying to recover the full RIR or the distance/velocity of the closest reflector using FMCW, we also pose the wall detection problem as an interference detection problem. First, sampling the interference function along frequency, we develop an algorithm for wall detection similar to [36]. Rather than using the drone's own noise, which was shown in [36] to be difficult, we show that by using a small affordable piezo buzzer emitting frequency sweeps in the audible range, we can detect the wall distance with high accuracy. We also show that we can get an estimate of the wall's angle, without resorting to a separate beamforming algorithm as in [182].

Then, we sample the interference function in space, leading to a method quite different from what has been studied before. Using only a single frequency emitted by the buzzer, we develop an algorithm to detect a wall and the angle at which we are approaching it. To the best of our knowledge, this is the first attempt at exploiting spatial interference for localization.

5.4. Wall localization based on the interference function

We derive a compact form of the interference function along frequency and distance in Section 5.4.1 and describe the two presently studied inverse problems, based on this function, in Section 5.4.2. We derive general probabilistic period recovery in 5.4.3, which is necessary for the solutions to the two problems in Sections 5.4.4 and 5.4.5, respectively. Notes on the practical implementation of these algorithms are given in 5.4.6. We finish by outlining recovery limits of the proposed methods 5.4.7.

Note that for the more general considerations in Section 5.4.1 to Section 5.4.3 we ignore the fact that we have multiple microphones. This keeps notation more lean and understandable. We introduce the notation for multiple microphones when it is necessary, starting from Section 5.4.4.

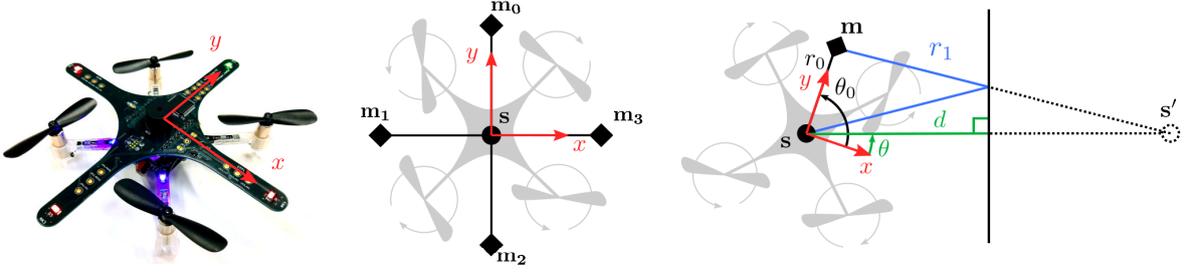


Figure 5.2.: *Geometry of used drone setup. The reference frame with respect to the developed audio deck is shown in (a), the microphone and source locations are shown in (b) and the notation related to wall localization is detailed in (c).*

5.4.1. Compact form of interference function

Since it is the main object of interest in this chapter, we do a few more manipulations to adapt equation (5.7) to our setup. First, we observe that the period of the cosine in (5.7) depends on the difference in path length. Since we are interested in the distance and angle of the wall, we rewrite the path length as a function of these two quantities. Without loss of generality, we set the robot's reference frame to coincide with the speaker location s , and parametrize the wall by its distance d and angle θ in the robot's reference frame, as shown in Figure 5.2. Then we can write the difference in path length as:

$$\Delta := r_1 - r_0 := g_{\Delta}(d, \theta) = \sqrt{r_0^2 + 4d^2 - 4dr_0 \cos(\theta_0 - \theta)} - r_0, \quad (5.8)$$

where $\theta_0 = \angle(m - s)$ denotes the angle of the vector from the microphone to the speaker in the robot's reference frame.²

Secondly, since the microphones and speakers are fixed, r_0 is constant. To simplify notation, we can thus express r_1 as $r_1 = \Delta + r_0$ and use only Δ as our free variable, dropping the arguments r_1 and r_0 .

Finally, as the gains $|S(f)|$ and $|G(f)|$ are indistinguishable unless independently calibrated, we introduce the combined gain function $T(f) := |S(f)G(f)|^2$. Putting everything together, we obtain the following interference function:

$$\zeta(f, \Delta) = T(f) (\rho_0^2 + \rho_1(\Delta)^2 + 2\rho_0\rho_1(\Delta) \cos 2\pi f \Delta / c), \quad (5.9)$$

with $\rho_1(\Delta) = (1 - \rho) / (\Delta + r_0)$. We will see that the oscillation amplitude of (5.9) is of importance for the performance of the subsequent methods, therefore we introduce:

$$a(\Delta, f) = 2T(f)\rho_0\rho_1(\Delta). \quad (5.10)$$

²Note that in the echo-localization literature, the simplification is often made that the distance of the wall is significantly bigger than the baseline r_0 , and that thus the path length difference simplifies to $r_1 - r_0 \approx 2d$. As we are mostly interested in walls that are close with respect to the baseline, we do not adopt this simplification.

5. Dynamical sound-based wall localization

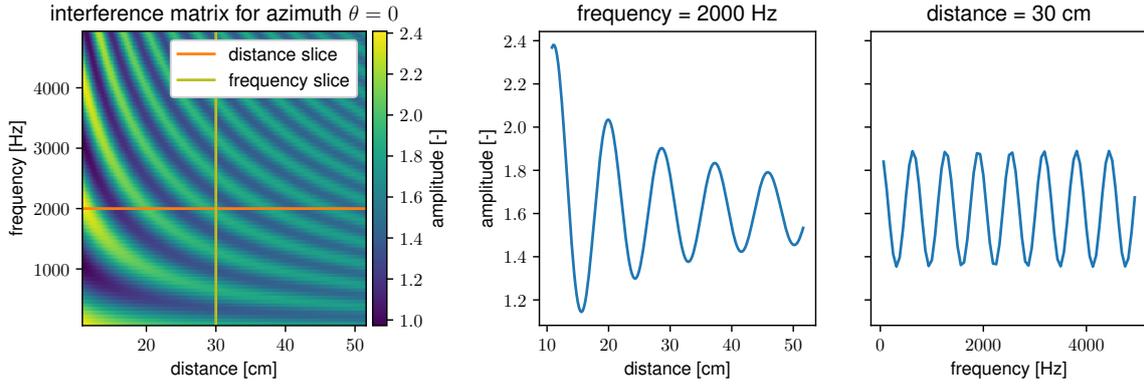


Figure 5.3.: Analytical interference matrix and two slices, obtained using equation (5.7).

5.4.2. Problem setup revisited

Depending on how we sample the function (5.9), we can try to solve different geometrical inference problems. In this chapter, we focus on two problems: one consists of retrieving the distance and angle of the wall from a static drone, and the other consists of retrieving what we denote the “angle of approach” of a moving drone. These two methods yield different “chirps”, similar to bats which adopt the shape of their ultrasound signal depending on the task at hand (see box in the beginning of the chapter). We formally derive these two problems in what follows, and discuss other possible use cases of the interference function.

We introduce the following measurement model for time t_n ,

$$z_n := \zeta(f_n, \Delta_n + \epsilon_n^\Delta) + \epsilon_n^z; n = 1 \dots N, \quad (5.11)$$

where ϵ_n^Δ and ϵ_n^z are zero-mean Gaussian measurement errors on the path difference and the interference function, respectively. Stacking up the measurements and parameters, we construct the vectors $\mathbf{z} \in \mathbb{R}^N$, $\mathbf{f} \in \mathbb{R}^N$ and $\mathbf{\Delta} \in \mathbb{R}^N$. When necessary, we add the superscript (i) to denote the measurements taken at microphone i : $\mathbf{z}^{(i)}$.

The two problems that we study use “frequency slices” or “distance slices”, respectively. We illustrate these two concepts in Figure 5.3, where we fix the angle at $\theta_n = 0$ for all n , which enables us to translate the path difference to the more intuitive distance of the wall. Then, we obtain samples of the interference function on a uniform grid of distances and frequencies; we refer to the resulting matrix as the interference matrix. When we fix the frequency and sample the distances uniformly, we obtain a distance slice. Equivalently, we can fix the distance to obtain a frequency slice. An example interference matrix and one of each of these slices are shown in Figure 5.3.

In the first sub-problem, we study distance inference from frequency slices. To achieve this in practice, a drone would hover at a fixed nominal location and record the response of a

frequency sweep, as shown in Figure 5.4 (a). The measurements are thus of the form:

$$\begin{aligned} z_n(f_n) &= \zeta(f_n, \Delta + \epsilon_n^\Delta) + \epsilon_n^z; n = 1 \dots N \\ &= \zeta(f_n, g_\Delta(d, \theta) + \epsilon_n^\Delta) + \epsilon_n^z; n = 1 \dots N, \end{aligned} \quad (5.12)$$

where $\Delta = g_\Delta(d, \theta)$ is the path difference corresponding to the drone's nominal location with respect to the wall, and ϵ_n^Δ incorporates lateral movements due to imperfect position control. As the values f_n simply correspond to the frequency bins of our frequency analysis, we can assume \mathbf{f} known. We can then formulate the first inverse problem:

Subproblem 1. *Given measurements $\mathbf{z} \in \mathbb{R}^N$ of the form (5.12) and for each microphone, estimate the distance d and angle θ to the closest wall.*

In the second sub-problem, we assume that the drone is approaching the wall at constant linear speed and playing a mono-frequency sound of frequency f , as visualized in Figure 5.4 (b). The orthogonal wall distance thus follows a linear model of the form $d_n = d_1 + l_n \sin \gamma$, where r_1 is the starting distance, γ is the angle of approach, and l_n is the cumulative relative movement of the drone: $l_n := \|\mathbf{r}_n - \mathbf{r}_1\|$, with $\mathbf{r}_n \in \mathbb{R}^D$ the position of the drone at time t_n . In this case, we obtain measurements of the form:

$$z_n(l_n) = \zeta\left(f, g_\Delta(d_1 + (l_n + \epsilon_n^l) \sin \gamma, \theta)\right) + \epsilon_n^z; n = 1 \dots N. \quad (5.13)$$

Determining the distance d_1 from measurements (5.13) is challenging and would require accurate knowledge of physical parameters such as the wall absorption. On the other hand, since the wall approach angle γ is directly related to the period of measurements, it can be reliably estimated as long as we are close enough to the wall to measure interference and we have a sufficiently accurate estimate of the relative movements l_n . In practice, this is sufficient for a simple navigation algorithm: as soon as we have a reliable estimate of γ we know that we are close to a wall and can estimate at what angle we are approaching it. Using this information, we can change direction to avoid crashing into the wall. Constructing the vector of relative movement estimates, $\mathbf{l} = [l_2, \dots, l_N]^\top \in \mathbb{R}^{N-1}$, the problem we aim to solve is:

Subproblem 2. *Given measurements $\mathbf{z} \in \mathbb{R}^N$ of the form (5.13) for each microphone, as well as relative movement estimates $\mathbf{l} \in \mathbb{R}^{N-1}$, estimate the angle of approach γ .*

Note that the framework studied here is not restricted to these two use cases, and in particular, it is not limited to drones: Figure 5.4 shows the connection of other experiments with the interference function, as well as the resulting slices when using a different robotic platform.

5.4.3. Probabilistic period estimation

As we have seen in the previous section, both inference problems require the estimation of the period of the interference function, either across frequencies or across distances. In this section, we thus introduce a probabilistic method for period estimation.

5. Dynamical sound-based wall localization

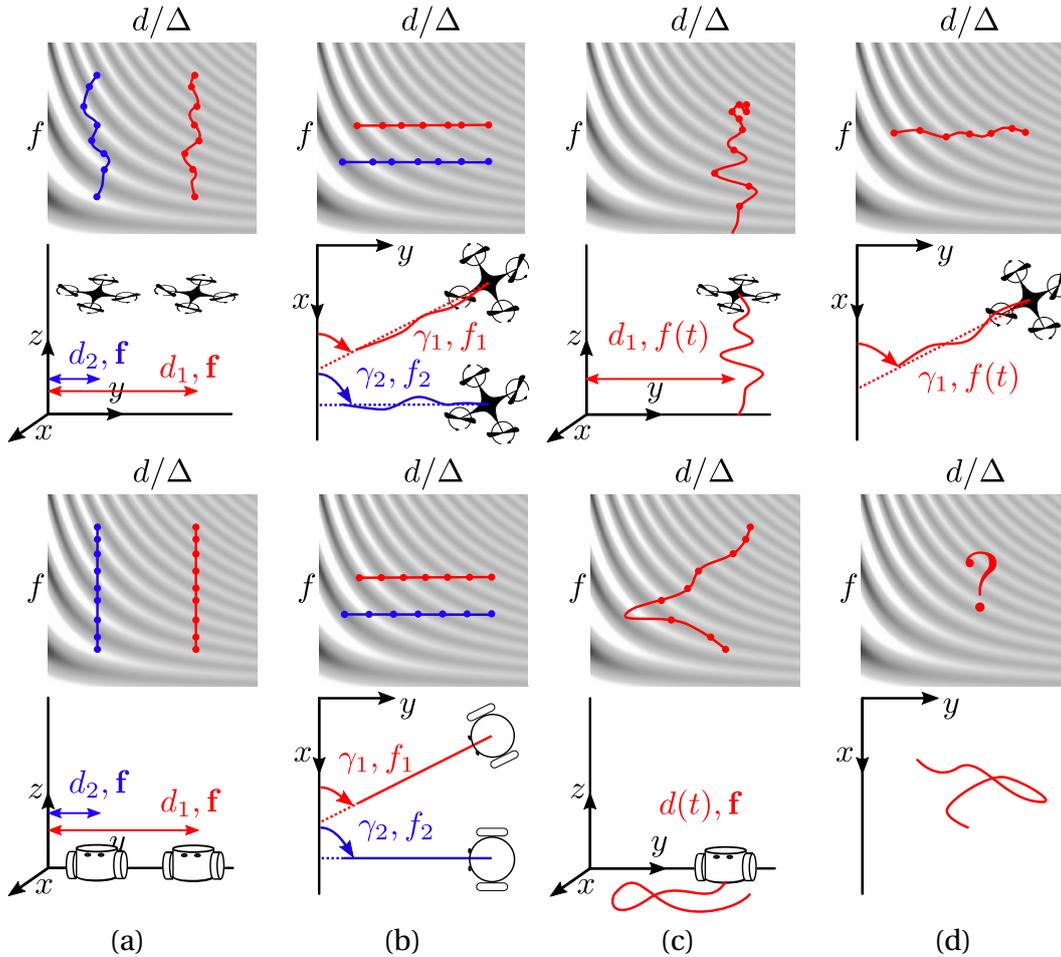


Figure 5.4.: Various ways of sampling the interference function. The top two rows show possible experiments using a drone, while the bottom rows show experiments with a ground-based robot. Depending on the experiment, we can aim to recover the distance d and angle of the wall (not drawn), or the angle of approach γ . Experiments (a) and (b) are studied in this chapter. Experiments (c) and (d) are possible extensions using the drone’s ego-noise, or a continuous movement and sweep by the ground-based robot.

We introduce the variable u_n , $n = 1 \dots N$ for measurement locations, δ_u for the average measurement spacing and v_k for candidate “frequencies”. We describe in Sections 5.4.4 and 5.4.5 what these quantities correspond to in the frequency-slice or distance-slice problem, respectively. Given our measurements z_n sampled at locations u_n , we can calculate the Schuster periodogram of the measurements, given by

$$P_k(z) := \frac{1}{N} \left| \sum_n z_n e^{j2\pi v_k u_n} \right|^2. \quad (5.14)$$

Note that if the samples u_n are close to uniform, a simple FFT can be used to calculate P_k . For non-uniform samples, other methods such as the Lomb-Scargle Periodogram can be used [123, 207]. Since the measurements z_n are real-valued, P_k is symmetric and we can only look at candidate interference frequencies $v_k = k/(N\delta_u)$, $k \in [0 \dots K]$, with $K = \lfloor \frac{N}{2} \rfloor$.

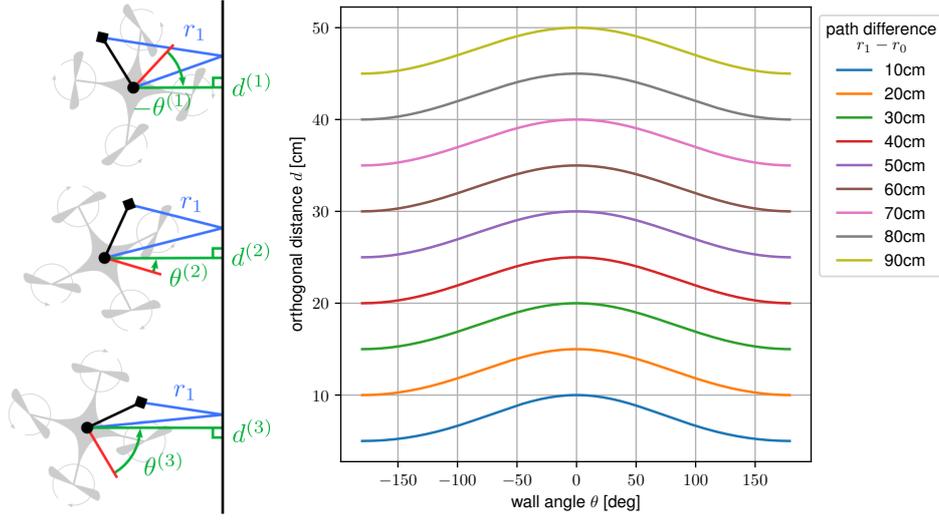


Figure 5.5.: Left: incomplete set of positions which yield the same path difference estimate but different orthogonal distances and angles. The complete range of possible orthogonal distances and angles for a list of example path differences is shown on the right.

From the Schuster periodograms, we would like to deduce a likelihood for a given interference frequency to be present in the signal. Quite interestingly, it was shown in [98] that the Schuster periodogram is in fact the optimal statistic for detecting a stationary sinusoidal signal in the presence of Gaussian noise. If an estimate of the amplitude noise σ_y is known, then the probability that our data is a sine of frequency v_k is governed by [28]:

$$p(v_k) \propto e^{P_k/\sigma_y^2}. \quad (5.15)$$

Alternatively, if the noise variance cannot be estimated, the probability for candidate frequency v_k is

$$p(v_k) \propto \left[1 - \frac{2P_k}{N\bar{z}^2} \right]^{\frac{2-N}{2}}, \quad (5.16)$$

where \bar{z}^2 denotes the mean-square measurements.

Connection with optimization Since we know that the model of the signal that we are sampling is a sinusoid at a given frequency, one might be tempted to solve an optimization problem rather than an FFT-based solution. For instance, the minimum least squares estimate of the frequency would be given by:

$$v_k = \arg \min_{v_k} \hat{\chi}^2(v_k) \quad (5.17)$$

$$\hat{\chi}^2(v_k) = \min_{A_k, \phi_k} \sum_n (z_n - A_k \sin(2\pi v_k t_n - \phi_k))^2,$$

where we have introduced $\hat{\chi}^2(f)$, the cost of the best least-squares fit of a sinusoid of frequency v_k to the data. Quite surprisingly, there is an equivalence between this optimization-based solution and the FFT-based solution. Indeed, it can be shown that the value of the

5. Dynamical sound-based wall localization

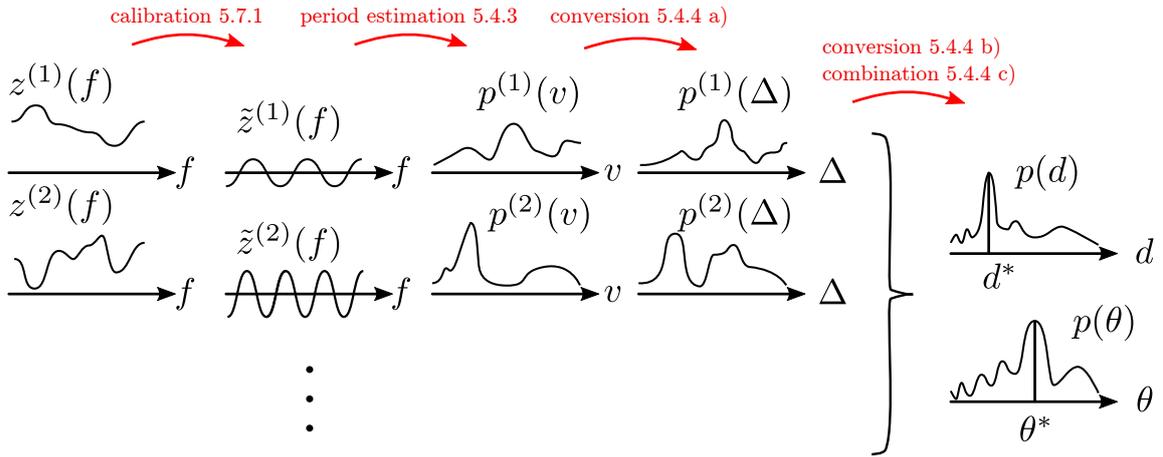


Figure 5.6.: Algorithm overview for frequency slices, or how to pass from the raw interference measurements $z^{(i)}(f)$ from microphones $i = 1 \dots m$ to probability distributions over distances and angles, $p(d)$ and $p(\theta)$.

Schuster periodogram for frequency v_k is also equal to [207]:

$$P(v_k) = \frac{1}{2} (\hat{\chi}^2(0) - \hat{\chi}^2(v_k)), \quad (5.18)$$

Note that extensions to other models exist, but in our case, the sine model is valid because of the nature of the interference function (5.9).

In other words, the FFT-based approach gives us the same result as we would get by solving the optimization for the best amplitude and phase offsets, for each frequency. This explains why the FFT-based approach is particularly useful when we want to be agnostic to phase and scaling discrepancies, which typically arise in our experimental setup.

5.4.4. Frequency slices (solution to Subproblem 1)

A sketch of the full algorithm pipeline for recovering the distance and angles from the closest wall is shown in Figure 5.6. We first apply offline calibration to the measurements (first step in Figure 5.6), after which we can obtain the probability distribution over periods $p^{(i)}(v)$ as explained in the preceding section (second step). We then need to translate this distribution to the variables of interest d and θ , using geometrical and probabilistic considerations. We proceed in two steps (corresponding to step 3 and 4, respectively, in Figure 5.6), as outlined below.

a) From frequency interference to path differences Once we have distributions of interference frequencies $p^{(i)}(v)$ for all microphones $i = 0 \dots m - 1$, we need to translate them to a path difference distributions $p^{(i)}(\Delta^{(i)})$ (third step in Figure 5.6). In this problem, the measurement locations u_n are the frequencies f_n , with average spacing δ_f . We know that for a given distance d , the interference period (across frequencies) is given by c/Δ . Therefore,

the probabilities in (5.15) and (5.16) can be translated directly to a probability distribution over path differences. Denoting by $p^{(i)}(v_k)$ these distributions obtained at microphone i using the periodogram $P_k(\mathbf{z}^{(i)})$, we have

$$p^{(i)}(v_k) = p(\Delta_k/c), \quad (5.19)$$

with $\Delta_k = ckK/\delta_f$ and therefore

$$p^{(i)}(\Delta) \propto p^{(i)}(v_k). \quad (5.20)$$

b) From path differences to distance and angle Denoting the number of microphones by m , the path difference at the i -th microphone at location \mathbf{m}_i , is given analogously to (5.8) by

$$\Delta^{(i)} = g_{\Delta}^{(i)}(d, \theta) = \sqrt{r_0^{2(i)} + 4d^2 - 4dr_0^{(i)} \cos(\theta^{(i)} - \theta)} - r_0^{(i)}, \quad i = 0 \dots m-1, \quad (5.21)$$

where we have introduced the direct path length and angle of microphone i , $r_0^{(i)} = \|\mathbf{m}_i - \mathbf{s}\|$ and $\theta^{(i)} = \angle \mathbf{m}_i - \mathbf{s}$, respectively.

Given the probabilistic solution of each path difference estimate, we are interested in similar probabilistic solutions for both the distance and the angle of the closest wall. This is of particular importance if one or more of the individual path difference distributions of the individual microphones exhibit multiple peaks. The simple approach of getting the maximum-likelihood path difference estimates and translating them to a single distance and angle by solving a system of equations, would be very sensitive to selecting the wrong peak. The probabilistic solution is more robust and provides us with an estimate as well as a measure of certainty.

Distance inference: We denote the probability distribution over path differences by $p^{(i)}(\Delta)$. Our aim is to translate this distribution to a distribution over distances. Using a change of variables, the following holds:

$$\begin{aligned} p^{(i)}(d; \theta) &= p^{(i)}(g_{\Delta}^{(i)}(d; \theta)) \left| \frac{\partial}{\partial d} g_{\Delta}^{(i)}(d; \theta) \right| \\ &= p^{(i)}(g_{\Delta}^{(i)}(d; \theta)) \left| \frac{4d - 2r_0^{(i)} \cos(\theta^{(i)} - \theta)}{\sqrt{r_0^{(i)2} + 4d^2 - 4dr_0^{(i)} \cos(\theta^{(i)} - \theta)}} \right|. \end{aligned} \quad (5.22)$$

We can thus compute the probability distribution over d by marginalizing over θ :

$$p^{(i)}(d) = \int_{-\pi}^{\pi} p^{(i)}(d; \theta) p(\theta) d\theta = \frac{1}{2\pi} \int_{-\pi}^{\pi} p^{(i)}(g_{\Delta}^{(i)}(d; \theta)) \left| \frac{4d - 2r_0^{(i)} \cos(\theta^{(i)} - \theta)}{\sqrt{r_0^{(i)2} + 4d^2 - 4dr_0^{(i)} \cos(\theta^{(i)} - \theta)}} \right| d\theta, \quad (5.23)$$

where we have used a uniform prior on θ . We refer to Section 5.4.6 for the implementation for discrete measurements.

5. Dynamical sound-based wall localization

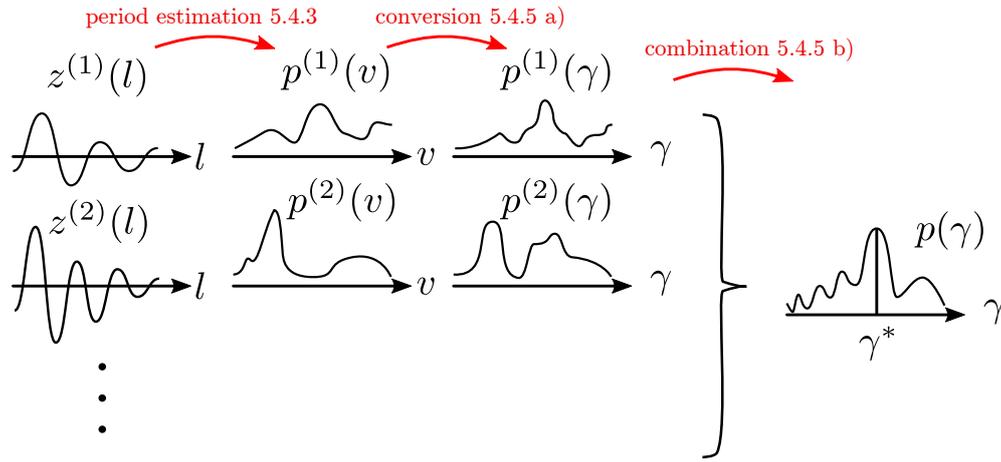


Figure 5.7.: Algorithm overview for distance slices, or how to pass from spatial interference measurements $z^{(i)}$ for microphones $i = 1 \dots m$ to the probability distribution over the angle of approach γ .

Angle inference: Angle inference is performed similarly to distance inference. Assuming a uniform prior over all distances and calculating the gradient of g with respect to θ , we get

$$\begin{aligned} p^{(i)}(\theta) &= \int_{d_{min}}^{d_{max}} p^{(i)}(\theta; r) p(r) dr \\ &= \frac{1}{d_{max} - d_{min}} \int_{d_{min}}^{d_{max}} p^{(i)}(g_{\Delta}^{-1(i)}(\theta; r)) \left| \frac{-2rr_0^{(i)} \sin(\theta^{(i)} - \theta)}{\sqrt{r_0^{(i)2} + 4r^2 - 4dr_0^{(i)} \cos(\theta^{(i)} - \theta)}} \right| dr, \end{aligned} \quad (5.24)$$

where we again refer to Section 5.4.6 for the practical evaluation of this integral.

c) Combination of individual microphones Finally, we combine the individual microphone measurements to obtain the total probability distribution over distances and angles. In the present implementation, we combine all measurements with equal weights:

$$p(d) = \eta \prod_{i=0}^{m-1} p^{(i)}(d), \text{ and } p(\theta) = \eta \prod_{i=0}^{m-1} p^{(i)}(\theta), \quad (5.25)$$

where η is the normalization factor to make $p(d)$ and $p(\theta)$ sum to one. Possible extensions of this scheme could use weighted averages, where microphones with unreliable measurements would be given less importance.

5.4.5. Distance slices (solution to Subproblem 2)

Given the distribution over periods in the distance slice, we want to infer the distribution over the angle of approach γ . We can use similar considerations as for the frequency slice here, with slightly fewer stages as no calibration is required and the conversion is more straightforward. The pipeline is sketched in Figure 5.7.

a) From distance interference to angle of approach To recover the angle of approach, we use the fact that there is a one-to-one relationship between the latter and the period of the distance slice. In fact, for a drone approaching the wall perpendicularly ($\gamma = \pi/2$), the interference period along the distance slice is given by $\Delta^* = c/f$. For all angles smaller than that, the interference period will be bigger by a factor dependent on the angle γ . We denote by δ_l the average relative distance spacing, and employ the simplification that the path differences spacing can be approximated by $\delta_\Delta \approx 2\delta_l$. Then, each candidate frequency v_k from (5.15) is related to the angle of approach γ_k by the following function g_γ :

$$\begin{aligned} \sin \gamma_k &= \Delta^* v_k \approx \frac{c v_k}{f} = \frac{c k K}{2 f \delta_l}, \\ v_k &:= g_\gamma(\gamma_k) = \frac{f \sin \gamma_k}{c}. \end{aligned} \quad (5.26)$$

Note that since only periods above Δ^* are physically feasible, we reduce the indices k to $k = 0 \dots 2\delta_l f / (Kc)$. The distribution over angles is given by

$$p^{(i)}(\gamma) = p^{(i)}(g_\gamma(\gamma)) \left| \frac{\partial}{\partial \gamma} g_\gamma(\gamma) \right| = p^{(i)}(g_\gamma(\gamma)) \left| \frac{2f \cos \gamma}{c} \right|. \quad (5.27)$$

Inspecting (5.27), we notice that the probability calculated as shown tends to zero as γ approaches π . We therefore sacrifice the ‘‘correct’’ form for a practically implementable form, where we drop the cosine factor.

b) Combination of multiple microphones Each microphone gives an individual estimate for the angle of approach γ . The naive approach to combine these estimates is via multiplication as in (5.25). Doing so would discard important information which allows to resolve from what direction we are approaching the wall. Instead, we split the microphones in two sets containing either the left or right microphones with respect to the axis of movement, called \mathcal{L} and \mathcal{R} , respectively. Then, we create two scores at the current estimate $\hat{\gamma} \in [0, 90^\circ]$ from the raw distributions $p^{(i)}(\hat{\gamma})$: $s_{left} = \sum_{i \in \mathcal{L}} \max(p^{(i)}(\hat{\gamma}))$ and $s_{right} = \sum_{i \in \mathcal{R}} \max(p^{(i)}(\hat{\gamma}))$. We choose the side corresponding to the higher score, and include only the corresponding probability distributions. If $s_{left} > s_{right}$, for instance, we compute

$$p(\gamma) = \prod_{i \in \mathcal{L}} p^{(i)}(\gamma), \quad (5.28)$$

and we derive that the wall is on the left side in the direction of travel.

5.4.6. Notes on practical implementation

In practice, we do not have continuous measurements, so we can not calculate the exact integrals above. We therefore discretize the state space using a fixed range and resolution: For distances, we evaluate $p(d)$ on a grid with one-centimeter resolution from the minimum

5. Dynamical sound-based wall localization

possible distance without collision, to 50 centimetres. For the wall angles $p(\theta)$, we create a grid from 0 to 360 degrees with one degree resolution, and for the angle of approach $p(\gamma)$ we choose a grid from 0 to 90 degrees with one degree resolution. Then, each probability distribution (5.23), (5.24) and (5.27) is evaluated on that grid. We use cubic interpolation between the discrete measurements. We use the same discretization to approximate the integrals by finite sums over the respective grids in (5.23) and (5.24).

Furthermore, both frequency and distance slice measurements might not be exactly uniform. Since we would like to use the FFT for evaluating the periodogram, as it is computationally cheap, we first interpolate the measurements on the uniform grid with the same minimum, maximum and average spacing.

5.4.7. Recovery limits

When fixing the number of measurements N , a given sampling spacing δ_u sets certain limits on the recoverable frequencies v_k (in the interference signal). By the Nyquist theorem we know that the maximum recoverable frequency is given by $\frac{1}{2\delta_u}$. In our two subproblems, this limit can be translated to recoverable distances and angles or usable source frequencies. We also know that the frequency resolution is given by $1/(N\delta_u)$. This is not a hard limit as we can always employ zero-padding to increase the apparent resolution. However, it provides a reasonable lower bound on the recoverable period, as for signals of a period higher than $N\delta_u$ we would not sample the full period.

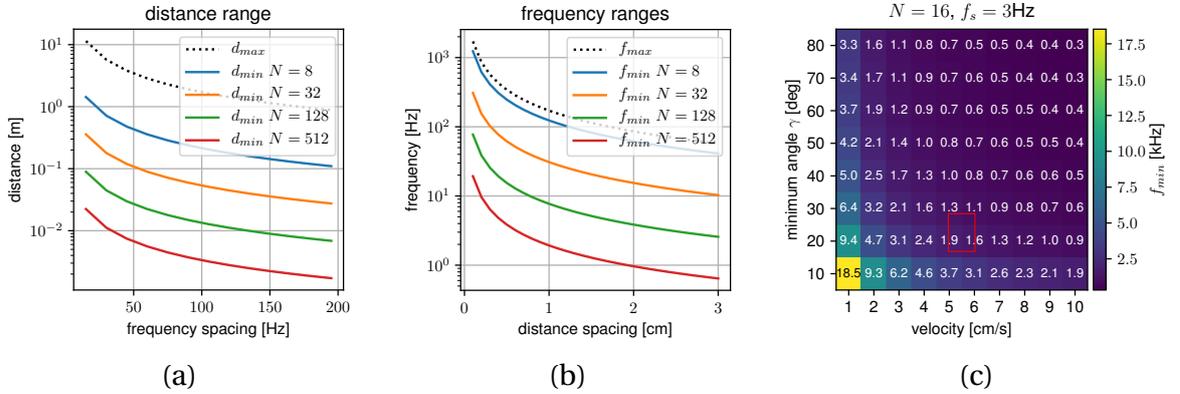


Figure 5.8.: Examples of recovery limits for frequency slice algorithm in (a) and distance slice algorithm in (b) and (c).

Recovery limits of Problem 1 The interference period gets smaller as the distance grows, and the smallest period we can recover is $2\delta_u$. This limits us to path differences below $\Delta_{max} := c/(2\delta_f)$. The distance resolution, or equivalently, the smallest path difference for which we have full-period sampling, is given by $\Delta_{min} := c/(N\delta_f)$. Using the simplification that $\Delta \approx 2d$, this translates to the recoverable distance range

$$d_{min} := \frac{c}{2N\delta_f} \leq d \leq \frac{c}{4\delta_f} := d_{max} \quad (5.29)$$

The recoverable distance ranges for different frequency spacings δ_f and measurement numbers N are shown in Figure 5.8 (a).

Recovery limits of Problem 2 The interference period is minimal at $\gamma = \pi/2$, given by c/f , and it increases, until it reaches infinity for a drone moving parallel to the wall $\gamma = 0$. For a fixed frequency, we can thus determine the minimum angle at which we can still sample a full period. The maximum recoverable period is given by $N\delta_\Delta$. Therefore, as long as $\sin \gamma \geq \frac{c}{fN\delta_\Delta}$, we sample more than one period.

For a fixed angle γ , we can determine the maximum buzzer frequency that we can use before reaching the Nyquist limit. The interference period at angle γ is given by $c/(f \sin \gamma)$. For it to be bigger than the Nyquist limit $2\delta_\Delta$, we need the buzzer frequency to be lower than $f_{max} := c/(2\delta_\Delta \sin \gamma)$. Similarly, in order to sample a full period, we need the frequency to be higher than $f_{min} := c/(N\delta_\Delta \sin \gamma)$. Using again the simplification that $\delta_\Delta \approx 2\delta_l$, we get the approximate usable frequency range for recovering angle $\gamma > 0$:

$$\frac{c}{2N\delta_l \sin \gamma} \leq f \leq \frac{c}{4\delta_l \sin \gamma}. \quad (5.30)$$

In Figure 5.8 (b), we show the usable frequency range for detecting angles between 10 and 90°. In Figure 5.8 (c), we assume that a device can take measurements at 3 Hz. For each linear velocity, this translates to a certain measurement spacing δ_Δ , which in turn comes down to what frequency f_{min} we can use, allowing the recovery of down to γ_{min} angles. For example, given a drone that moves at 5 cm/s, if we wanted to detect angles down to 20°, then we would need to use a frequency above 1.9 kHz (highlighted square).

5.5. Simulation results

In this section, we study the performance of the proposed algorithms in simulation. To this end, we generate data of the form (5.11) and solve for the distance or angle of approach. For the experiments with noise, we independently sample from the noise distributions a 100 times and we report both the mean and the standard deviation of the absolute error of all realizations.

5.5.1. Frequency slice algorithm

We study the simulated performance of the frequency slice algorithm using parameters suitable for our experimental framework: we use a uniform frequency sweep of 16 tones between 3000 Hz and 5000 Hz, and we study distances up to 50 cm.

Note that although we can infer both the distance and angle of the wall with the frequency slice algorithm, we report here only the distance, corresponding to the path difference

5. Dynamical sound-based wall localization

at a fixed angle, for a single microphone. This is sufficient to gain intuition on when the methods will break down, as an erroneous path difference estimates will evidently lead to an erroneous angle and distance estimates as well.

Amplitude noise We show how the method performs in the presence of amplitude noise, *i.e.* ϵ_n^z in (5.11), in Figure 5.9. Looking at both the mean and standard deviation of the error, we see a sharp transition from almost zero error for low distances and noise values to significantly higher error at higher values. An analysis of the point of this performance breakdown shows that it occurs roughly when the amplitude noise is of the order of magnitude of the oscillation amplitude of the cosine $a(d)$ (5.10). This amplitude decreases with the inverse of the distance d , suggesting that the level of amplitude noise determines up to which distance we can hope to get reliable distance estimates.

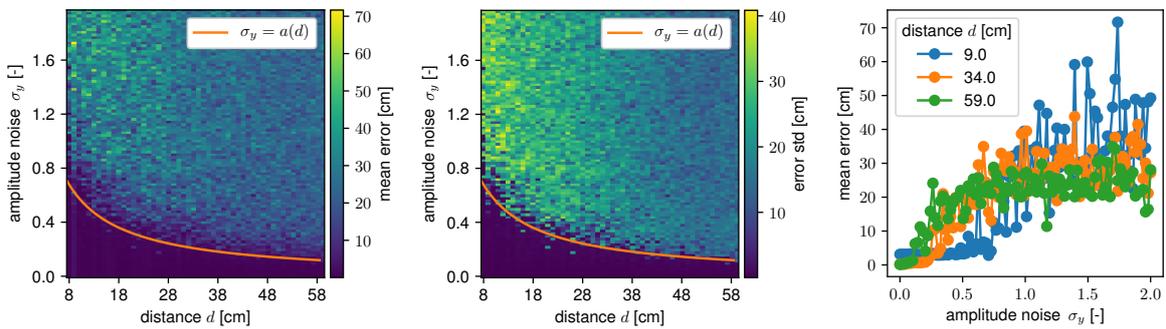


Figure 5.9.: Robustness study for amplitude noise σ_y .

Non-uniform frequency bins We study the effect of non-uniform frequency bins on the performance of the FFT-based method in Figure 5.10. For this experiment, we do not interpolate on a uniform grid before applying the FFT as described in 5.4.6. We have again a sharp transition in terms of performance starting from a frequency standard deviation of ca. 70 Hz at 40-50 cm, and at more than 150 Hz at 20 cm or closer. This is expected as frequency slices at lower frequencies are smoother, thus the amplitude error between the non-uniform bins and uniform bins becomes negligible as the distance decreases. Depending on the use case, the interpolation procedure could thus be omitted.

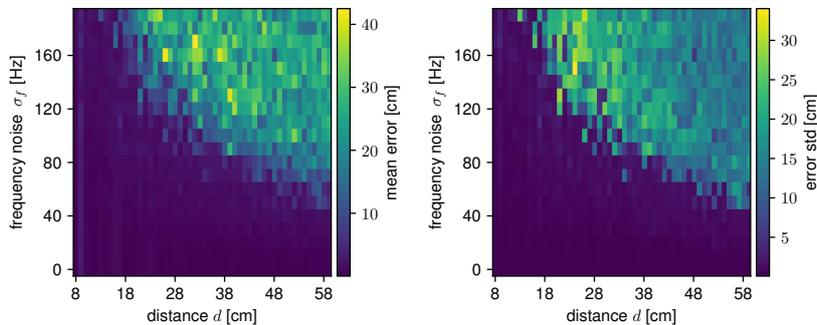


Figure 5.10.: Sensitivity of our method (without interpolation) to non-uniform frequency bins.

Lateral movement We choose to study both distance and angle noise jointly by adding noise to the path difference Δ . How this noise translates to either distance noise or angle noise is shown in the left plot of Figure 5.11. For this kind of noise, we observe in the two right plots of Figure 5.11 that the absolute error is directly proportional to the noise, and there is little dependence on the distance.

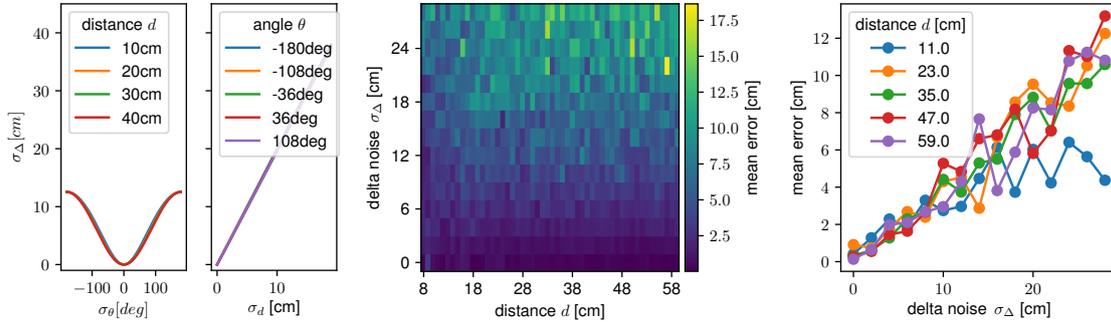


Figure 5.11.: *Left: correspondence between delta noise and angle and distance noise, respectively. Middle: robustness study for movement noise σ_Δ . Right: different slices of the movement noise matrix.*

5.5.2. Distance slice algorithm

The results reported in this section correspond to a simulation setup with $d_1 = 50$ cm and $l = [0, 1 \text{ cm}, \dots, 19 \text{ cm}]$.

Recovery limits The results of the distance slice algorithm show a strong dependence on the used frequency: the higher the frequency, the higher the accuracy of the angle estimate. This is visualized in Figure 5.12, where no amplitude or movement noise are added.

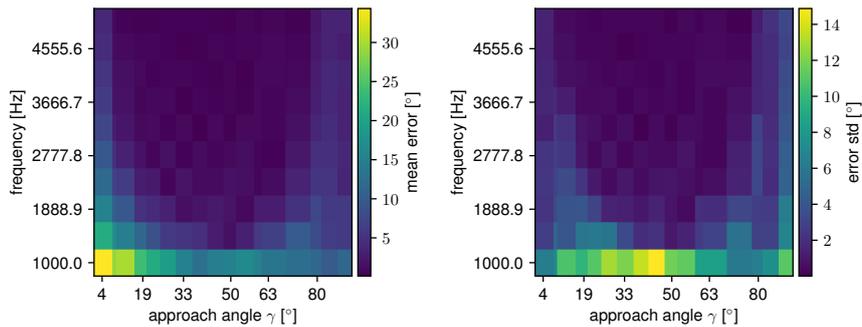


Figure 5.12.: *Performance of distance slice algorithms for different frequencies.*

Amplitude noise A number of interesting observations can be made on the simulations with amplitude noise, shown in Figure 5.13 for 3 different frequencies. We observe again a sharp transition from close-to-zero errors, below noise around $10^{-0.7}$, and high error beyond that level. The method breaks down more quickly at lower angles, which corresponds to

5. Dynamical sound-based wall localization

regions where we sample a smaller portion of the interference pattern. As we increase the frequency, we sample bigger portions of the pattern, thus the performance for low angles increases. However, for the high angles, we approach the aliasing limit as we increase the frequency, which explains a performance deterioration for high frequencies and angles, respectively.

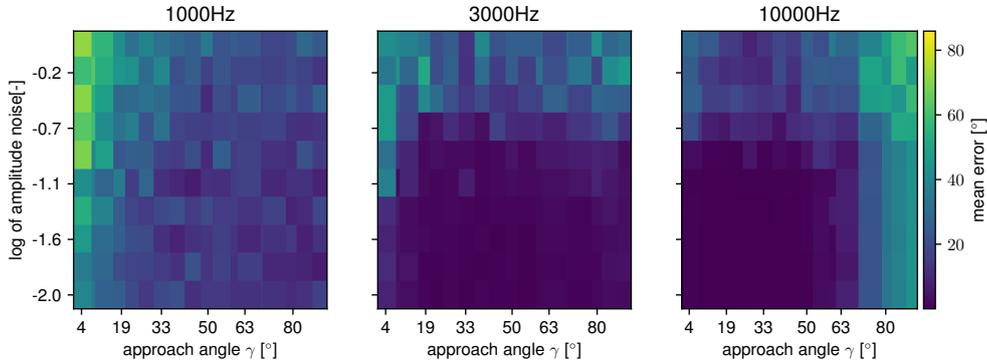


Figure 5.13.: Performance of distance slice algorithm for different levels of amplitude noise.

Non-uniform distance spacing In real experiments, we will not be able to sample the distance slice perfectly uniformly, as a robot’s movement is always subject to uncertainties. We study the effect of non-uniform sampling by adding Gaussian movement noise onto each step. The resulting performance for our 3 example frequencies is shown in Figure 5.14. The performance gets worse as we increase the frequency or increase the angle of approach. Both elements lead to a smaller spacing of samples with respect to the period of the signal to be detected. Therefore, a fixed movement noise will have a higher effect with higher frequencies and higher angles.

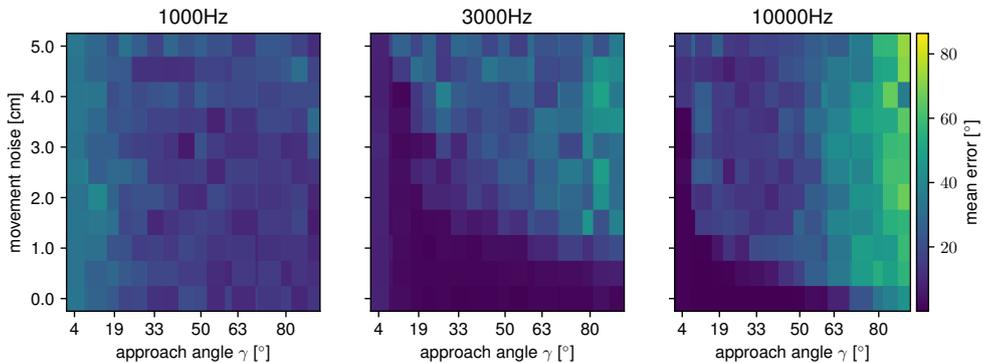


Figure 5.14.: Performance of distance slice algorithm for different levels of movement noise.

5.6. Experimental framework

We investigate the performance of the proposed algorithms on a microdrone. Among the many available options, we choose the *Crazyflie* drone by *Bitcraze* [75], as it is affordable,

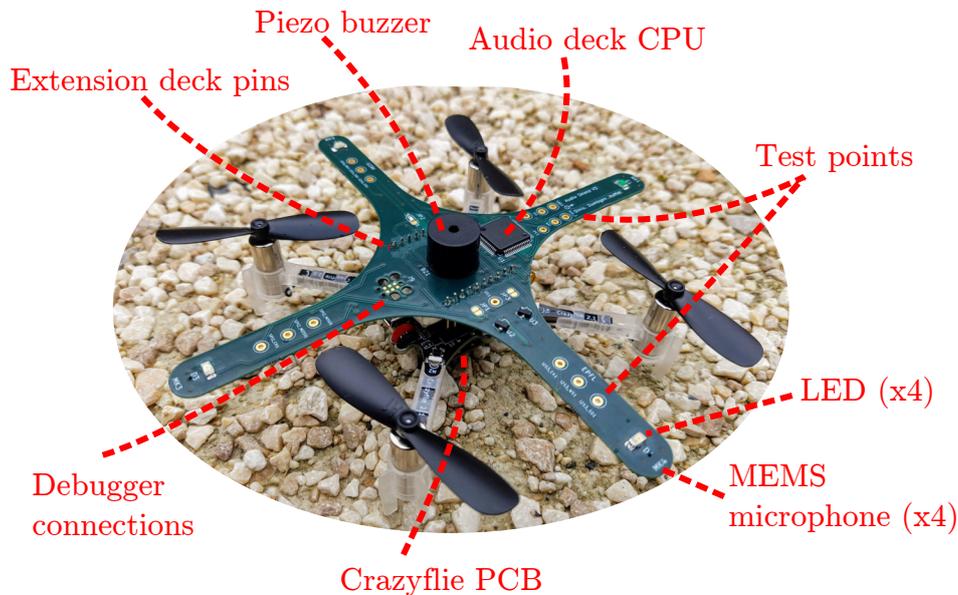


Figure 5.15.: Crazyflie drone including the custom audio deck used for the experimental validation.

developer-friendly and widely used in the research community. As none of the available extension decks include microphones, we develop a custom audio deck which allows us to record, pre-process and send the audio signals from four embedded microphones to the *Crazyflie* processor. The audio deck also contains a piezo buzzer which is more powerful and precise than the one included in the buzzer deck provided by *Bitcraze*. The drone including audio deck is shown in Figure 5.15. The ground station runs our custom Robot Operating Systems (ROS) processing pipeline. It includes means to record and replay measurements, as well as a real-time pipeline for wall detection. Additional modules such as path planning and obstacle avoidance could be easily integrated.

A visualization of the full pipeline can be found in Figure 5.16. In the following subsections, we give more technical details on each component of the pipeline.

5.6.1. Hardware design

In order to reduce the added weight and maximize flight agility, we create a compact printed circuit board PCB including the microphones, the buzzer, an onboard microcontroller STM32F405 and additional debugging modalities³. A picture of the drone including audio deck can be found in Figure 5.15. The most important components for this work are four Micro-Electro-Mechanical System (MEMS) microphones and a piezo buzzer as a sound source. Different microphone placements were tested with respect to the influence of propeller noise and flight dynamics, resulting in the shown compromise of microphones close enough to the drone to reduce off-center weight, but far enough from propellers for an acceptable SNR. The buzzer is in the exact center of the drone and microphones. The

³All documentation, including the PCB schematics, mechanical drawings and bills of materials, are available online <https://github.com/LCAV/crazyflie-audio/>

5. Dynamical sound-based wall localization

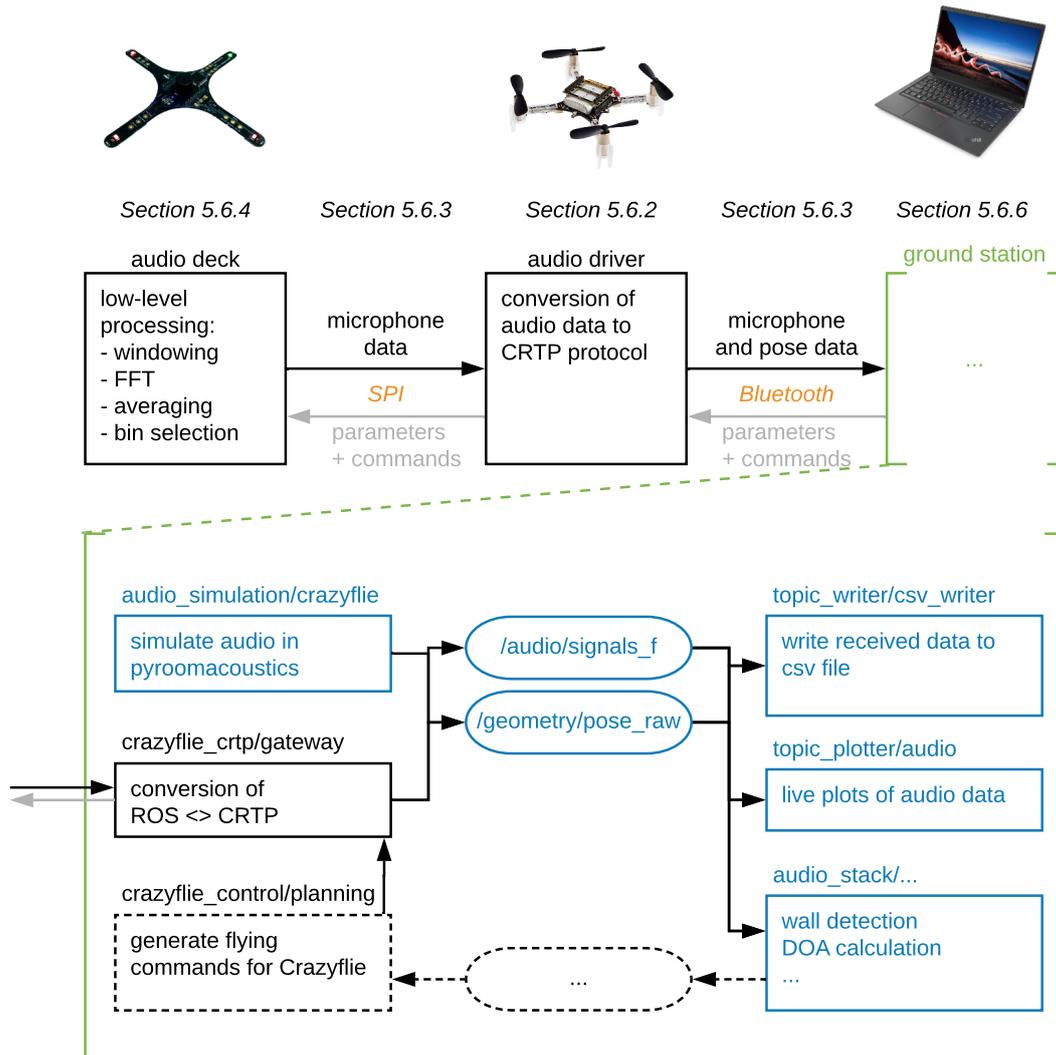


Figure 5.16.: Overview of the full experiment pipeline. The parts that are agnostic to the platform choice are marked in blue: the algorithms presented in this and the following chapter, for instance, can be applied to any robotic platform, as long as audio data and pose data are published to the correct ROS topics.

setup is thus highly symmetrical, meaning the performance of the proposed algorithms is not affected by the drone's orientation with respect to the wall.

5.6.2. Crazyflie firmware

To keep the *Crazyflie*'s modularity, most of the firmware is left untouched, and all audio-related functionalities are performed on a novel audio deck driver. Other minor modifications include the disabling of the battery-low buzzer signal, which could disturb experiments, functionalities to read the current buzzer frequency and motor commands. We also added

the new port for audio communication (see Section 5.6.3). The adapted firmware is available as a public fork of the official firmware.⁴ The driver is activated automatically when the drone registers the new audio deck through its EEPROM memory.

5.6.3. Communication

The audio deck driver communicates to the audio deck via a Serial Peripheral Interface (SPI), sending the current settings while receiving the newest audio data. The audio data is then converted to Crazy RealTime Protocol (CRTP) messages and sent to the ground station. The settings are communicated to and from the ground station via the standard logging and parameter framework of the *Crazyflie*.

The *Crazyflie* communicates with the ground station through the custom Bluetooth communication protocol CRTP, developed by *Bitcraze*. This communication channel is the bottle-neck of the whole system in terms of data throughput, and its capacity thus determines the rate at which we can obtain audio measurements. Since we are not interested in the full audio signals, as would be the case for speech processing applications, for instance, we can achieve a higher rate by filtering out the most important parts of the signals before sending, as described in the next section. The extracted audio signals are serialized and distributed across multiple CRTP messages. We use channels zero to two to signal the start, end, and type of the message, as explained in more detail in the documentation of our ROS package.

5.6.4. Audio preprocessing

The audio deck is equipped with a STM32F446 processor, which we use to pre-process incoming audio data and extract the most important features.

The signals at the four microphones are sampled at $F_S = 64\,000$ Hz and processed in four buffers of size $N = 2048$, which are filled via direct memory access (DMA). Using intermediate buffers of twice the size, we trigger an interrupt every time we have received N samples (when the intermediate buffer is half-full), which starts the processing, as outlined in what follows.

We start by doing a windowed FFT of each buffer. Different window types can be chosen from, depending on the use case. For wall detection, we are interested in an exact measurement of the amplitude, for which the flattop window is a good candidate. For applications requiring a high frequency resolution, a window such as the Hann window, which results in a thinner main lobe, is recommended. A comparison of the implemented window types is given in Appendix 5.A.

We choose the $K = 32$ most important frequency bins based on the algorithm of choice. In

⁴The latest version is available at <https://github.com/LCAV/crazyflie-firmware>.

5. Dynamical sound-based wall localization

our case, the frequency of interest is known: it is the current buzzer frequency, which we control. In the present analysis, we use a bin selection scheme filling the buffer with the K bins corresponding to the played buzzer frequencies. For applications where the frequency of interest is not known, one can choose the K bins of highest amplitude, or bins uniformly sampled in a given frequency range. Switching between different bin selection schemes is possible through the parameter setting framework of the *Crazyflie*.

For the selection of the highest-amplitude bins, we perform averaging of the total magnitude of the four microphone signals over time, using either a moving-average filter or exponential smoothing.

5.6.5. Buzzer signal design

We use a simple piezo-buzzer, which is driven by a square pulse-width modulation PWM signal. The chosen buzzer is more powerful than the one provided on the “buzzer deck” by *Bitcraze*, as shown in Figure 5.17.

To emulate a sine wave, we set the compare and capture register (CCR) value to half the auto-reload register (ARR) value. For a fixed ARR value, we can then play with different timer prescalers (PSC) to achieve different frequencies. The frequency in Hz is given by

$$f = \frac{f_{clock}}{PSC \times ARR}, \quad (5.31)$$

where f_{clock} is the clock frequency in Hz. Since both PSC and ARR are integer-valued, we see from (5.31) that we have a lower resolution at higher frequencies. This explains why the spectrum becomes more sparse at higher frequencies, as shown in Figure 5.18. A higher frequency resolution can be obtained by tuning both PSC and ARR, which is implemented on the custom audio-deck. We generate all possible values of PSC and ARR given a selected range of frequencies, and store one chosen pair per frequency in a lookup table.

Since the piezo acts as a low-pass filter, the audio signal produced is not square, but it is also not a perfect sine wave. Rather than having a single peak in frequency domain, we observe a series of harmonics. The harmonics do not harm us as we only send the frequency bins around the fundamental frequency.

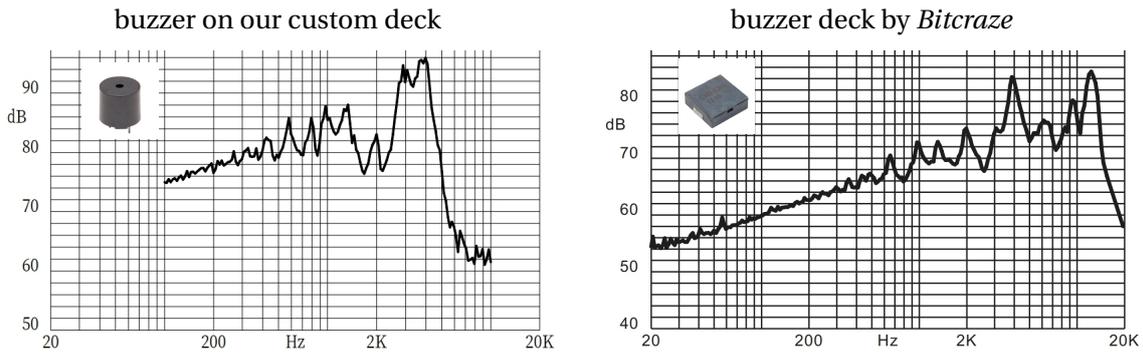


Figure 5.17.: Buzzer frequency responses of the used buzzer (left) compared to the buzzer on the deck provided by Bitcraze (right), retrieved from the respective datasheets (note the scale difference between the plots). Our own experimental evaluation of the Crazyflie's frequency response is shown in Figure 5.18.

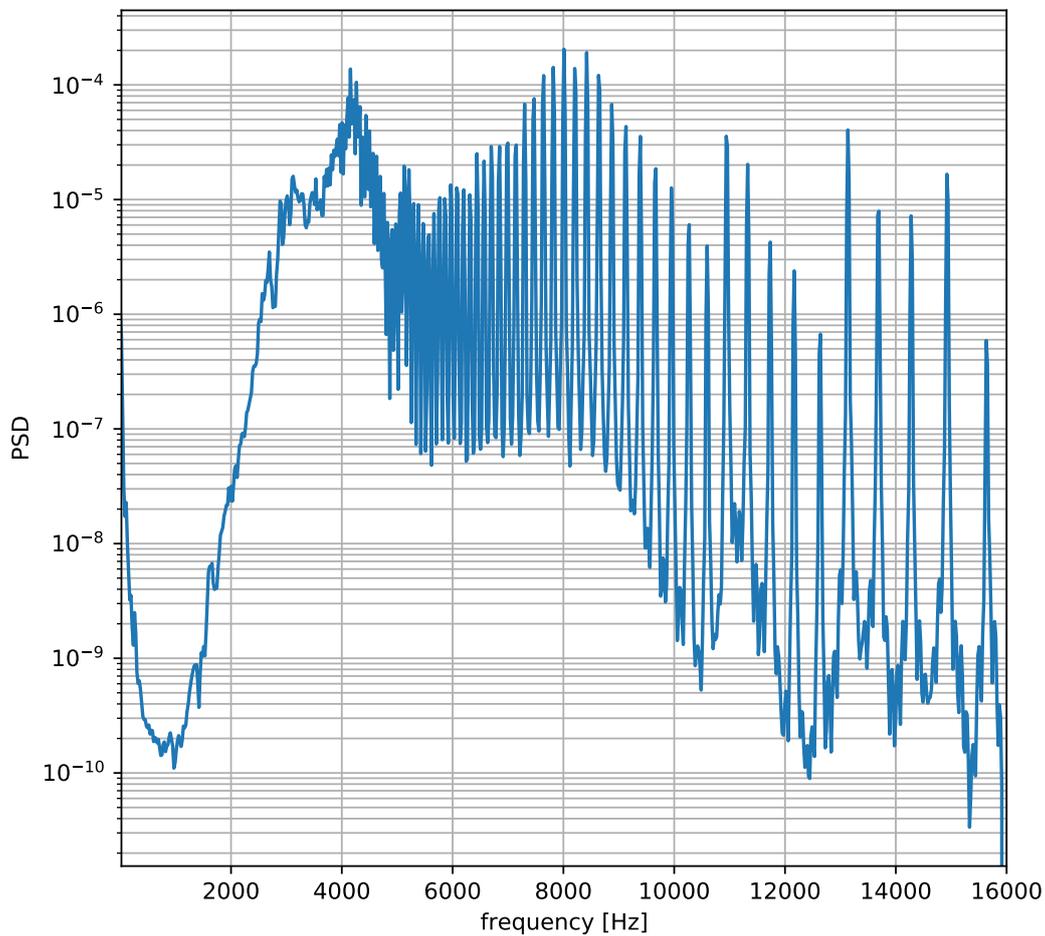


Figure 5.18.: Experimental evaluation of the Crazyflie's buzzer frequency response (power spectral density). We clearly see a decreased resolution at higher frequencies due to the relationship (5.31). We can also verify that the envelope roughly corresponds to the one provided by the suppliers, shown in Figure 5.17.

5.6.6. Ground station

We use as ground station a *ThinkPad Lenovo T470s laptop*, which runs the ROS processing pipeline. The capabilities of the package go beyond the algorithms described in this chapter: they include direction-of-arrival estimation, live plotting and logging, as well as packages allowing for the simulation of the experiments using the library *Pyroomacoustics* [184]. Most of the modules used are agnostic to the chosen robotic platform, and can thus be readily used for other use cases, as demonstrated by the successful adaptation of the pipeline to the ground-based *e-puck2* robot in Section 5.8. Indeed, all nodes marked in blue in Figure 5.16 are independent of to the platform choice, and only the communication conversion and the motion planning modules need to be adapted when porting to new platforms.

5.7. Experimental results

We separate our experiments in three parts. First, we experimentally validate that we can detect the interference pattern predicted by theory, in Section 5.7.1. Then, we test the frequency-slice and distance-slice algorithms in a variety of conditions in Sections 5.7.2 and 5.7.3, respectively.

5.7.1. Interference matrix

Stepper motor setup To sample the interference function, we use the stepper motor setup shown in Figure 5.19. The setup allows us to fix the drone at a given angle and move it in 1 cm steps perpendicularly away from the wall. At each distance, we play three buzzer sweeps, with frequencies chosen between 3000 and 5000 Hz. To study the effect of propeller noise, we repeat the experiment while giving constant thrust commands to all four motors, but keeping the drone in place.

Obtaining amplitude measurements Depending on the bin selection scheme, we either use fixed frequency bins, or we extract the strongest frequency bin at each time instant. In the latter case, the buzzer tone can vary slightly in both frequency and amplitude over time, because of scalloping and measurement noise. To get a consistent estimate, we merge frequency bins lying less than 50Hz apart, we remove sparse measurements (defined as frequencies for which we have less than two measurements) and remove outliers (amplitudes lying more than two standard deviations away from the median). The measurements before and after cleaning, with and without propeller noise, are shown in Appendix 5.B.

Calibration Before applying our algorithms, we need to calibrate our measurements (5.9) for the unknown gain function $T(f)$, which introduces a frequency-dependent scaling.

For the frequency slices, we first calibrate the function $T(f)$ using the measurements from all



Figure 5.19.: Visualizations of the stepper motor setup, where the Crazyflie drone is attached to a rotating platform mounted on a linear rail, both of which can be operated with stepper motors.

distances for each used buzzer frequency f_n . The raw data and fitted curve for an example frequency are shown in the left plot of Figure 5.21. For gain calibration, we are merely interested in the offset of this function. Because of the symmetry of the model around the gain, and since we are sampling more than a full period, we can simply take the median over all measurements as an estimate of the gain. The obtained gain functions $T(f)$, calibrating either all microphones individually or doing one global calibration, are shown in Figure 5.22. On the right of Figure 5.21, we see that the fitted amplitude is above the average standard deviation of the magnitude measurements up to ca. 30 cm, so we expect good performance in this regime according to our simulation results.

Results Combining the amplitude measurements for different distances, we generate the experimental interference matrices. The result, shown in the left-most plot of Figure 5.20, clearly shows a frequency-dependent gain. After calibration, the resulting interference matrices after calibration are close to the values predicted by theory.

Having experimentally validated the interference pattern, we can now make use of the methods to localize walls using these measurements.

5. Dynamical sound-based wall localization

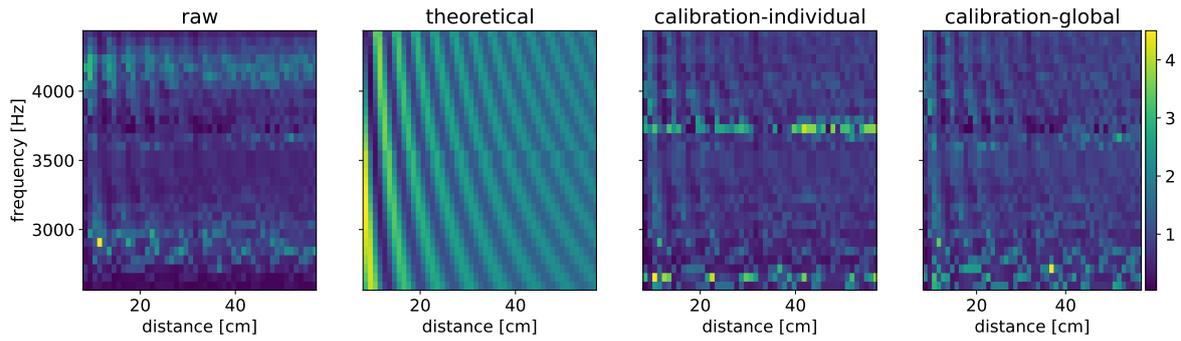


Figure 5.20.: *Interference matrices for fixed angle and microphone 1. From left to right: raw values, theoretical values, values after calibrating each microphone individually, values after global calibration.*

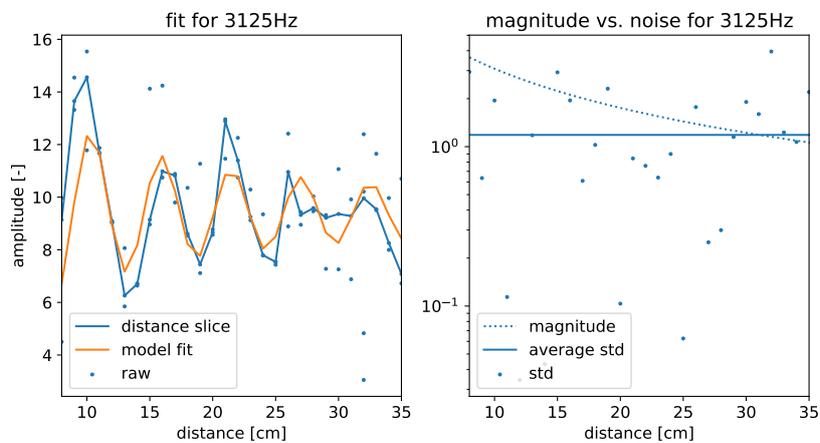


Figure 5.21.: *Left: fit of model to experimental data along distances. Right: comparison of magnitude of measurements with their standard deviation per distance.*

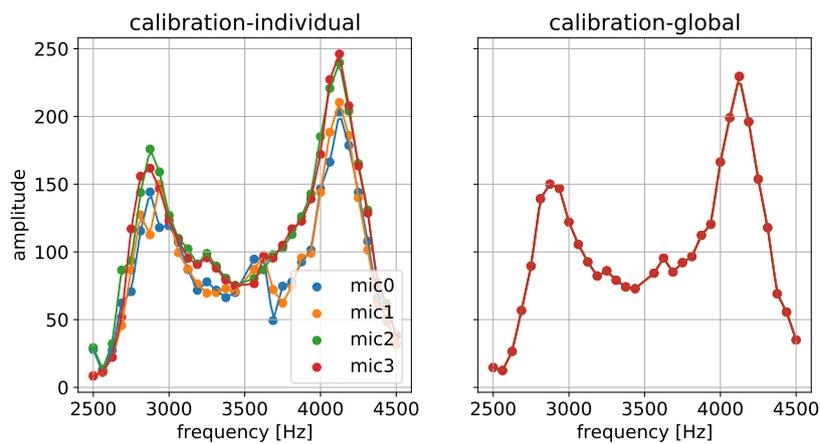


Figure 5.22.: *Calibrated joint microphone-speaker gain $T(f)$, using the median over all distance measurements, and calibrating all microphones individually (left) or globally (right).*

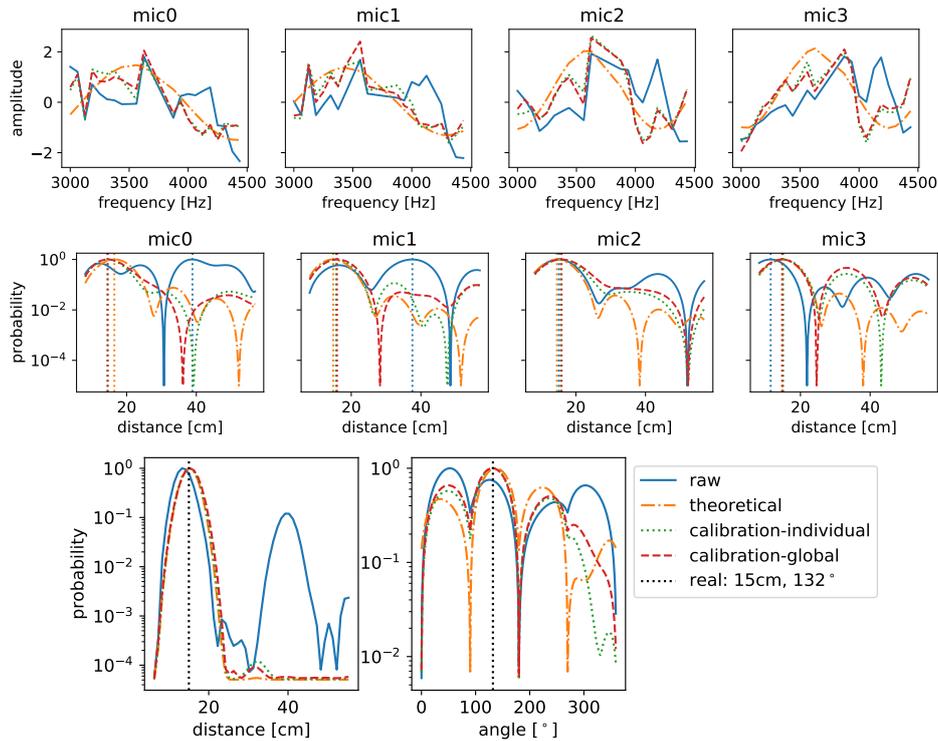


Figure 5.23.: *Raw analysis of frequency slice algorithm, for the example distance of 15 cm. We impose a uniform prior on the azimuth angle, and use the distance estimate for the angle algorithm*

5.7.2. Frequency slice algorithm

Stepper motor study First, we study the performance of the frequency slice algorithm on the stepper motor data. Therefore, we apply our algorithm for each distance from the wall. In other words, we traverse the distance-frequency matrix in Figure 5.20, performing inference on vertical slices.

To visualize the algorithm, we first show the pipeline for one example distance in Figure 5.23. Comparing the raw data with the calibrated data in the top row, we observe that calibration is crucial to remove wrong oscillations (for instance for microphone 1). After calibrating, we see that each microphone yields probability distributions that are roughly consistent with the true distance (second row). Note that we assume a known angle for plotting, since the distance is more intuitive than the path difference. Combining all distributions, the final estimates for the distance and angle of the wall are shown in the last row. The distance distribution is unambiguous, while the angle distribution exhibits strong sidelobes. Since the same sidelobes are present when using theoretical (noise-free) data, we believe that this is a consequence of the symmetry of the setup.

Finally, we perform the pipeline from Figure 5.23 for all distances, and extract the absolute error of the peak distance and peak angle compared to the ground truth. The results are shown in the first row of Figure 5.24. As expected, the distance error deteriorates with distance, but we have close-to-perfect estimates up to ca. 30 cm from the wall. Note that the theoretical data does not always yield zero error due to limited frequency resolution at low

5. Dynamical sound-based wall localization

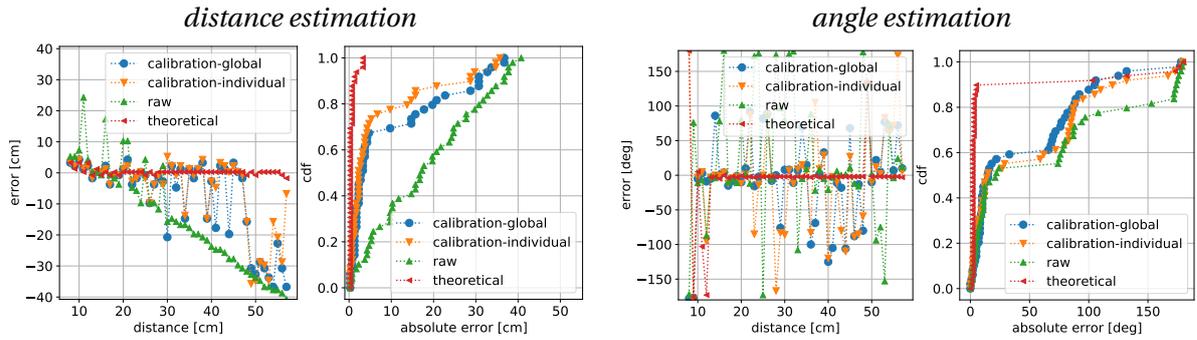


Figure 5.24.: Performance of frequency slice algorithm. We impose a uniform prior on the azimuth angle, and we use the distance estimate as delta prior for the angle estimation.

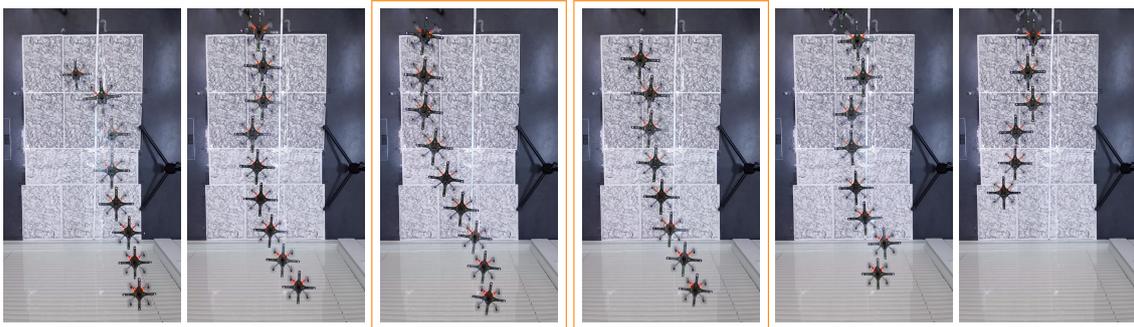


Figure 5.25.: Overlaid pictures of drone's movement during 6 different wall approach experiments. The wall is at the bottom of the pictures. To visualize the movement, we extract the dynamic part of the picture and overlay a selection of frames. The experiments highlighted in orange are analyzed in Figure 5.26.

distances. The angle estimate is less reliable, but for more than 50% of the cases, we obtain an error close to zero. When the angle is off, it tends to be off by a lot: we explain this from the strong sidelobes at 90 degrees spacing observed in Figure 5.23.

Flying drone results We apply the frequency-slice algorithm in the real-world setting of a flying drone. This setting bears two additional difficulties compared to the stepper motor study. First, because of stabilizing movements, the propeller sound during hovering is less stable than during the stepper motor study (in which the propellers rotate at constant speed). Secondly, even when there is no nominal speed, since this stabilization is not perfect, there is significant lateral motion during each frequency sweep. Effectively, this means that we sample the interference function with a spread around the vertical slice corresponding to the nominal distance. We had predicted this behaviour in Figure 5.4.

We perform the following experiment: the drone approaches the wall at constant speed while continuously emitting frequency sweeps. Figure 5.25 shows different instances of this experiment. For each frequency sweep, we infer a probability distribution over distances, assuming the drone is static during the sweep. When using the same calibration scheme as for the stepper motor experiments, the results are poor: looking at Figure 5.26, we see

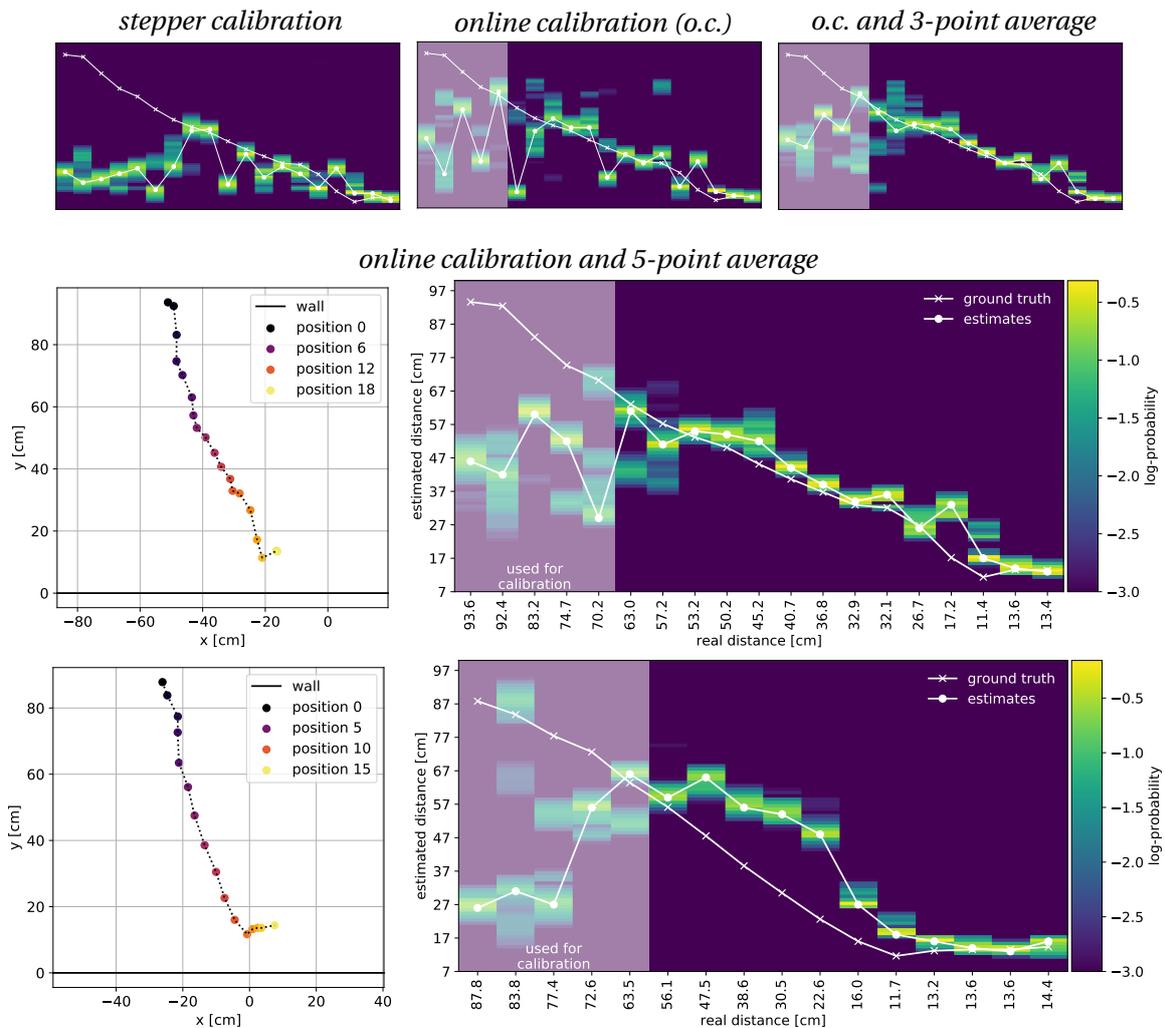


Figure 5.26.: Results from drone approaching the wall at constant speed, emitting frequency sweeps. The bottom two rows show on the left the location of the drone at the beginning of each sweep and on the right, for each distance, the resulting probability distribution. We perform online calibration (o.c.), and average over the first 5 latest measurements. The top row shows the resulting distributions for the first experiment, when using standard calibration and no or less averaging, respectively.

that the probability distributions do not peak at the correct distances. We assume that this is due to the propeller noise altering the characteristic microphone-buzzer gains, rendering the calibration from the stepper motor experiments invalid.

The remedy lies in performing online calibration. Assuming the drone takes off at a safe spot, *i.e.* far from a wall, we can calculate its current frequency response by averaging over the first few responses while hovering. Since the hovering thrust is different depending on the current battery level and drone payload, this ensures a more appropriate calibration. Indeed, using the new calibration, the probability distribution take the correct shape after calibration. After combining measurements over a fixed time window, we obtain a sufficiently high accuracy in distance estimation for safe operation of the drone, as shown in the bottom two rows of Figure 5.26.

5. Dynamical sound-based wall localization

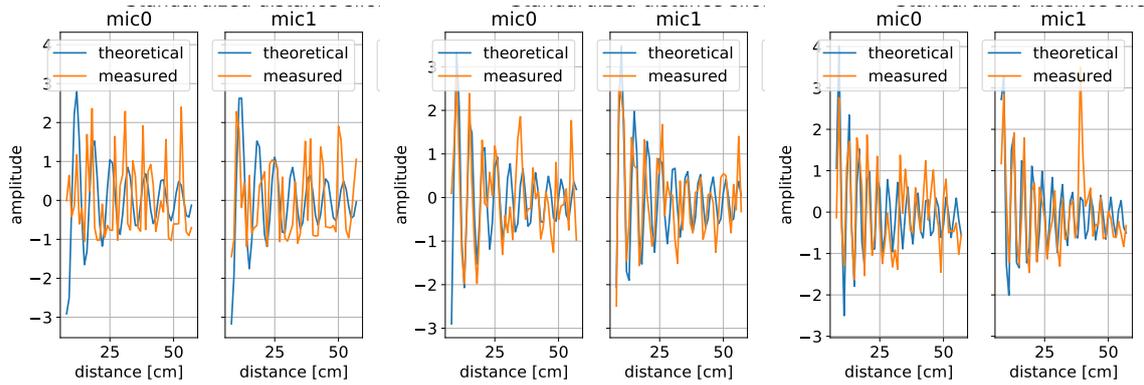


Figure 5.27.: Distance slice for two different microphones at frequencies 2562 Hz (left pair), 3375 Hz (middle pair) and 4500 Hz (right pair).

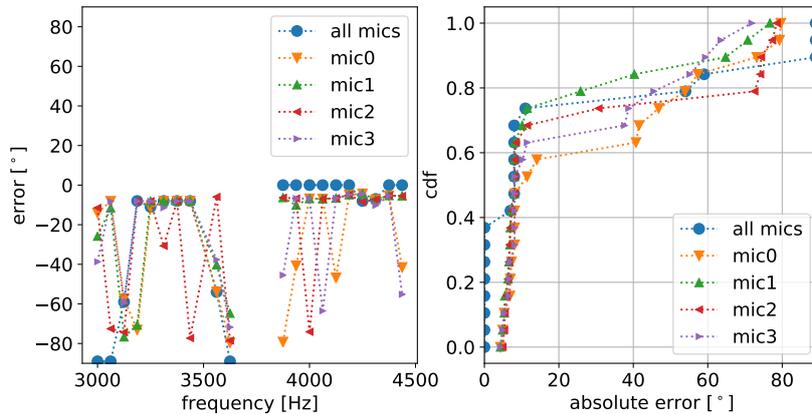


Figure 5.28.: Performance of distance slice algorithm, per microphone. Absolute error of angle of approach over all frequencies (left) and the corresponding cumulative distribution function (right).

5.7.3. Distance slice algorithm

Stepper motor study Similarly to above, we first test the distance-slice algorithm on the stepper motor data. The standardized distance slices for two different frequencies are shown in Figure 5.27. We note that for the two higher frequencies, the period of the distance slices almost perfectly match the theoretical predictions, but that there is a slight phase offset. Since we only extract the strongest interference frequency, the phase offset is no problem. For the lowest frequency, the slice is significantly more noisy, which might be due to a lower signal-to-noise ratio.

Next, we compute the angle error over all frequencies and for all microphones, reported in Figure 5.28. We again observe poor performance for the lower frequencies, but starting from 3000 Hz the angle is recovered almost perfectly, and the remaining small errors are eliminated once all microphones are combined.

Flying drone results As a final experiment, we apply the algorithm on a flying drone. The drone approaches the wall with a constant speed, starting from a bit more than 1 m distance, and at different angles. Two sample experiments are shown in Figure 5.29. Figure 5.30 shows, in the top row, the 5 different trajectories from consecutive experiments. We split each experiment in three to investigate the effect of wall distance. We observe that all the angle of the closest sections (bottom lines) are detected with almost zero error, but that the left *vs.* right side is not correctly resolved. As we get further from the wall, the detection becomes less reliable and breaks down completely above ca. 50 cm.



Figure 5.29.: Pictures of distance slice experiment. Left: side view of the drone flying (starting position). Right: drone approaching the wall at two different angles.

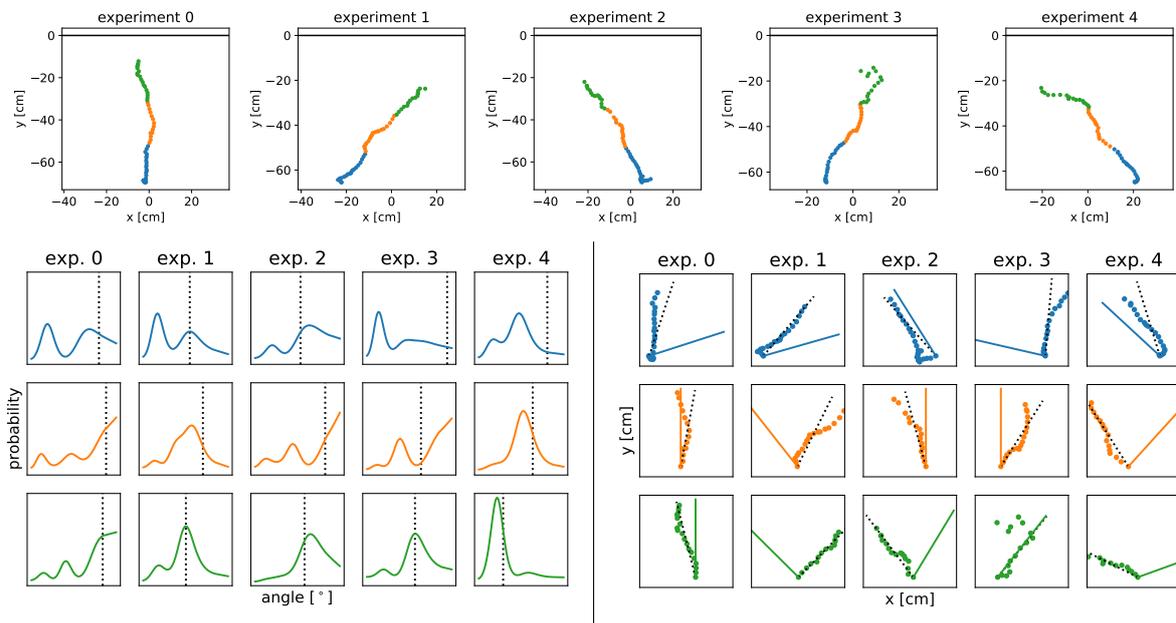


Figure 5.30.: Flying drone results of distance slice algorithm. The top row shows the trajectory of 5 different experiments, where the wall is along x with $y=0$. Below, we plot on the left the probability distributions and on the right the portions of the trajectory they correspond to, including the real distance (dotted line) and the estimate (solid line).

5. Dynamical sound-based wall localization



Figure 5.31.: Left: the *e-puck* version 2 [143], a robot for education originally developed at EPFL and maintained by GCTronic. Right: the robot in action during the wall approach experiment (overlaid frames).

5.8. Preliminary results on the *e-puck2* robot

Since the drone's unstable motion renders the problem very challenging, we use the proposed algorithms to equip the educational *e-puck2* robot (see Figure 5.31) with audio-based wall detection capabilities. This application might be less appealing than the bat-like drone, but since the robot comes with four microphones and a little buzzer, it comes at no additional cost. In addition, this application allows us to showcase the generalizability of our modular ROS implementation.

Just like for the drone, we found that the interference matrices of the *e-puck* robot show strong variations across frequencies, and that they vary strongly between microphones. After using the same calibration schemes as for the *Crazyflie* drone, we do however get excellent results for the angle and the distance estimation, as presented in Figure 5.32. We believe that by studying the algorithm on this platform, in particular by pushing the limits of movement while sensing, we will gain further insights in the behavior of the algorithms on the drone.

5.9. Conclusion and ongoing work

In this chapter, we have proposed a solution for audio-based wall detection on a microdrone. Building on a simple model for sound propagation and reflection, we have introduced an algorithm that can recover the distance and angle of the closest wall based on interference patterns across frequency. Implementing this algorithm on the *Crazyflie* drone equipped with our custom audio deck, we obtain excellent wall localization results in controlled conditions, when moving the drone in a well-behaved manner. Using additional measures including an online calibration scheme and smoothing, we can extend these results to a flying drone.

Using the same model, we have developed an algorithm for determining the angle at which the drone is approaching a wall. While movement is a nuisance for the frequency-slice algo-

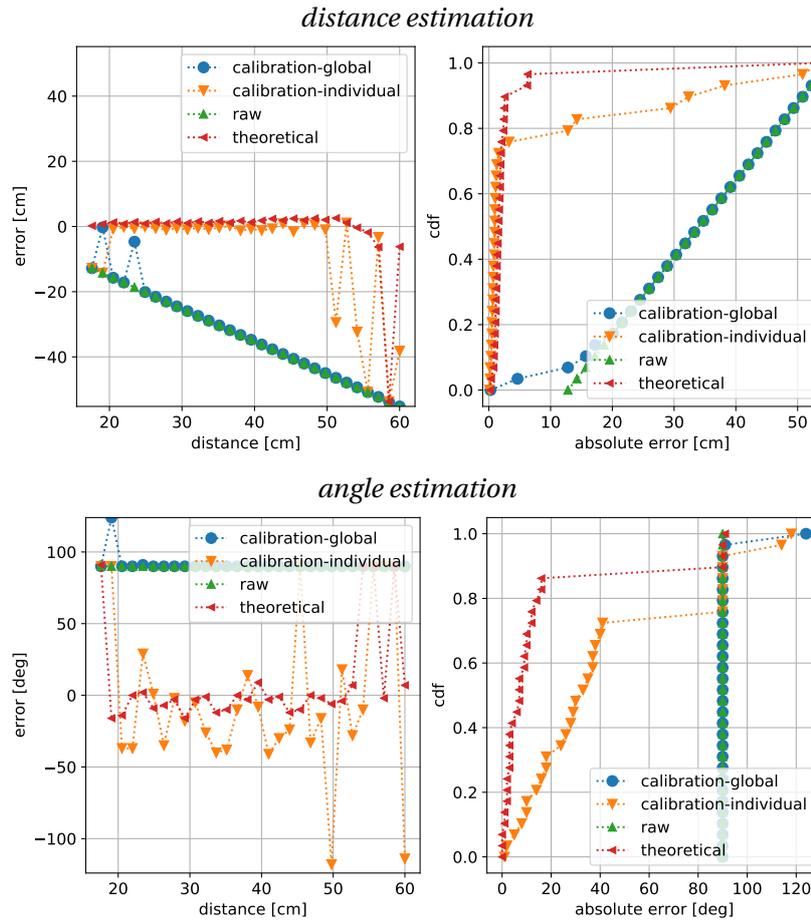


Figure 5.32.: Performance of frequency slice algorithm on the *e-puck* robot for distance and angle estimation. Shown are on the left the absolute errors and on the right the corresponding cumulative distribution functions.

rithm, it can be exploited for sampling the interference function in space. We obtain robust results in both controlled and flying experiments, up to an ambiguity whose resolution is the topic of ongoing research.

Compared to common methods for echolocation, the proposed algorithm works with magnitude measurements only and does not require exact knowledge of the input signal, in particular its phase. Therefore, the algorithm overcomes synchronization requirements and is amenable to low-key hardware. Indeed, its functioning is tested on the *Crazyflie* drone and the ground-based *e-puck2* robot, both of which are small platforms (less than 13 cm in diameter) with limited computational power and sensing capabilities. In particular, both platforms are only equipped with cheap MEMS microphones and piezo buzzers, but the proposed algorithms can extract enough information for wall localization from the noisy signals.

Because of significantly better results on the stepper-motor setup and on the *e-puck* robot compared to the flying experiments, we believe that improving the stability of the drone's flight would be very beneficial for the algorithm's performance. If a performance gap persists

5. Dynamical sound-based wall localization

even with more stable flights, a frequency-slice algorithm which exploits movement could be designed.

In ongoing work we are investigating how to incorporate the wall estimates into a full navigation solution, including localization and room mapping through factor graph inference. This opens up interesting questions for future research, such as the extension of factor graphs from point features to planes, and the incorporation of non-Gaussian measurement models. Finally, we are investigating how the method behaves in the presence of more than one reflecting close-by wall or obstacle, which might be required to generalize the navigation solution to more complex environments.

Appendix

5.A. Window study

Figure 5.33 shows the received audio signals for the first 12 played notes of a uniform sweep. Each note is plotted in a different color, and we study the effect of different windows applied to the time signal, before performing the FFT. Note that we also study the result with and without propeller noise.

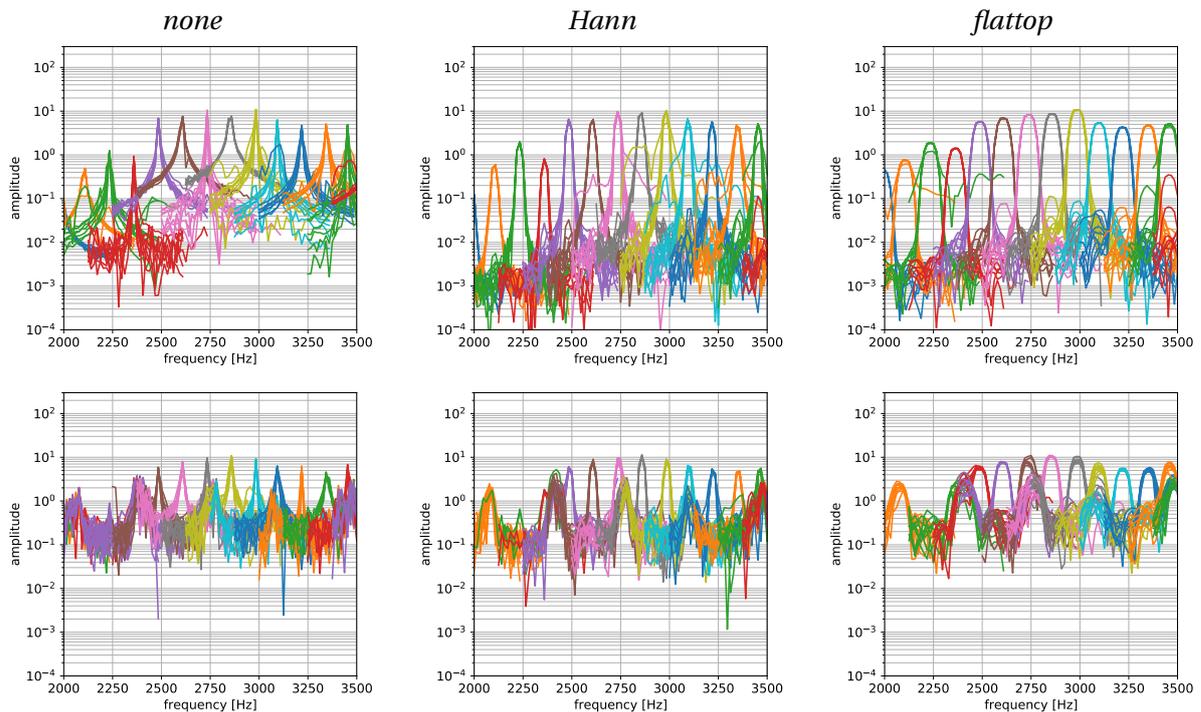


Figure 5.33.: Analysis of performance of different windows on frequency responses received, without motors (top row) and with motors (bottom row). Different colors correspond to different tones, and we plot multiple buffers per note.

5.B. Interference matrix cleaning

Figure 5.34 shows the result of the interference matrix before and after the post-processing described in Section 5.7.1. We can clearly distinguish the interference patterns predicted by theory, even after turning on the motors.

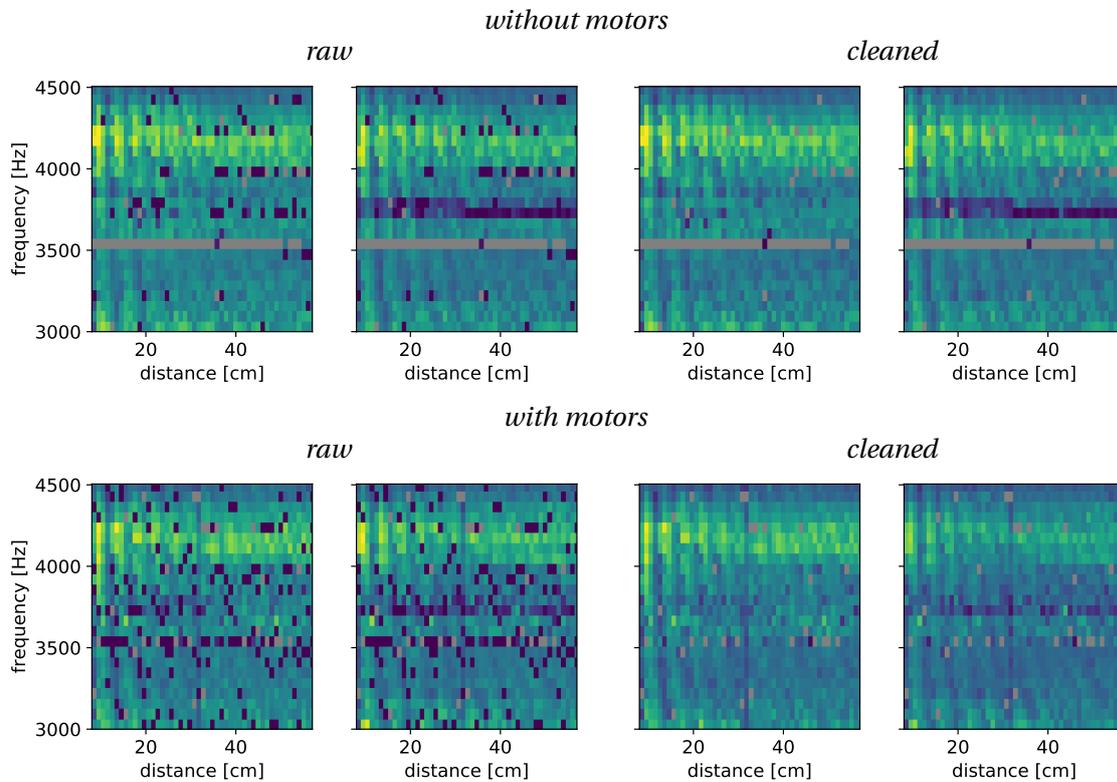


Figure 5.34.: Measured interference matrices for the first two microphones before and after cleaning.

6 | Dynamical sound-source localization

И људе послушај, и своје памети имај
Listen to people, but have your own ears.

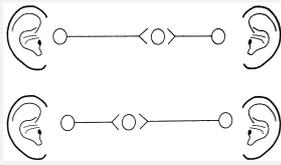
— Serbian saying

While I am writing this very sentence, I hear a fly buzzing around my desk. Even though my gaze is focused on the screen, I can roughly tell where it is, at least as long as it is flying and making this annoying sound — the task becomes significantly harder when I try to locate the fly once it has landed on my cluttered desk.

How do I perform this task? Researchers have investigated for centuries how humans locate a target using sound information only. They have found three main ingredients for sound source localization (SSL), and all of them exploit the geometry of our ears and head. The first two, inter-aural time differences (ITD) and inter-aural level differences (ILD), use the parallax between the ears. Sound waves propagate through space with a fixed velocity and their intensity decreases as they travel further from the source. Therefore, sound impinges on our two ears with an ITD and an ILD that are both characteristic of the sound's direction. For sound coming from the right, for instance, the right ear hears the signal louder and earlier than the left ear. The way we actually compute these inter-aural differences is a fascinating topic in itself, and covered in the info box for the interested reader. The third main cue that we use for SSL is the way our head geometry affects the sound response from different directions. This property can be captured in the so-called head-related transfer function (HRTF), and is used by most hearing animals to determine the direction [187].

This chapter contains unpublished work. It is included here because we believe it to contain promising directions of research. Of particular interest is the tight coupling of motion with sensing, which allows to view sensor motion as an aid rather than a detrimental factor for localization. The method is directly applicable to any moving device, in particular the drone platform in Chapter 5, and is included in the open-source software described therein.

Delay and subtract/sum for vision and audition in the human brain



Visualization of binaural neurons in the auditory pathway, introducing two different delays [81].

It is crucial for our survival to detect and track moving objects. In modern days, an approaching car or a burglar in the dark are examples of moving dangers, in cavemen and cavewomen days an approaching predator or a hiding prey were targets that needed to be detected. To detect movement, we need to compare what we see now to what we saw before. Thinking about the brain as a computer, we could somehow store the previous image frame and take its difference with the current frame. What evolution has taught us is much more memory efficient: Put very simply (details can be found in [81]), there are many neural paths from the eyes to the visual cortex, each of which takes a slightly different time. Therefore, at each instant, the brain receives stimuli from different times in the past, and it can “subtract” them to figure out if and where something has moved. The same concept is thought to be applied for delaying and comparing auditory stimuli to retrieve the angle of a sound source (see Section 6.3.1): we can infer the inter-aural time difference by detecting which paths lead to simultaneous spiking [81].

Many algorithms have been designed for microphone arrays to perform SSL which use the same physical principles as evolution has taught us. For instance, some methods use a known or learnable scattering medium around the microphones, similar to the HRTF [61, 160]. More common are approaches that use the time delay between multiple microphones to localize sound. Some of these methods operate in time domain and rely on finding the time-difference-of-arrival using zero-level-crossing or onset times. More methods work in the frequency domain and with phase differences, leading to so-called cross-correlation techniques [173].

There is one more trick which humans were found to use to locate sound which has not been adopted in common algorithms. It was shown that in certain cases, to disambiguate where a sound is coming from, humans use their head movement. With only two auditory sensors there is a front-back ambiguity, resulting in what is called the cone of confusion. When no other stimuli or priors exist, we have difficulties determining if a sound is coming from the front or from the back, as both yield the same ITD and ILD (ignoring the relatively small HRTF effects). It was shown in extensive studies that we can use our head movement to resolve this ambiguity [211]. Other studies show that when a low-frequency source is placed in the upper median vertical plane, its elevation can only be estimated by using horizontal head motion [164]. When the source has higher frequency contents, head movement helps with localizing it in any region, as shown in the same study.

To this day, most existing algorithms which integrate motion in SSL are based on the typical Bayesian filtering paradigm from robotics: we obtain a first estimate, we move, we improve the initial estimate (see Chapter Chapter 2). In this chapter, we seek to integrate motion more tightly into the SSL process: given the raw sound signals from subsequent positions, a movement estimate and timing information, we seek to obtain a more accurate localization

Notation for this chapter	
N	number of poses
m	number of microphones
M	number of sources
$s^{(l)}(t)$	sound source signal from source l
$\theta_n^{(l)}$	DOA of source l at pose n
$x_n^{(i)}(t)$	received signal at microphone i and pose n
T	buffer duration
$y(t)$	beamformed output signal
$\mathbf{a}(\theta, f) \in \mathbb{C}^m$	steering vector in direction θ at frequency f
$\mathbf{h}(f) \in \mathbb{C}^m$	beamforming weights
$\mathbf{R}(f) \in \mathbb{C}^{m \times m}$	covariance matrix

result than what we could obtain by a higher-level, iterative scheme. We do this by creating a virtual microphone array, consisting of all locations the microphone has visited during a certain time window. By artificially increasing the number of microphones through this process, we reduce ambiguities, obtain better angular resolution, and are thus able to detect more sound sources simultaneously (or conversely, distinguish desired from interfering sources more effectively).

Using movement for enhanced sensing capabilities might be an under-explored idea for audio, but it is quite common in other sensing modalities. For example, as mentioned in the introduction of this thesis, most smartphones nowadays use the fact that the camera is not perfectly still during a hand-held capture to increase the sharpness of an image [216] or to estimate a depth image [142]. The principle of obtaining higher-resolution from movement is also widely used in so-called synthetic aperture radar [13] and sonar [91].

In this short work-in-progress chapter, we present a method to use the movement of a microphone array to achieve better angular resolution and higher localization accuracy than static SSL methods. Our proposed method is based on standard and low-cost algorithms for SSL and thus suitable for implementation on embedded platforms with limited computing power.

6.1. Problem setup

We denote by M the number of sound sources, emitting sound signals denoted by $s^{(l)}(t)$, with $l = 1 \dots M$. We measure sound from m microphones, placed in a known configuration. We denote the signal recorded at the n -th pose by $x_n^{(m)}(t)$, $m = 1 \dots M$, where $t \in [t_n, t_n + T]$, with T the buffer duration. We denote the DOA of source l in the robot's reference frame at pose n by $\theta_n^{(l)}$. We accumulate the relative movement estimates between poses n and $n + 1$, $l_i \in \mathbb{R}$ and $\beta_i \in SO(D)$ for $n \leq i < n + 1$, to yield the overall movement estimates l_n and β_n , as described in Chapter 3. From these estimates, we can calculate the rotation matrix $\mathbf{R}_n \in SO(D)$ and relative movement vector $\Delta_n \in \mathbb{R}^D$ between every pair of audio

6. Dynamical sound source localization

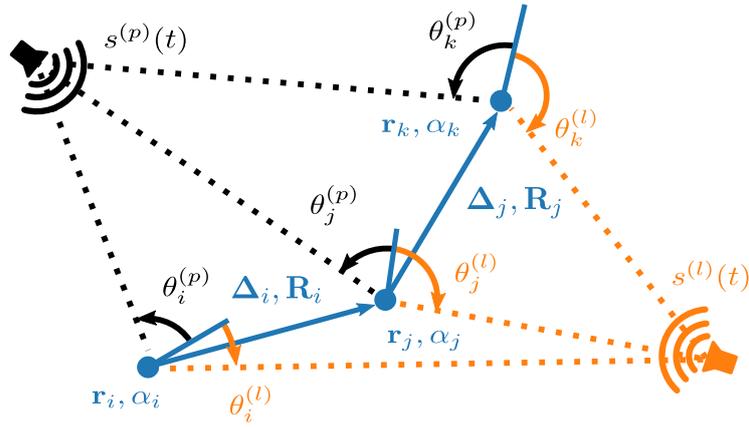


Figure 6.1.: *Visualization of problem setup. We use the approximately known movement of a microphone array (in blue, each dot corresponds to one array pose r_n, α_n) to estimate the direction of arrival (DOA) of M sound sources with unknown, but self-correlated, signals $s^{(l)}(t), l = 1 \dots M$.*

measurement times. The above quantities are visualized in Figure 6.1. The problem we address in this chapter is the following:

Problem 4. *Given the m microphone signals $x_n^{(i)}(t)$ for $i = 1 \dots m$, relative movement estimates Δ_n and \mathbf{R}_n , at known measurement times t_n for $n = 1 \dots N$, determine the angles of the M sound sources, $\theta_n^{(l)}$, for $l = 1 \dots M$.*

6.2. Related work

The idea of using the movement of a sensor array to emulate a larger virtual array can be found in a variety of areas. Starting with the highest-frequency signals, in Synthetic Aperture Radar (SAR), radar scans from a sequence of positions of the carrier vehicle, for instance an aircraft or a satellite, are combined to create a sharper estimate of the target. In Synthetic Aperture Imaging Lidar SAIL [19], image resolution limits imposed by diffraction are overcome by combining multiple positions. In burst processing, which runs on most smartphones nowadays to enhance image quality, images taken from slightly different poses are combined to improve resolution, dynamic range, or signal-to-noise level. Moving from electromagnetic waves to pressure waves, in underwater sonar applications, ideas of SAR have found their way into a variety of now widely-used Synthetic Aperture Sonar (SAS) methods [91].

The above approaches work on signals with considerably higher frequencies than audible sound. For instance, Radar works with electromagnetic waves between 3 MHz and 110 GHz, and since SAS is an imaging technique it usually operates in the ultrasonic range (other than passive sonars used to detect distant targets). Rather than combining the raw signals in a coherent way after correcting for phase shifts induced by movement and different recording times, these systems thus operate mostly on the formed “images”, combining them in a way

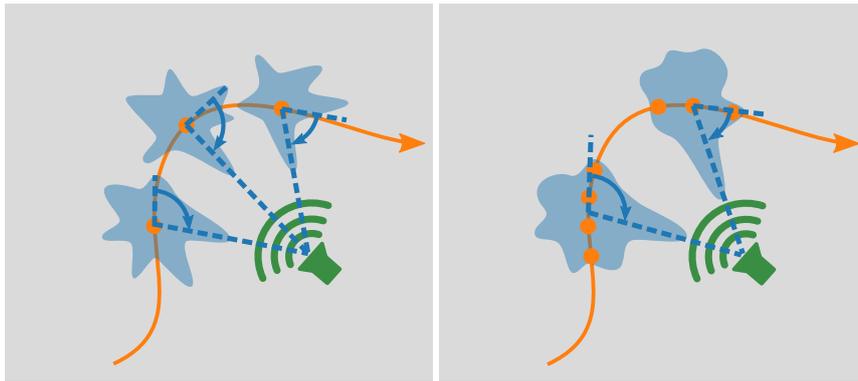


Figure 6.2.: Visualization of classic DOA (left) on a moving device and the proposed method (right).

as to sharpen images. In the audible range, since we can easily implement microsecond timers, we can hope to have a good enough synchronization to actually time-align the raw signals, thus truly creating a virtual array.

Some people have tried to solve the same problem, but while doing so, have imposed constraints on the movement of the device. The assumption of ETAM (extended towed array measurements) [195] for instance, used for towed sonar arrays attached to submarines, is that the array moves in such a way that some new positions overlap with previous positions. Using this fact, the phase shift can be corrected for, assuming the signal is correlated in different time intervals. The sensor overlap and signal correlation constraints are relaxed in [106], a method developed for both ultrasound and radio-frequency signals. Here, an auxiliary sensor is used in addition to the moving array, and beamforming is done between the array and the auxiliary sensor at each time instant. It is desirable not to impose such constraints as to not limit the versatility of the robot.

Moving over to SSL, a plethora of methods have investigated how to smooth angle estimates obtained on a moving platform, in order to yield consistent estimates, using classical filtering techniques from robotics (see Section 2.3). Existing methods use for instance particle filters [127, 179] or extended Kalman filters [70, 227]. As sketched in the left of Figure 6.2, these methods get spectra and angle estimates at each position individually, and then combine these results in hindsight. Some of these methods are again restricted to certain movement types such as constant velocity [70] or a predefined sequence of motions [107]. More importantly, these methods do not exploit the available information optimally: for instance, because of the resolution limits imposed by number and spacing of microphones, we may not be able to resolve all sources at each time stamp individually, and nothing guarantees that a sequence of such suboptimal beamforming steps would fix this information loss. To exploit the full power of the motion, it is thus necessary to work on the raw data. An approach proposed by Cheung *et al.* goes in this direction, using the factor graph representation of the SLAM problem with factors for each audio measurement [42]. However, rather than using powerful beamforming algorithms on the raw data, the factor graph is solved using a standard, suboptimal nonlinear solver.

In this chapter we aim to bridge the gap between the somewhat restrictive movements of the

6. Dynamical sound source localization

synthetic aperture literature and the filtering-based algorithms from SSL mobile robotics. We show how, by simply keeping track of the recording times of all microphone signals, we can create a virtual microphone array that achieves almost identical performance to an equivalent, larger, static array. We thus get one spectrum and angle estimate, exploiting a sequence of positions, as shown in Figure 6.2 on the right. Our method works for any kinds of movement as long as an accurate measurement of the movement can be obtained and as long as the source signal is correlated with itself. Since we generate signals that resemble those of a virtual microphone array, well-established algorithms from the SSL literature can be applied to the time-aligned microphone signals. In particular, we study two standard beamforming approaches which are appropriate for miniature robotics applications as they are computationally cheap and non-parametric *i.e.* do not require any knowledge of the source signal. We explain them in the next Section.

6.3. Background

In this section, we introduce two DOA algorithms for a fixed microphone setup which we will later exploit for the moving-microphone setup. We study their properties, in particular the spatial aliasing behavior and angular resolution, which can both be improved with movement. We focus on one pose n and thus drop the index in what follows. In the frequency domain, we can write the received signals at the M microphones by [197]:

$$\mathbf{x}(f) = \begin{bmatrix} x^{(1)}(f) \\ x^{(2)}(f) \\ \vdots \\ x^{(m)}(f) \end{bmatrix} = \begin{bmatrix} \mathbf{a}(\theta^{(1)}, f) & \dots & \mathbf{a}(\theta^{(M)}, f) \end{bmatrix} \begin{bmatrix} s^{(1)}(f) \\ \vdots \\ s^{(M)}(f) \end{bmatrix} + \mathbf{e}(f) \quad (6.1)$$

$$= \mathbf{A}\mathbf{s}(f) + \mathbf{e}(f), \quad (6.2)$$

where $\mathbf{a}(\theta_i, f) \in \mathbb{R}^m$ is the array steering vector at frequency f , and $\mathbf{e}(f) \in \mathbb{R}^m$ contains independent realizations of random noise. Assuming the microphones have unit gain in each direction and for each frequency, it is given by

$$\mathbf{a}(\theta_i, f) = \begin{bmatrix} e^{-j2\pi f \tau_{i1}(\theta_i)} & e^{-j2\pi f \tau_{i2}(\theta_i)} & \dots & e^{-j2\pi f \tau_{im}(\theta_i)} \end{bmatrix}^\top, \quad (6.3)$$

where $\tau_{ik}(\theta_i)$ denotes the propagation time from source i , placed at angle θ_i , to microphone k .

We are interested in steered-response DOA algorithms (see Section 6.3.3), which rely on the concept of beamforming. Put simply, these algorithms steer the “attention” of the microphone array in all directions and record the received power. To steer the attention to different directions, a beamformer applies different weights $\mathbf{h} \in \mathbb{C}^m$, consisting of the applied gain and delay, to the different microphones, yielding

$$y(f) = \mathbf{h}^* \mathbf{x}(f), \quad (6.4)$$

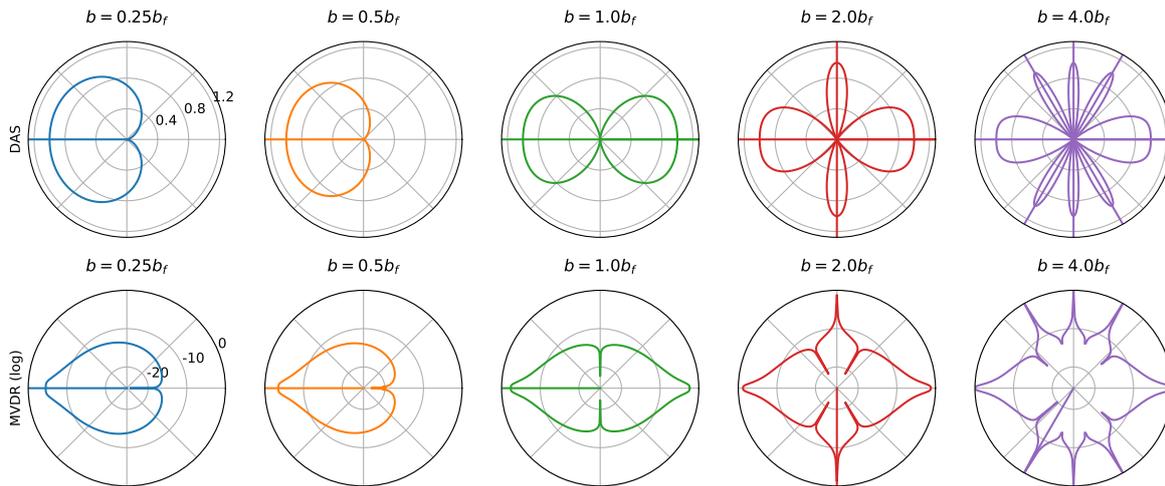


Figure 6.3.: *Angular spectra for two microphones, distanced with increasing baselines b , and a source playing at 0 degrees and 1000 Hz. The maximum baseline b_f is given by 0.1715 m, as by (6.14).*

where $*$ denotes the Hermitian transpose.

We will study reasonably low-key beamforming algorithms that could run on any robotic platform. For readability, we drop the parameter f . For each frequency bin, we create a beamformer independently, and we denote the resulting spatial energy spectrum by $E(\theta)$. In general, this energy is given by

$$E(\theta) = \mathbb{E}\{|y|^2\} = \mathbf{h}^* \mathbb{E}\{\mathbf{x}\mathbf{x}^*\} \mathbf{h} = \mathbf{h}^* \mathbf{R} \mathbf{h} = \mathbf{h}^* \mathbf{A} \mathbf{s} \mathbf{s}^* \mathbf{A}^* \mathbf{h} + \mathbf{h}^* \boldsymbol{\sigma} \mathbf{I} \mathbf{h}, \quad (6.5)$$

where we have introduced $\mathbf{R} := \mathbb{E}\{\mathbf{x}\mathbf{x}^*\} \in \mathbb{R}^{m \times m}$, also known as the covariance matrix. We explain in Section 6.3.5 how it can be estimated in practice.

6.3.1. Delay-and-Sum (DAS)

The delay-and-sum (DAS) filter time-aligns all microphone signal so that they would be in phase were they coming from a certain direction θ . It is one of the most commonly used beamforming filters and sometimes called “conventional beamformer”. To allow for comparison with the MVDR beamformer, we use a less common derivation here [197]. In fact, the DAS beamformer can be interpreted as the solution to the following optimization problem:

$$\begin{aligned} \mathbf{h}^{DAS} &= \arg \min_{\mathbf{h}} \mathbf{h}^* \mathbf{h} \\ &\text{s.t. } \mathbf{a}(\theta)^* \mathbf{h} = 1, \end{aligned} \quad (6.6)$$

which has as solution

$$\mathbf{h}^{DAS} = \mathbf{a}(\theta)/m. \quad (6.7)$$

6. Dynamical sound source localization

The corresponding angular energy spectrum is given by

$$E^{DAS}(\theta) = \mathbf{a}(\theta)^* \mathbf{R} \mathbf{a}(\theta). \quad (6.8)$$

The DAS beamformer is simple to implement but has relatively poor resolving power as it exhibits a relatively wide main peak and high side lobes: see for instance Figure 6.3, top row.

6.3.2. Minimum Variance Distortionless Response (MVDR)

For better resolving power, the MVDR beamformer, also commonly called the Capon beamformer, can be used: by exploiting the actual contents of the signal, we create a beamformer with an overall smallest possible energy (thus “minimum variance”), thus suppressing side lobes, without affecting the signal coming from the desired direction (thus “distortionless”).

We formulate the MVDR beamformer as a special case of the Linearly Constrained Minimum Variance (LCMV) beamformer. This beamformer is obtained by solving the problem

$$\begin{aligned} \mathbf{h}^{LCMV} &= \arg \min_{\mathbf{h}} \mathbf{h}^* \mathbf{R} \mathbf{h} \\ \text{s.t. } \mathbf{C}^* \mathbf{h} &= \mathbf{c}, \end{aligned} \quad (6.9)$$

which admits the closed-form solution

$$\mathbf{h}^{LCMV} = \mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^* \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{c} \quad (6.10)$$

The MVDR beamformer is obtained by setting

$$\mathbf{C} = \mathbf{a}(\theta) \text{ and } \mathbf{c} = 1. \quad (6.11)$$

The corresponding angular energy spectrum is given by

$$E^{MVDR}(\theta) = \frac{1}{\mathbf{a}(\theta)^* \mathbf{R}^{-1} \mathbf{a}(\theta)}. \quad (6.12)$$

Compared to the DAS beamformer, the MVDR method has a significantly narrower main peak and reduced side lobes, as can be confirmed for instance in Figure 6.3, bottom row.

6.3.3. Steered-response DOA

Both DAS and MVDR provide us with beamformer coefficients and an estimated energy for a given direction, in the chosen frequency band f . We can repeat this process for a number of frequency bands in a chosen set, $f \in \mathcal{F}$. Depending on the application at hand, this

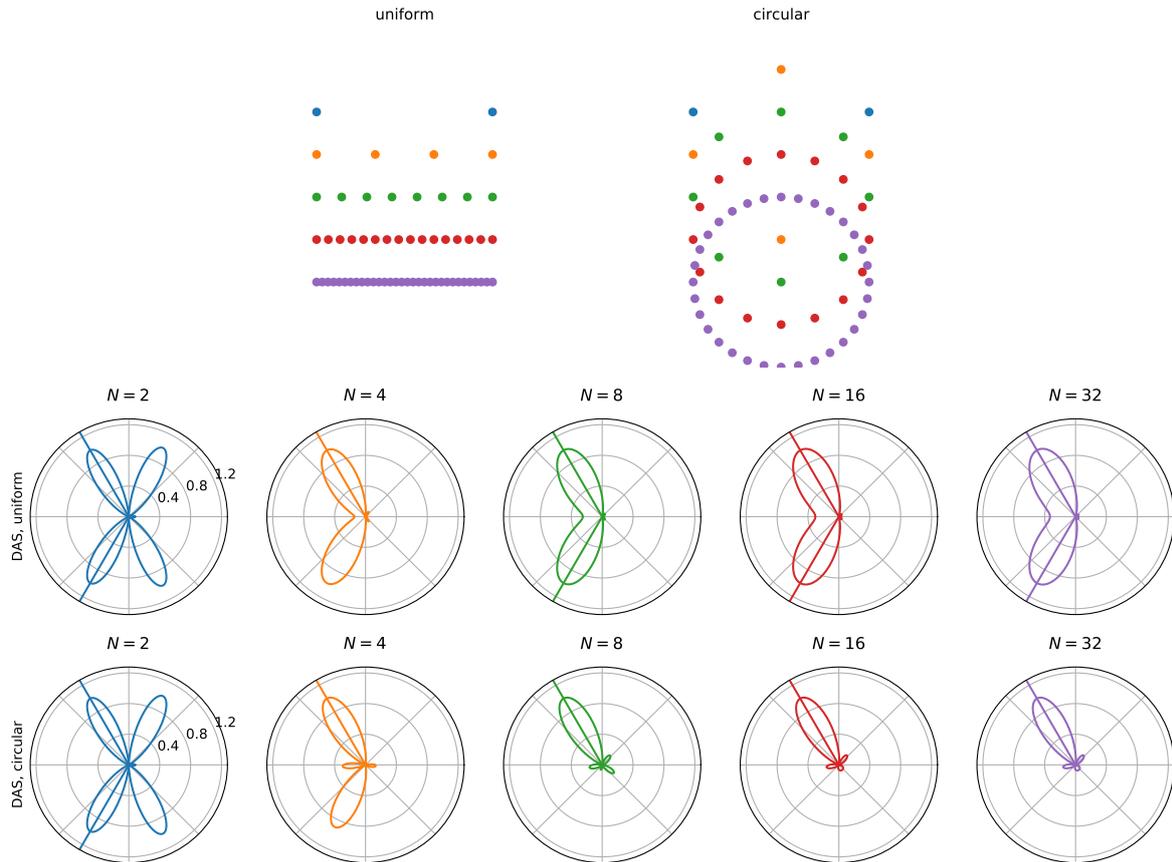


Figure 6.4.: *Angular spectra of linear and circular arrays with increasing number of microphones (denoted by N here) and fixed overall baseline. There is one far-field source at 120 degrees, playing a mono-frequency sound of 1000 Hz.*

information can be exploited in different ways. For (acoustic) imaging, we can report the energy for all possible azimuth and elevation angles for a chosen set of frequencies, yielding one 2D image per frequency (similar to one channel of an RGB image). If we want to listen in a certain direction, we can simply combine the signals using the beamformer coefficients for this direction. In the current chapter, we are interested in DOA estimation, which is obtained by finding the maximum-energy direction $\hat{\theta}$ over all frequencies:

$$\hat{\theta} = \arg \max_{\theta} \sum_{f \in \mathcal{F}} E(\theta, f), \quad (6.13)$$

where we have made the dependency in f explicit for this section only. In other words, we steer our beamformer in different directions and record the received energy. The direction with most energy corresponds to the direction of the source.

6.3.4. Recovery limits

The performance of a steered-response DOA algorithm depends on several factors, which we explain in more detail in this section.

6. Dynamical sound source localization

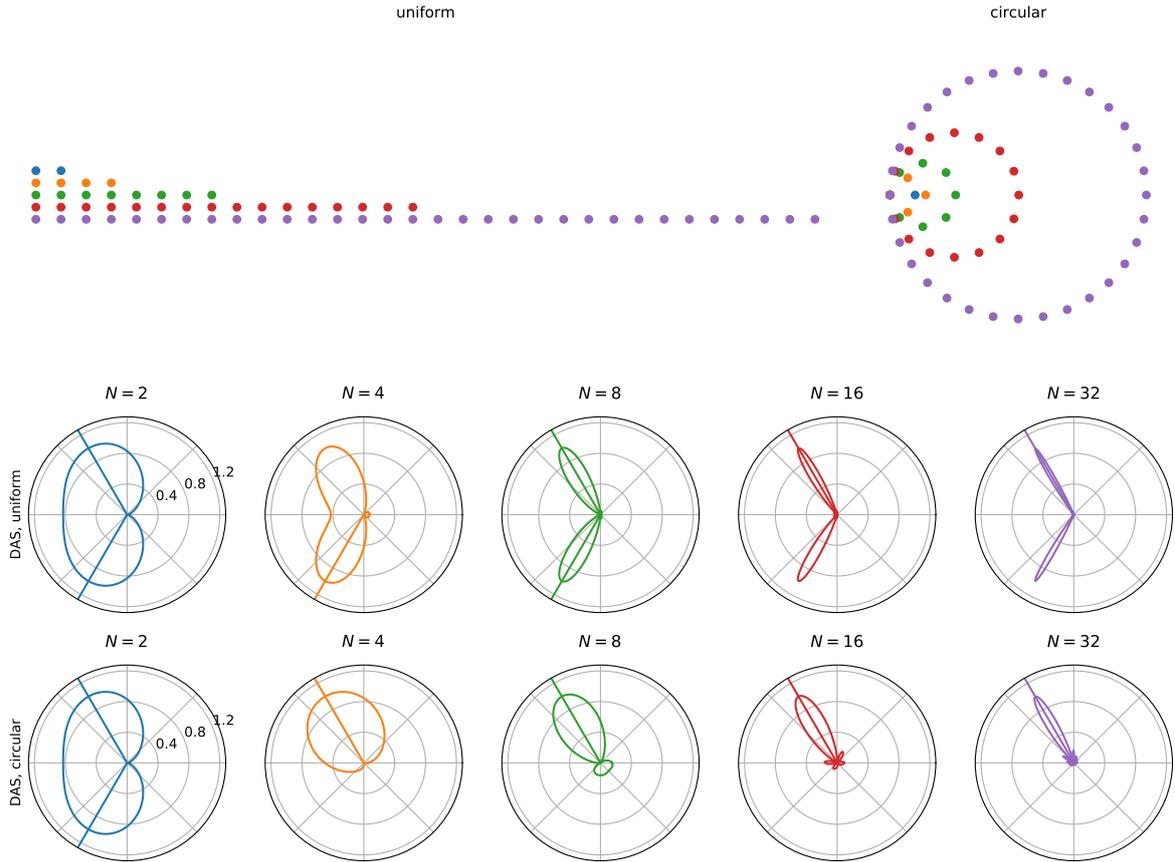


Figure 6.5.: *Angular spectra of linear and circular arrays with increasing number of microphones (denoted by N here) and fixed between-microphone baseline. There is one far-field source at 120 degrees, playing a mono-frequency sound of 1000 Hz.*

Baseline and frequency content First, the width of the main lobe of the beamformer for a fixed frequency is inversely proportional to the baseline of the microphone array. This affects the source separation capacity of the beamformer: If the order of magnitude of the source separation in degrees is smaller than the beamwidth, they do not appear as distinct peaks in our estimated spectrum.

Conversely, the width of the main lobe for a fixed baseline is inversely proportional to the frequency. Therefore, the higher the frequency content of the sources, the easier it is to separate them.¹

Ideally, we would thus like to use very large-baseline microphone arrays and high-frequency signals for source localization. However, there is a limit in how far we can push both parameters. For a given frequency, we start to “undersample” the signal in space above a certain baseline, after which we see aliased peaks appearing in our spectrum. The valid range of baselines b is

$$0 < b \leq b_f := \lambda = c/2f, \quad (6.14)$$

¹ For delay-and-sum on linear arrays, for instance, the beamwidth null to null is given by: $\theta_{BW} = 2 \sin^{-1} \left(\frac{c}{b_0 M f} \right)$ with b_0 the spacing between two microphones and M the number of microphones [124].

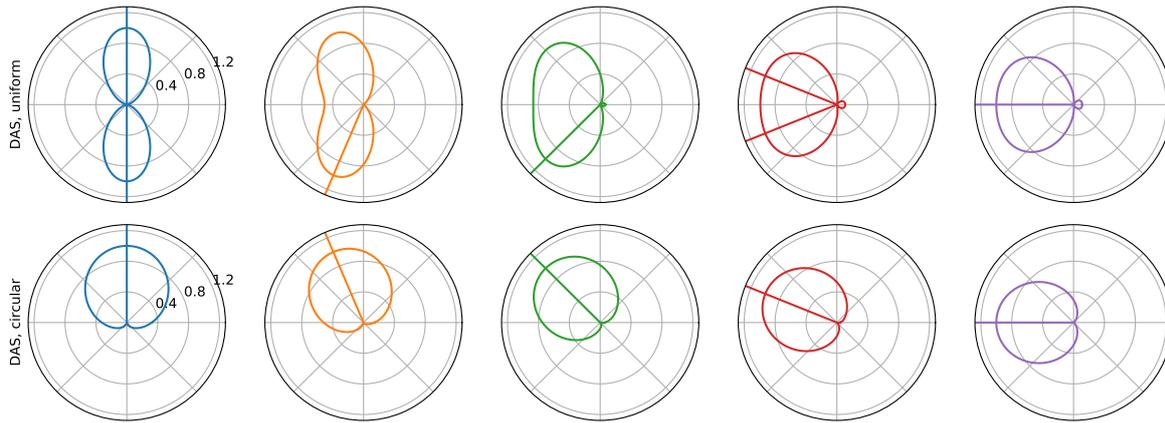


Figure 6.6.: *Angular spectra for four microphones in a uniform (top) or circular (bottom) array, for a mono-frequency source at 1000 Hz, placed at different directions.*

where c is the speed of sound and f the source frequency. We have introduced b_f for the maximum baseline without spatial aliasing. Equivalently, for a fixed baseline, we can only sense up to the threshold frequency $f_b := c/2b$. This phenomenon is visualized in Figure 6.3 that we saw earlier, where we plots baselines around the aliasing limit, for a fixed frequency $f = 1000$ Hz.²

Number of microphones and array geometry Another important factor on the performance of beamforming algorithms is the number of microphones used and their geometry. We study the performance of two typical array geometries, uniform linear and uniform circular. We fix the overall size to the limit baseline at frequency 1000 Hz, and increase the number of microphones fitted within that size. The source is fixed at an angle 120° . The results, shown in Figure 6.4, show no significant performance increase with number of microphones. The main advantage in oversampling without increasing the overall baseline stems from increasing SNR in case the signals are noisy, which we do not study in this simulation. However, if we fix the spacing between two microphones and grow the overall size of the array as we increase the number of microphones, shown in Figure 6.5, then the beamshape improves significantly as we add more microphones, ensuring a better angular resolution and source separation performance.

Finally, it is interesting to note the dependence of the beamshape on the angle. This is much more pronounced for the uniform array than for the circular array, as shown in Figure 6.6. Since we do not control from which direction the sound is coming from, it is desirable to have a uniform beamshape, so the array geometry is important when “growing” our virtual array. This point is of particular importance when we control the robot’s movements in order to achieve a certain performance, as further discussed in Section 6.7.

²For these simulations, we generate signals according to the far field assumption. This is done to investigate the desired effects only. For the more realistic assumptions, we simulate signals without this assumption (but with a source reasonably far).

6.3.5. Estimating the covariance matrix

Having only finite measurements of the signal, we cannot compute the expected value in $\mathbf{R} = \mathbb{E}\{\mathbf{x}\mathbf{x}^*\}$ exactly, but need to resort to an approximation. Using the method laid out in [153], we compute an estimate for the covariance matrix as follows: For a fixed buffer duration T , we compute the periodogram of the signal $P_T^{(mn)}(\omega) = \frac{1}{2T} X^{(m)}(j\omega)^* X^{(n)}(j\omega)$, where $X^{(m)}(j\omega)$ is the discrete-time Fourier transform (DTFT) of the signal $x^{(m)}(t)$. We repeat the process for a number of realizations of the signal. In our case, we cannot “go back in time” to get a new realization, so we use subsequent buffers. If the signal is ergodic, then both variants lead to the same result. The Einstein-Wiener-Khinchin theorem states that the value of $P_T^{(mn)}(\omega)$, as T goes to infinity, approaches the Fourier transform of $R_{mn}(\tau)$ in the frequency band ω . Finally, since we cannot compute the true DTFT, we resort to a simple discrete Fourier transform (DFT).

6.4. Incorporation of movement

We have seen that the main limiting factor for the performance of classical DOA algorithms is the size of the microphone array. To maximize performance, an engineer is thus faced with the difficult trade-off of choosing a large enough microphone array, ensuring the desired DOA performance, while satisfying computational constraints (more microphones means more data to process) and limiting the size and weight. Both are typically very constrained in mobile robotics, in particular on drones, as they can be directly related to the power consumption, stability and agility of the robot.

The key insight is that by using the mobility of our platform, we get away with fewer microphones without compromising DOA performance. As we have seen, existing virtual array methods typically use a predefined and regular movement of the microphone array, which is not practical for real-life robot dynamics. The regular movement in these methods is needed in order to correctly time-align measurements from different microphones. We can get rid of this requirement by exploiting timing information directly. As we will show in the next section, we can combine measurements at different positions and times by aligning them using the time information. Under some assumptions on the source signal, which we discuss in the next section, the time-aligned signals provide a good approximation of the signals from the equivalent virtual microphone array composed of all visited microphone positions.

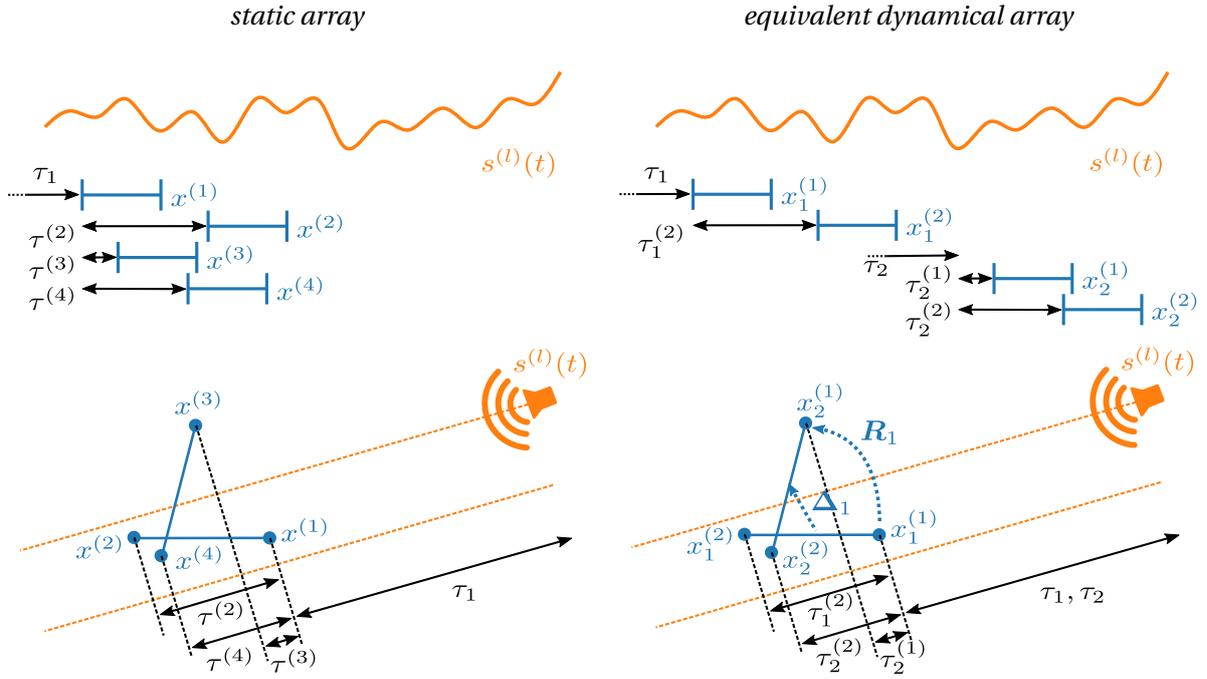


Figure 6.7.: Static microphone array with 4 microphones compared to equivalent dynamical microphone array with two microphones and two positions.

6.4.1. Time alignment method

The microphone signals at each time index $n = 1 \dots N$ can be written as:

$$\begin{aligned}
 x_n^{(1)}(t) &= \sum_{l=1}^M s_l(t - \tau_n^{(1,l)} - \tau_n) \\
 &\dots \\
 x_n^{(m)}(t) &= \sum_{l=1}^M s_l(t - \tau_n^{(m,l)} - \tau_n),
 \end{aligned} \tag{6.15}$$

where we have introduced $\tau_n := t_n - t_0$, the measurement delay with respect to some global offset t_0 (e.g. the time the device was switched on), and $\tau_n^{(k,l)}$ for the time delay from the source signal l to microphone k with respect to some absolute reference point, e.g. the array center at the first position. This delay is calculated using the movement estimates R_n and Δ_n , respectively.

To explain our method, we show in Figure 6.7 an example of what portions of the source signal we measure, using (6.15). We compare with a static microphone array at all visited positions, for which we get an instantaneous signal from all microphones (*i.e.* the delay $\tau_n = \tau_1$ for all n).

The intuition behind the proposed method is that, after removing the delays corresponding to the different recording times (τ_1 to τ_N), the “leftover” delays in the signals correspond to

6. Dynamical sound source localization

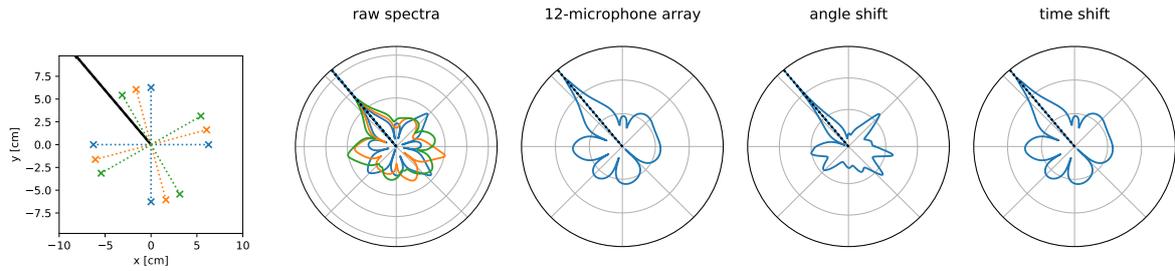


Figure 6.8.: *Performance in locating a high-frequency source (4000 Hz) for different beamforming methods, where the minimum microphone spacing is above the spatial aliasing limit. We compare the individual spectra, independently computed at each pose and aligned to the first pose (“raw spectra”), with a 12-microphone array at the visited positions, the naive combination of the angle-aligned spectra (“angle shift”, see Section 6.4.2) and the proposed combination of the time-aligned spectra (“time shift”, see Section 6.4.1).*

the relative delays due to the different relative microphone positions. However, for this to work, there needs to be a correlation between the signals recorded at different time instances. If the signal was white noise, for instance, then by definition there is no correlation between two different portions of the signal. In this chapter, we focus on mono-frequency signals; for which the assumption holds.

6.4.2. Angle alignment method

In the following sections, we compare the time alignment method to a simple angle alignment method. The angle alignment method follows the Bayesian filtering paradigm from robotics: we get a first angular spectrum from our initial position (prior distribution), we move and update the prior, and then we correct the prior according to the new measured angular spectrum, yielding the posterior distribution. Concretely, we combine a set of angular spectra $E(\theta_n)$, $n = 1 \dots N$, each expressed in their local reference frames, by first expressing them in the first reference frame $n = 1$, using the relative rotation estimates R_n for $n = 1 \dots N - 1$. Then, we combine the angle-aligned spectra by summing over n .

6.5. Performance evaluation

Before investigating the performance of our algorithm in realistic conditions, we showcase three aspects of it in the ideal scenario without any noise. We conduct the simulation with the microphone array geometry of the Crazyflie drone equipped with a custom audio deck. The specifications of this platform are explained in more detail in Chapter 5. For the present analysis, what counts is that we have $m = 4$ microphones placed in a uniform array with a radius of 6.27 cm. Different poses of the array are shown for instance in the left plot of Figure 6.8. To reduce space, we report the results using the (more powerful) MVDR algorithm, but all trends discussed can be equivalently observed with the DAS algorithm.

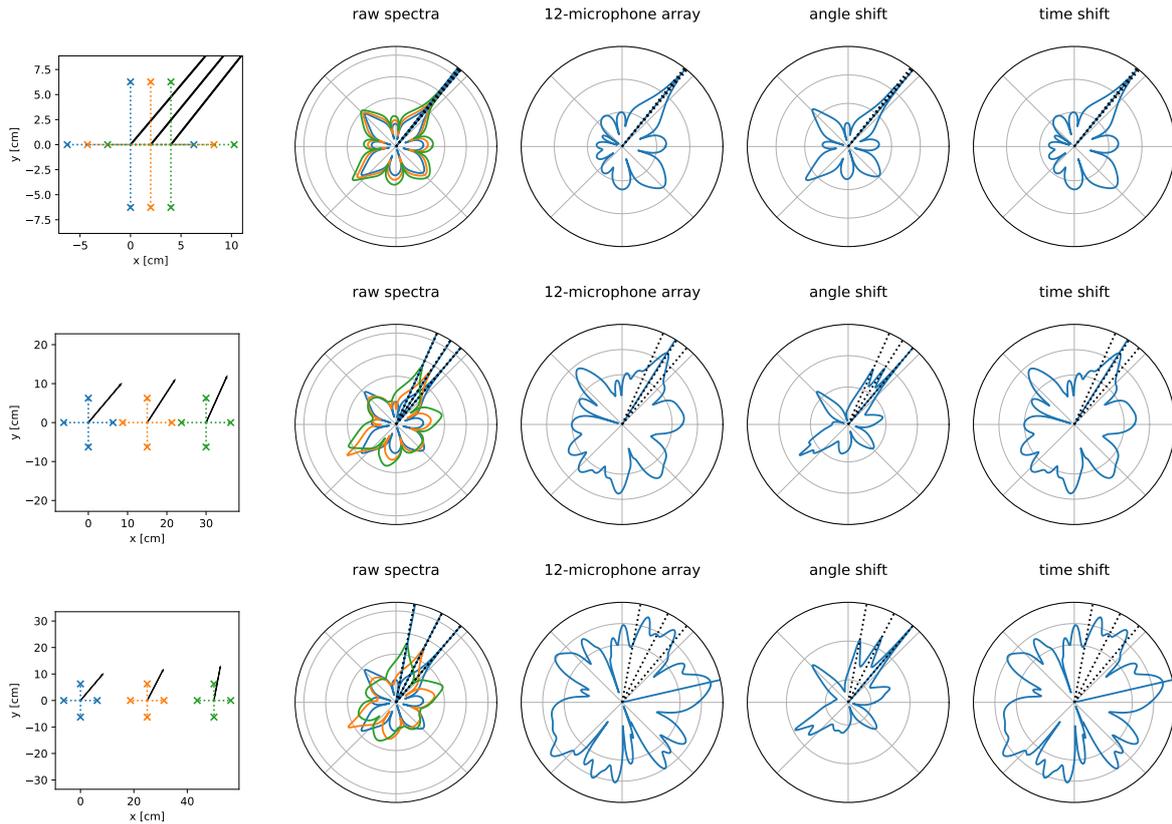


Figure 6.9.: Performance of different algorithms in the presence of lateral movement. We study spacings of 2 cm, 15 cm, and 25 cm, and the source plays at 4000 Hz. We compare the same methods as in Figure 6.8.

Resolving higher frequencies through movement The first case we study is a purely rotational movement and a source with a frequency above the admitted frequencies, given the array layout. In that case, after compensating for the measurement times, the angular spectrum resulting from our algorithm is identical to the angular spectrum from a static microphone array with as many microphones as visited positions. Figure 6.8 shows one such scenario. The minimum spacing between microphones for each individual microphone array is 8.8 cm and thus above the spatial aliasing limit: $b_f(4000) = 4.3$ cm. Thanks to the movement, we add baselines of down to 1.6 cm and remove the aliases.

We could achieve the same increased spatial sampling by using lateral movement. Lateral movement is even better than rotational movement as it increases the overall size of our virtual microphone array, leading to a better angular resolution. However, a significant lateral movement compared to the source distance changes to the perceived direction of arrival, and can thus have a tendency to blur out the estimates. In Figure 6.9, we show that a small lateral movement helps in locating a high-frequency source, similarly to a small rotational movement. We then increase the movement to investigate the tolerance. A movement of 15 cm is still tolerable, but after that, the estimate becomes blurred out. The individual and angle-shifted methods cope better with lateral movement, yielding simply a few peaks for the relative source directions at each position. We study the effect of lateral movement in

6. Dynamical sound source localization

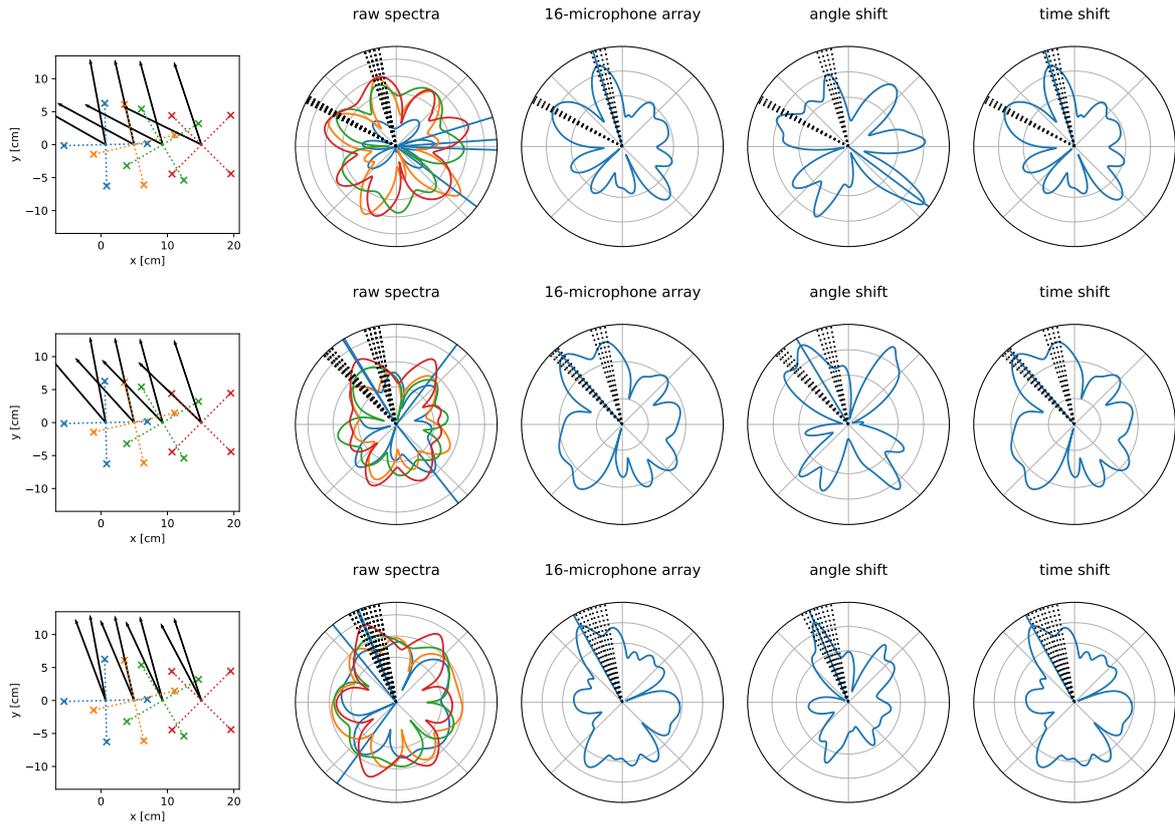


Figure 6.10.: Performance of different beamforming algorithms as we move two sound sources at 4000 Hz closer together, with respectively 50° , 30° and 10° spacing. We compare the same methods as in Figure 6.8.

more detail in the next section.

In Figure 6.10, we decrease the spacing of two sound sources until the methods break down. We see that the individual beamformers already struggle at a spacing of 50° , leading to a poor estimate of the angle alignment method. Our time alignment method gives a spectrum equivalent to the 12-microphone array, where the two sources are clearly resolved. As the spacing approaches 10° , the sources collapse to one estimate only.

Resolving multiple sources through movement Another advantage of the virtual microphone array comes to play when localizing many sound sources compared to the number of microphones. In Figure 6.11, we increase the number of sources from 3 to 5, spaced uniformly around the microphone array. Up to 4 sources, the time alignment method successfully identifies all sources, just like the equivalent static microphone array. The angle alignment method, on the other hand, fails to separate the sources at each position individually, leading to a blurred-out overall angular spectrum.

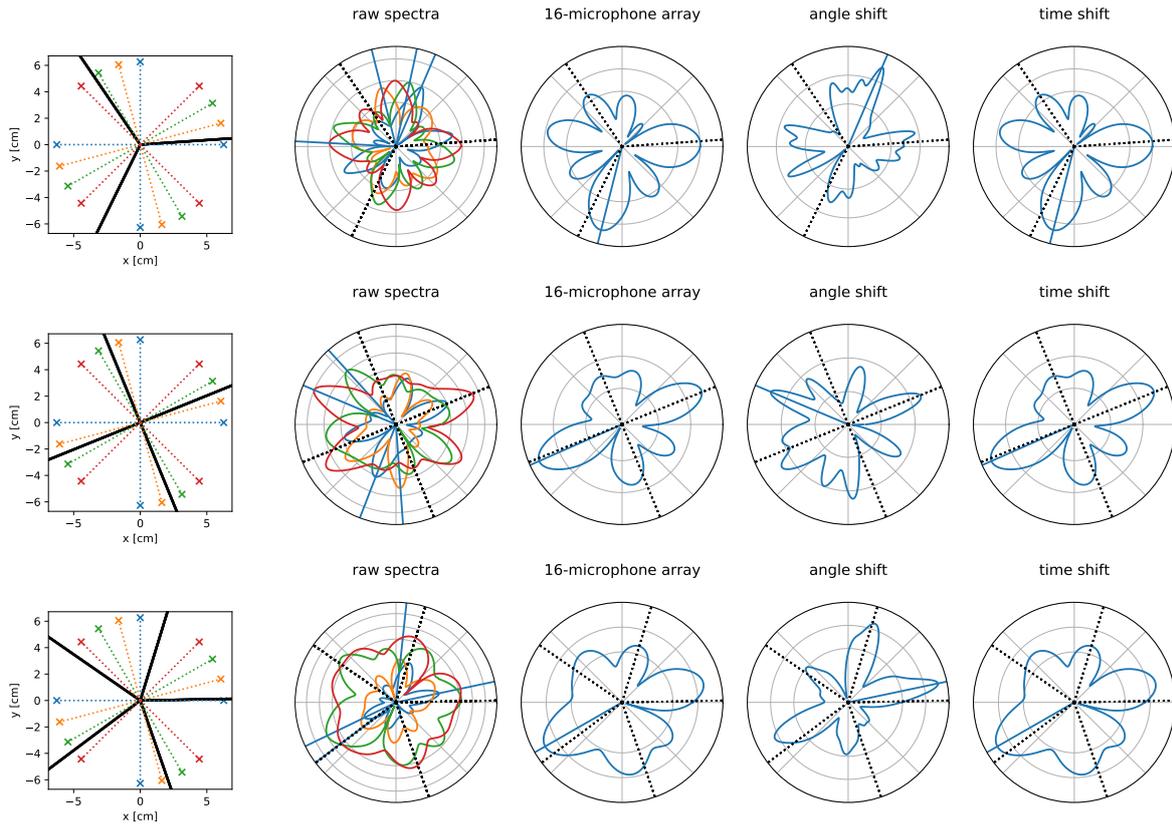


Figure 6.11.: Performance of different beamforming algorithms in the presence of multiple sound sources at 4000 Hz. We compare the same methods as in Figure 6.8.

6.6. Simulation results

We have identified two core problems in which the movement of the array can help. In this section, we investigate the tolerable range of movement and timestamp noise, respectively, for which combining different estimates is still beneficial.

Resolving higher frequencies through movement First, we estimate how accurate our position estimate needs to be for our method to work. We use the same setup as in Figure 6.8, and we add different levels of Gaussian angle noise σ_α to each position. We use source frequencies from 1000 Hz to 9000 Hz.

For the same experiment, we also study the robustness against time errors. We assume a time quantization of $1e - 6$, which corresponds to a microseconds-timer, and increasing levels of noise, added before quantization. Figure 6.12 (left plot) shows that there is a relatively sharp transition where the time-alignment method breaks down. As expected, the higher the frequency, the smaller is the tolerance for timing errors.

Finally, we study the effect of lateral movement on the performance. Figure 6.12 (right plot) shows the performance of the experiment described in Figure 6.9, for different magnitudes

6. Dynamical sound source localization

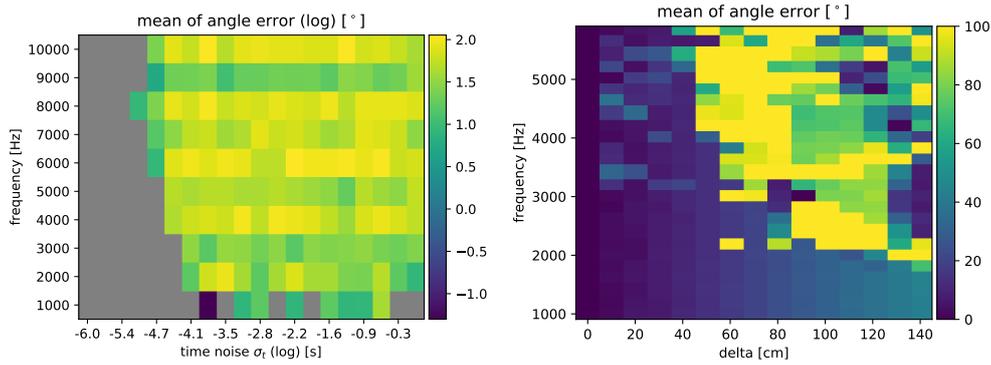


Figure 6.12.: Study of two detrimental effects for the time-alignment method. The left plot shows the effect of timing noise, using the setup in 6.8. The right shows the effect of lateral movements on the performance, using the same kinds of movement as in Figure 6.9.

of movement and different source frequencies. Again, tolerance to lateral movement gets smaller as the frequency increases. For the lowest frequencies, even movements up to 1 m are tolerable, for a source at the same distance.

Resolving multiple sources through movement Finally, we study the resolving power of multiple sources under more realistic conditions. Using the canonical example from Figure 6.10 with 40° source spacing, we study the success rate of resolving both sound sources with an accuracy of 10° . Figure 6.13 shows the performance for different source frequencies and angle noise levels, respectively.

From the conducted simulations, we observe that a major potential of the proposed time-alignment method lies in resolving multiple and closely-spaced sources. However, the method simulations are by no means exhaustive, and depending on the use case, other factors such signal-to-noise ratio, interfering sources or multi-frequency sources should be considered.

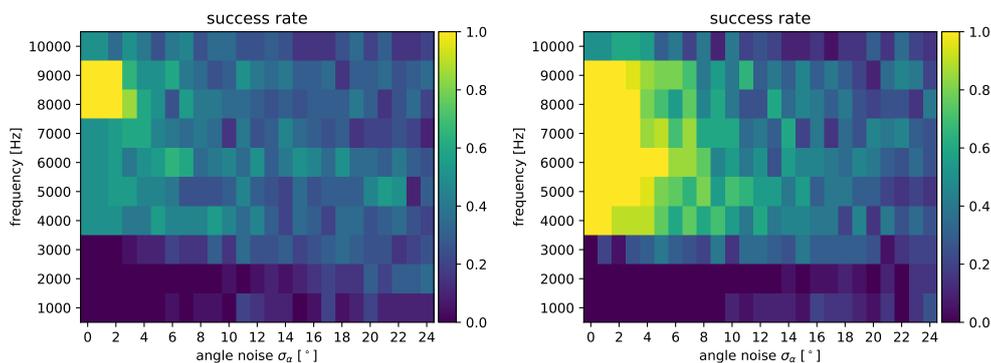


Figure 6.13.: Study of the success rate of resolving two sources at 10° and 50° , respectively, using the angle-alignment method (left) and time-alignment method (right). We define the success rate as the percentage of correctly recovered sources (0%, 50% or 100%), and average over all 10 trials.

6.7. Conclusion and ongoing work

In this chapter, we have explored the potential of using movement for increased DOA performance. Our method consists of time-aligning the signals based on the measurement times and estimated relative movement, thus creating a microphone array with as many microphones as individual positions visited. We show that, under some assumptions on the source signal, the same DOA resolution can be obtained as with the equivalent static multi-microphone array. We can thus successfully multiplex the number of microphone in software rather than hardware, which is particularly useful when limitations on the size, weight and computational resources of the microphone array are imposed. This is the case in many real-world applications, in particular when using mobile robotic platforms like drones.

We have shown in extensive simulations that by using the proposed method, we can localize sound sources more reliably than when using only static positions, in particular when the sources are closely spaced, their frequency is above the spatial aliasing frequency, or when there are more sources than microphones in the array.

Note that, although we have tested the explored methods on frequencies and wavelengths corresponding to sound signals, the arguments remain valid for any signals, as long as the assumptions on self-correlation is met, and as long as a precise enough timestamp can be obtained along with the data. Applying the methods with a variety of signal types is thus an interesting direction for future research, promising to bring performance improvements to a variety of applications where the movement of a sensor array has so far blurred rather than enhanced results.

Secondly, we emphasize that we have used two of the most simple beamforming algorithms, MVDR and DAS, because they are computationally cheap while offering good intuition on the achievable performance. However, for increased performance, other methods can be applied to the time-aligned signals, for instance parametric DOA algorithms such as MUSIC or ESPRIT [197]. These methods represent the state of the art in many DOA scenarios, and are thus expected to further improve the results reported in this chapter.

Thirdly, we recall that the emphasis of this chapter is on exploiting movement of a microphone array, where we have no control over the movement. Follow-up questions include for example how to move the array to improve accuracy of the source localization. In particular, we can move in a way that creates a uniform beamshape (see Section 6.3.4), or move perpendicularly to the current direction estimate, which increases the effective baseline, narrowing the beamshape and thus increasing resolution.

Finally, just as in the previous chapter, the solution presented here can become a part of a full navigation algorithm, by integrating the angle distributions in standard SLAM algorithms, which is being done in ongoing research.

Part II.

Embedding based on angle and distance measurements

7 | Background of the second part

In this introductory chapter, we lay out the foundations for distance and angle-based embedding, treated in this second part of the thesis.

7.1. Distance-based embedding

When measuring dissimilarities of entities in terms of Euclidean distances, a rich literature on distance geometry can be exploited for purposes such as denoising the given distances, finding unknown distances, and recovering a low-dimensional point embedding that best satisfies these distances [120]. In the following, we review the results most relevant for this thesis, in particular the concept of Euclidean distance matrices (EDMs). We follow the comprehensive tutorial by Dokmanić *et al.* [53].

7.1.1. Problem statement

For N entities that we wish to localize, we are given a set of pairwise distance measurements $d_{ij} \in \mathbb{R}^+$. We seek to embed the input data in a low-dimensional Euclidean space, and denote the resulting coordinates by $x \in \mathbb{R}^D$. An example point set and some corresponding distance measurements are shown in the top-left of Figure 7.1. Stacking all coordinates in the columns of a matrix, we obtain $X \in \mathbb{R}^{D \times N}$. The Euclidean distance matrix of these N entities, denoted by $D \in \mathbb{R}^{N \times N}$, is simply the matrix composed of the square distances. It

Section 7.3 of this chapter is based on the publication:

G. Baechler, F. Dümbgen, G. Elhami, M. Krekovic, R. Scheibler, A. Scholefield, and M. Vetterli. “Combining range and direction for improved localization”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2018, pp. 3484–3488

Detailed contributions: FD designed the presented algorithm, performed simulations and created first versions of the plots, which were improved by MK. The paper was written by GB, GE, MK, AS and FD, and the research was designed by AS, RS and MV.

Implementations of methods discussed in this chapter are available at github.com/lcav/localization-icassp2018.

7. Background of the second part

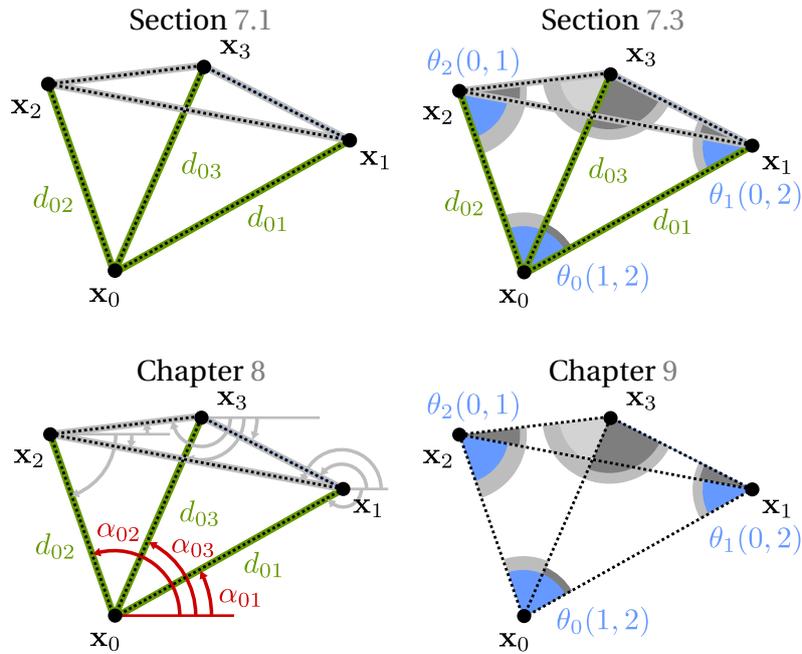


Figure 7.1.: Overview of four different problem setups studied in the second part of the thesis. We draw in gray the measured quantities from which we infer the locations of the points x_n , and provide the labels for a few instances in color. Distances, inner angles and absolute angles are marked in green, blue and red, respectively.

can be written as [53]

$$D = [d_{ij}^2]_{i,j=0}^N = \mathbf{1} \operatorname{diag}(\mathbf{X}^\top \mathbf{X}) - 2\mathbf{X}^\top \mathbf{X} + \operatorname{diag}(\mathbf{X}^\top \mathbf{X}) \mathbf{1}^\top, \quad (7.1)$$

with $\mathbf{1}$ the column vector of all ones and $\operatorname{diag}(\mathbf{A})$ the column vector composed of the diagonal entries of \mathbf{A} .

As we will see below, when an embedding in D exists and all distances between points are known exactly, then forming the EDM and factorizing it via an eigenvalue decomposition EVD leads to an exact point estimate (up to rigid motions). However, in real-world applications, the set of distances can be 1) noisy, 2) incomplete and 3) unlabeled, which is summarized in the following problem statement:

Problem 5. Given a set of (noisy) pairwise Euclidean distances d_k , (and the mapping from k to index pairs i, j), recover the point embedding \mathbf{X} that (best) matches these distances.

Note that in this and the following chapters, we use the vague formulation “best match”. For this to be well-defined, we need to introduce the objective. In order to keep the problem statement clear and general, we allow for vagueness at this level, and explain within each section what we mean by a “best match”.

Notation for this part	
N	number of entities
D	embedding point dimension
$\mathbf{x} \in \mathbb{R}^D$	embedding point coordinates
$\mathbf{X} \in \mathbb{R}^{D \times N}$	matrix of point coordinates
$d_{ij} \in \mathbb{R}^+$	Euclidean distance between two points
$\mathbf{D} \in \mathbb{R}^{N \times N}$	Euclidean distance matrix
$\mathbf{J} \in \mathbb{R}^{N \times N}$	geometric centering matrix
$\alpha_{ij} \in [0, 2\pi]$	absolute azimuth angle of vector $\mathbf{x}_j - \mathbf{x}_i$
$\theta_k(i, j) \in [0, \pi]$	inner angle between points i, k and j
\mathcal{M}	index set of all inner or absolute angles
$M = \mathcal{M} $	number of all inner or absolute angles

7.1.2. Lost information

We have already hinted a few times at the fact that the absolute orientation and translation of points are lost in distance-based embedding. Intuitively, this makes sense: consider we have a few points connected only between them with rods of fixed length. As we move the whole rigid assembly around, we change its global orientation and translation, but not its distances. Using equation (7.1), this observation can be more formally derived. Indeed, we can replace \mathbf{X} by $(\mathbf{Q}\mathbf{X} + \mathbf{b}\mathbf{1}^\top)$, where $\mathbf{Q} \in \mathbb{R}^{D \times D}$ is an orthogonal matrix representing rotation/reflection and $\mathbf{b} \in \mathbb{R}^D$ is an arbitrary translation vector. Plugging the transformed points into (7.1) and simplifying, we observe that the EDM remains unchanged.

7.1.3. Realizability

A set of distances is D -realizable if there exists a point embedding in dimension D which gives rise to exactly these distances. Realizability is a theoretical concept: as soon as distances are noisy, no such embedding is guaranteed to exist. However, realizability conditions can be exploited for denoising, point recovery and labeling algorithms, as shown in the following sections.

We can, for instance, characterize when a matrix is an EDM for points in a given dimension. If a matrix is an EDM, then by definition, we can find a point embedding that generates exactly the squared distances of the matrix (and therefore, also the distances). Thus, if the distances squared lead to an EDM with rank $D + 2$, the distances are realizable in some dimension D . It was shown that a matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is an EDM if and only if [77]

$$-\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J} \succcurlyeq 0, \quad (7.2)$$

where $\succcurlyeq 0$ means the matrix is positive semi-definite, and $\mathbf{J} \in \mathbb{R}^{N \times N} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is the geometric centering matrix.¹ In fact, if \mathbf{D} is an EDM, then (7.2) describes the Gram matrix

¹The original results uses $\mathbf{I} - \mathbf{s}\mathbf{s}^\top$ for any \mathbf{s} such that $\mathbf{s}^\top\mathbf{1} = 1$ and $\mathbf{s}\mathbf{D} \neq \mathbf{0}$, of which \mathbf{J} is a special case.

7. Background of the second part

$$\mathbf{G} := \mathbf{X}^\top \mathbf{X}.$$

Since in practice we are interested in low-dimensional embeddings ($D < N$), this condition can be extended by a rank constraint $\text{rank}(\mathbf{G}) \leq D$.

A different necessary and sufficient realizability condition which directly incorporates d , based on the Cayley-Menger determinants, is given in [120]. This realizability condition is however more technical and to the best of our knowledge, has not been employed for efficient embedding algorithms.

7.1.4. Recovery algorithms

EDM-based algorithms When all distances are given, point coordinates can be recovered through a simple EVD. To derive this, we note that the Gram matrix, being positive-semidefinite (PSD), can be factorized as

$$\mathbf{G} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top, \quad (7.3)$$

where $\mathbf{\Lambda}$ is the diagonal matrix of decreasing eigenvalues and \mathbf{U} is orthonormal. Then, an estimate of \mathbf{X} is given by:

$$\hat{\mathbf{X}} = \left[\text{diag} \left(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_d} \right), \mathbf{0}_{d \times (N-d)} \right] \mathbf{U}^\top. \quad (7.4)$$

Note that when the distances are noiseless, only the first D eigenvalues are non-zero, so (7.4) could have been written as $\hat{\mathbf{X}} = \mathbf{\Lambda} \mathbf{U}^\top$ and we know that $\hat{\mathbf{X}} = \mathbf{Q} \mathbf{X} + \mathbf{b} \mathbf{1}^\top$ for some rigid transformation given by \mathbf{Q} and \mathbf{b} . When the distances are noisy, the last $N - D$ eigenvalues might not be zero. In this case, (7.4) actually gives the orthogonal projection to rank- D matrices (thus the minimum error in Frobenius norm), by the Eckart-Young-Mirsky theorem [60]. The above algorithm has been proposed for data analysis in psychology, under the name of multidimensional scaling (MDS).

In most applications, we are not given all pairwise distance measurements. In these cases, properties of the EDM can be used to denoise and complete the distance measurements. This yields a complete and realizable EDM, which can be fed into the *classical MDS* algorithm. As an example, the property described in (7.2) in combination with the rank condition naturally leads to rank-constrained semidefinite programs. Using a standard rank relaxation, such as the trace or nuclear norm, the problem can be solved efficiently.

Classical optimization One might be tempted to compare the EDM-based algorithms with more classical optimization algorithms, such as the RLS or SMLS estimates that we encountered in Chapter 4. Using the notation of this chapter and translating these concepts to joint

rather than anchor-based localization, we get

$$\mathbf{X}^{RLS} = \arg \min_{\mathbf{X}} \sum_{i,j} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})^2, \text{ or} \quad (7.5)$$

$$\mathbf{X}^{SRLS} = \arg \min_{\mathbf{X}} \sum_{i,j} (\|\mathbf{x}_i - \mathbf{x}_j\|^2 - d_{ij}^2)^2. \quad (7.6)$$

Note that the cost functions in (7.5) and (7.6) also go under the names raw and s-stress, respectively. The advantage of solving these optimization problems rather than using the EVD-based solution is that they use measurements more transparently, and in fact (7.5) corresponds to the maximum likelihood estimate for additive Gaussian noise on distances. In contrast, the *classical MDS* technique actually finds the best approximation, in Frobenius norm, to the Gram matrix, so rather than optimizing over distances, it optimizes over inner products.

Although they are in that sense more “optimal”, problems (7.5) and (7.6) are difficult to solve, and to this date there exists no algorithm that guarantees to find the global minimum. However, efficient methods based on alternating descent have been proposed and shown to perform well on simulated data. [46, 159]

Build-up algorithms Finally, in certain applications, distance measurements are very sparse, in which case the matrix denoising and completion algorithms tend to break down. One well-known application of this kind is the experimental reconstruction of molecular conformation. In a long string of works tackling these problems, the distance-based embedding problem was solved using build-up algorithms [120]: arbitrarily fixing the first point, subsequent points are added by (approximate) sphere intersection from previously recovered points. When multiple solutions exist, branches are multiplied, and when a branch does not yield meaningful sphere intersections, it is dropped.

7.1.5. Unlabeled sensing

Although EDMs are designed to solve assigned distance problems, they have proved to be useful as labeling tools in various applications. As an example, the rank property of EDMs has been used to recover distance labels in room geometry reconstruction from echoes [54]. In this problem, discussed in Chapter 5, we need to assign different peaks in the room impulse response to the different walls. Looping over different label assignments, the best assignments can be identified by comparing rank approximation errors. A similar algorithm can be used for sparse phase retrieval, briefly outlined in [53] and in more detail in [9].

7.1.6. Number of measurements

For noiseless EDMs, there is no exact formula that describes the minimum number of measurements, but we can lower-bound it by a counting argument [53]. We have ND unknowns (N D -dimensional points) to recover, but these points are only recovered up

7. Background of the second part

to rigid motions, which encompass D degrees of freedom for translations, and $D(D - 1)/2$ degrees of freedom for orthogonal transformations. We conclude that the number of measurements is lower bounded by $ND - (D + 1)D/2$. Moreover, to ensure that every point is rigidly connected to the others, at least $D + 1$ measurements are needed for every point.

This concludes our brief overview of distance geometry and corresponding tools. In the following sections, we introduce the types of angle measurements studied in this thesis, for which we seek to derive similar characterizations as for the distance measurements above.

7.2. Introducing angles: notation and definitions

We distinguish between inner and absolute angles, which we define as follows. The absolute angle of two points x_k and x_l in $D = 2$ is given by

$$\alpha_{kl} = \angle(x_l - x_k + j(y_l - y_k)) \in [0, 2\pi], \quad (7.7)$$

where j is the imaginary number and \angle returns the phase of a complex number. Examples of absolute angles are given in the bottom left of Figure 7.1. The index set \mathcal{M} of absolute angle measurements of N points contains all pairs of points with $i \neq j$. Note that we do not assume antisymmetry: in general, $\alpha_{ij} \neq -\alpha_{ji}$. Therefore, we have $M = |\mathcal{M}| = N(N - 1)$.

There are several scenarios of practical relevance where we measure absolute angles. For instance, when using sensors such as turning lidar or radar range finders, we localize landmarks by their distance and absolute angle in the current reference frame. A microphone array can measure the absolute angle of a sound source, as described in Chapter 6. Finally, a directional proximity detector triggers whenever there is an object in its (narrow) field of view, in which case the location of the object can be approximated by the view direction.

Moving over to inner angles, these measurements are defined between three points of any dimension, as

$$\theta_i(j, k) = \cos^{-1} \frac{(\mathbf{x}_j - \mathbf{x}_i)^\top (\mathbf{x}_k - \mathbf{x}_i)}{\|\mathbf{x}_j - \mathbf{x}_i\| \|\mathbf{x}_k - \mathbf{x}_i\|}, \quad (7.8)$$

where we use the notation $\theta_i(j, k)$ to describe the angle between points j and k , as seen from point i . Examples of inner angles can be found in the right column of Figure 7.1. The inner angle always measures the smallest angle between three points. In three dimensions, it corresponds to the angle in the plane formed by the two vectors. We assume the following symmetry: $\theta_i(j, k) = \theta_i(k, j)$, and conclude that the total number of angles of a given point set equals $M = N(N - 1)(N - 2)/2$.

Inner angles are also relevant in many practical applications. For instance, when a collection of robots each measure absolute angles to other robots, but each robot does not have an orientation estimate of itself, then we can translate these angles to a collection of inner angles instead. Inner angles are also closer related to similarity measures than absolute angles. In fact, the cosine similarity (*i.e.* the argument in (7.8)) is a commonly used similarity

measure in information retrieval from documents [140, 141]. When using the cosine, the measure is not a proper metric anymore (the triangular inequality is not satisfied), but it still performs well for point embedding. It is also worth noting that when the vector differences in (7.8) are normalized, then the cosine of $\theta_i(j, k)$ is equal to

$$\cos(\theta_i(j, k)) = 1 - \frac{1}{2} \|\mathbf{x}_j - \mathbf{x}_k\|. \quad (7.9)$$

The introduced notation is summarized in the notation box of this chapter, and visualized in Figure 7.1.

The overview of distance geometry has hopefully convinced the reader that understanding fundamental properties of distance measurements allows the design of efficient recovery algorithms. Compared to distance geometry, only few results exist regarding angle measurements. For instance, there is no equivalent to the EDM for inner angles. Since angles involve three points, such an object would be a three-dimensional tensor and thus probably more complex to handle. However, we argue that similar to distances, understanding the properties of such a sensor would lead to many interesting algorithms.

We do not provide a full characterization of such a tensor. However, focusing on certain aspects of angle-based measurements, we provide results that we hope to be useful in approaching such a theory:

- In the next section, we present a method for integrating angle measurements in the *classical MDS* algorithm described above. Building on work presented in [131], we show that adding a few simple constraints to improve the realizability of the recovered angles also improves the localization result.
- In Chapter 8, we investigate how to combine absolute angle measurements of type (7.7) with distance measurements. This study leads us to introduce coordinate difference matrices, conceptually similar to Euclidean distance matrices, which we study thoroughly.
- In Chapter 9, we study the localization from inner angle measurements of the form (7.8). We provide realizability conditions, which we exploit for angle denoising. This in turn allows to use a simple reconstruction algorithm for point embedding.

7.3. Distance and angle-based embedding: preliminary results

Already in 2007, De Abreu [48] proposed “super MDS” or “Edge MDS” (E-MDS), an extension of MDS to inner angle measurements. In this and follow-up works [74, 131], the authors exploit what they call the “Edge Gram Kernel”. It is constructed by forming the inner products between edges, defined by $\mathbf{v}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. Stacking all $M = N(N - 1)/2$ such edges into a

7. Background of the second part

matrix $\mathbf{V} \in \mathbb{R}^{M \times D}$, the Edge Gram Kernel $\mathbf{K}_E \in \mathbb{R}^{M \times M}$ can be written as:

$$\mathbf{K}_E = \mathbf{V}\mathbf{V}^\top = \mathbf{d}_E \mathbf{d}_E^\top \circ \mathbf{\Omega}, \quad (7.10)$$

where we have introduced the vector of edge lengths $\mathbf{d}_E \in \mathbb{R}^M$, and the matrix $\mathbf{\Omega} \in \mathbb{R}^{M \times M}$ contains the cosines of inner angles for each edge pair. It is straight-forward to see why decomposition in (7.10) holds, by comparing it with the definition of the inner angle in (7.8). Similar to *classical MDS*, the edge vectors \mathbf{V} are recovered as follows:

1. Given noisy distance and angle measurements, construct the noisy edge Kernel $\tilde{\mathbf{K}}_E$
2. Use an EVD to recover the denoised edge matrix $\hat{\mathbf{V}}$, using the D biggest eigenvalues.
3. Using $\hat{\mathbf{V}}$ and anchoring one point, recover the coordinates by solving a sparse linear system (the algorithm used in fact resembles our algorithms in Chapter 8).

The above solution is termed “super MDS” or “E-MDS” and outperforms previous approaches, which include *classical MDS* and SDP-based methods using either distances only or hybrid measurements [21]. However, it neglects certain geometric constraints, which means the recovered vectors are not guaranteed to be geometrically consistent. Using the same terms as in distance geometry, the resulting edge vectors are not necessarily *realizable* by a point set in D dimensions. For instance, nothing ensures that the obtained edges satisfy geometric constraints such as the triangle equality:

$$\mathbf{v}_{ik} = \mathbf{v}_{ij} + \mathbf{v}_{jk} \quad \forall (i, j, k) \subseteq \mathcal{M}. \quad (7.11)$$

We can illustrate the non-realizability of the recovered vectors by the following observation. In step 3 described above, the authors simply use an arbitrary anchor point and the first $N - 1$ rows of $\hat{\mathbf{V}}$ to reconstruct the point set. Since their $\hat{\mathbf{V}}$ is not geometrically consistent, and any other choice of rows could lead to a different reconstruction. As we will show, after rendering the estimated edge kernel consistent, any choice of rows leads to the same reconstruction.

Based on this insight, we propose a new method, which we call the *constrained edge-kernel* method. We introduce the notion of a *triangle constraints matrix* $\mathbf{M} \in \mathbb{R}^{C \times M}$. This matrix incorporates the mutual dependencies of edge vectors forming a triangle, with $M_{li} = 1$, $M_{lk} = M_{lj} = -1$, assuming that the vectors \mathbf{v}_i , \mathbf{v}_j and \mathbf{v}_k form the l -th triangle. The number of constraints C equals to the number of triangles in the point set, $C = \binom{N}{3}$. The Edge Gram kernel recovery problem can then be written as the rank-constrained optimization,

$$\begin{aligned} \hat{\mathbf{K}}_E = \arg \min_{\mathbf{K}} \left\| \tilde{\mathbf{K}}_E - \mathbf{K} \right\|_F \\ \text{subject to } \mathbf{K} \in \mathcal{C} \text{ and } \mathbf{K} \in \mathcal{R}, \end{aligned} \quad (7.12)$$

where we introduce the feasibility sets

$$\mathcal{R} = \{ \mathbf{X} \in \mathbb{R}^{E \times E} : \mathbf{X} \succeq 0, \text{rank}(\mathbf{X}) = D \}, \quad (7.13)$$

$$\mathcal{C} = \{ \mathbf{X} \in \mathbb{R}^{E \times E} : \mathbf{M}\mathbf{X} = \mathbf{0} \}. \quad (7.14)$$

Because of the rank constraint, Problem (7.12) is non-convex and, furthermore, it has no analytic solution. In such situations, it is typical to relax the non-convex constraint. To do this, we implement the widely used nuclear norm relaxation from [47]. Since the solution is constrained to be positive-semi-definite, its nuclear norm is equal to its trace, resulting in the following relaxation:

$$\begin{aligned} \hat{\mathbf{K}}_E = \arg \min_{\mathbf{K}} \left\| \tilde{\mathbf{K}}_E - \mathbf{K} \right\|_F + \lambda \text{trace}(\mathbf{K}) \\ \text{subject to } \mathbf{K} \in \mathcal{C}, \mathbf{K} \succeq 0, \end{aligned} \quad (7.15)$$

where λ is a regularization parameter, which controls to what extent the solution satisfies the rank property and the linear constraints, respectively.

As an alternative to relaxation, we can apply the lift-and-project method from [82]. It is an iterative method which consists of projecting the estimate at iteration k ($\hat{\mathbf{K}}_{E,k}$) onto the feasible sets \mathcal{C} and \mathcal{R} in an alternating fashion. The optimal projection onto the non-convex set \mathcal{R} is equivalent to the super MDS solution. The orthogonal projection of $\tilde{\mathbf{K}}_E$ onto the convex set \mathcal{C} is given by

$$\hat{\mathbf{K}}_{E,k+1} = \left(\mathbf{I} - \mathbf{M}^\top (\mathbf{M}\mathbf{M}^\top)^{-1} \mathbf{M} \right) \hat{\mathbf{K}}_{E,k}. \quad (7.16)$$

Experimentally, we found that this second approach shows excellent convergence rate in simulations (convergence is always achieved in at most two iterations), and no parameter tuning is required, so we rely on this approach for the presented results. We call the resulting algorithm “super-duper” MDS. Indeed, extensive simulation results, reported in the paper [7], show that this approach outperforms the unconstrained approach.

7.4. Alternative embeddings and metrics

Since in this thesis, our main interest is localization, we choose the Euclidean space of two or three dimensions for embedding. Depending on the application, other embedding spaces can be more appropriate. For tree-like data, for instance, hyperbolic spaces convey the structure more faithfully in a given dimension than a Euclidean embedding [115].

Tree-like data is common in applications with clear hierarchies, for instance in the study of the evolution of biological cells, or in text semantics embedding [199]. More generally, we may seek a tree-like embedding when, rather than having similarity estimates, we have similarity comparisons. Such measurements are the core component of so-called non-metric dimensional scaling [113], which was introduced roughly at the same time as MDS. Although not treated in this thesis, we want to point out that similarity comparisons can be an interesting alternative in practical systems. For instance, RSSI measurements of a non-calibrated anchor can be treated as non-metric: the position with higher signal strength is likely to be closer to the anchor than the one with lower signal strength. Similarly, rather than converting step counts to exact distances as done in Chapter 3, we could use step counts to

7. Background of the second part

compare distances over short time windows. In these cases, non-metric dimensional scaling can avoid the necessity for calibrating anchors or step lengths.

7.5. Outlook

The algorithm in Section 7.3 gives a first taste of the potential of exploiting angle information, and in particular, of using geometric constraints to impose realizability.

In Chapter 8, we propose a novel framework for point recovery from distance and absolute angle measurements. Since such measurements can be combined to vectors, the setup is similar to point 3 in the “super MDS” algorithm above. However, we propose an algorithm that uses all available information, and we extend the framework to non-Cartesian projections, relevant for applications such as structure from motion or geometric microphone array calibration.

In Chapter 9, we explore geometric angle constraints without using any distance information. In our aim to fully characterize realizable angles, we find necessary conditions, which, backed by simulation, we conjecture to be sufficient too. Similar to the method described above, these conditions can be used to design a constrained optimization problem for denoising angle measurements, yielding realizable angles. Once the angles are realizable, a simple build-up algorithm can be used to reconstruct the point set.

8 | Embedding based on coordinate differences

بيت الضيق ببساع الف صديق

A tiny house fits a thousand friends.

— Lebanese saying

Out of many problems that can be phrased as a general embedding problem, one of great scientific value is the molecular conformation problem. Both distances and angles between molecules are quantities that can be extracted from nowadays widely used nuclear magnetic resonance (NMR) spectroscopy [64, 90]. Even in earlier days, Patterson [162] realized that x-ray data can be exploited to extract interatomic vector measurements. Given these similarity measurements, we can try to find the three-dimensional embedding — the molecule geometry. Building on the mature field of distance geometry, many researchers have investigated build-up algorithms for reconstructing the molecule geometry from distance measurements only [121]. The available information on angles, on the other hand, remain mostly unused [20].

In this chapter, we study a method that can exploit this additional angular information. We illustrate the basic idea of the method in Figure 8.1: we build the vectors from distance and angle measurements, and project these vectors to the D directions of the Cartesian reference

This chapter is based on the publication:

G. Baechler, F. Dümbgen, G. Elhami, M. Krekovic, and M. Vetterli. “Coordinate Difference Matrices”. In: SIAM Journal on Matrix Analysis and Applications 41.1 (2020), pp. 332–363

Detailed contributions: MK, GB, GE, MMC and MV designed the project based on sports ranking, FD joined the project bringing in the localization perspective. FD, MK, GB, and GE proposed CDMs and derived their properties as well as their connection with EDMs (Section 8.2.3). MK and GB derived and implemented the point recovery in 1D (Section 8.2). MK derived the generalization to higher dimensions and performed the overall statistical analysis (Section 8.3). FD performed the experiments in Section 8.4.

8. Embedding based on coordinate differences

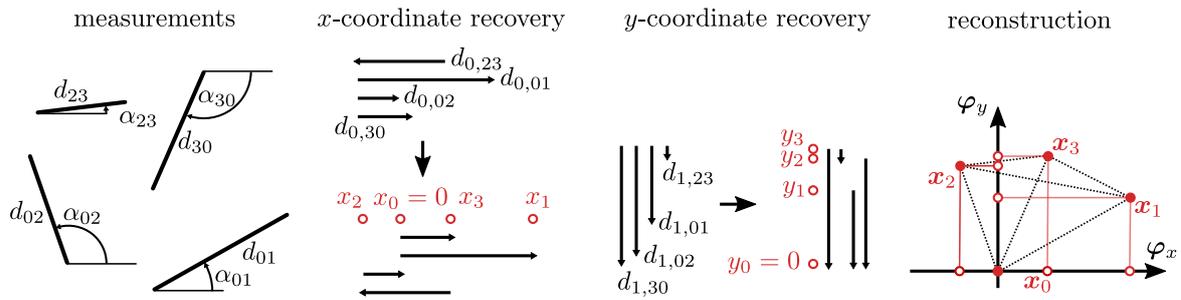


Figure 8.1.: Sketch of example of coordinate-difference-based reconstruction in two dimensions. Given distances and absolute angles, we create the projections to x and y axis. We recover the best estimate of coordinates in each direction independently, fixing the first point to the origin. Then, we recover the point set by simple concatenation of coordinates.

frame. This yields coordinate differences in each frame, which we combine to estimate the coordinates. Putting together the estimates from each frame, we recover the point locations. Because of the reduction of the problem to difference measurements, we call this chapter “Difference-based embedding”. In analogy with distance geometry, we introduce new objects that we call coordinate difference matrices (CDMs). Rather than containing Euclidean distances, as for EDMs, CDMs contain differences between the coordinates of pairs of points. In our molecular conformation example, for instance, we obtain three CDMs containing, respectively, the x , y and z coordinates of the edges between points. In Section 8, we characterize the CDMs and describe recovery algorithms and uniqueness conditions.

Rather than measuring projections to D Cartesian vectors as above, in some applications, we can measure projections to $K \geq D$ vectors in non-orthogonal directions. In this case, we could still use the same simple recovery algorithm per direction, and then obtain a best estimate of the point coordinates by solving a least-squares problem. We call this the “splitting” approach. Although computationally cheap, this approach is not always guaranteed to yield the best estimates, and it cannot exploit measurements from directions where only few measurements are available. Instead, we show that the MSE estimate is obtained by combining all measurements in a global estimation problem, which we simply denote by the “optimal approach”. We study the individual performance of these two algorithms and the connection between them in Section 8.3. Finally, we showcase and solve a variety of applications which can be formulated as coordinate difference problems in Section 8.4, focusing on sensor array localization and calibration.

8.1. Problem setup

In this chapter, we are interested in finding point embedding from coordinate differences. The most general formulation of the problem is as follows. We consider $K \geq D$ frame vectors $\varphi_{1:K}$ [208], in which we measure coordinate differences. We define the coordinate difference along frame vector k by $d_{k,ij} = x_{k,i} - x_{k,j}$, where $x_{k,i}$ describes the coordinate of point x_i in

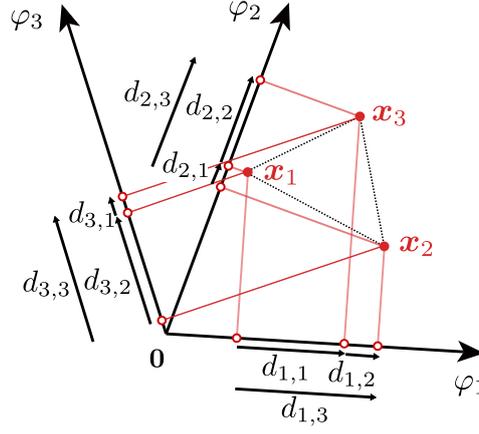


Figure 8.2.: An general example of point recovery in 2D from $K = 3$ projections. We use $d_{k,m}$ for $k = 1, 2, 3$ and $m = 1, 2, 3$, to indicate the measured pairwise differences between the coordinates of the points $\{x_{1:3}\}$.

frame vector k . To simplify notation, we introduce an arbitrary ordering on the measured differences, which we assume to form the set $\mathcal{M}_k \subseteq \{(0, 0), (0, 1), (1, 0), \dots, (N-1, N)\}$, and denote the m -th difference measurement by $d_{k,m}$, with $m = 1 \dots M_k$, with $M_k := |\mathcal{M}_k|$.

Finally, we introduce the noisy coordinate differences $\tilde{d}_{k,m} := d_{k,m} + z$, where z is zero-mean Gaussian noise, and the corresponding measurement vectors $\tilde{\mathbf{d}}_k \in \mathbb{R}^{M_k}$.

Before describing what kind of setups lead to such measurements, we formally state the problem addressed in this chapter:

Problem 6. Given the noisy coordinate-wise differences $\tilde{\mathbf{d}}_k$ observed in the known frames φ_k , with $k = 1 \dots K$, recover the points embedding \mathbf{X} that best matches them.

There is a strong connection between the proposed framework and node localization from relative measurements [16]. The novelty of our work lies in the formulation of the problem in arbitrary coordinate directions, which allows for an elegant formulation from which the “splitting” and “optimal” solutions can be readily derived. More importantly, this formulation finds an immediate application in problems such as sensory array calibration, where coordinate difference measurements are obtained directly in $K \geq D$ frames. Furthermore, splitting the problem in different directions allows us to introduce Coordinate Difference Matrices, which have interesting parallels with the Euclidean Distance Matrices we encountered earlier.

We derive how to solve Problem 6 in each frame direction independently in Section 8.2. Building on these results, we propose the “splitting approach” in Section 8.3, which consists of applying the procedure in each coordinate direction independently. We compare it with the “optimal approach” in the same section, and investigate when the two algorithms are equivalent.

8.2. Solution in one dimension: Coordinate Difference Matrices

In this Section, we take a closer look at Problem 6 in one dimension, which is equivalent to looking at each frame vector independently.

8.2.1. Related work

The work in this chapter started from the study of a problem in one dimension: ranking sports teams. In that context, the Massey method [137] is commonly used to rank a collection of sport teams based on their scores, and it is in fact equivalent to the formulation we present in Section 8.3.1. This formulation was also used to design tournaments that maximally improve the informativeness of a ranking for a given number of future comparisons [154]. The novelty of our work lies in the formulation of the 1D problem in a Coordinate Difference Matrix, which provides an interesting parallel to Euclidean Distance Matrices, and in the generalization to higher dimensions.

8.2.2. Coordinate Difference Matrices

Inspired by the powerful framework provided by EDMs, we define a *coordinate difference matrix* (CDM) in frame k by $C_k \in \mathbb{R}^{N \times N}$, with entries $C_{ij} = d_{k,ij}$. Since we are only interested in one direction, we drop the index k in what follows. Defining the absolute coordinate vector of the considered direction by $\mathbf{x} := [x_0, \dots, x_{N-1}]^\top \in \mathbb{R}^N$, the CDM can be written as

$$C = \mathbf{x}\mathbf{1}^\top - \mathbf{1}\mathbf{x}^\top. \quad (8.1)$$

A CDM can be viewed as an EDM in one dimension which preserves the sign of the distance. A more comprehensive overview of the connections between CDMs and EDMs is given in Section 8.2.5.

As in Problem 6, we allow partial and noisy coordinate differences. We introduce the weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ with non-negative entries, where $W_{ij} = 0$ indicates that the entry (i, j) is missing. Introducing the noise matrix \mathbf{Z} , whose entries are independent noise realizations, we can define an incomplete and noisy CDM as:

$$\tilde{C} = (C + \mathbf{Z}) \circ \mathbf{W}. \quad (8.2)$$

Then we can formulate the one-dimensional version of Problem 6 in terms of CDMs:

Problem 7. *Given the noisy and incomplete coordinate difference matrix \tilde{C} , recover the set of coordinates \mathbf{x} that best matches them.*

	Properties	Description
P.1	Rank-2	$\text{rank}(\mathbf{C}) = 2$ for $N > 1$
P.2	Triangular equality	$C_{ij} = C_{ik} + C_{kj}$
P.3	Skew-symmetry	$\mathbf{C} = -\mathbf{C}^\top$
P.4	Hollowness	$\text{diag}(\mathbf{C}) = \mathbf{0}$
P.5	Zero-sum	$\mathbf{1}^\top \mathbf{C} \mathbf{1} = \mathbf{0}$
P.6	Columns as a solution set	CDM of $\mathbf{x} = \{C_{ij} \mid \forall i \leq N\}$ is \mathbf{C}
P.7	Row averaging	$\mathbf{x} + c = \frac{1}{N} \mathbf{C} \mathbf{1}$, where c is a constant
P.8	Translation invariance	\mathbf{x} and $\mathbf{x} + c$ have the same CDM \mathbf{C}
P.9	Imaginary eigenvalues	$\text{Re}(\lambda_i) = 0$ for $i = \{1, 2\}$
P.10	Anti-symmetric eigenvalues	$\lambda_1 = -\lambda_2$

Table 8.1.: Properties of CDMs.

8.2.3. Properties of CDMs

We first study the properties of CDMs. In Table 8.1, we provide a non-exhaustive list of properties of CDMs (for derivations, see [8]). These properties are necessary conditions for a matrix to be a CDM, but not all sufficient. The following is a necessary and sufficient condition:

Proposition 1. A matrix \mathbf{C} is a CDM if and only if its elements satisfy the triangular equality $C_{ij} = C_{ik} + C_{kj}$ for all triples (i, j, k) .

Proof. A matrix \mathbf{C} whose entries satisfy the triangular equality implies both hollowness and skew-symmetry: $C_{ij} = C_{ii} + C_{ij}$ ensures that $C_{ii} = 0$, while $C_{ii} = C_{ij} + C_{ji} = 0$ implies $C_{ij} = -C_{ji}$. Combining the triangular equality with the skew-symmetry, we can express all elements as $C_{ij} = C_{ik} - C_{jk}$. Without loss of generality, we define $\mathbf{x} = [C_{1k} \ C_{2k} \ \dots \ C_{Nk}]^\top$. Using this definition, we can rewrite C_{ij} as $C_{ij} = x_i - x_j$ for every i and j ; therefore, \mathbf{C} is of the form (8.1). \square

An alternative way to check if the sufficient condition is satisfied is based on consistent positive reciprocal matrices [190], described below for completeness.

Definition 1. The matrix $\mathbf{R} \in \mathbb{R}^{N \times N}$ is positive reciprocal if $R_{ij} > 0$ and $R_{ij} = R_{ji}^{-1}$, for any $i, j = 1, \dots, N$. If $R_{ik} = R_{ij} R_{jk}$, it is said to be consistent.

We can transform a CDM \mathbf{C} into a reciprocal matrix \mathbf{R} with element-wise exponentiation, $\mathbf{R} = \exp(\mathbf{C})$. The consistency property is then the natural extension of the triangular equality to reciprocal matrices and it provides an alternative way of testing if a matrix is a CDM.

It is known that a reciprocal matrix is consistent if and only if it has $N - 1$ zero eigenvalues and one eigenvalue equal to N [190]. As a consequence, we can verify if a matrix is a CDM by examining the eigenvalues of its corresponding reciprocal matrix.

8.2.4. Reconstructing coordinates based on CDMs

To solve Problem 7 we propose to estimate the points from a measured subset of their pairwise differences as:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} f(\mathbf{x}) = \arg \min_{\mathbf{x}} \left\| \mathbf{W} \circ (\mathbf{x}\mathbf{1}^\top - \mathbf{1}\mathbf{x}^\top - \tilde{\mathbf{C}}) \right\|_F^2. \quad (8.3)$$

In [8] we prove that $f(\mathbf{x})$ is convex. Thus, we can find the optimal solution by setting the first derivative of $f(\mathbf{x})$ to zero:

$$x_i = \frac{1}{\Lambda_{ii}} \sum_{j=1}^N (x_j + \tilde{C}_{ij}) W_{ij}, \quad (8.4)$$

where $\Lambda = \text{diag}(\mathbf{W}\mathbf{1})$. We can rewrite this result in matrix form

$$\mathbf{A}\mathbf{x} = \tilde{\mathbf{v}}, \quad (8.5)$$

where $\mathbf{A} = \Lambda - \mathbf{W}$ and $\tilde{\mathbf{v}} = (\tilde{\mathbf{C}} \circ \mathbf{W})\mathbf{1}$.

The matrix \mathbf{A} has a particular structure and belongs to the class of so-called *M-matrices* [100, 165]. It is of rank at most $N - 1$, which is related to the fact that any translation of the point set leads to the same CDM. In order to anchor the translation and obtain a full-rank matrix, we can arbitrarily fix one point of the embedding. Without loss of generality, we set x_0 to zero. Then, we remove the first entry of \mathbf{x} and $\tilde{\mathbf{v}}$ and denote the new vectors by \mathbf{x}' and $\tilde{\mathbf{v}}'$. Similarly, we remove the first row and column of \mathbf{A} to get \mathbf{A}' , and we define the matrices Λ' , \mathbf{W}' and $\tilde{\mathbf{C}}'$ analogously. This yields the following linear system:

$$\mathbf{A}'\mathbf{x}' = \tilde{\mathbf{v}}'. \quad (8.6)$$

We denote the solution of this linear equation as $\hat{\mathbf{x}}' \in \mathbb{R}^{N-1}$, leading to the final solution

$$\hat{\mathbf{x}} = \begin{bmatrix} 0 \\ \hat{\mathbf{x}}' \end{bmatrix} = \begin{bmatrix} 0 \\ (\mathbf{A}')^{-1} \tilde{\mathbf{v}}' \end{bmatrix}. \quad (8.7)$$

We study the invertibility of \mathbf{A}' , required for (8.7) to be defined, in the following two paragraphs.

Complete CDM In the special case where we measure all pairwise differences and assign them the same weight—we call it a *complete CDM*—an analytic solution for $(\mathbf{A}')^{-1}$ exists. Indeed, with $\Lambda' = (N - 1)\mathbf{I}$ and $\mathbf{W}' = \mathbf{1}\mathbf{1}^\top - \mathbf{I}$, we obtain

$$(\mathbf{A}')^{-1} = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top - \mathbf{I}}{N - 1} \right)^{-1} \frac{1}{N - 1} = \frac{\mathbf{I} + \mathbf{1}\mathbf{1}^\top}{N}. \quad (8.8)$$

This can be easily verified by direct computation. To recover \mathbf{x} from a complete CDM, we substitute (8.8) into (8.6):

$$\hat{\mathbf{x}}' = (\mathbf{A}')^{-1} \tilde{\mathbf{v}}' = \frac{\mathbf{I} + \mathbf{1}\mathbf{1}^\top}{N} \tilde{\mathbf{C}}' \mathbf{1} = \frac{1}{N} \tilde{\mathbf{C}}' \mathbf{1} + c\mathbf{1}, \quad (8.9)$$

where $c = \frac{1}{N} \sum_{ij} \tilde{C}'_{ij}$ is a constant that only translates the solution. We conclude that the optimal point recovery in the complete case corresponds to a simple average of the rows of a CDM.

Incomplete and weighted CDM In order to study the invertibility of \mathbf{A}' when the CDM is incomplete, we define the weighted graph $G = (\mathbf{x}, \mathbf{W})$, where vertices are represented by the points \mathbf{x} and their connecting edges are given by \mathbf{W} .¹

We say that the CDM \mathbf{C} is *connected* if and only if its underlying graph G is connected, or in other words, if for all indices $i \neq j$ there is a path of indices i_1, i_2, \dots, i_m such that $W_{i_1 i_1} \neq 0, W_{i_1 i_2} \neq 0, \dots, W_{i_{m-1} i_m} \neq 0, W_{i_m j} \neq 0$. According to Kirchhoff's matrix tree theorem [33], the number of spanning trees of G is given by $\kappa(G) = \det(\mathbf{A}')$. Obviously, G is connected if and only if $\kappa(G) \neq 0$, or equivalently, if and only if \mathbf{A}' is nonsingular.

It is not hard to see that \mathbf{A}' is weakly diagonally dominant, $|A'_{ii}| \geq \sum_{j=1, j \neq i}^{N-1} |A'_{ij}|$, as $|A'_{ii}| = \sum_{j=1}^N W_{i+1, j}$, $|A'_{ij}| = W_{i+1, j+1}$, and $\sum_{j=1}^N W_{i+1, j} \geq \sum_{j=2}^N W_{i+1, j}$ for every row i . When \mathbf{C} is connected, then \mathbf{A}' is reducible and for at least one row i we have a strict inequality $\sum_{j=1}^N W_{ij} > \sum_{j=2}^N W_{ij}$; hence, \mathbf{A}' is irreducibly diagonally dominant. This proves the same result as above, that \mathbf{A}' is nonsingular [180], but it provides two additional insights: it shows that we can solve (8.6) in nearly-linear time in N [194], and it enables a simple derivation for the inverse of \mathbf{A}' : for a connected matrix \mathbf{C} , it is given by Neumann series

$$(\mathbf{A}')^{-1} = \left(\sum_{k=0}^{\infty} ((\mathbf{\Lambda}')^{-1} \mathbf{W}')^k \right) (\mathbf{\Lambda}')^{-1}. \quad (8.10)$$

The proof follows directly from the Gershgorin circle theorem [73], which proves that the eigenvalues of $(\mathbf{\Lambda}')^{-1} \mathbf{W}'$ are inside the unit circle, and the Neumann series theorem [95]. We proof that the infinite sum (8.10) converges to (8.8) in the case of complete measurements, in [8].

Clearly, we cannot apply (8.7) when \mathbf{C} is not connected. What we can do in such a case is recover the points by invoking (8.7) within each *connected component* of \mathbf{C} . As the recovered connected components can shift independently, we have an infinite number of solutions.

¹Using the graph representation, we observe that \mathbf{A} is the Laplacian matrix of its corresponding graph G , as it is the difference of the degree matrix $\mathbf{\Lambda}$ and the adjacency matrix \mathbf{W} . This offers an alternative explanation its rank is at most $N - 1$ [88]

8. Embedding based on coordinate differences

Properties	CDM	EDM
Rank	$\text{rank}(\mathbf{C}) = 2$	$\text{rank}(\mathbf{D}) = D + 2$
Triangular (in)equality	$C_{ij} = C_{ik} + C_{kj}$	$\sqrt{D_{ij}} \leq \sqrt{D_{ik}} + \sqrt{D_{kj}}$
Symmetry	$\mathbf{C} = -\mathbf{C}^\top$	$\mathbf{D} = \mathbf{D}^\top$
Hollowness	$\text{diag}(\mathbf{C}) = \mathbf{0}$	$\text{diag}(\mathbf{D}) = \mathbf{0}$
Invariance to	translations	all rigid motions

Table 8.2.: Comparison of CDMs \mathbf{C} and EDMs \mathbf{D} .

8.2.5. Comparison with EDMs

In this section, we relate the introduced framework of CDMs with the more mature field of EDMs. Oftentimes, an EDM is the tool of choice simply because the coordinate information is not available. However, due to the maturity of the field of Distance Geometry, vector measurements are sometimes converted to Euclidean distances, which results in a loss of information and prevents the use of the more adequate CDM framework. The molecular conformation application is one such example: even though coordinate information is available through different angle measurements, more focus was given to distance-based methods to this date [20].

We compare the two frameworks with respect to matrix properties, number of required measurements and available algorithms.

First, there are some obvious algebraic differences between CDMs and EDMs, summarized in Table 8.2. Two properties are particularly interesting: First, the rank of both matrices is independent of the number of points. For CDMs constructed for 1D points, the rank is equal to 2, while for EDMs of D -dimensional points it is equal to $D + 2$. Secondly, when measuring coordinate differences, information about the absolute translation of the points is irremediably lost, but rotation and reflection are preserved. When measuring Euclidean distances, we can only recovery the point set up to a rigid motion.

Secondly, we compare the required number of measurements. For CDMs, the lower bound of number of measurements is achieved when we have $N - 1$ connected measurements per dimension, resulting in a total of $D(N - 1)$ measurements (more on this in Section 8.3.7). We have seen in Section 7.1.6 that for EDMs, we need at least $ND - (D + 1)D/2$ measurements. As the dimension of the space in most applications is typically 2 or 3, the contribution of the second term is negligible and we can conclude that the same number of measurements is required for both cases.

Finally, the most important advantage of coordinate differences over Euclidean distances is that an optimal recovery algorithms exists. For EDMs, no algorithm guarantees an optimal solution when distances are noisy and/or missing, which manifests itself in the plethora of suboptimal iterative and relaxed algorithms presented in Section 7.1.4. In particular, the intermediate completion and denoising steps are not required for CDMs, as the point recovery is immediate.

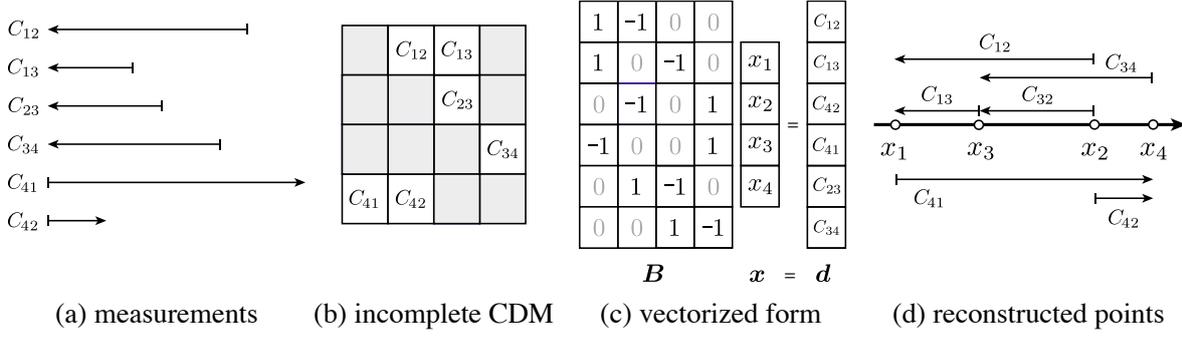


Figure 8.3.: An instance of Problem 7. (a) Assume that we can measure a subset of noisy 1D differences between the points. (b) We arrange them in a CDM and use it either directly, or (c) its vectorized form to (d) reconstruct the points that give rise to measurements.

8.3. Solution in multiple dimensions: optimal vs. splitting

8.3.1. Vector form in one and multiple dimensions

To discuss the multidimensional solution, we introduce the “vector form” of CDMs. We first note that the coordinate difference vector \mathbf{d}_k in frame φ_k can be written in terms of the points $\mathbf{x} = [x_1^\top \dots x_N^\top]^\top \in \mathbb{R}^{ND}$ as

$$\mathbf{B}_k \Phi_k \mathbf{x} = \tilde{\mathbf{d}}_k. \quad (8.11)$$

Here, we have constructed the matrix $\Phi_k = \mathbf{I}_N \otimes \varphi_k^\top \in \mathbb{R}^{N \times ND}$, and we have introduced a sparse measurement matrix $\mathbf{B}_k \in \mathbb{R}^{M_k \times N}$ as follows: assuming row m corresponds to the difference $x_i - x_j$, we have $B_{k,mi} = 1$, $B_{k,mj} = -1$ and 0 otherwise. \mathbf{B}_k has identical rows for every repeated measurement and $\tilde{\mathbf{d}}_k$ contains all of their realizations. An example of \mathbf{B}_k in one direction and the corresponding CDM are shown in Figure 8.3. As mentioned earlier, problem (8.11) is closely related to node localization from relative measurements [16], with the exception that we make the fact that we may measure the differences in arbitrary frame directions explicit. Furthermore, we do not consider fixed anchors here, but they could be easily integrated by splitting the linear system into known and unknown elements of \mathbf{x} .

Recall that in Section 8.2.2 we defined \mathbf{W} to be a non-negative matrix whose entries are the weights of each difference measurement. To be aligned with such a definition, we should allow not only multiple measurements of each difference (identical rows in \mathbf{B}_k), but also different weights for these measurements. To do so, we can simply scale rows of \mathbf{B}_k and $\tilde{\mathbf{d}}_k$ with the same factors. To keep the notation clean, in the rest of the chapter we assume that \mathbf{B}_k and $\tilde{\mathbf{d}}_k$ encompass these weights.

Each coordinate direction $k = 1 \dots K$ gives us one system of equations of the form (8.11). We combine them by introducing

$$\mathbf{B} = \text{diag}(\mathbf{B}_1, \dots, \mathbf{B}_K) \in \mathbb{R}^{M \times NK}, \text{ and } \Phi = [\Phi_1^\top, \dots, \Phi_K^\top]^\top \in \mathbb{R}^{NK \times ND}. \quad (8.12)$$

8. Embedding based on coordinate differences

The final system equations is then given by

$$\mathbf{B}\Phi\mathbf{x} = \tilde{\mathbf{d}}, \quad (8.13)$$

8.3.2. Optimal solution

Analogously to (8.5) and due to the translation invariance in each coordinate, the system (8.13) is non-invertible. Again, we fix the first coordinate of every dimension to zero. This corresponds to removing every K th column of \mathbf{B} , as well as every K th row and the first D columns of Φ , resulting in the new matrices $\mathbf{B}' \in \mathbb{R}^{M \times (N-1)K}$ and $\Phi' \in \mathbb{R}^{(N-1)K \times (N-1)D}$, respectively. Then, we can reconstruct the absolute coordinates of the points as

$$\hat{\mathbf{x}}'_o = (\mathbf{B}'\Phi')^\dagger \tilde{\mathbf{d}}. \quad (8.14)$$

This solution is optimal in the least-squares sense, and reduces to the best linear unbiased estimator (BLUE) discussed in [16], when considering only Cartesian coordinates and no prior knowledge of noise. Noise information could in fact be incorporated in (8.14) using the covariance matrix as in [16], but this extension is straight-forward and omitted here to avoid cluttered notation.

8.3.3. Splitting algorithm

To reduce the complexity of the solution, we propose and study an alternative approach that decomposes Problem 6 into many instances of Problem 7 of smaller dimension. We divide the multidimensional problem into K one-dimensional CDM recovery problems, and estimate the points from their differences separately in each frame direction. Therefore, we first recover the expansion coefficients of the points for each frame vector, given by $(\mathbf{B}')^\dagger \tilde{\mathbf{d}}$, and then we find their canonical coordinates by a change of basis:

$$\hat{\mathbf{x}}'_s = (\Phi')^\dagger (\mathbf{B}')^\dagger \tilde{\mathbf{d}}. \quad (8.15)$$

8.3.4. Statistical Analysis

In this section, we provide a closed-form expression for the difference in the estimation errors of the optimal and the splitting algorithm.

We assume that we add independent Gaussian noise to all the differences, such that $\tilde{\mathbf{d}} \sim \mathcal{N}(\mathbf{d}, \sigma^2 \mathbf{I})$. We first estimate the points $\hat{\mathbf{x}}'_o$ and $\hat{\mathbf{x}}'_s$ from (8.14) and (8.15), and add the first point $\mathbf{x}_0 = \mathbf{0}$. Then, we create $\mathbf{J}'_D \in \mathbb{R}^{ND \times ND}$ by generalizing the centering matrix \mathbf{J} (see Section 7.1) to D dimensions, such that the sum of the point coordinates in every separate dimension is 0:

$$\mathbf{J}'_D = \mathbf{J} \otimes \mathbf{I}_D. \quad (8.16)$$

It follows that the centered estimated points, $\hat{\mathbf{x}}_o = \mathbf{J}'_D \hat{\mathbf{x}}'_o$ and $\hat{\mathbf{x}}_s = \mathbf{J}'_D \hat{\mathbf{x}}'_s$ have Gaussian distributions with the following parameters:

$$\begin{aligned} \hat{\mathbf{x}}_o &\sim \mathcal{N}(\mathbf{x}, \Sigma_{\hat{\mathbf{x}}_o}), \text{ where } \Sigma_{\hat{\mathbf{x}}_o} = \sigma^2 \mathbf{J}_D (\mathbf{B}\Phi)^\dagger ((\mathbf{B}\Phi)^\dagger)^\top (\mathbf{J}_D)^\top, \\ \hat{\mathbf{x}}_s &\sim \mathcal{N}(\mathbf{x}, \Sigma_{\hat{\mathbf{x}}_s}), \text{ where } \Sigma_{\hat{\mathbf{x}}_s} = \sigma^2 \mathbf{J}_D \Phi^\dagger \mathbf{B}^\dagger (\mathbf{B}^\dagger)^\top (\Phi^\dagger)^\top (\mathbf{J}_D)^\top. \end{aligned} \quad (8.17)$$

For a less cluttered notation, in (8.17) and the rest of the section, we omit the prime symbol ' on $\mathbf{J}_D, \mathbf{B}, \Phi$.

We define the estimation error vectors of the optimal and splitting algorithms as $\mathbf{e}_o = \hat{\mathbf{x}}_o - \mathbf{x}$ and $\mathbf{e}_s = \hat{\mathbf{x}}_s - \mathbf{x}$, respectively. The expectations of the mean squared errors are $\mathbb{E}\{\frac{1}{N} \|\mathbf{e}_o\|^2\} = \frac{1}{N} \text{tr}(\Sigma_{\hat{\mathbf{x}}_o})$ and $\mathbb{E}\{\frac{1}{N} \|\mathbf{e}_s\|^2\} = \frac{1}{N} \text{tr}(\Sigma_{\hat{\mathbf{x}}_s})$.

Simulation results We consider Gaussian noise with 0-mean and $\sigma = 0.01$ added to the difference measurements, and we assume the complete case in 2D, such that both approaches are optimal. Then, for a given pair (N, K) , we generate K directions of frame vectors uniformly at random from $[0, 2\pi)$ and the complete measurements matrix \mathbf{B} . Figure 8.4 shows that the estimation error decreases with the number of frame vectors K and the number of points N .

8.3.5. The cost of splitting

To evaluate the performance of the proposed splitting algorithm with respect to the optimal solution, we define the *cost of splitting* c , as the normalized squared norm of the difference between the two estimators, $c = \frac{1}{N} \|\hat{\mathbf{x}}_o - \hat{\mathbf{x}}_s\|^2$.

We can compute the expected cost of splitting as:

$$\begin{aligned} \mathbb{E}\{c\} &= \frac{\sigma^2}{N} \text{tr} \left[\left(\mathbf{J}_D \left((\mathbf{B}\Phi)^\dagger - \Phi^\dagger \mathbf{B}^\dagger \right) \right)^\top \left(\mathbf{J}_D \left((\mathbf{B}\Phi)^\dagger - \Phi^\dagger \mathbf{B}^\dagger \right) \right) \right] \\ &= \frac{\sigma^2}{N} \left(\text{tr} \left[(\mathbf{J}_D)^\top \mathbf{J}_D \Phi^\dagger (\mathbf{B}^\top \mathbf{B})^{-1} (\Phi^\dagger)^\top \right] - \text{tr} \left[(\mathbf{J}_D)^\top \mathbf{J}_D \left((\mathbf{B}\Phi)^\top (\mathbf{B}\Phi) \right)^{-1} \right] \right) \\ &= \frac{\sigma^2}{N} \text{tr}(\Psi(\Phi, \mathbf{B})) - \frac{\sigma^2}{N^2} \mathbf{1}^\top \Psi(\Phi, \mathbf{B}) \mathbf{1}, \end{aligned} \quad (8.18)$$

where $\Psi(\Phi, \mathbf{B}) = \Phi^\dagger (\mathbf{B}^\top \mathbf{B})^{-1} (\Phi^\dagger)^\top - (\Phi^\top \mathbf{B}^\top \mathbf{B}\Phi)^{-1}$.

As the splitting approach leads to a more efficient algorithm, identifying cases in which we can apply the splitting algorithm and still obtain an optimal solution is valuable. From (8.18) it follows that the cost is equal to 0 if $(\mathbf{B}\Phi)^\dagger = \Phi^\dagger \mathbf{B}^\dagger$, or equivalently, if $\Psi(\Phi, \mathbf{B}) = \mathbf{0}$.

For any matrices of appropriate dimensions \mathbf{A} and \mathbf{B} , $(\mathbf{A}\mathbf{B})^\dagger = \mathbf{B}^\dagger \mathbf{A}^\dagger$ holds if, for instance [79],

- \mathbf{A} has orthonormal columns; or

8. Embedding based on coordinate differences

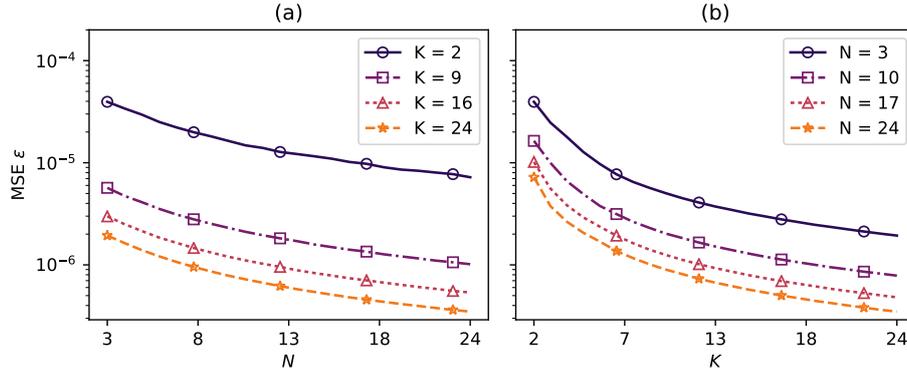


Figure 8.4.: Estimation error ϵ in 2D. We assume complete CDM for every frame vector and Gaussian noise with 0-mean and $\sigma = 0.01$ on the measurements. (a) Dependence of ϵ on N for fixed K . (b) Dependence of ϵ on K for fixed N .

- B has orthonormal rows; or
- A has linearly independent columns and B has linearly independent rows.

For B and Φ as defined in (8.12), none of the above hold in general. In the following, we prove however that there are at least two practical cases for which $(B\Phi)^\dagger = \Phi^\dagger B^\dagger$: First, when $K = D$ and second, when $B_k = B_0$ for every $k = 1, \dots, K$.

Case $K = D$: When the number of frame vectors is the same as the dimension of the space, Φ is invertible, thus $\Phi^\dagger = \Phi^{-1}$. The splitting algorithm results in the optimal solution:

$$\begin{aligned} \hat{x}_o &= (B\Phi)^\dagger \tilde{d} \\ &= (\Phi^\top B^\top B\Phi)^{-1} \Phi^\top B^\top \tilde{d} \\ &= \Phi^{-1} (B^\top B)^{-1} B^\top \tilde{d} = \hat{x}_s. \end{aligned}$$

Case $B_k = B_0$ for every k : Assume that all measurement matrices for every frame direction are equal to the matrix B_0 . For instance, this is the case when we observe all pairwise differences on all frame vectors k . In that case, $B = I \otimes B_0$, where \otimes is the Kronecker product. We change the order of the entries in \tilde{d} and the order of the corresponding columns in Φ , so that we can write $\Phi = \Phi_0 \otimes I$, where the k th row of $\Phi_0 \in \mathbb{R}^{K \times D}$ is φ_k^\top , for every $k = 1, \dots, K$. Note that this does not influence the estimation of the points, but it provides a simpler expression for Φ via the Kronecker product. Then, we can show that the optimal

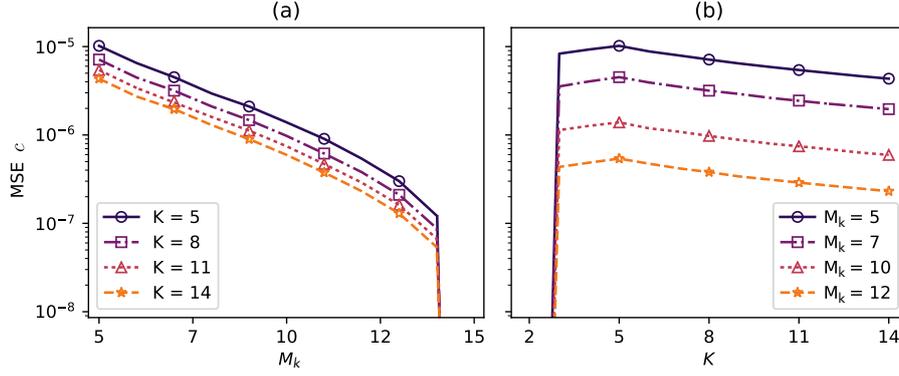


Figure 8.5.: The cost of splitting c in 2D. We consider the setup of $N = 6$ points and assume Gaussian noise with 0-mean and $\sigma = 0.01$ on the measurements. (a) Dependence of c on M_k for fixed K . (b) Dependence of c on K for fixed M_k .

solution is equal to the solution of the splitting algorithm:

$$\begin{aligned}
 \hat{x}_o &= (\Phi^\top B^\top B \Phi)^{-1} \Phi^\top B^\top d \\
 &= \left((\Phi_0^\top \otimes I) \left(I \otimes (B_0^\top B_0) \right) (\Phi_0 \otimes I) \right)^{-1} \Phi^\top B^\top d \\
 &= \left((\Phi_0^\top \Phi_0)^{-1} \otimes (B_0^\top B_0)^{-1} \right) \Phi^\top B^\top d \\
 &= (I \otimes (B_0^\top B_0)^{-1}) \left((\Phi_0^\top \Phi_0)^{-1} \otimes I \right) (\Phi_0 \otimes I) B^\top d \\
 &= \Phi_0^\dagger \otimes B_0^\dagger d = \Phi^\dagger B^\dagger d = \hat{x}_s.
 \end{aligned}$$

Simulation results To visualize the difference between the two approaches, in Figure 8.5 we plot the cost of splitting (8.18) for different number of measurements M_k and different number of frame vectors K . We consider $N = 6$ points in 2D drawn from $[0, 1]^2$, so the number of measurements ranges from the smallest value $M_k = N - 1 = 5$ to the complete case $M_k = N(N - 1)/2 = 15$. We assume that M_k is the same for every direction k . We have seen that the cost should be zero for $K = D = 2$ and for the complete case; Figure 8.5 confirms these two special cases. In addition, we observe that the cost decreases with M_k and K .

8.3.6. Algorithmic considerations

In this section, we compare the proposed algorithms in terms of their numerical complexity and accuracy on simulated data. We experiment with three different implementations of the optimal and splitting methods. Two of the implementations are related to the vector form of Section 8.3, and one is a direct implementation of our initial CDM formulation from Section 8.2.4, generalized to higher dimensions. We describe them in more details

8. Embedding based on coordinate differences

below:². The *standard* and *CDM* implementations are done using `numpy` and LAPACK's DGELSD solver. The *sparse* implementations use the `scipy.sparse` package and the sparse iterative LSQR solver [156].

- *Standard implementation.* We implement the optimal and the splitting algorithms as described in Section 8.3.2 and Section 8.3.3, respectively. The matrices \mathbf{B} , \mathbf{B}_k , Φ are represented with a standard linear algebra library. Least-squares solvers are implemented with the same library and use SVD to solve the linear problem.
- *Sparse implementation.* The splitting and the optimal algorithms solve the same linear problem as in the standard implementation, but the matrices \mathbf{B} , \mathbf{B}_k , Φ are represented as Compressed Sparse Row matrices (CSR).
- *CDM implementation.* To avoid costly computations resulting from the sparse and large matrices of the vector form, we leverage the matrix representation offered by CDMs. For the splitting strategy, we create K CDMs $\tilde{\mathbf{C}}_k$, one for each frame vector. By counting the number of pairwise comparisons in every \mathbf{B}_k , we compose K weight matrices \mathbf{W}_k . For every k , we compute the matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ from $\tilde{\mathbf{C}}_k$ and \mathbf{W}_k as in Section 8.2.4. Similarly, when implementing the optimal algorithm, we directly compute a matrix equivalent to the product $\Phi^\top \mathbf{B}^\top \mathbf{B} \Phi \in \mathbb{R}^{DN \times DN}$, without having to set up Φ and \mathbf{B} , whose dimension M can be arbitrarily large. This is done through derivations analogous to Section 8.2.4, where (8.3) is generalized to take into account all contributions from the frame,

$$f(\mathbf{x}) = \sum_{k=1}^K \left\| \mathbf{W}_k \circ (\Phi_k \mathbf{x} \mathbf{1}^\top - \mathbf{1} \mathbf{x}^\top \Phi_k^\top - \tilde{\mathbf{C}}_k) \right\|_F^2. \quad (8.19)$$

In all implementations, we exploit the distributed nature of the splitting approach by parallelizing it on 8 CPU cores.

Simulation setup Throughout all experiments, N 2D points are chosen uniformly at random from the interval $[0, \sqrt{N}]^2$, such that the average density is one point per unit area. The directions of K frame vectors are picked uniformly at random from $[0, 2\pi]$. We varied the number of points from $N = 10$ to $N = 1000$, and we chose an average of 4 measurements per frame vector and per point, while ensuring that that the CDM for every k is connected. For such a connectivity value, the large memory requirement restricted us from simulating the standard implementations for $N \geq 100$.

The time and error averages are taken over 100 independent geometry and noise realizations. Note that the reported times do not include the time it takes to simulate the measurements. It is therefore a measure of how fast we can solve the different least squares problems with the proposed implementations.

²The methods are implemented using standard python libraries and solvers on a Linux server with processor model Intel(R) Xeon(R) CPU X5675 @ 3.07GHz

8.3. Solution in multiple dimensions: optimal vs. splitting

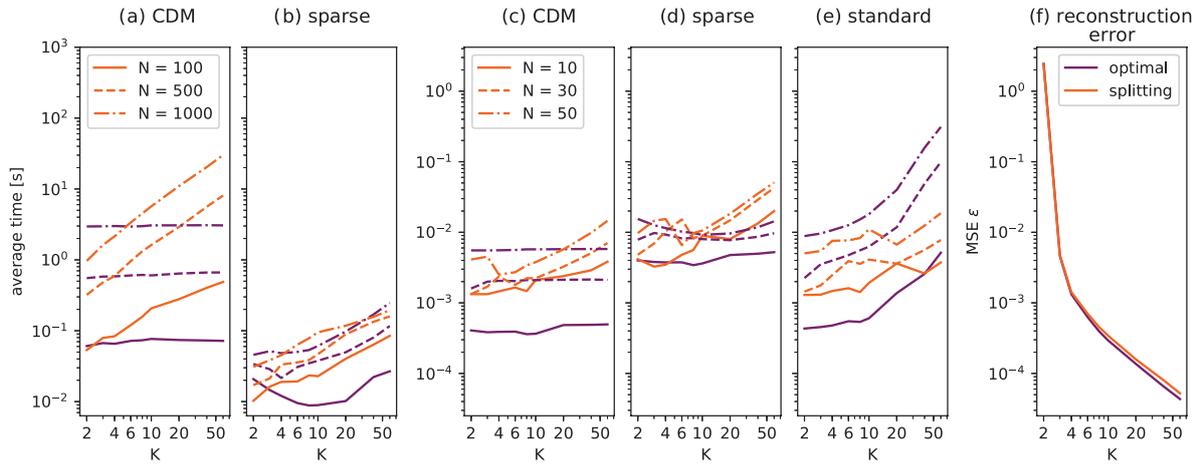


Figure 8.6.: *Evaluation of proposed methods in terms of efficiency (average running time in seconds) and reconstruction error (MSE). Figures (a) and (b) show results for high N , for which the standard implementation is infeasible due to memory issues. The number of directions ranges from the minimum required $K = D = 2$ to $K = 60$. For each implementation, the optimal and splitting approaches are highlighted in purple and orange, respectively. The reconstruction error in (f) is independent of the implementation strategy and the number of points N . The standard deviation of the Gaussian noise added is $\sigma = 0.1$.*

Observations The plots in Figure 8.6 show the simulation results of the three described implementations. There is no single method that outperforms the others for every combination of N and K , thus we discuss below multiple important findings.

Figure 8.6a and Figure 8.6c confirm that the CDM implementation of the optimal method does not depend on K . This is a significant advantage over the implementation strategies that rely on the vector form, for which the execution times grow with K . Moreover, to optimally benefit from the CDM implementation, one should choose the splitting algorithm since it is faster than the optimal method when K is below a certain threshold dependant on N .

When N is larger than 100 and K is smaller than 60, the sparse implementations of the vector form exhibit shorter execution time than the CDM approach; this can be seen by comparing Figure 8.6a and Figure 8.6b. However, as the execution times of sparse implementations increase with K , the CDM implementation of the optimal method surpasses its sparse implementation after a certain value of K .

In the standard implementation, the optimal solution relies on the SVD of a matrix whose number of rows equals the total number of measurements; this becomes prohibitively large with increasing K . As a consequence, the splitting algorithm becomes attractive for higher values of K ; for more than 30 points splitting is on average always faster than the optimal solution. However, the standard implementation does not compare favorably neither with the CDM implementation, nor with the sparse implementation for any N and K .

The last column shows that we lose little in terms of accuracy by applying the splitting algorithm instead of the optimal solution. The reconstruction error is independent of the number of points N and the type of implementation.

8. Embedding based on coordinate differences

Finally, we would like to stress that more efficient and robust solvers for the systems of linear equations can be used. In particular, methods that avoid squaring the condition number and exploit the diagonally dominant structure of \mathbf{A} are preferable for applications requiring fast performance. We expect that the splitting algorithm would specifically benefit from such methods, as the corresponding linear system involves the inversion of irreducibly diagonally dominant matrices, which is not the case for the optimal formulation.

8.3.7. Uniqueness and number of solutions

Given a set of coordinate differences, there is an infinite number of valid point sets which are generated by translating the original point set. Here, we refer to a problem with a *non-unique* solution when at least two different point sets that are not translated versions of each other are both valid solutions.

For the multidimensional point reconstruction to be possible, we need to extend the connectivity argument from Section 8.2.4. For a unique solution to exist, every point needs to be connected with the others by at least D frame measurements. Since we have N points and therefore at least $N - 1$ connections, the minimum number of measurements for a unique solution is $D(N - 1)$. When using the splitting algorithm, we require more measurements: the CDM corresponding to every frame vector needs to be connected for a solution to exist, and hence, we need at least $(N - 1)K$ measurements. This is summarized in the following proposition.

Proposition 2. We refer to the ensemble of all translated reconstructions as one single solution. Then, the CDM problem in 1D can have either one or an infinite number of solutions. It has one solution if and only if the CDM is connected. The CDM problem in D dimensions can have either one solution or an infinite number of solutions. If the CDMs of at least D independent frame vectors are connected, a unique solution exists.

Note that the uniqueness condition for $D > 1$ is sufficient, but not necessary. The study of the exact number of solutions is tightly connected with the (global) graph rigidity problem [120], and is out of the scope of this thesis.

8.4. Applications

Coordinate differences can be measured in a variety of applications from diverse fields. For this thesis, we study three setups. First, we can do position calibration by sending a calibration signal from known directions φ of a source in far field, and measure its time of arrival at each node (see Figure 8.7 (a)). Since we cannot usually ensure synchronization between the source and the nodes, we can work with time differences of arrival (TDOAs) with respect to one reference node. Knowing the speed of propagation of the signal, these TDOAs can be translated to coordinate differences along the given direction. Although D directions are enough in theory, we can introduce redundancy by measuring $K > D$

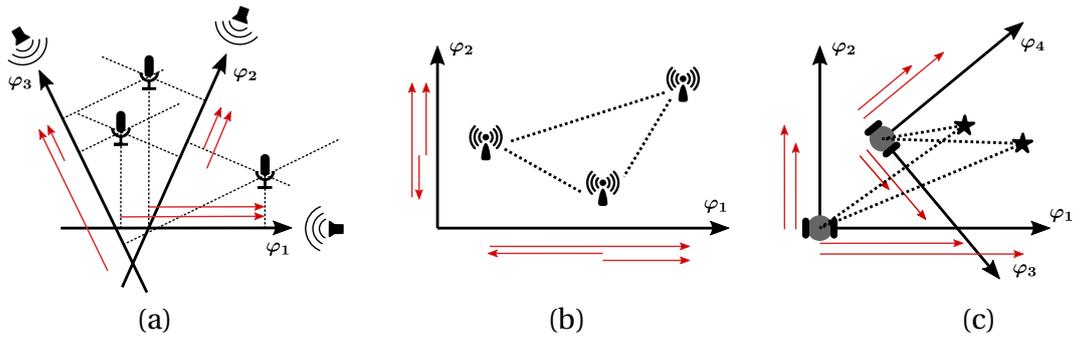


Figure 8.7.: Examples of real-world situations in which we can measure coordinate differences (in red). (a) TDOA measurements from a moving sound source, useful for e.g. sensor array calibration, (b) static sensor network localization with distance and angle measurements, and (c) structure-from-motion with fixed landmarks.

directions, which improves accuracy. In Section 8.4.2, we present experiments of this kind for microphone array calibration.

The second application, shown in Figure 8.7 (b), involves measuring distances and angles between nodes, as discussed previously for the molecular conformation problem. Such measurements can be translated to coordinate differences by considering the projection to the Cartesian coordinates ($K = D$). We localize a static ad-hoc sensor network using such measurements in Section 8.4.1.

Finally, in ongoing work we study how the structure-from-motion problem can be framed in terms of coordinate differences: a robot moves along an approximately known path and measures the range and bearing of fixed landmarks. Again, we can project to the local Cartesian coordinates, but for a multitude of different poses, yielding $K > D$ projections. We depict an example setup in Figure 8.7 (c) and briefly discuss this problem in Section 8.5.

In [8], we present additionally the molecular conformation problem, the sports ranking problem, and phase retrieval problem. A more detailed treatment of the phase retrieval problem can also be found in [9].

We emphasize that our goal with these applications is not to demonstrate the superior performance of our method over state-of-the-art techniques, but rather to provide evidence for a wide range of applications of the coordinate-difference framework.

8.4.1. Multi-modal sensor array localization

In this section, we consider the joint localization of nodes that can measure distances and angles between each other. This problem is of relevance in many application areas, such as wireless sensor networks [16, 17, 32, 151] or robot systems with relative measurements between poses [39, 40, 176] or between multiple robots [136]. We note that in some of these works, the node orientations are not known, leading to a more challenging optimization problem.

In many areas of interest the devices can however be equipped with compasses, in which

8. Embedding based on coordinate differences

case the positioning problem reduces to a linear system, which is what we study here. The same setup is considered in [16], where distributed algorithms are investigated, while studying how errors propagate using graph-theoretic notions. The core algorithm is equivalent to our method, but does not include extension to frame vectors, which is required for the next section's application. Similarly, [151] equip nodes with compasses and propose both centralized and distributed approaches, but use distance and angle estimates in an interleaved fashion. Instead, we work with a centralized approach and tightly integrate distance and angle estimates. We compare our algorithm with E-MDS [131], discussed in Section 7.3, and demonstrate its superior performance. Considering that our method uses absolute orientations while E-MDS uses inner angles only, this finding confirms that absolute orientation should be exploited whenever available.

Our goal is to estimate points $\hat{\mathbf{x}}_i$ from a noisy subset of measured distances \tilde{d}_{ij} and angles $\tilde{\alpha}_{ij}$. The noisy coordinate differences are recovered from \tilde{d}_{ij} and $\tilde{\alpha}_{ij}$ in the x and y direction, using $x_{ij} = \tilde{d}_{ij} \cos \tilde{\alpha}_{ij}$ and $y_{ij} = \tilde{d}_{ij} \sin \tilde{\alpha}_{ij}$.

We observe that this is a 2D point recovery problem, which can be optimally solved with the splitting algorithm from Section 8.3.3; we can decompose the problem into two independent sub-problems and solve each independently with (8.7). In this case, we always work in the canonical basis, hence $\Phi = \mathbf{I}$.

We compare the CDM algorithm with E-MDS [131] and MDS [112]. E-MDS uses inner angles, not absolute angles, and MDS uses no angles at all. Therefore, this is not an algorithmic comparison, but it allows us to investigate the utility of absolute angle measurements. We consider $N = 10$ points chosen uniformly at random from $[0, 1]^D$. As in most real-world applications the measurements of distances and angles are obtained in an independent manner from time-of-arrival and angle-of-arrival estimates, we also generate independent additive noise for these quantities. We assume Gaussian noise with 0-mean and standard deviation σ_d and σ_α , respectively. Note that noise exceeding $\pm\pi$ would distort the angle noise distribution, but for the range of standard deviations chosen in these experiments, this effect is negligible.

We evaluate the performance of the three algorithms using the root mean squared error (RMSE) between the original and the estimated point sets and illustrate its dependence on the noise levels in Figure 8.8. For a more convenient comparison, we slice the graphs at four different values of σ_d and σ_α . Figure 8.8 (a) shows the dependence of the RMSE on the distance noise level for two chosen levels of angle noise: low ($\sigma_\alpha = 0.11$), and high ($\sigma_\alpha = 0.5$). We observe that for low σ_α it is advantageous to include angle measurements, since both multimodal methods achieve smaller error than MDS. For higher σ_α , our method still outperforms MDS, as long as the distance noise is not too small. The dependence of the RMSE on the angle noise level for two chosen levels of distance noise is shown in Figure 8.8 (b). For low distance noise ($\sigma_d = 0.05$), we observe that using multimodal methods is beneficial only when angle noise is low; otherwise the angle information becomes detrimental and one should rely on the distance-based method MDS. However, for higher distance noise ($\sigma_d = 0.15$), using angles significantly improves the result for all considered noise levels.

Numerical simulations show that the proposed algorithm based on CDMs surpasses E-MDS

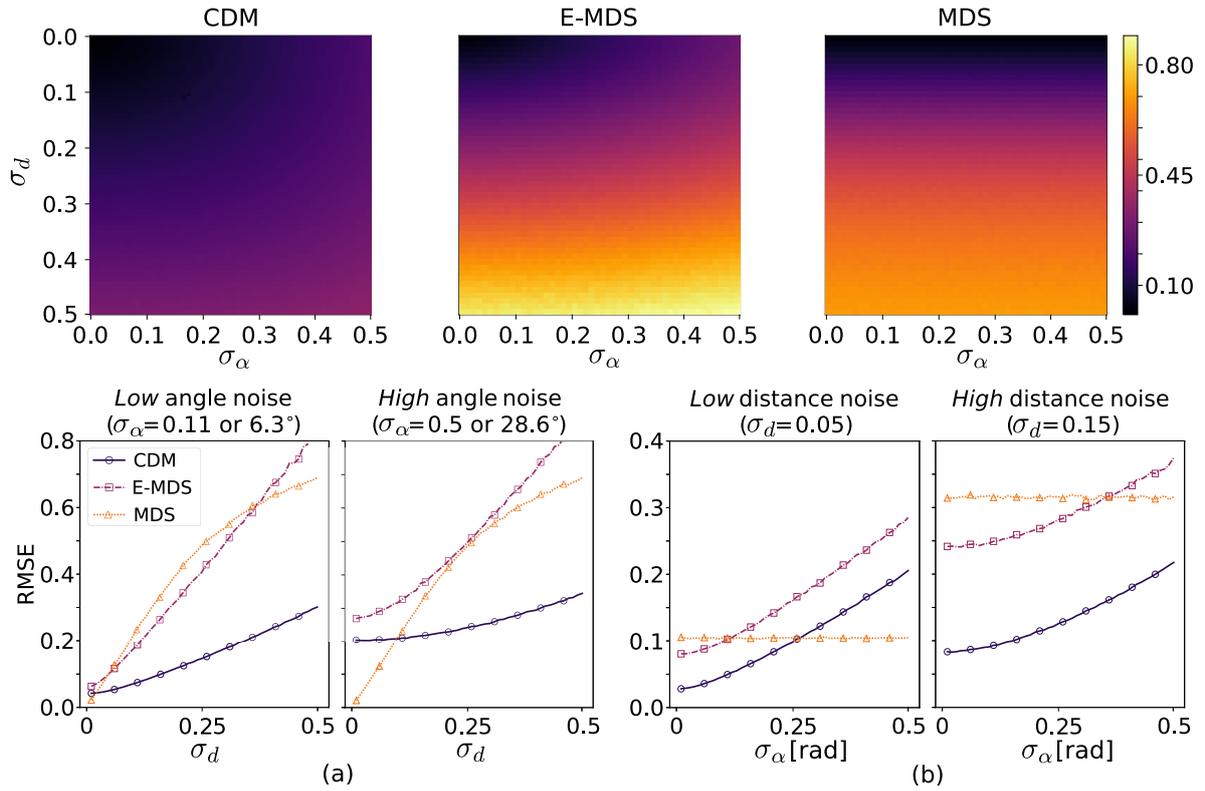


Figure 8.8.: Comparison of the CDM with E-MDS and MDS for different distance and angle noise. (a) RMSE vs. σ_d for two fixed σ_α . (b) RMSE vs. σ_α for two fixed σ_d .

for every pair of $(\sigma_d, \sigma_\alpha)$. This shows that absolute angle information, when available, should be preferred over inner-angle information. It also compares favorably with range-only based method MDS, except for the case of high σ_α and low σ_d .

8.4.2. Sensor array calibration

We aim to localize N sensors from the measurements of K calibration sources placed at angles $\{\varphi_k\}_{k=1}^K$ in the far field, with $K \geq D$. The far-field assumption implies that the sources emit plane waves and the incident angle for a given source location is the same for all sensors. This notation is illustrated in Figure 8.7 (a), and the experimental setup is shown in 8.9. The setup includes a microphone array called *Pyramic* [183] and three loudspeakers in an anechoic chamber. Pyramic (see Figure 8.9b) is a pyramidal array composed of six branches, each containing eight microphones (see Figure 8.9c). The shortest distance between two microphones is 8 mm, while their maximum distance is 200 mm. As the loudspeakers are located as far as possible to emulate far-field conditions, it is more practical to place the Pyramic array on a turntable and rotate it instead of the speakers, yielding the same desired relative orientation.

The measurements from the calibration sources are conducted in the following way: The speakers produce waves at unknown times, and the sensors register the absolute times

8. Embedding based on coordinate differences

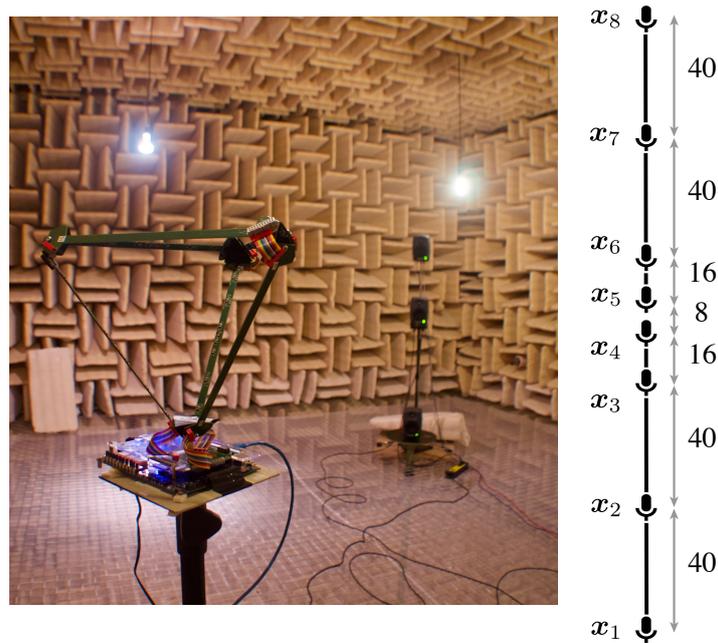


Figure 8.9.: Calibration of sensor arrays: geometry and real experiment. Left: Pyramic array (foreground) and speakers (background) in an anechoic chamber. Right: eight microphones placed in one branch of the Pyramic array, where distances are in mm.

of arrival (TOA) of the waves, denoted τ_{nk} for the n th sensor and the k th source. We use sensor $n = 0$ as the reference and subtract its TOA from all other measurements, yielding TDOA measurements. Knowing the propagation speed, these measurements can be seen as coordinate differences of the sensors' locations, projected onto the directions of the frame vectors φ_k . In accordance with the theory developed in Section 8.3, we can thus reconstruct the sensors' locations in D dimensions as long as we have $K \geq D$ non-collinear calibration sources, and enough measurements for the connectivity requirement (see Section 8.3.7).

We localize the 24 microphones of the top triangle of the Pyramic array using 90 calibration directions uniformly spaced in $[0, 180^\circ)$. We use three loudspeakers at different heights in the far field (3.5 meters away). The measurement directions correspond to the directions of arrival of the sound played by the loudspeakers, so we have a total of $K = 3 \times 90 = 270$ vectors φ_k .

All microphones worked properly throughout the experiment, so for every $k = 1, \dots, 270$ we can compute the pairwise differences of the sound detection times for all pairs of $N = 21$ microphones, arrange them in a measurement vector $\tilde{\mathbf{d}}$ and create a corresponding matrix $\mathbf{B}_k \in \mathbb{R}^{210 \times 21}$, where 210 is the number of differences: $N(N-1)/2$. Then, we use the splitting algorithm (8.15) to find the locations of the microphones. As $\mathbf{B}_k = \mathbf{B}_0$ for every k , it leads to the optimal solution. Our method localizes the microphones with a MSE of $4.49 \mu\text{m}$ when using all speakers and all directions. The error remains almost unchanged ($4.45 \mu\text{m}$) when considering only the middle speaker (placed at the same height as the top triangle of the Pyramic array), and localizing the microphones in 2D.

8.5. Conclusion and ongoing work

We have formulated and analysed coordinate difference matrices, simple tools that enable us to introduce an efficient optimization framework for reconstructing point sets from their noisy and partial coordinate differences. We have established the properties of CDMs and compared them to the classical EDMs, highlighting important differences, in particular the existence of optimal recovery algorithms for CDMs. The structure of CDMs also relates to graph theory which in turn helps us provide necessary and sufficient conditions for the proposed framework to work.

Porting CDMs to multiple dimension, we have seen that many different problems can be solved with the proposed framework; either by exploiting CDMs in each distance separately (what we call the “splitting” approach), or by formulating a combined optimization problem, leading to the “optimal” approach. We have shown when the two approaches are equivalent, and have showcased their performance on two applications. First, we have studied the sensor network localization problem, where distance and absolute angle measurements can be projected to the D Cartesian coordinates, and shown that the proposed solution outperforms E-MDS. Then, we have applied the framework to microphone array calibration, where we directly measure coordinate differences in K calibration directions, and have shown that we can obtain millimeter-level accuracy.

One interesting application that we have not exploited yet, is structure from motion (SfM). SfM is a fundamental problem in robotics and computer vision, where we seek to find the positions of landmarks and camera poses from images of the landmarks at said poses. In the considered instance of SfM, we assume that we have solved the data association problem, so we are given a set of (labeled) distances and angles to a set of fixed and passive landmarks. We can optionally incorporate proprioceptive measurements, bringing the problem closer to visual SLAM; these measurements also consist of distance and angle measurements between points. This measurement setup leads to a particular form of the localization problem, visualized in Figure 8.7 (c). We note that this problem is a combination of the previous two applications: we project distance and angle measurements to Cartesian coordinates, but the frame moves over time, yielding $K > D$ different projections. A few interesting questions arise in this setup:

- If we use proprioception to estimate the relative movement, the pose estimates will drift over time, yielding inaccurate projection directions and coordinate offsets. This calls for a study of the sensitivity of the proposed methods to noise on the directions, and potentially even the extension to unknown directions.
- We have mentioned briefly that CDMs can be used for labeling, but we have not fully explored this avenue yet. In the SfM problem, labeling (called data association here) is crucial and the study of CDMs in this context is therefore promising.

The answers to these and more questions are subject to further investigation.

9 | Embedding based on inner angles

Ce n'est pas la distance qui mesure l'éloignement
— *Antoine de Saint-Exupéry*

Up to this point, we have seen how to embed point sets from either only inter-point distances or from joint distance and angle measurements. The last missing piece of the puzzle of this thesis is point embedding from inner-angle measurements only. No exhaustive theory exists to denoise and complete angle measurements in a similar way to EDMs. In this chapter, we attempt to fill this gap by characterizing inner angles between points. In particular, we are interested in the question of realizability (see Section 7.1.3), which tells us if we can find a point set in a given dimension which exactly satisfies a given set of angle measurements.

We have seen how a moving drone can find the angle of a sound source (Chapter 6), or how it can determine the angle of a wall from interference measurements (Chapter 5). Building on these results, we could imagine a drone swarm, in which each drone estimates the directions of the sound sources present in its vicinity, corresponding for instance to other buzzing drones. While doing so, the drone could detect reflections of its own sound off of walls and estimate their angle. Secondly, building on top of the sensor network localization in Section 8.4.1, we can imagine use cases in which each sensor measures only the angle to the neighboring devices. Since angle-of-arrival (AOA) has recently been added to the Bluetooth standards [215], this is a plausible use case, and particularly interesting since certain angle measurements degrade less with distance than range measurements. The theory presented in this chapter is useful for both applications, and potentially many more.

This chapter is based on the publication:

F. Dümbgen, M. El Helou, and A. Scholefield. "Realizability of Planar Point Embeddings from Angle Measurements". In: IEEE International Conference on Acoustics, Speech and Signal Processing. 2020, pp. 5710–5714

The code to reproduce the results of this chapter is available at github.com/duembgen/AngleRealizability.

9. Embedding based on inner angles

This chapter makes three main contributions.

- a) We find necessary conditions for inner angle measurements to be realizable, including linear and non-linear constraints which we exactly characterize.
- b) We observe which of the necessary conditions are sufficient in extensive simulations.
- c) By exploiting the above conditions, we construct an angle denoising and point recovery algorithm. Finally, we characterize in simulation the noise regimes in which inner angle measurements are preferred over distance measurements.

9.1. Problem setup

Recall that the non-directed inner angle measured at point x_i between two points x_j and x_k is denoted by $\theta_i(j, k) \in [0, \pi]$. As we only consider inner angles, we drop the “inner” attribute in what follows. We stack all angles in a vector denoted by $\theta \in \mathbb{R}^M$. In this work we do not address the labeling problem, so we assume we know the indices (i, j, k) for each element θ_m .

For the scope of this chapter, we rule out degenerate point sets in which any angle θ_m is exactly either zero or π . When randomly choosing points, such setups occur with probability zero. We define the notion of a point set’s *equivalency class* as follows: Two point sets belong to the same equivalency class if each node has the same circular clockwise sequence of edges in both point sets. The same notion was introduced in [51]. As we will see later, this helps in setting up linear constraints since the linear constraints matrix \mathbf{A} is the same for all point sets in the same class.

Then we seek to solve the following problem:

Problem 8. *Given a set of noisy angle measurements $\tilde{\theta}$, recover the point embedding \mathbf{X} that best matches them.*

In order to solve the above problem, we will exploit the notion of realizable angles, which we define as:

Definition 2. A inner-angle vector θ is realizable if and only if there exists a point embedding $\mathbf{x} \in \mathbb{R}^{N \times D}$ such that $\angle(\mathbf{x}_j - \mathbf{x}_i, \mathbf{x}_k - \mathbf{x}_i) = \theta_{m(i,j,k)}$ for all index triplets in \mathcal{M} .

We focus on the two-dimensional case, but we use the general notation D where applicable.

9.2. Related work

Localization from angles has been studied thoroughly. In vision-based systems, bearing-only localization is solved through triangulation [89]. In robot and sensor networks, distributed

methods [17, 30, 221], SLAM-based solutions [49, 125], and centralized approaches [22, 83] have been proposed. In the latter, a central processing unit collects data from connecting nodes and recovers their locations in a global manner. In distributed approaches, the nodes are self-localizing, and in most SLAM-based algorithms, the location of an active device and a (usually passive) map of landmarks are iteratively updated.

In this chapter, we focus on the centralized localization problem, meaning that we globally process all inner angles at once. We argue that as devices are increasingly connected, the central processing of measurements is feasible in many applications. On the other hand, we do not require knowledge of the node's orientations or any anchor nodes.

Since it is designed for one or multiple cameras observing a passive scene, the theory of triangulation from computer vision [89] can not be directly applied for the general node embedding problem studied in this chapter, where we wish to incorporate measurements from all involved nodes. The methods from bearing-only robot and sensor network localization, on the other hand, incorporate such measurements, but they fall short in using the available information optimally. For instance, many methods [96] use linearization of the bearing measurements of a moving device to incrementally refine a solution (see also Section 2.3), which is prone to local minima and requires appropriate initialization. In [225], a system of nonlinear equations in trigonometric functions of the angles is solved; here, the optimal solution can be found, but the cost function involved is sub-optimal. The localization method we propose, on the other hand, can first find the minimum least-squares estimate of angle measurements, and then recover the points, which are consistent with these measurements.

By phrasing our problem as an instance of the general embedding problem, we seek a solution for localizing points from angle measurements that uses the available information more optimally than the above approaches.

Our approach relies on realizability conditions, formalized in Definition 2. In the field of graph rigidity, people are also interested in “localizability conditions”, which determine whether the obtained solution is uniquely determined by its measurements and anchors [63, 99, 205]. The objects of study in these papers are the minimal set of measurements which uniquely define the shape of a configuration. Since the focus is on theoretical uniqueness, measurements are assumed noiseless, and although tested with noise in simulation, there is no guarantee that the developed algorithms are optimal given noise. Note also that the aforementioned methods rely on absolute angle measurements, and knowledge of anchor nodes.

Rather than characterizing the minimal set of angles and anchors for unique localization, we aim to find realizability conditions which can be used to denoise and localize overdetermined systems. Our work is close in nature to an early work on angle-based planar graph embedding [51]. The authors prove conditions for an angle set to correspond to a planar and convex planar graph with no crossing edges. We relax these assumptions to non-convex graphs, lying in 2 dimensions, and with arbitrary edge layouts. By doing so, we can construct an optimization problem on angles only, which characterizes the maximum likelihood estimator under Gaussian noise assumption, and ensures realizability.

9.3. Degrees of freedom of realizable angles

When we measure the relative dissimilarities of points using angles, some information is irrecoverably lost, as for distances and differences. As we have seen in Section 7.1.6, pairwise differences only retain $ND - D - D(D - 1)/2 = ND - D(D + 1)/2$ degrees of freedom of the original point set, due to invariance to orientation and translation. When measuring only angles, we additionally lose scale information, leading to one degree of freedom less:

$$\text{DOF} = ND - \frac{D(D + 1)}{2} - 1. \quad (9.1)$$

We can conclude that we need at least DOF angle measurements to accurately recover a point embedding, up to a rigid transformation and scale — the best that can be done from angle measurements.

9.4. Recovering points from angles

To recover point sets from the given angles θ , we apply the following standard build-up algorithm. Similar algorithms have been proposed for distances [3] or distances and angles [20]. We start by fixing the first two points (x_1, x_2) arbitrarily. The subsequent points $x_n, n = 3 \dots D + 1$ can be found using $n - 1$ angles from the previous points. Finally, all remaining points are uniquely defined using $D + 1$ angle measurements.

Note that the minimum number of measurements required in the described algorithm is

$$\sum_{i=2}^D i + (N - D + 1)(D + 1) = N(D + 1) - \frac{D(D + 1)}{2} - 1, \quad (9.2)$$

which is exactly N more than the lower bound DOF .

We stress that, given a full angle vector θ , this build-up algorithm only uses a subset of them for reconstruction. Unless the angles are realizable, the reconstruction accuracy thus depends on the choice of angles used for reconstruction, and there might be a discrepancy between the non-used and the reconstructed angles. In the next section, we provide constraints on the angles that must be satisfied for them to correspond to a valid point set. These constraints can be used to find the closest realizable angle vector. By doing this denoising before applying the build-up algorithm, every selection of angles results in the same reconstruction (up to scale and rigid transformation).

9.5. Characterization of realizable angles

The intuition of our realizability conditions is the following: Since an angle vector has M elements but only DOF degrees of freedom, $M - \text{DOF}$ of the measurements are redundant. In what follows, we construct a set of L linear constraints and K non-linear constraints on the angles, such that the $M - \text{DOF}$ residual degrees of freedom are correctly eliminated ($L + K = M - \text{DOF}$).

9.5.1. Linear constraints

We can impose two types of linear constraints on angles: the first type addresses angles measured at a single point, and the second type addresses angles in convex polygons. For all angles measured from one point, any subset of adjacent angles must sum up to form the bigger angle. In convex polygons of size $m \geq 3$, the angles must sum up to $(m - 2)\pi$. Numerous such constraints exist, but not all of them are linearly independent. We provide a method that constructs exactly the maximum number of linearly-independent constraints, defined by $L = L_{single} + L_{poly}$, where L_{single} denotes the number of linear constraints at a single node, and L_{poly} denotes the number of polygon constraints.

To create the linear constraints, in particular the *single* constraints, we impose $D = 2$. The analytical derivation becomes significantly more complex when $D = 3$ and it might be favorable to use the learned constraints instead, as described in Section 9.5.2.

For the *single* constraints at each node n , we define an order of the outgoing edges, $k = 1 \dots N - 1$, and the number of considered angles $\ell = 2 \dots N - 1$. Then, for each starting index $k = 1 \dots N - 1 - \ell$, we impose that

$$\begin{cases} \sum_{i=k}^{k+\ell} \theta_n(i, i+1) = \theta_n(k, k+\ell) & \text{if } \sum_{i=k}^{k+\ell} \theta_n(i, i+1) < \pi, \text{ or} \\ \sum_{i=k}^{k+\ell} \theta_n(i, i+1) + \theta_n(k, k+\ell) = 2\pi & \text{else.} \end{cases} \quad (9.3)$$

The order of angles can be determined through a combinatorial algorithm or from prior knowledge. For illustration purposes, we enumerate the *single* constraints for the first node ($n = 0$) of a point set of $N = 6$ points, assuming that the biggest angle $\theta_0(1, 5) < \pi$, in which case we are always in the first condition of (9.3):

$$\begin{aligned} \ell = 2 : & \quad k = 1 : \theta_0(1, 2) + \theta_0(2, 3) = \theta_0(1, 3) \\ & \quad k = 2 : \theta_0(2, 3) + \theta_0(3, 4) = \theta_0(2, 4) \\ & \quad k = 3 : \theta_0(3, 4) + \theta_0(4, 5) = \theta_0(3, 5) \\ \ell = 3 : & \quad k = 1 : \theta_0(1, 2) + \theta_0(2, 3) + \theta_0(3, 4) = \theta_0(1, 4) \\ & \quad k = 2 : \theta_0(2, 3) + \theta_0(3, 4) + \theta_0(4, 5) = \theta_0(2, 5) \\ \ell = 4 : & \quad k = 1 : \theta_0(1, 2) + \dots + \theta_0(4, 5) = \theta_0(1, 5) \end{aligned} \quad (9.4)$$

9. Embedding based on inner angles

This leads to the total number of constraints

$$L_{single} = N \sum_{i=1}^{N-3} i = \frac{N(N-2)(N-3)}{2}. \quad (9.5)$$

Moving to the *polygon* constraints, one can find all constraints linearly independent from the *single* constraints by choosing one single corner and imposing the constraint in all triangles involving that corner. All higher-order polygons lead to constraints that are linearly dependent of the former. This leads to:

$$L_{poly} = \binom{N-1}{2} = \frac{(N-1)(N-2)}{2}, \quad (9.6)$$

and the total number of constraints:

$$L = L_{single} + L_{poly} = \frac{1}{2} (N^3 - 4N^2 + 3N + 2). \quad (9.7)$$

We represent the obtained linear constraints in matrix form $\mathbf{A}\boldsymbol{\theta} = \mathbf{b}$, with $\mathbf{A} \in \mathbb{R}^{L \times M}$, $\mathbf{b} \in \mathbb{R}^L$. In Appendix 9.A, we prove why it is enough to consider all triangles involving one corner only.

Note that in above considerations, we have arbitrarily chosen to first impose the *single* and then the *polygon* constraints. It would have been possible to proceed the other way around, in which case we would have a higher proportion of *polygon* constraints. We visualize example constraints matrices obtained using either first the *single* or first the *polygon* constraints, in Figure 9.1.

9.5.2. Automatic creation of linear constraints

Constructing the constraints matrix \mathbf{A} , with the previously proposed combinatorial algorithm, can be prohibitively time consuming. If we know the equivalency class of our point set, we can greatly speed up the process by learning the constraints matrix from simulations. If the equivalency class is not known a priori, it can be determined through a rough initial point estimate using the noisy or partially denoised angles.

We learn the constraints matrix by randomly generating at least $T \geq L$ point sets in the correct equivalency class, and reading off the angles $\boldsymbol{\theta}_t$, $t = 1 \dots T$. Since all points sets have the same order of edges per node, they share the same linear conditions and $\mathbf{A}\boldsymbol{\theta}_t = \mathbf{b}$, must be satisfied for each angle vector. Therefore we can write $\boldsymbol{\Theta}\mathbf{C}^\top = \mathbf{0}$, with

$$\boldsymbol{\Theta} = \left[\boldsymbol{\theta}_t^\top \quad -1 \right]_{t=1}^T \in \mathbb{R}^{C \times (M+1)}, \text{ and } \mathbf{C} = \left[\mathbf{A} \quad \mathbf{b} \right] \in \mathbb{R}^{L \times (M+1)} \quad (9.8)$$

In order to determine \mathbf{C}^\top , we can find a basis of the null space of the matrix $\boldsymbol{\Theta}$. A valid basis can be found by extracting the L last columns of \mathbf{V} from the SVD given by $\mathbf{C}^\top = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. As opposed to the analytically obtained matrix \mathbf{A} , the learned matrix is not sparse. Since each

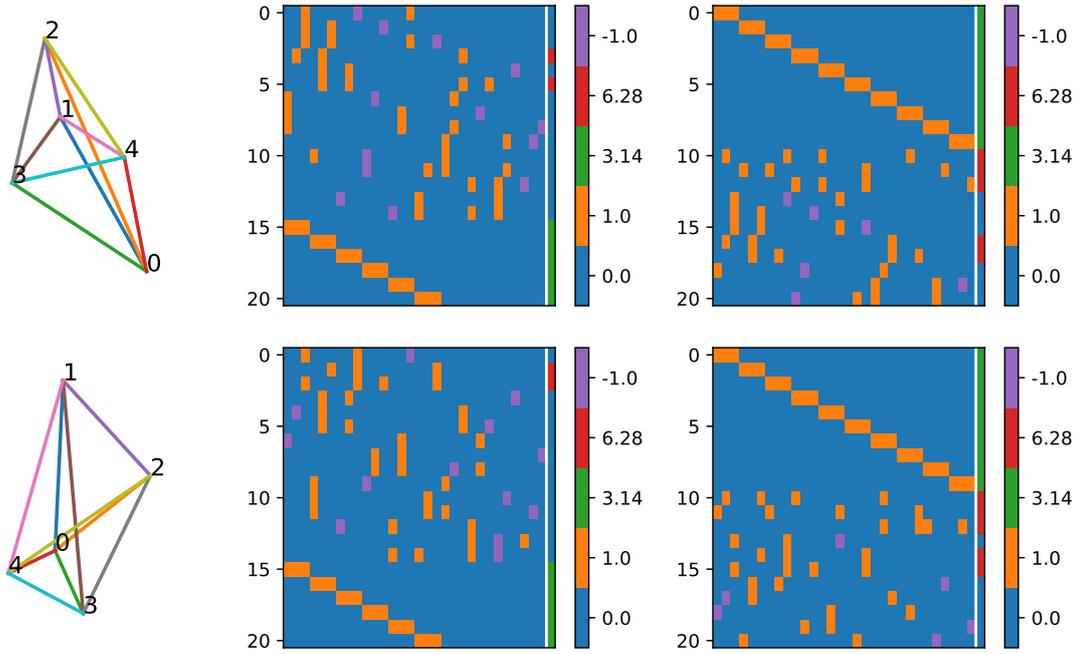


Figure 9.1.: Example setups (left) and corresponding full-rank constraints matrices obtained using first the single constraints (middle) or first the polygon constraints (right). Shown attached to the matrices A are the vector b , separated by a white line.

constraint acts on all angles, this leads to a smoother denoising behavior when incrementally adding constraints, as confirmed in Section 9.7.1.

9.5.3. Non-linear constraints

The linear constraints are not enough to eliminate the residual degrees of freedom. To see this, we compare the number of linear constraints with the residual degrees of freedom. Assuming that there is a number K of unconstrained degrees of freedom, we pose:

$$\begin{aligned}
 K &= M - \text{DOF} - L \\
 &= \frac{N(N-1)(N-2)}{2} - ND + \frac{D(D+1)}{2} + 1 - \frac{1}{2}(N^3 - 4N^2 + 3N + 2) \quad (9.9) \\
 &= \frac{N(N-1)}{2} - ND + \frac{D(D+1)}{2}
 \end{aligned}$$

For $D = 2$, K simplifies to:

$$K = \frac{(N-3)(N-2)}{2} \quad (9.10)$$

To fill the gap K , we need to resort to non-linear constraints. We can for instance impose the sine law (or equivalently, the cosine law) in each triangle. In order to remove the distance parameters, we process two adjacent triangles together, which allows to eliminate the

9. Embedding based on inner angles

distances and obtain equations of the form

$$f_k(\boldsymbol{\theta}) = \frac{\sin \theta_b(a, c) \sin \theta_c(a, d) \sin \theta_d(a, b)}{\sin \theta_b(a, d) \sin \theta_c(a, b) \sin \theta_d(a, c)} - 1 = 0, \quad (9.11)$$

where a to d denote any four corners of the k th quadrilateral. The remaining question is which constraints of this form are independent, and if we can find exactly K of them.

Considering $D = 2$, we note that $K = \binom{N-2}{2}$. Choosing two points as “anchors”, for instance points $n = 0$ and $n = 1$, we can create exactly K nonlinear constraints by imposing one constraint of the form (9.11) for each quadrilateral involving these two points. It remains to be proven that this choice of constraints is in fact the “optimal” one; but as we see in the following, there is some evidence that it is, since it successfully eliminates the remaining degrees of freedom.

In $D = 3$ dimensions, we do not have an expression for L_{single} , and we cannot derive the form of K . However, the above argument holds in two or three dimensions, so we conjecture that the same nonlinear constraints can be added to constrain, at least partially, the remaining degrees of freedom.

Table 9.1.: Comparison of number of constraints compared to the number of points N , angles M and degrees of freedom DOF

N	M	DOF	necessary	linear	nonlinear
4	12	4	8	7	1
5	30	6	24	21	3
6	60	8	52	46	6
7	105	10	95	85	10
8	168	12	156	141	15
9	252	14	238	217	21
10	360	16	344	316	28
11	495	18	477	441	36
12	660	20	640	595	45
13	858	22	836	781	55
14	1092	24	1068	1002	66

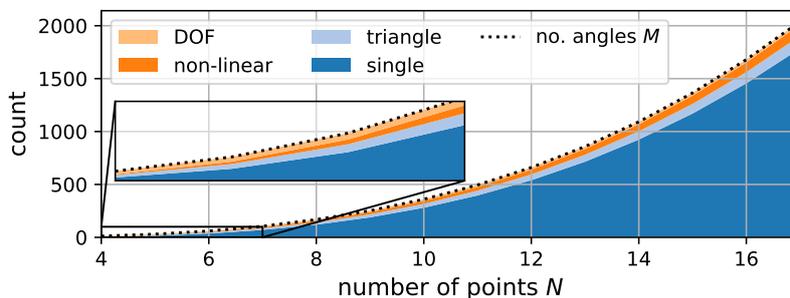


Figure 9.2.: Visualization of the number of angles M as the sum of the degree of freedom (DOF) and the number of different constraint types, for increasing numbers of points N .

9.5.4. Guarantees

Summing the number of linear and non-linear constraints found with the DOF of realizable angles gives the number of angles M , as desired:

$$M = \text{DOF} + L_{\text{single}} + L_{\text{triangle}} + K. \quad (9.12)$$

Figure 9.2 visualizes the different components of this equation. In Table 9.1, we also provide the numerical examples for point number up to $N = 10$. We can see that the majority of measurements can be constrained through linear constraints, so one might argue that the non-linear constraints are negligible. As we show in Section 9.7, these constraints are however crucial for angle realizability, and for good reconstruction accuracy.

We conclude by establishing necessary and sufficient guarantees for angle realizability. The proposed constraints are induced from well-known geometry laws, thus the following proposition holds:

Proposition 3. If θ is realizable, then both linear and non-linear constraints are satisfied: $A\theta = b$ and $f_i(\theta) = 0$ for $i = 1 \dots K$.

We have seen that by combining the proposed constraints, we get exactly $L + K = M - \text{DOF}$ constraints. We thus constrain all redundant measurements, and we can put forward the following conjecture.

Conjecture 1. If both linear and non-linear constraints are satisfied, then the angle set θ is realizable.

We provide extensive simulation results that empirically verify this conjecture in Section 9.7.1. We also provide the sketch of a proof in Appendix 9.B. If the conjecture is true, it follows that the constraints provided here are sufficient and necessary, as stated in the following result.

Conjecture 2. An angle set θ is realizable if and only if both linear and non-linear constraints are satisfied.

9.6. Denoising and recovery algorithm

Possible applications of Conjecture 2 include outlier rejection (removing spurious measurements which do not satisfy the conditions), labeling (finding an order of angles such that the constraints are satisfied) and completion (filling in missing angle measurements such that they satisfy the constraints). For the purpose of this chapter, we propose a denoising and point recovery algorithm.

Given a noisy angle vector $\tilde{\theta}$, we want to find a point embedding with angles as close to the measurements as possible. Thanks to Conjecture 2 we can solve this in two steps: first, we

9. Embedding based on inner angles

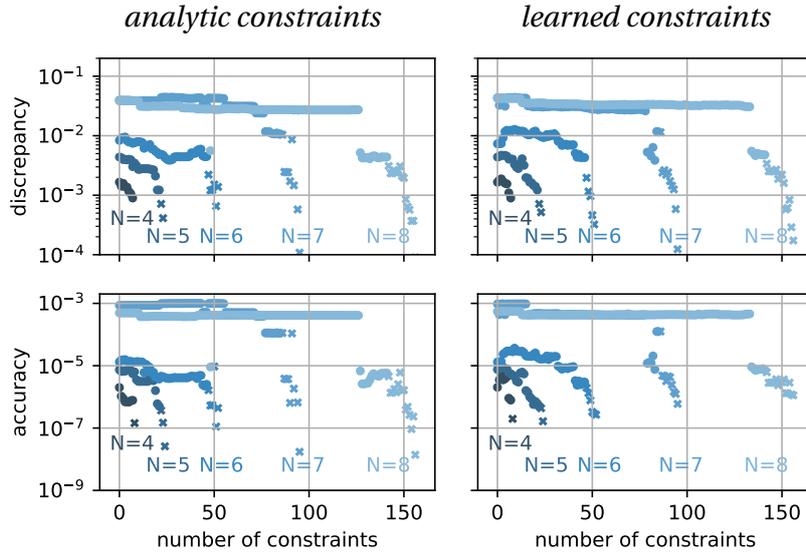


Figure 9.3.: First row: discrepancy between reconstructed and denoised angles as a function of number and type of imposed constraints, for different numbers of points N , drawn uniformly from a unit square. Second row: reconstruction accuracy for the same setups.

find the maximum likelihood estimate (under zero-mean Gaussian noise assumption) of a realizable angle set by solving

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^M} \left\| \tilde{\theta} - \theta \right\|, \quad (9.13)$$

such that $A\theta = \mathbf{b}$, $f_k(\theta) = 0$ for $k = 1 \dots K$.

The non-linear constraints render this problem non-convex, and no existing algorithm is guaranteed to find its optimum. As we see in Section 9.7, standard solvers do however show good performance and convergence properties.

Knowing that the resulting angle set $\hat{\theta}$ is realizable, we use the simple build-up algorithm as described in Section 9.4 for point recovery. If knowledge of some anchor points is provided, we align and scale the obtained point set to the anchor points using the orthogonal Procrustes transform [186].

9.7. Simulation results

In this section, we validate the theory established in Section 9.5 through extensive numerical simulations and evaluate the algorithm's performance. We generate random point sets by uniformly picking points from a square of given size. Simulated noise is always assumed zero-mean Gaussian. To avoid numerical issues we ignore point sets that have at least one angle smaller than 1×10^{-3} rad in absolute value. We use Sequential Least Squares

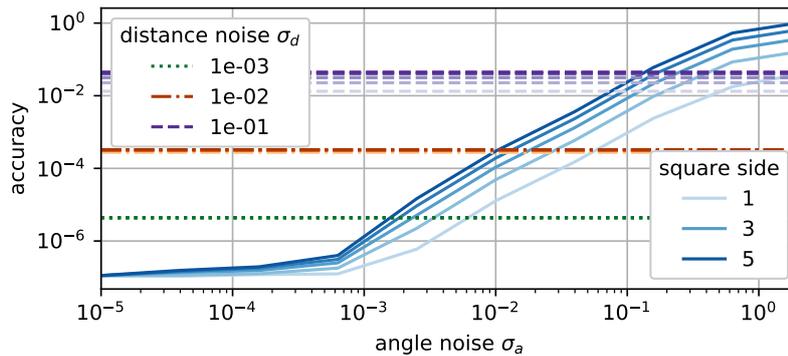


Figure 9.4.: Point recovery accuracy using either distances (dashed lines) or angles (solid lines). The shading corresponds to different sizes of the experiment setup.

Programming (SLSQP) to solve (9.13), provided in the `scipy` package.¹

9.7.1. Realizability study

To study the validity of Conjecture 2, we design the following experiment: Given a measured noisy angle set, we denoise the angles using (9.13) and recover the point set using the build-up algorithm from Section 9.4. From the obtained point set, we read off the angles. By construction, the angles used for reconstruction equal the denoised angles, but the unused angles will only equal the denoised angles if the constraints in Conjecture 2 are sufficient.

We report the results of this experiment in the top row of Figure 9.3. We define the *discrepancy* error as the mean absolute error (MAE) between the denoised and reconstructed angles, and calculate this error for 20 random realizations of each parameter set. We confirm that the discrepancy only approaches zero when all constraints are added. The learned constraints (right column) lead to a more gradual decrease than analytical constraints, because they are not sparse. Finally, we note that the difference in errors can be explained by the convergence to different local minima of (9.13). We conclude that the combination of the proposed constraints leads to realizable angles.

9.7.2. Performance evaluation

We also evaluate the performance of the proposed recovery algorithm in terms of point reconstruction *accuracy*, and report results in the second row of Figure 9.3. To this end, we fit the obtained point set with orthogonal Procrustes [186] to the true locations (including rescaling), and calculate the MSE. We observe that the reconstruction accuracy increases with the number of constraints. Indeed, the plot in the bottom of Figure 9.3, shows a clear correlation between the number of added constraints and reconstruction accuracy.

Finally, we provide insight into when angle measurements are preferable over distances. For

¹The code to reproduce the results will be provided upon publication.

9. Embedding based on inner angles

the distance-based reconstruction, we implement the standard MDS algorithm [53]. We assume a fixed noise level on distances, denoted by σ_d , and plot the accuracy for a range of angle noise σ_a . As angle-based recovery is sensitive to the scale of the setup, we show results for different area sizes. Figure 9.4 confirms that accuracy decreases with area size and noise, but that for the lower regimes of angle noise, angle-based localization is more accurate than distance-based localization.

9.8. Conclusion and ongoing work

We have proposed necessary and sufficient constraints for the realizability of inner angle measurements. The constraints are related to well-known trigonometric properties and we have shown how to construct them both analytically and numerically, given minimal prior knowledge of the point set's topology. While all constraints are necessary, we conjecture which of the constraints are sufficient, and provide numerical evidence as well as a sketch of a proof. We have exploited the conditions for a denoising and recovery algorithm and studied their performance on a variety of simulated problem setups. Finally, we have identified regimes in which angle-based recovery leads to more accurate localization than distance-based recovery.

The required prior knowledge of an equivalency class can be constraining in certain situations, therefore future work could focus on establishing end-to-end algorithms which exploit different levels of constraints in a hierarchical fashion to recover the equivalency class along the way. For instance, using only a set of linear constraints might be enough for cheaply finding the topology of the point set, before imposing the more expensive non-linear constraints. Another natural extension of the present work is the application of the realizability conditions for other methods such as outlier rejection, completion and labeling.

Appendix

9.A. Minimal set of linear constraints

The following proposition motivates our choice of imposed linear constraints.

Proposition 4. If we impose the *polygon* constraints in all triangles involving one (arbitrarily) chosen point, and impose all *single* constraints, then the *polygon* constraints for all triangles of the point set are satisfied.

Proof. Let us assume that we impose the linear constraints for all triangles involving n . Then, we will show that for any triangle (i, j, k) , where neither of the corners equals n , the inner

angle sum has to equal π . Indeed, we can rewrite the inner angle sum of triangle (i, j, k) as

$$\begin{aligned}
\theta_i(k, j) + \theta_j(k, i) + \theta_k(i, j) &\stackrel{(a)}{=} \theta_i(k, n) - \theta_i(n, j) + \theta_j(k, n) - \theta_j(n, i) + \theta_k(i, n) - \theta_k(n, j) \\
&\stackrel{(b)}{=} \pi - \theta_n(i, k) + \pi - \theta_n(j, k) - \theta_i(n, j) - \theta_j(n, i) \\
&\stackrel{(c)}{=} 2\pi - \theta_n(i, j) - \theta_i(n, j) - \theta_j(n, i) \stackrel{(d)}{=} 2\pi - \pi = \pi,
\end{aligned} \tag{9.14}$$

where we have used the *single* constraints of points i, j and k in (a), the *polygon* constraints in (b), the *single* constraint of n in (c) and finally the *polygon* constraint of (n, i, j) in (d). \square

Thanks to this proposition, we can assume that the *polygon* constraint is imposed for every triangle in the point set, even though only a subset is explicitly covered with $A\theta = \mathbf{b}$.

9.B. Sketch of proof

In this section, we outline the sketch of a formal proof of Conjectures 1 and 2.

We want to show that if $A\theta = \mathbf{b}$ and $f_k(\theta) = 0 \forall k = 1 \dots K$ then there exists a point set \mathbf{X} of which the inner angles match θ . To prove this, we will use a subset of angles to construct a valid point set, and then show that all angles of this point set, in particular the non-used ones, match the vector θ . We denote the inner angles in the reconstructed point set by $\hat{\theta}$, so that we want to show $\theta = \hat{\theta}$.

Note that we impose the triangle constraints on all triangles involving point x_0 , and we choose points x_0, x_1 as the basis for nonlinear constraints. Since the numbering is arbitrary, these choices do not matter.

Starting with $N = 3$, we construct x_2 by intersecting the angles $\theta_0(1, 2)$ and $\theta_1(0, 2)$. It is trivial to show that the non-used vector $\hat{\theta}_1(0, 2)$ equals $\theta_1(0, 2)$ because of the triangle constraint.

Now assume we want to construct point $n, n > 3$. We use again point x_0 and x_1 and the angles $\theta_0(1, n)$ and $\theta_1(0, n)$ to construct the point x_n . There are two choices, but only one is consistent with the points constructed thus far. We can pick the correct one by choosing, for instance, the one satisfying the *single* constraints at point x_0 .

Then, we show that for all points k with $k = 1 \dots n - 1$, the angles $\hat{\theta}_k(0, n)$ and $\hat{\theta}_k(1, n)$ are, although not used for the reconstruction, equal to $\theta_k(0, n)$ and $\theta_k(1, n)$, respectively. We define the point $x_{n'}$ as the point of intersection between the line formed by $\hat{\theta}_k(1, n)$ and $\theta_0(1, n)$, as shown in Figure 9.5 (a). We can show that since the nonlinear constraint holds for any quadrilateral involving $x_0, x_1, x_n = x_{n'}$ or, in other words, there is a unique point of intersection defined by the three angles $\theta_0(1, n), \theta_1(0, n)$ and $\theta_k(1, n)$.

9. Embedding based on inner angles

We first impose the sine laws in the relevant triangles:

$$\begin{aligned}
 \frac{l_{1n}}{\sin \theta_0(1, n)} &= \frac{l_{01}}{\sin \theta_n(0, 1)} \quad (\text{sine law in triangle } (0, 1, n)) \\
 \frac{l_{1n'}}{\sin \hat{\theta}_k(1, n')} &= \frac{l_{1k}}{\sin \hat{\theta}_{n'}(1, k)} \quad (\text{sine law in triangle } (1, k, n')) \\
 \frac{l_{01}}{\sin \theta_k(0, 1)} &= \frac{l_{1k}}{\sin \theta_0(1, k)} \quad (\text{sine law in triangle } (0, 1, k))
 \end{aligned} \tag{9.15}$$

We can reformulate these equations to get

$$\frac{l_{1n}}{l_{1n'}} = \frac{\sin \theta_0(1, n) \sin \hat{\theta}_{n'}(1, k) \sin \theta_k(0, 1)}{\sin \theta_n(0, 1) \sin \hat{\theta}_k(1, n) \sin \theta_0(1, k)} = f_k(\boldsymbol{\theta}) + 1 = 1, \tag{9.16}$$

where we denote by f_k the linear constraint imposed for the choice of points x_k, x_n . We conclude that since the non-linear constraint holds, then $l_{1n} = l_{1n'}$ and thus $x_n = x_{n'}$. This confirms that $\hat{\theta}_k(n, 1) = \theta_k(n, 1)$ and $\hat{\theta}_n(k, 0) = \theta_n(k, 0)$.

The last step is to show that all other reconstructed angles also match the (non-used) initial angles. This can be easily seen through the linear constraints: We can use the *single* constraints for points 0, 1, k and n, respectively, to show that $\hat{\theta}_0(k, n) = \theta_0(k, n)$, $\hat{\theta}_1(k, n) = \theta_1(k, n)$, $\hat{\theta}_k(0, n) = \theta_k(0, n)$, and $\hat{\theta}_n(0, k) = \theta_n(0, k)$. This is visualized in Figure 9.5 (b).

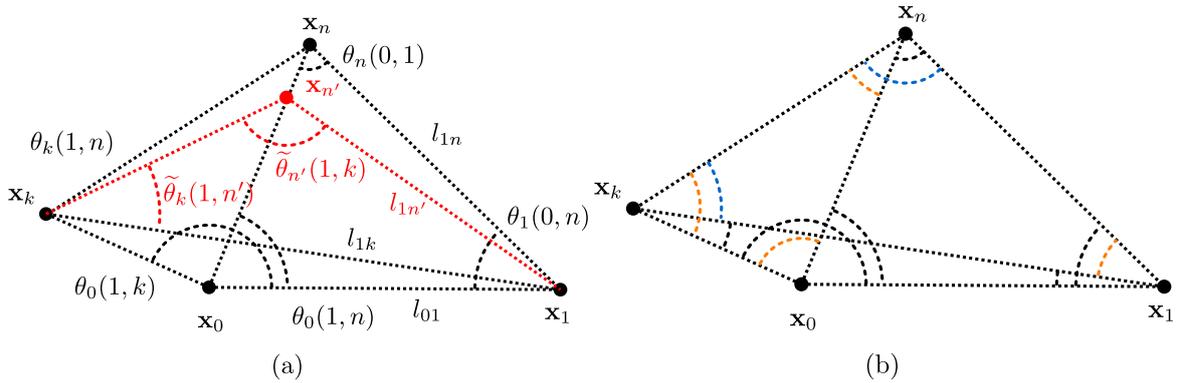


Figure 9.5.: Visualization of proof sketch: in (a), we sketch the setup studied to determine that thanks to the non-linear constraint, there is a unique point of intersection ($x_n = x_{n'}$). In (b), we show in black the consistent angles, in blue the angles found to be consistent in the previous step, and in orange the remaining angles. Clearly, the remaining angles are uniquely determined by the linear constraints.

10 | Conclusion and future work

We have covered different aspects of spatial perception without sight, ranging from localization based on radio-frequency signals, to wall localization from sound and sound-source localization. Since these signals typically allow for inferring distances or angles from the corresponding access points, sources or walls, we have devoted the second part of the thesis to providing some missing puzzle pieces regarding the theory of localization from different combinations of distance and angle measurements.

The nature of contributions of the thesis includes entire systems, such as the indoor localization solutions now deployed in our partner's software, and the custom audio deck including software stack for performing audio experiments on the *Crazyflie* drone. In other parts, we provide novel algorithms for localization which do not rely on classical linear and Gaussian assumptions, both of which tend to be inaccurate when dealing with distances or angle measurements. In particular, we provide a closed-form solution for range-only trajectory recovery, a problem which is typically solved using sub-optimal iterative solvers. Furthermore, we provide algorithms which exploit distance or angle measurements to yield geometrically consistent point estimates, which we successfully apply for joint localization problems. Exploiting audio signals, we provide algorithms to perform wall and sound-source localization which are tailored to moving and low-cost devices, and showing novel ways of using motion for enhanced sensing.

We have pointed to some ongoing research directions in the “Conclusion and ongoing research” sections throughout this thesis. In this final chapter, we would like to seize the opportunity to discuss more ambitious and general research directions which came up during the PhD work and while writing this thesis. We also use this space to discuss directions that combine findings from a variety of chapters of the thesis.

Continuous localization on a drone using sound In Chapter 4, we have discussed how to localize a moving object from non-synchronized distance measurements to fixed anchors. In Chapter 5, we have discussed how a drone can measure the distance and angle of the closest wall. The natural follow-up question is: can we perform continuous trajectory recovery and wall mapping, using sound only, on a drone?

The reason continuous recovery is interesting is because the entities we seek to localize are smooth: as the drone moves in the vicinity of the wall, we know that all wall estimates should

lie on a line (or plane, when localizing in three dimensions). We also know that the trajectory of the drone, because of physical constraints, is (at least piecewise) smooth. Therefore, jointly localizing the wall and the drone's trajectory could be phrased as a parametrized localization problem, or using a more flexible representation as discussed in Chapter 4.

There are plenty of reasons why using sound for this endeavour is interesting. Besides the already mentioned advantages of sound in terms of ubiquity and omnidirectionality, sound can provide a more minimal parametrization of the wall. A wall's location can be characterized with very few parameters, but in classical visual SLAM, we instead recover either the locations of a set of features on the wall, or a full three-dimensional representation such as a mesh. Of course, we can perform dimensionality reduction in hindsight, but it seems unnecessarily costly to recover all this unnecessary detail along the way. Using audio SLAM, we reduce ourselves, from the beginning, to only the degrees of freedom of interest of the wall: its angle and distance from the drone, and we can therefore perform localization more cheaply, and potentially, more accurately.

Embedding higher-dimensional entities Although of practical relevance, localization is arguably a less interesting instance of the general embedding problem, as it does not uncover any hidden structure of the data; its goal is merely to find point locations as close to their true locations as possible. When applied to higher-dimensional entities of all sorts, low-dimensional embedding can be used to find hidden connections in the studied entities, clusters, and correlations, that are hard to see in the original space. We have discussed a few interesting problems such as clustering biomedical images and text; but even more esoteric examples such as the geometry of music can be studied [50]. When extending our horizon to higher-dimensional data, there is a never-ending range of interesting applications of the theory developed in the second part of the thesis, which we have not explored to this date. In particular, angle measurements are an interesting alternative to distance measurements, as seen by the wide adaptation of the cosine distance for word embedding, for instance. However, to the best of our knowledge, existing approaches put little emphasis on finding an optimal embedding — a rough embedding is good enough for applications such as data visualization and finding higher-level. An interesting direction of research is thus to explore to what extent the developed theory in Section 7.3 and Chapter 9 yields embeddings more faithful to the original data, thus allowing to uncover more subtle and complex relationships between the entities.

Distance and angle geometry for semantic SLAM With great advances in scene understanding thanks to machine learning, more and more SLAM algorithms seek to incorporate semantic information, *i.e.* to reconstruct a map consisting of a few labeled objects in the environment, rather than a collection of abstract features or a dense representation [181]. Such a treatment has at least two advantages: first, semantic information is an enabling factor for higher-level tasks and decision making. A robot that detects doors as doors rather than flat surfaces can open them, for instance, and an autonomous car that distinguishes people from trees can take life-saving decisions when it comes to a crash. Another advantage of the semantic SLAM formulation, which is more important for the

current work, is that using semantics significantly reduces the problem size. Rather than localizing thousands of features or three-dimensional points, the mapping and localization problem can be reduced to a few objects of interest. Advances have been made to determine the pose of such objects optimally, given their three-dimensional scans and a dictionary of labeled models [188]. This decreased problem size allows for more efficient labeling algorithms [27], novel visual SLAM algorithms [149] and it allows to revisit simplifying assumptions such as staticity of landmarks [93].

We postulate that representing our world using a few labeled objects rather than high-dimensional maps also allows for global localization algorithms, such as the ones studied in this thesis, to become widely applicable. Indeed, the algorithms studied in Chapter 4 and the second part of this thesis perform best on a handful of data points rather than thousands of data points typical in visual or lidar-based SLAM. As semantics become more readily available, it would be very exciting to stress-test these methods on such use cases.

To summarize, we believe that continuing to create tighter connections between signal processing and robotics can bring novel algorithms, improved performance and a better understanding to many open problems in spatial perception.

Bibliography

- [1] B. P. Abbott *et al.* “GW151226 : Observation of Gravitational Waves from a 22-Solar-Mass Binary Black Hole Coalescence”. In: *Physical Review Letters* 116.241103 (2016), pp. 1–14.
- [2] T. Ajdler, L. Sbaiz, and M. Vetterli. “The Plenacoustic Function and Its Sampling”. In: *IEEE Transactions on Signal Processing* 54.10 (2006), pp. 3790–3804.
- [3] J. Alencar, C. Lavor, and L. Liberti. “Realizing Euclidean distance matrices by sphere intersection”. In: *Discrete Applied Mathematics* 256 (2019), pp. 5–10.
- [4] J. B. Allen and D. A. Berkley. “Image method for efficiently simulating small-room acoustics Image method for efficiently simulating small-room acoustics”. In: *The Journal of the Acoustical Society of America* April (1979).
- [5] I. Alyafawi, S. Kiener, and T. Braun. “Hybrid indoor localization using multiple Radio Interfaces”. In: *17th International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, 2016, pp. 1–7.
- [6] S. R. Arnott, L. Thaler, J. L. Milne, D. Kish, and M. A. Goodale. “Shape-specific activation of occipital cortex in an early blind echolocation expert”. In: *Neuropsychologia* 51.5 (2013), pp. 938–949.
- [7] G. Baechler, F. Dümbgen, G. Elhami, M. Krekovic, R. Scheibler, A. Scholefield, and M. Vetterli. “Combining range and direction for improved localization”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2018, pp. 3484–3488.
- [8] G. Baechler, F. Dümbgen, G. Elhami, M. Krekovic, and M. Vetterli. “Coordinate Difference Matrices”. In: *SIAM Journal on Matrix Analysis and Applications* 41.1 (2020), pp. 332–363.
- [9] G. Baechler, M. Kreković, J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli. “Super Resolution Phase Retrieval for Sparse Signals”. In: *IEEE Transactions on Signal Processing* 67.18 (2019), pp. 4839–4854.
- [10] G. Baechler, A. Latty, M. Pacholska, M. Vetterli, and A. Scholefield. “Shedding light on 19th century spectra by analyzing Lippmann photography”. In: *PNAS* 118.17 (2021), pp. 1–7.

- [11] A. Bahillo, S. Mazuelas, J. Prieto, P. Fernández, R. M. Lorenzo, and E. J. Abril. “Hybrid RSS-RTT localization scheme for wireless networks”. In: *International Conference on Indoor Positioning and Indoor Navigation*. 2010, pp. 1–7.
- [12] T. Baltrusaitis, P. Robinson, and L.-P. Morency. “Continuous Conditional Neural Fields for Structured Regression”. In: *European Conference on Computer Vision*. 2014, pp. 593–608.
- [13] R. Bamler and P. Hartl. “Synthetic aperture radar interferometry”. In: *Inverse Problems* 14 (1998), R1–R54.
- [14] T. D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.
- [15] T. D. Barfoot, C. H. Tong, and S. Särkkä. “Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression”. In: *Robotics: Science and Systems*. 2014.
- [16] P. Barooah and J. P. Hespanha. “Estimation on graphs from relative measurements”. In: *IEEE Control Systems Magazine* 27.4 (2007), pp. 57–74.
- [17] A. Basu, J. S. Mitchell, J. Gao, and G. Sabhnani. “Distributed localization using noisy distance and angle information”. In: *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. 2006, pp. 262–273.
- [18] A. Beck, P. Stoica, and J. Li. “Exact and approximate solutions of source localization problems”. In: *IEEE Transactions on Signal Processing* 56.5 (2008), pp. 1770–1778.
- [19] S. M. Beck, J. R. Buck, W. F. Buell, R. P. Dickinson, D. A. Kozlowski, N. J. Marechal, and T. J. Wright. “Synthetic-aperture imaging laser radar: laboratory demonstration and signal processing”. In: *Applied Optics* 44.35 (2005), pp. 7621–7629.
- [20] S. J. L. Billinge, P. M. Duxbury, D. S. Gonçalves, C. Lavor, and A. Mucherino. “Recent results on assigned and unassigned distance geometry with applications to protein molecules and nanostructures”. In: *Annals of Operations Research* 271 (2018), pp. 161–203.
- [21] P. Biswas. “Semidefinite programming approaches to distance geometry problems”. PhD thesis. Stanford University, 2007.
- [22] P. Biswas, H. Aghajan, and Y. Yinyu. “Semidefinite programming algorithms for sensor network localization using angle information”. In: *Conference Record - Asilomar Conference on Signals, Systems and Computers 2005* (2005), pp. 220–224.
- [23] J. L. Blanco, J. A. Fernández-Madrigal, and J. González. “Efficient probabilistic range-only SLAM”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 1017–1022.
- [24] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. “CodeSLAM — Learning a Compact, Optimisable Representation for Dense Visual SLAM”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2560–2568.
- [25] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nuetzi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart. *A Primer on the Differential Calculus of 3D Orientations*. Tech. rep. ETH Zurich, 2016.

- [26] M. Boutin and G. Kemper. “A drone can hear the shape of a room”. In: *SIAM Journal on Applied Algebra and Geometry* 4.1 (2020), pp. 123–140.
- [27] S. L. Bowman, N. Atansov, K. Daniilidis, and G. J. Pappas. “Probabilistic Data Association for Semantic SLAM”. In: *IEEE International Conference on Robotics and Automation*. 2017, pp. 1722–1729.
- [28] G. L. Bretthorst. “Bayesian Spectrum Analysis and Parameter Estimation”. In: *Lecture Notes in Statistics*. Ed. by J. Berger, S. Fienberg, J. Gani, K. Krickeberg, and B. Singer. Springer-Verlag Berlin Heidelberg, 1988.
- [29] L. Brown and R. H. March. “A Radar History of World War II: Technical and Military Imperatives”. In: *Physics Today* 53.10 (2000), pp. 82–83.
- [30] J. Bruck, J. Gao, and A. A. Jiang. “Localization and routing in sensor networks by local angle information”. In: *ACM Transactions on Sensor Networks* 5.1 (2009), pp. 1–31.
- [31] S. L. Brunton and M. B. Si. “Modern Koopman Theory for Dynamical Systems”. In: *arXiv preprint arXiv:2102.12086v1* (2021).
- [32] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi. “Collaborative Sensor Network Localization : Algorithms and Practical Issues”. In: *Proceedings of the IEEE* 106.6 (2018), pp. 1089–1114.
- [33] F. Buekenhout and M. Parker. “The number of nets of the regular convex polytopes in dimension ≤ 4 ”. In: *Discrete Mathematics* 186.1-3 (1998), pp. 69–94.
- [34] K. Burnett, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot. “Radar Odometry Combining Probabilistic Estimation and Unsupervised Feature Learning”. In: *Robotics: Science and Systems*. 2021, pp. 1–10.
- [35] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age”. In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332.
- [36] L. Calkins, J. Lingeitch, L. Mcguire, J. Geder, M. Kelly, M. M. Zavlanos, D. Sofge, and D. M. Lofaro. “Bio-Inspired Distance Estimation using the Self-Induced Acoustic Signature of a Motor-Propeller System”. In: *IEEE International Conference on Robotics and Automation*. 2020, pp. 5047–5053.
- [37] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. *BRIEF: Binary robust independent elementary features*. Springer Berlin Heidelberg, 2010, pp. 778–792.
- [38] A. Canepa, Z. Talebpour, and A. Martinoli. “Automatic Calibration of Ultra Wide Band Tracking Systems Using A Mobile Robot : A Person Localization Case-study”. In: *International Conference on Indoor Positioning and Indoor Navigation*. 2017, pp. 1–8.
- [39] L. Carlone and A. Censi. “From Angular Manifolds to the Integer Lattice : Guaranteed Orientation Estimation With Application to Pose Graph Optimization”. In: *IEEE Transactions on Robotics* 30.2 (2014), pp. 475–492.
- [40] L. Carlone and F. Dellaert. “Duality-based verification techniques for 2D SLAM”. In: *IEEE International Conference on Robotics and Automation*. IEEE, 2015, pp. 4589–4596.

- [41] E. Ceolini, I. Kiselev, and S.-C. Liu. “Audio classification systems using deep neural networks and an event-driven auditory sensor”. In: *IEEE Sensors*. 2019.
- [42] M. Y. Cheung, D. Fourie, N. R. Rypkema, P. V. Teixeira, H. Schmidt, and J. Leonard. “Non-Gaussian SLAM utilizing Synthetic Aperture Sonar”. In: *IEEE International Conference on Robotics and Automation*. 2019, pp. 3457–3463.
- [43] K. S. Chong and L. Kleeman. “Feature-based Mapping in Real, Large Scale Environments Using an Ultrasonic Array”. In: *International Journal of Robotics Research* 18.1 (1999), pp. 3–18.
- [44] J. H. Christensen, S. Hornauer, and S. Yu. “BatVision: Learning to See 3D Spatial Layout with Two Ears”. In: *IEEE International Conference on Robotics and Automation* (2020).
- [45] Compulab. *WILD Minimal - WiFi RTT proof-of-concept for Android 9 Pie*.
- [46] J. A. Costa, N. Patwari, and A. O. Hero. “Distributed weighted-multidimensional scaling for node localization in sensor networks”. In: *ACM Transactions on Sensor Networks* 2.1 (2006), pp. 39–64.
- [47] J. Dattorro. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing USA, 2015.
- [48] G. T. F. De Abreu and G. Destino. “Super MDS: Source location from distance and angle information”. In: *IEEE Wireless Communications and Networking Conference* (2007), pp. 4433–4437.
- [49] M. C. Deans. “Bearings-Only Localization and Mapping”. PhD thesis. Carnegie Mellon University, 2005.
- [50] E. D. Demaine, F. Gomez-martin, H. Meijer, D. Rappaport, P. Taslakian, G. T. Toussaint, T. Winograd, and D. R. Wood. “Computational Geometry : Theory and Applications The distance geometry of music”. In: *Computational Geometry: Theory and Applications* 42.5 (2009), pp. 429–454.
- [51] G. Di Battista and L. Vismara. “Angles of planar triangular graphs”. In: *SIAM Journal on Discrete Mathematics* 9.3 (1996), pp. 349–359.
- [52] J. Djughash, B. Hamner, and S. Roth. “Navigating with Ranging Radios: Five Data Sets with Ground Truth”. In: *Journal of Field Robotics* 26.9 (2009), pp. 689–695.
- [53] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli. “Euclidean Distance Matrices: Essential theory, algorithms, and applications”. In: *IEEE Signal Processing Magazine* 32.6 (2015), pp. 12–30.
- [54] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli. “Acoustic echoes reveal room shape”. In: *Proceedings of the National Academy of Sciences* 110.30 (2013), pp. 12186–12191.
- [55] J. Dong, M. Mukadam, B. Boots, and F. Dellaert. “Sparse Gaussian processes on matrix lie groups: A unified framework for optimizing continuous-time trajectories”. In: *IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 6497–6504.

- [56] D. Droschel and S. Behnke. “Efficient continuous-time SLAM for 3D lidar-based online mapping”. In: *IEEE International Conference on Robotics and Automation*. 2018, pp. 5000–5007.
- [57] F. Dümbgen, M. El Helou, N. Gucevka, and S. Süsstrunk. “Near-infrared fusion for photorealistic image dehazing”. In: *IS and T International Symposium on Electronic Imaging Science and Technology*. 2018, pp. 3211–3215.
- [58] F. Dümbgen, M. El Helou, and A. Scholefield. “Realizability of Planar Point Embeddings from Angle Measurements”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2020, pp. 5710–5714.
- [59] F. Dümbgen, C. Oeschger, M. Kolundzija, A. Scholefield, E. Girardin, J. Leuenberger, and S. Ayer. “Multi-Modal Probabilistic Indoor Localization on a Smartphone”. In: *International Conference on Indoor Positioning and Indoor Navigation*. 2019, pp. 1–8.
- [60] C. Eckart and G. Young. “The approximation of one matrix by another of lower rank”. In: *Psychometrika* 1.3 (1936), pp. 211–218.
- [61] D. El Badawy and I. Dokmanic. “Direction of Arrival with One Microphone, a Few LEGOs, and Non-Negative Matrix Factorization”. In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 26.12 (2018), pp. 2436–2446.
- [62] D. Faccio, A. Velten, and G. Wetzstein. “Non-line-of-sight imaging”. In: *Nature Reviews Physics* 2 (2020), pp. 318–327.
- [63] X. Fang, X. Li, and L. Xie. “Angle-Displacement Rigidity Theory With Application to Distributed Network Localization”. In: *IEEE Transactions on Automatic Control* 66.6 (2021), pp. 2574–2587.
- [64] X. Feng, P. J. E. Verdegem, Y. K. Lee, D. Sandström, M. Edén, P. Bovee-Geurts, W. J. de Grip, J. Lugtenburg, H. J. M. de Groot, and M. H. Levitt. “Direct Determination of a Molecular Torsional Angle in the Membrane Protein Rhodopsin by Solid-State NMR”. In: *Journal of the American Chemical Society* 119.29 (1997), pp. 6853–6857.
- [65] M. B. Fenton, A. D. Grinnell, A. N. Popper, and R. R. Fay. “Bat Bioacoustics”. In: *Springer Handbook of Auditory Research*. Ed. by M. B. Fenton, A. D. Grinnell, A. N. Popper, and R. R. Fay. 1992.
- [66] P. Foster, Z. Sun, J. J. Park, and B. Kuipersl. “VisAGGE : Visible Angle Grid for Glass Environments”. In: *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2213–2220.
- [67] P. Furgale and T. D. Barfoot. “Visual Teach and Repeat for Long-Range Rover Autonomy”. In: *Journal of Field Robotics* 27.5 (2010), pp. 534–560.
- [68] P. Furgale, T. D. Barfoot, and G. Sibley. “Continuous-time batch trajectory estimation using temporal basis functions”. In: *IEEE International Conference on Robotics and Automation*. 2012, pp. 2088–2095.
- [69] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley. “Continuous-time batch trajectory estimation using temporal basis functions”. In: *International Journal of Robotics Research* 34.14 (2015), pp. 1688–1710.

BIBLIOGRAPHY

- [70] D. Gala, N. Lindsay, and L. Sun. “Realtime Active Sound Source Localization for Unmanned Ground Robots Using a Self-Rotational Bi-Microphone Array”. In: *Journal of Intelligent & Robotic Systems* 95 (2019), pp. 935–954.
- [71] R. Galambos. “The Avoidance of Obstacles by Flying Bats: Spallanzani’s Ideas (1794) and Later Theories”. In: *Isis* 34.2 (1942), pp. 132–140.
- [72] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. “Event-based Vision : A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)* (2020), pp. 1–26.
- [73] S. Gerschgorin. “Über die Abgrenzung der Eigenwerte einer Matrix”. In: *Bulletin de l’Académie des Sciences de l’URSS. Classe des sciences mathématiques et naturelles* 6 (1931), pp. 749–754.
- [74] A. Ghods and G. Abreu. “Complex domain super MDS: Computationally efficient localization via ranging and angle information”. In: *IEEE Wireless Communications and Networking Conference 2018-April* (2018), pp. 1–6.
- [75] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Koziński. “Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering”. In: *22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*. 2017, pp. 37–42.
- [76] E. A. Gómez, J. Rodríguez-Gómez, J. Martínez-de Dios, and A. Ollero. “Asynchronous Event-based Line Tracking for Time-to-Contact Maneuvers in UAS”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2020, pp. 5978–5985.
- [77] J. C. Gower. *Euclidean distance geometry*. Vol. 7. 1. 1982, pp. 1–14.
- [78] S. Greif and B. M. Siemers. “Innate recognition of water bodies in echolocating bats”. In: *Nature Communications* 1.106 (2010), pp. 1–6.
- [79] T. N. E. Greville. “Note on the generalized inverse of a matrix product”. In: *SIAM Review* 8.4 (1966), pp. 518–521.
- [80] D. R. Griffin and R. Galambos. “The sensory basis of obstacle avoidance by flying bats”. In: *Journal of Experimental Zoology* 86.3 (1941), pp. 481–506.
- [81] J. M. Groh. *Making space : how the brain knows where things are*. The Belknap Press of Harvard University Press, 2014.
- [82] C. Grussler, A. Rantzer, and P. Giselsson. “Low-rank optimization with convex constraints”. In: *IEEE Transactions on Automatic Control* 63.11 (2018), pp. 4000–4007.
- [83] A. Guerra, D. Dardari, and P. M. Djuric. “Joint Indoor Localization and Navigation of UAVs for Network Formation Control”. In: *Conference Record - Asilomar Conference on Signals, Systems and Computers 2018-October* (2018), pp. 13–19.
- [84] V. Guimarães, L. Castro, S. Carneiro, M. Monteiro, T. Rocha, M. Barandas, J. Machado, M. Vasconcelos, H. Gamboa, and D. Elias. “A motion tracking solution for indoor localization using smartphones”. In: *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2016, pp. 1–8.

- [85] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al. “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs”. In: *Jama* 316.22 (2016), pp. 2402–2410.
- [86] H. Gupta, J. Fageot, and M. Unser. “Continuous-Domain Solutions of Linear Inverse Problems with Tikhonov Versus Generalized TV Regularization”. In: *IEEE Transactions on Signal Processing* 66.17 (2018), pp. 4670–4684.
- [87] G. Hakobyan and B. Yang. “High-Performance Automotive Radar: A Review of Signal Processing Algorithms and Modulation Schemes”. In: *IEEE Signal Processing Magazine* 36.5 (2019), pp. 32–44.
- [88] K. M. Hall. “An r -Dimensional Quadratic Placement Algorithm”. In: *Management Science* 17.3 (1970), pp. 219–229.
- [89] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, 2004.
- [90] T. F. Havel. “Distance Geometry: Theory, Algorithms and Chemical Applications”. In: *In Encyclopedia of Computational Chemistry*. John Wiley & Sons, 1998, pp. 723–742.
- [91] M. P. Hayes and P. T. Gough. “Synthetic Aperture Sonar : A Review of Current Status”. In: *IEEE Journal of Oceanic Engineering* 34.3 (2009), pp. 207–224.
- [92] K. He, Q. Liu, D.-M. Xu, F.-Y. Qi, J. Bai, S.-W. He, P. Chen, X. Zhou, W.-Z. Cai, Z.-Z. Chen, Z. Liu, X.-L. Jiang, and P. Shi. “Echolocation in soft-furred tree mice”. In: *Science* 372.6548 (2021), eaay1513.
- [93] M. Henein, J. Zhang, R. Mahony, and V. Ila. “Dynamic SLAM: The Need For Speed”. In: *IEEE International Conference on Robotics and Automation*. 2020.
- [94] M. Holler, J. Raabe, A. Diaz, M. Guizar-Sicaros, R. Wepf, M. Odstrcil, F. R. Shaik, V. Panneels, A. Menzel, B. Sarafimov, S. Maag, X. Wang, V. Thominet, H. Walther, T. Lachat, M. Vitins, and O. Bunk. “OMNY—A tOMography Nano crYo stage”. In: *Review of Scientific Instruments* 89.043706 (2018), pp. 1–14.
- [95] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [96] G. P. Huang, K. X. Zhou, N. Trawny, and S. I. Roumeliotis. “Bearing-only target tracking using a bank of MAP estimators”. In: *IEEE International Conference on Robotics and Automation*. 2011, pp. 4998–5005.
- [97] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai. “Verification: Accuracy evaluation of WiFi fine time measurements on an open platform”. In: *Proceedings of the Annual International Conference on Mobile Computing and Networking*. 2018, pp. 417–427.
- [98] E. T. Jaynes and G. L. Bretthorst. *Probability Theory – The Logic of Science*. Cambridge University Press, 1987.
- [99] G. Jing, C. Wan, and R. Dai. “Angle-Based Sensor Network Localization”. In: *IEEE Transactions on Automatic Control* (2021), pp. 1–16.
- [100] C. R. Johnson. “Inverse M -matrices”. In: *Linear Algebra and its Applications* 47 (1982), pp. 195–216.

BIBLIOGRAPHY

- [101] M. Kaczmarek, J. Ruminski, and A. Bujnowski. “Accuracy analysis of the RSSI BLE SensorTag signal for indoor localization purposes”. In: *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems* 8 (2016), pp. 1413–1416.
- [102] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. “iSAM2: Incremental smoothing and mapping using the Bayes tree”. In: *International Journal of Robotics Research* 31.2 (2012), pp. 216–235.
- [103] M. Kaess, S. Member, A. Ranganathan, and S. Member. “iSAM: Incremental Smoothing and Mapping”. In: *IEEE Transactions on Robotics* 24.6 (2008), pp. 1365–1378.
- [104] E. Kaufmann, M. Gehrig, P. Foehn, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza. “Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing”. In: *IEEE International Conference on Robotics and Automation*. January 2019. 2018.
- [105] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. “Deep Drone Acrobatics”. In: *Robotics: Science and Systems*. 2020, pp. 1–10.
- [106] J. R. Kendra. “Motion-Extended Array Synthesis—Part I: Theory and Method”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.4 (2017), pp. 2028–2044.
- [107] L. Kneip and C. Baumann. “Binaural model for artificial spatial sound localization based on interaural time delays and movements of the interaural axis”. In: *The Journal of the Acoustical Society of America* 124.5 (2008), pp. 3108–3119.
- [108] R. Koch, S. May, P. Murmann, and A. Nüchter. “Identification of transparent and specular reflective material in laser scans to discriminate affected measurements for faultless robotic SLAM”. In: *Robotics and Autonomous Systems* 87 (2017), pp. 296–312.
- [109] M. Krekovic. “Point , Plane , Set , Match ! Winning the Echo Grand SLAM”. PhD thesis. EPFL, 2020, pp. 77–112.
- [110] M. Kreković, I. Dokmanić, and M. Vetterli. “EchoSLAM: Simultaneous localization and mapping with acoustic echoes”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2016, pp. 11–15.
- [111] M. Kreković, I. Dokmanić, and M. Vetterli. “Shapes From Echoes : Uniqueness From Point-to-Plane Distance Matrices”. In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 2480–2498.
- [112] J. B. Kruskal. “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis”. In: *Psychometrika* 29.1 (1964), pp. 1–27.
- [113] J. B. Kruskal. “Nonmetric multidimensional scaling: a numerical method”. In: *Psychometrika* 29.2 (1964), pp. 115–129.
- [114] J. Lafferty, A. McCallum, and F. C. N. Pereira. “Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data Conditional Random Fields : Probabilistic Models for Segmenting and”. In: *International Conference on Machine Learning (ICML)*. 2001, pp. 282–289.
- [115] J. Lamping and R. Rao. “Laying out and visualizing large trees using a hyperbolic space”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. 1994, pp. 13–14.

- [116] A. Ledergerber, M. Hamer, and R. D'Andrea. "A robot self-localization system using one-way ultra-wideband communication". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2015, pp. 3131–3137.
- [117] T. Li, H. Chen, S. Sun, and J. M. Corchado. "Joint Smoothing and Tracking Based on Continuous-Time Target Trajectory Function Fitting". In: *IEEE Transactions on Automation Science and Engineering* 16.3 (2019), pp. 1476–1483.
- [118] X. Li, X. Cai, Y. Hei, and R. Yuan. "NLOS identification and mitigation based on channel state information for indoor WiFi localisation". In: *IET Communications* 11.4 (2017), pp. 531–537.
- [119] Z. Li, Z. Tian, M. Zhou, Z. Zhang, and Y. Jin. "Awareness of Line-of-Sight Propagation for Indoor Localization Using Hopkins Statistic". In: *IEEE Sensors Journal* 18.9 (2018), pp. 3864–3874.
- [120] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. "Euclidean distance geometry and applications". In: *SIAM Review* 56.1 (2014), pp. 3–69.
- [121] L. Liberti, C. Lavor, and A. Mucherino. *The discretizable distance geometry problem*. Vol. 6. 8. Springer, 2012, pp. 1671–1686.
- [122] F. Liu, C. Shen, G. Lin, and I. Reid. "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.10 (2016), pp. 2024–2039.
- [123] N. R. Lomb. "Least-Squares Frequency Analysis of Unequally Spaced Data". In: *Astrophysics and Space Science* 39 (1976), pp. 447–462.
- [124] T. Long, I. Cohen, B. Berdugo, Y. Yang, and J. Chen. "Window-Based Constant Beamwidth Beamformer". In: *Sensors* 19.2091 (2019), pp. 1–20.
- [125] P. Lourenço, P. Batista, P. Oliveira, and C. Silvestre. "A globally exponentially stable filter for bearing-only simultaneous localization and mapping with monocular vision". In: *Robotics and Autonomous Systems* 100 (2018), pp. 61–77.
- [126] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, A. Markham, J. A. Stankovic, N. Trigoni, and A. M. See. "See Through Smoke : Robust Indoor Mapping with Low-cost mmWave Radar". In: *MobiSys*. 2020, pp. 14–27.
- [127] Y.-c. Lu and M. Cooke. "Motion strategies for binaural localisation of speech sources in azimuth and distance by artificial listeners". In: *Speech Communication* 53.5 (2011), pp. 622–642.
- [128] Z. Q. Luo, W. K. Ma, A. So, Y. Ye, and S. Zhang. "Semidefinite relaxation of quadratic optimization problems". In: *IEEE Signal Processing Magazine* 27.3 (2010), pp. 20–34.
- [129] D. Lymberopoulos and J. Liu. "The Microsoft Indoor Localization Competition: Experiences and Lessons Learned". In: *IEEE Signal Processing Magazine* 34.5 (2017), pp. 125–140.
- [130] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Science & Business Media, 2003.

BIBLIOGRAPHY

- [131] D. Macagnano and G. T. F. De Abreu. “Algebraic approach for robust localization with heterogeneous information”. In: *IEEE Transactions on Wireless Communications* 12.10 (2013), pp. 5334–5345.
- [132] A. Magassouba, N. Bertin, and F. Chaumette. “Aural Servo: Sensor-Based Control from Robot Audition”. In: *IEEE Transactions on Robotics* 34.3 (2018), pp. 572–585.
- [133] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. *Adaptive and generic corner detection based on the accelerated segment test*. Springer Berlin Heidelberg, 2010, pp. 183–196.
- [134] E. Marchand. “Direct Visual Servoing in the Frequency Domain”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 620–627.
- [135] A. T. Mariakakis, S. Sen, J. Lee, and K. H. Kim. “SAIL: Single access point-based indoor localization”. In: *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services* (2014), pp. 315–328.
- [136] A. Martinelli, F. Pont, and R. Siegwart. “Multi-Robot Localization Using Relative Observations”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2005, pp. 2808–2813.
- [137] K. Massey. *Statistical models applied to the rating of sports teams*. Bluefield College, 1997.
- [138] A. McCallum, D. Freitag, and F. Pereira. “Maximum Entropy Markov Models for Information Extraction and Segmentation”. In: *International Conference on Machine Learning (ICML)*. 2000, pp. 591–598.
- [139] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello. “Range-only SLAM with a mobile robot and a wireless sensor networks”. In: *IEEE International Conference on Robotics and Automation*. 2009, pp. 8–14.
- [140] T. Mikolov, K. Chen, G. Corrado, and J. Dean. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).
- [141] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed representations of words and phrases and their compositionality”. In: *Advances in Neural Information Processing Systems*. 2013, pp. 3111–3119.
- [142] K. J. Mitchell, F. Dümbgen, and S. Liu. “Dense Reconstruction for Narrow Baseline Motion Observations”. In: *U.S. Patent*. 2020.
- [143] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klapotocz, J.-C. Zufferey, D. Floreano, and A. Martinoli. “The e-puck , a Robot Designed for Education in Engineering”. In: *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*. 2009, pp. 59–65.
- [144] A. Monod, J. Delon, and T. Veit. “An Analysis and Implementation of the HDR + Burst Denoising Method”. In: *Image Processing On Line* 11 (2021), pp. 142–170.
- [145] J. J. Moré. *The Levenberg-Marquardt algorithm: Implementation and theory*. 1978, pp. 105–116.
- [146] E. Mueggler, G. Gallego, and D. Scaramuzza. “Continuous-time trajectory estimation for event-based vision sensors”. In: *Robotics: Science and Systems*. 2015.

- [147] J. L. Mueller and S. Siltanen. “Nonlinear inversion”. In: *Linear and Nonlinear Inverse Problems with Practical Applications*. Society for Industrial and Applied Mathematics, 2012, pp. 139–158.
- [148] R. Ng. “Digital light field photography”. PhD thesis. Stanford University, 2006.
- [149] L. Nicholson, M. Milford, and N. Sunderhauf. “QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM”. In: *IEEE Robotics and Automation Letters* 4.1 (2019), pp. 1–8.
- [150] *Number of smartphone users from 2016 to 2021, report published by S. O’Dea*. <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. Accessed: 2021-10-28.
- [151] G. Oberholzer, P. Sommer, and R. Wattenhofer. “SpiderBat : Augmenting Wireless Sensor Networks with Distance and Angle Information”. In: *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE, 2011, pp. 211–222.
- [152] G. Oguntala, R. Abd-Alhameed, S. Jones, J. Noras, M. Patwary, and J. Rodriguez. “Indoor location identification technologies for real-time IoT-based applications: An inclusive survey”. In: *Computer Science Review* 30 (2018), pp. 55–79.
- [153] A. V. Oppenheim and G. C. Verghese. “Power Spectral Density”. In: *Signals, Systems and Inference*. 2010, pp. 183–194.
- [154] B. Osting, C. Brune, and S. J. Osher. “Optimal data collection for informative rankings expose well-connected graphs”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 2981–3012.
- [155] M. Pacholska, F. Dümbgen, and A. Scholefield. “Relax and Recover : Guaranteed Range-Only Continuous Localization”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2248–2255.
- [156] C. C. Paige and M. A. Saunders. “LSQR : An Algorithm for Sparse Linear Equations and Sparse Least Squares”. In: *ACM Transactions on Mathematical Software* 8.1 (1982), pp. 43–71.
- [157] H. Pan, R. Scheibler, E. Bezzam, I. Dokmanic, and M. Vetterli. “FRIDA: FRI-based DOA estimation for arbitrary array layouts”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2017, pp. 3186–3190.
- [158] H. Pan, M. Simeoni, P. Hurley, T. Blu, and M. Vetterli. “LEAP : Looking beyond pixels with continuous-space EstimAtion of Point sources”. In: *Astronomy & Astrophysics* 608.A136 (2017), pp. 1–14.
- [159] R. Parhizkar. “Euclidean distance matrices: Properties, algorithms and applications”. PhD thesis. Ecole Polytechnique Federale de Lausanne (EPFL), 2013.
- [160] Y. Park, A. Choi, and K. Kim. “Monaural Sound Localization Based on Reflective Structure and Homomorphic Deconvolution”. In: *Sensors* 17 (2017), pp. 1–19.

BIBLIOGRAPHY

- [161] M. Paton, K. Mactavish, M. Warren, and T. D. Barfoot. “Bridging the Appearance Gap: Multi-Experience Localization for Long-Term Visual Teach and Repeat”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 1918–1925.
- [162] A. L. Patterson. “A Fourier series method for the determination of the components of interatomic distances in crystals”. In: *Physical Review* 46.5 (1934), pp. 372–376.
- [163] J. Peng, B. Liefeng, and J. Xu. “Conditional Neural Fields”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 1–9.
- [164] S. Perrett and W. Noble. “The effect of head rotations on vertical plane sound localization”. In: *The Journal of the Acoustical Society of America* 102.2325 (1997).
- [165] G. Poole and T. Boullion. “A Survey on M-Matrices”. In: *SIAM Review* 16.4 (1974), pp. 419–427.
- [166] Pozyx Labs. *Pozyx Accurate Positioning*. 2019.
- [167] A. Prorok, A. Arfire, A. Bahr, J. R. Farserotu, and A. Martinoli. “Indoor Navigation Research with the Khepera III Mobile Robot : An Experimental Baseline with a Case-Study on Ultra-Wideband Positioning”. In: *International Conference on Indoor Positioning and Indoor Navigation*. 2010, pp. 1–9.
- [168] A. Prorok and A. Martinoli. “Accurate indoor localization with ultra-wideband using spatial models and collaboration”. In: *International Journal of Robotics Research* 33.4 (2014), pp. 547–568.
- [169] K. Qin. “General matrix representations for B-splines”. In: *The Visual Computer* 16 (2000), pp. 177–186.
- [170] R. J. Radke. *Computer Vision for Visual Effects*. Cambridge University Press, 2012.
- [171] V. Radu and M. K. Marina. “HiMLoc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced WiFi fingerprinting”. In: *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2013, pp. 1–10.
- [172] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. “Zee: Zero-effort crowdsourcing for indoor localization”. In: *Proceedings of the Annual International Conference on Mobile Computing and Networking*. 2012, pp. 293–304.
- [173] C. Rascon and I. Meza. “Localization of sound sources in robotics: A review”. In: *Robotics and Autonomous Systems* 96 (2017), pp. 184–210.
- [174] M. Rea, A. Fakhreddine, D. Giustiniano, and V. Lenders. “Filtering Noisy 802.11 Time-of-Flight Ranging Measurements From Commoditized WiFi Radios”. In: *IEEE/ACM Transactions on Networking* 25.4 (2017), pp. 2514–2527.
- [175] D. M. Rosen, C. DuHadway, and J. J. Leonard. “A convex relaxation for approximate global optimization in simultaneous localization and mapping”. In: *IEEE International Conference on Robotics and Automation*. June. 2015, pp. 5822–5829.
- [176] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard. “SE-Sync : A certifiably correct algorithm for synchronization over the special”. In: *The International Journal of Robotics Research* 58.2-3 (2019), pp. 95–125.

- [177] E. Rosten and T. Drummond. “Erickson - 2008 - Methods of treatment using anti-erbb antibody-maytansinoid conjugates.pdf”. In: *European Conference on Computer Vision*. Springer Berlin Heidelberg, 2006, pp. 430–443.
- [178] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. “ORB: An efficient alternative to SIFT or SURF”. In: *International Conference on Computer Vision*. 2011, pp. 2564–2571.
- [179] N. R. Rypkema, E. M. Fischell, and H. Schmidt. “Closed-Loop Single-Beacon Passive Acoustic Navigation for Low-Cost Autonomous Underwater Vehicles”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2018, pp. 641–648.
- [180] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2003.
- [181] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison. “SLAM++: Simultaneous localisation and mapping at the level of objects”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1352–1359.
- [182] U. Saqib and J. R. Jensen. “A Model-based Approach to Acoustic Reflector Localization with a Robotic Platform”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2020, pp. 4499–4504.
- [183] R. Scheibler, J. Azcarreta, R. Beuchat, and C. Ferry. “Pyramic: Full Stack Open Microphone Array Architecture and Dataset”. In: *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. 2018, pp. 226–230.
- [184] R. Scheibler, E. Bezzam, and I. Dokmanic. “Pyroomacoustics: A Python package for audio room simulations and array processing algorithms”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2018, pp. 351–355.
- [185] F. Schilling, J. Lecoœur, F. Schiano, and D. Floreano. “Learning Vision-based Flight in Drone Swarms by Imitation”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4523–4530.
- [186] P. H. Schoenemann. “A Generalized Solution of the Orthogonal Procrustes Problem”. In: *Psychometrika* 31.1 (1966), pp. 1–10.
- [187] J. Schupp, I. Nelken, and A. J. King. *Auditory Neuroscience: Making Sense of Sound*. MIT Press, 2010.
- [188] J. Shi, H. Yang, and L. Carlone. “Optimal Pose and Shape Estimation for Category-level 3D Object Perception”. In: *Robotics: Science and System*. 2021.
- [189] O. Shih and A. Rowe. “Can a Phone Hear the Shape of a Room?” In: *International Conference on Information Processing in Sensor Networks*. 2019, pp. 277–288.
- [190] S. Shiraishi, T. Obata, and M. Daigo. “Properties of a positive reciprocal matrix and their application to AHP”. In: *Journal of the Operations Research Society of Japan* 41.3 (1998), pp. 404–414.
- [191] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao. “Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing”. In: *IEEE Journal on Selected Areas in Communications* 33.7 (2015), pp. 1443–1457.

BIBLIOGRAPHY

- [192] M. Simeoni, S. Kashani, P. Hurley, and M. Vetterli. “DeepWave: A Recurrent Neural Network for Real-Time Acoustic Imaging”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 15274–15286.
- [193] Y. Song, M. Guan, W. P. Tay, C. L. Law, and C. Wen. “UWB/LiDAR Fusion For Co-operative Range-Only SLAM”. In: *IEEE International Conference on Robotics and Automation*. 2019, pp. 6568–6574.
- [194] D. A. Spielman and S.-H. Teng. “Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems”. In: *SIAM Journal on Matrix Analysis and Applications* 35.3 (2014), pp. 835–885.
- [195] S. Stergiopoulos and E. J. Sullivan. “Extended towed array processing by an overlap correlator”. In: *The Journal of the Acoustical Society of America* 86.1 (1989), pp. 158–171.
- [196] P. Stoica and J. Li. “Source Localization from Range-Difference Measurements”. In: *IEEE Signal Processing Magazine* 63 (2006), pp. 63–66.
- [197] P. Stoica and R. Moses. *Spectral Analysis of Signals*. Prentice Hall, Inc., 2004.
- [198] N. B. Suryavanshi, K. Viswavardhan Reddy, and V. R. Chandrika. “Direction Finding Capability in Bluetooth 5.1 Standard”. In: *International Conference on Ubiquitous Communications and Network Computing*. Springer International Publishing, 2019, pp. 53–65.
- [199] P. Tabaghi and I. Dokmanić. “Hyperbolic Distance Matrices”. In: *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*. 2020, pp. 1728–1738.
- [200] P. Tabaghi, I. Dokmanić, and M. Vetterli. “Kinetic Euclidean Distance Matrices”. In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 452–465.
- [201] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.
- [202] D. S. W. Ting, C. Y.-L. Cheung, G. Lim, G. S. W. Tan, N. D. Quang, A. Gan, H. Hamzah, R. Garcia-Franco, I. Y. San Yeo, S. Y. Lee, et al. “Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes”. In: *Jama* 318.22 (2017), pp. 2211–2223.
- [203] C. H. Tong, P. Furgale, and T. D. Barfoot. “Gaussian Process Gauss-Newton for non-parametric simultaneous localization and mapping”. In: *International Journal of Robotics Research* 32.5 (2013), pp. 507–525.
- [204] A. Torres-González, J. R. Martínez-de-Dios, and A. Ollero. “Range-only SLAM for robot-sensor network cooperation”. In: *Autonomous Robots* 42.3 (2018), pp. 649–663.
- [205] M. H. Trinh, S. Zhao, Z. Sun, D. Zelazo, B. D. Anderson, and H. S. Ahn. “Bearing-Based Formation Control of a Group of Agents with Leader-First Follower Structure”. In: *IEEE Transactions on Automatic Control* 64.2 (2019), pp. 598–613.
- [206] P. D. Turney and P. Pantel. “From frequency to meaning: Vector space models of semantics”. In: *Journal of Artificial Intelligence Research* 37 (2010), pp. 141–188.
- [207] J. T. Vanderplas. “Understanding the Lomb-Scargle Periodogram”. In: *The Astrophysical Journal Supplement Series* 236.1 (2018), p. 16.

- [208] M. Vetterli, J. Kovačević, and V. K. Goyal. *Foundations of Signal Processing*. Cambridge University Press, 2014.
- [209] Vidinoti. *Pixlive SDK Release 6.0*.
- [210] P. Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272.
- [211] H. Wallach. “On Sound Localization HE two factors governing sound localization”. In: *The Journal of the Acoustical Society of America* 10 10.270 (1939).
- [212] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.
- [213] M. Warren, M. Greeff, B. Patel, J. Collier, A. P. Schoellig, and T. D. Barfoot. “There’s No Place Like Home : Visual Teach and Repeat for Emergency Return of Multirotor UAVs During GPS Failure”. In: *IEEE Robotics and Automation Letters* 4.1 (2019), pp. 161–168.
- [214] W. Winterhalter, F. Fleckenstein, B. Steder, L. Spinello, and W. Burgard. “Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 2015-Decem. 2015, pp. 3138–3143.
- [215] M. Woolley. *Bluetooth Core Specification Version 5.1 Feature Overview*. <https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-v5-1-feature-overview/>. 2019.
- [216] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, C.-k. Liang, M. Levoy, and P. Milanfar. “Handheld Multi-Frame Super-Resolution”. In: *ACM Transactions on Graphics* 38.4 (2019), 28:1–28:18.
- [217] C. Wu, J. Liu, X. Huang, Z.-P. Li, C. Yu, J.-T. Ye, J. Zhang, Q. Zhang, D. Xiankang, V. K. Goyal, F. Xu, and J.-W. Pan. “Non-line-of-sight imaging over 1.43 km”. In: *Proceedings of the National Academy of Sciences* 118.10 (2021), pp. 1–7.
- [218] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. “Lightweight map matching for indoor localisation using conditional random fields”. In: *International Symposium on Information Processing in Sensor Networks*. 2014, pp. 131–142.
- [219] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. “Robust indoor positioning with life-long learning”. In: *IEEE Journal on Selected Areas in Communications* 33.11 (2015), pp. 2287–2301.
- [220] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik. “Non-Line-of-Sight Identification and Mitigation Using Received Signal Strength”. In: *IEEE Transactions on Wireless Communications* 14.3 (2015), pp. 1689–1702.
- [221] S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang, and S. Lin. “Graph embedding and extensions: A general framework for dimensionality reduction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.1 (2007), pp. 40–51.

BIBLIOGRAPHY

- [222] B. Yang, D. Jayaraman, G. Berseth, A. Efros, and S. Levine. “Morphology-Agnostic Visual Robotic Control”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 766–773.
- [223] M. Yang, C.-H. Chien, T. Delbruck, and S.-C. Liu. “A 0.5 V 55 μ W 64 \times 2 Channel Binaural Silicon for Event-Driven Stereo-Audio Sensing”. In: *IEEE Journal of Solid-State Circuits* 51.11 (2016), pp. 2554–2569.
- [224] Z. Yang, C. Wu, and Y. Liu. “Locating in fingerprint space: Wireless indoor localization with little human intervention”. In: *Proceedings of the Annual International Conference on Mobile Computing and Networking*. 2012, pp. 269–280.
- [225] M. Ye, B. D. Anderson, and C. Yu. “Bearing-Only Measurement Self-Localization, Velocity Consensus and Formation Control”. In: *IEEE Transactions on Aerospace and Electronic Systems* 53.2 (2017), pp. 575–586.
- [226] D. J. Yoon, H. Zhang, M. Gridseth, H. Thomas, and T. D. Barfoot. “Unsupervised Learning of Lidar Features for Use in a Probabilistic Trajectory Estimator”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2130–2138.
- [227] X. Zhong, L. Sun, and W. Yost. “Active binaural localization of multiple sound sources”. In: *Robotics and Autonomous Systems* 85 (2016), pp. 83–92.
- [228] B. Zhou, M. Elbadry, R. Gao, and F. Ye. “BatMapper : Acoustic Sensing Based Indoor Floor Plan Construction Using Smartphones”. In: *MobiSys*. 2017, pp. 42–55.
- [229] B. Zhou, M. Elbadry, R. Gao, and F. Ye. “Towards Scalable Indoor Map Construction and Refinement using Acoustics on Smartphones”. In: *IEEE Transactions on Mobile Computing* 19.1 (2020), pp. 217–230.

FREDERIKE DÜMBGEN

EDUCATION

École polytechnique fédérale de Lausanne, Switzerland (EPFL)

PhD in Computer and Communication Sciences *Sep 2016 – Sep 28 2021 (oral defense date)*

Audiovisual Communications Laboratory, directors: Prof. M. Vetterli, Dr. A. Scholefield.

PhD thesis title: **Blind as a bat: spatial perception using little visual information**. Performed multiple research projects on the theory and applications of localization and mapping using radio-frequency signals, acoustic signals and images. Also active in teaching and student supervision.

Eidgenössische Technische Hochschule Zürich, Switzerland (ETHZ)

Master's Thesis *Feb 2016 – Jun 2016*

Autonomous Systems Lab, directors: Prof. R. Siegwart, A. Karimi, P. Krüsi, M. Bloesch, D. Schindler.

Title: **Local Spline-Based Dense Stereo Reconstruction and Pose Estimation**. Designed and implemented iterative visual feature-free localization and mapping scheme based on B-splines in *ROS*.

École polytechnique fédérale de Lausanne, Switzerland (EPFL)

Master of Science in Mechanical Engineering *Sep 2014 – Jun 2016*

Bachelor of Science in Mechanical Engineering *Sep 2010 – Jun 2014*

Specialization in Control and Mechatronics, minor in Computational Science and Engineering. Exchange year in **Heriot Watt University, Edinburgh, UK** from Sep 2013 – Jun 2014.

WORK EXPERIENCE

Disney Research Los Angeles, California, US

Mar 2018 – Nov 2018

Lab Associate

Implemented learning-based view synthesis algorithm for motion capture using a linear camera array. Successfully published and presented the accomplished work at conference *ICIP 2019*.

voCHabular, Switzerland

Mar 2017 – Mar 2019

Leader of Application Development Team

Directed the app development team of non-for-profit organization *voCHabular* for teaching Swiss German to refugees. Active in app development, recruitment, volunteer management, and fundraising.

LCAV, EPFL, Switzerland

June 2015 – August 2015

Research Assistant

Set up and programmed webcams for visual robot tracking and created robot operation pipeline in python for collecting odometric, visual and acoustic data in python.

Bystronic Laser AG, Berne, Switzerland

Mar 2014 – Sep 2014

Project Intern Total Cost of Ownership

Designed and implemented global sales tool calculating and visualizing the total cost of ownership of laser and waterjet cutting machines. Organized the launching of the tool in global sales offices.

ABB High Voltage Products, Zurich, Switzerland

Sep 2013 – Mar 2014

Technical Training Intern

Designed online e-Learning courses for supplier admission and high voltage factory safety and conducted hands-on training sessions for mechanical assembly of gas-insulated switch gear systems.

PUBLICATIONS

(* means equal contributions)

M. Pacholska*, [F. Dürnberg*](#), and A. Scholefield. Relax and Recover: Guaranteed Range-Only Continuous Localization. *Robotics and Automation Letters*, 5(2):2248–2255, 2020. presented at ICRA. [link](#).

G. Baechler*, [F. Dürnberg*](#), G. Elhami*, M. Krekovic*, and M. Vetterli. Coordinate Difference Matrices. *SIAM Journal on Matrix Analysis and Applications*, 2020. [link](#).

[F. Dürnberg](#), M. El Helou, and A. Scholefield. Realizability of Planar Point Embeddings from Angle Measurements. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. [link](#).

[F. Dürnberg](#), C. Oeschger, M. Kolundzija, A. Scholefield, E. Girardin, J. Leuenberger, and S. Ayer. Multi-Modal Probabilistic Indoor Localization on a Smartphone. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2019. [link](#).

[F. Dürnberg](#), C. Schroers, and K. Mitchell. Light Field Synthesis Using Inexpensive Surveillance Camera Systems. In *IEEE International Conference on Image Processing (ICIP)*, pages 744–748, 2019. [link](#).

M. El Helou, [F. Dürnberg](#), and S. Süsstrunk. AL2: Progressive Activation Loss for Learning General Representations in Classification Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, under submission, 2020. [link](#).

M. El Helou, [F. Dürnberg](#), and S. Süsstrunk. AAM: An Assessment Metric of Axial Chromatic Aberration. In *IEEE International Conference on Image Processing (ICIP)*, pages 2486–2490, 2018. [link](#).

M. El Helou, [F. Dürnberg](#), R. Achanta, and S. Süsstrunk. Fourier-domain optimization for image processing. *arXiv preprint*, 2018. [link](#).

G. Baechler*, [F. Dürnberg*](#), G. Elhami*, M. Krekovic*, et al. Combining Range and Direction for Improved Localization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3484–3488, 2018. [link](#).

[F. Dürnberg*](#), M. El Helou*, N. Gucevskaja, and S. Süsstrunk. Near-Infrared Fusion for Photorealistic Image Dehazing. *IS&T EI Proceedings*, 2018. [link](#).

AWARDS AND HONORS

May 2020: Received the *2020 Google Women Techmaker Scholarship*.

December 2018: Obtained *EDIC Distinguished Service* award for my work as student representative.

2017: Finalist at multiple hackathons (coding competitions): *StartHack* (top 10 of 100), *HackZurich* (top 25 of 145) and *Laushack* (overall 3rd out of 54 teams, competition winner of start-up Gamaya).

September 2016: Awarded *EDIC fellowship* for my first year of PhD (given to < 7% of PhD applicants).

March 2016: Granted the *NCCR Robotics Scholarship for Women* for Master’s Thesis at ETHZ.

September 2011: Accepted into the *Swiss Study Foundation*.